



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD ZACATENCO  
DEPARTAMENTO DE CONTROL AUTOMÁTICO**

**Ludibot: un robot diseñado desde la robótica,  
las ciencias del juego y la didáctica de lenguas y culturas**

tesis que presenta

**MANUEL ALEJANDRO OJEDA MISSES**

para obtener el grado de

**DOCTOR EN CIENCIAS**

en la Especialidad de

**CONTROL AUTOMÁTICO**

Directores de tesis:

**Dr. Alberto Soria López**

**Dra. Haydée Silva Ochoa**

Agradezco a toda mi familia por siempre creer en mí, por apoyarme en cada decisión que he tomado y por brindarme ejemplos de constancia para cumplir metas y objetivos en la vida; a mi abuelita, mi mamá, mis hermanos, mis tíos y mis primos por apoyarme, quererme y siempre estar pendiente de mí. Su apoyo incondicional es una fuente inagotable de motivación.

Agradezco al Dr. Alberto Soria por compartir su conocimiento, sus consejos y, sobre todo, por su paciencia. Gracias por su apoyo y por contribuir a formarme como un futuro investigador y un profesional de calidad.

Agradezco también a la Dra. Haydée Silva por compartir su conocimiento, sus consejos, su paciencia, su guía durante la redacción y por los aprendizajes que me permitió alcanzar en las áreas de la didáctica de lenguas y culturas y de las ciencias de juego.

Agradezco al Departamento de Control Automático del Centro de Investigación y Estudios Avanzados la oportunidad de estudiar el doctorado.

Agradezco a la Universidad Nacional Autónoma de México el haberme abierto sus puertas a través de la Facultad de Filosofía y Letras, pues me dio la oportunidad de reforzar mi formación como doctor. Fue sumamente valioso también el apoyo otorgado en el marco del Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica a cargo de la Dirección General de Asuntos del Personal Académico.

Finalmente, agradezco al Consejo Nacional de Ciencia y Tecnología que, por medio de su Programa de Posgrados de Calidad, me otorgó la beca gracias a la cual pude dedicarme de lleno a mi trabajo de tesis.

## Resumen

En esta tesis se presenta el desarrollo de un robot móvil, llamado Ludibot, basado en el sensor Kinect v2 y diseñado desde una óptica pluridisciplinar. Se trata de un prototipo que permite el desarrollo de aplicaciones ludoeducativas con propósitos de aprendizaje de lenguas, a partir de la interacción gestual humano-máquina.

La principal contribución de esta tesis consiste en la combinación poco usual de la robótica móvil, la didáctica de lenguas y culturas y las ciencias del juego, desde la teoría y desde la práctica. Desde la teoría, el capítulo 1 presenta un marco teórico y conceptual pluridisciplinar que apuesta por la construcción del conocimiento por parte del aprendiente, a partir de su propia experiencia de la interacción humano-máquina, buscando que el aprendizaje sea atractivo, significativo y eficaz.

Desde la práctica, el capítulo 2 expone la arquitectura integrada por objetos a partir de la cual se diseñaron, construyeron y pusieron a prueba dos prototipos ludoeducativos destinados a complementar y facilitar la enseñanza y el aprendizaje de lenguas mediante el uso de la tecnología. En ese mismo capítulo, se exploran las propiedades, las relaciones y los comportamientos que, en conjunto, otorgan el carácter ludoeducativo, y se describen pormenorizadamente las primeras dos versiones de un robot ludoeducativo con una interfaz gráfica, una máquina de estados, acciones multimodales y estrategias de control.

Finalmente, en el capítulo 3, se presentan las aplicaciones desarrolladas durante el proyecto para ambas versiones de Ludibot y se discuten las perspectivas abiertas por la investigación.

En suma, este trabajo está orientado a la búsqueda de una solución novedosa de interacción humano/máquina funcional, de fácil manejo y abierta a una personalización de mecanismos y contenidos, en función de contextos variados para el aprendizaje lúdico de idiomas.

## Abstract

*This thesis presents the development of a mobile robot, called Ludibot, based on the Kinect v2 sensor and designed from a multidisciplinary perspective. It is a prototype that allows the development of ludo-educational applications for language learning purposes, from human-machine gesture interaction.*

*The main contribution of this thesis is the unusual combination of mobile robotics, the language and culture teaching, and game science from theory and practice. From theory, chapter 1 presents a multidisciplinary theoretical and conceptual framework that bets on the knowledge construction by the learner from his own experience of human-machine interaction, aiming an attractive, meaningful, and effective learning.*

*From practice, chapter 2 exposes the object-integrated architecture from which two ludo-educational prototypes were designed, built, and tested to complement and facilitate the language teaching and learning through the use of technology. In the same chapter, the properties, the relationships, and the behaviors are explored, that together give the ludo-educational character, and the first two versions of a ludo-educational robot with a graphical interface, a state machine, multimodal actions, and control strategies are discussed in details.*

*Finally, chapter 3 introduces the applications developed during the project for both versions of Ludibot, and the opened prospects by the investigation are discussed.*

*In addition, this work is oriented to the search of a novel human/machine interaction functional, easy to use, and open to mechanisms and contents personalization, according to varied contexts for playful language learning.*

# Índice general

RESUMEN.....	2
ABSTRACT.....	3
ÍNDICE DE FIGURAS.....	7
ÍNDICE DE TABLAS.....	11
INTRODUCCIÓN .....	12
ANTECEDENTES .....	14
PROBLEMÁTICA .....	15
CONTRIBUCIONES .....	16
HIPÓTESIS .....	18
OBJETIVOS .....	19
<i>Objetivo general</i> .....	19
<i>Objetivos de investigación</i> .....	19
<i>Objetivos de desarrollo</i> .....	20
ESTRUCTURA DE LA TESIS .....	21
<b>CAPÍTULO 1. MARCO TEÓRICO Y CONCEPTUAL PLURIDISCIPLINAR PARA LUDIBOT .....</b>	<b>23</b>
1.1 ROBÓTICA.....	24
1.1.1 <i>Definiciones de “robot” y de “robot móvil”</i> .....	25
1.1.2 <i>Elementos de los robots móviles interactivos</i> .....	27
<i>Estrategias de control</i> .....	31
1.1.3 <i>Interacción desde la robótica</i> .....	37
1.2 DIDÁCTICA DE LENGUAS Y CULTURAS .....	41
1.2.1 <i>Teorías psicológicas del aprendizaje aplicadas a la enseñanza de lenguas</i> .....	42
1.2.2 <i>Nociones clave de la didáctica de lenguas y culturas</i> .....	53
1.2.3 <i>Nuevos aportes de las ciencias de la educación</i> .....	75
1.3 CIENCIAS DEL JUEGO.....	80
1.3.1 <i>Gamificación, ludificación y ludicización</i> .....	81
1.3.2 <i>Regiones metafóricas de juego</i> .....	83
1.3.3 <i>Interacción desde el juego</i> .....	87
1.4 ESTADO DEL ARTE DE LA ROBÓTICA, APRENDIZAJE DE LENGUAS Y JUEGO .....	89
1.4.1 <i>Robótica y aprendizaje de lenguas</i> .....	90
1.4.2 <i>Robótica y juego</i> .....	91
1.4.3 <i>Juego y aprendizaje de lenguas</i> .....	98

1.5. ROBÓTICA LUDOEDUCATIVA .....	101
1.5.1 Definición de la robótica ludoeducativa.....	101
1.5.2 Robot ludoeducativo .....	103
1.5.3 Aplicación ludoeducativa.....	104
<b>CAPÍTULO 2. LUDIBOT: ARQUITECTURA Y CARÁCTER LUDOEDUCATIVO .....</b>	<b>108</b>
2.1 ARQUITECTURA DE SISTEMAS PARA UN ROBOT LUDOEDUCATIVO .....	108
2.1.1 Sistema.....	108
2.1.2 Arquitectura de sistemas .....	109
2.1.3 Arquitectura de un robot ludoeducativo.....	115
2.2 ARQUITECTURA DE LUDIBOT V1 Y LUDIBOT V2 .....	125
2.2.1 Arquitectura de Ludibot v1.....	126
2.2.2 Arquitectura de Ludibot v2.....	135
2.2.4 De Ludibot v1 a Ludibot v2: principales semejanzas y diferencias .....	144
<b>CAPÍTULO 3. INTERFACES Y APLICACIONES DE LUDIBOT .....</b>	<b>155</b>
3.1 INTERFAZ DE LUDIBOT V1 .....	156
3.1.1 Estados principales de la interfaz de Ludibot v1.....	157
3.1.2 Elementos de la interfaz humano-máquina de Ludibot v1 .....	162
3.2 INTERFAZ DE LUDIBOT V2.....	165
3.2.2 Descripción general de la interfaz de Ludibot v2.....	165
3.2.2 Elementos de la interfaz humano-máquina de Ludibot v2 .....	167
3.3 APLICACIÓN POSICIÓNNAME.....	170
3.4 APLICACIÓN ¡TE SIGO! .....	173
3.5 JUEGO DE LA TÊTE AUX PIEDS.....	176
<b>CONCLUSIONES.....</b>	<b>181</b>
<b>BIBLIOGRAFÍA.....</b>	<b>185</b>
<b>APÉNDICE A. MODELADO Y CONTROL DE LUDIBOT V1 .....</b>	<b>200</b>
Control de seguimiento de usuario.....	205
Control de seguimiento de trayectorias .....	209
<b>APÉNDICE B. LENGUAJE DE PROGRAMACIÓN PARA IMPLEMENTAR LA MÁQUINA DE ESTADOS DE LUDIBOT V2 MEDIANTE UN LENGUAJE DE MARCADO TIPO HTML.....</b>	<b>214</b>
SINTAXIS HTML.....	214
SUBDIRECTORIOS O CARPETAS .....	215
CONFIGURACIÓN.....	216

VARIABLES EN EL LENGUAJE.....	217
<i>Variables</i> .....	217
<i>Variable</i> .....	217
<i>Variables de Posicion XYZ</i> .....	218
<i>Variables Booleana</i> .....	218
<i>Variables Reales</i> .....	219
<i>Variables de audio</i> .....	219
<i>Variables UDPVar</i> .....	219
<i>Variable Gesto</i> .....	219
MÁQUINA DE ESTADOS .....	219
ESTADOS .....	221
<i>Estado</i> .....	221
<i>Estado Inicial</i> .....	221
<i>Gestos</i> .....	222
<i>Gesto</i> .....	222
<i>Transiciones</i> .....	222
<i>Transición</i> .....	222
<i>Acciones</i> .....	223
<i>Acción</i> .....	223
<i>Acciones con variables</i> .....	223
<i>Acciones con audio</i> .....	225
<i>Acciones Kinect</i> .....	226
<i>Acciones con Cozmo</i> .....	227
<i>Acciones con Loomo</i> .....	228
<i>Otras acciones</i> .....	229
<b>APÉNDICE C. ENTRENAMIENTO DE GESTOS USANDO APRENDIZAJE MÁQUINA MEDIANTE</b>	
<b>KINECT STUDIO Y VISUAL GESTURE BUILDER.....</b>	<b>230</b>
KINECT STUDIO.....	232
VISUAL GESTURE BUILDER.....	236
<b>APÉNDICE D. USO, CONEXIÓN Y ANIMACIONES DE COZMO .....</b>	<b>248</b>

## Índice de figuras

Figura 1.1. Diagrama de funcionamiento de un robot móvil .....	32
Figura 1.2. Diagrama de transición de la máquina de estados de Ludibot.....	34
Figura 1.3. Diagrama de control de planificación de trayectorias de un robot móvil.....	36
Figura 1.4. Esquema del acto de comunicación verbal humana y sus componentes básicos .....	59
Figura 1.5. Diagrama del salón de clase y la adquisición de una lengua extranjera .....	74
Figura 1.6. Competencias del siglo XXI.....	77
Figura 1.7. Diferencia entre gamificación, ludificación y ludicización .....	83
Figura 1.8. Robots Nao como asistentes para el aprendizaje del idioma inglés .....	91
Figura 1.9. Esquema del brazo robótico aplicado al juego de gato con un humano .....	92
Figura 1.10. Robot manipulador diseñado para un juego de damas mediante sistema de visión ..	93
Figura 1.11. Robots móviles usados para un juego de ajedrez chino.....	93
Figura 1.12. Robots móviles desarrollados por Faust <i>et al.</i> .....	94
Figura 1.13. Robots Nao empleados como asistentes en un juego basado en signos de lenguaje .	95
Figura 1.14. Robot destinado a la rehabilitación de personas de la tercera edad.....	95
Figura 1.15. Robot Justina y laboratorio donde es usado para la navegación.....	96
Figura 1.16. Plataforma de un robot móvil dentro de un laberinto .....	96
Figura 1.17. Arquitectura orientada al servicio para juegos serios .....	97
Figura 1.18. Juego basado en movimientos del cuerpo con el sensor Kinect .....	97
Figura 1.19. Juego interactivo para niños de preescolar basado en Kinect.....	99
Figura 1.20. Juego para el aprendizaje del idioma inglés basado en videojuegos .....	99
Figura 1.21. La intersección disciplinar de la robótica ludoeducativa.....	103
Figura 2.1. Ludibot v1 .....	126
Figura 2.2. Diagrama de la arquitectura de Ludibot v1 .....	128
Figura 2.3. Puntos de las articulaciones del cuerpo detectados por Kinect y ejes de referencia .	131
Figura 2.4. Componentes del sensor Kinect V2.....	132
Figura 2.5. Estados de Ludibot v1.....	134
Figura 2.6. Ludibot v2.....	136
Figura 2.7. Diagrama de la arquitectura de Ludibot v2 .....	137
Figura 2.8. Robot Cozmo .....	143



Figura 2.9. Ludibot v1 y Ludibot V2 .....	149
Figura 2.10. Diagramas de la arquitectura de Ludibot v1 y Ludibot v2 .....	150
Figura 3.1. Diagrama de estados de la interfaz de Ludibot V1 .....	158
Figura 3.2. Gestos disponibles en la función <i>Menú</i> de la interfaz .....	159
Figura 3.3. Gestos disponibles en la función <i>Posicióname</i> de la interfaz.....	160
Figura 3.4. Gestos disponibles en la función <i>¡Te sigo!</i> de la interfaz.....	160
Figura 3.5. Gestos disponibles en la función <i>Juguemos</i> de la interfaz.....	162
Figura 3.6. Interfaz humano-máquina de Ludibot v1.....	163
Figura 3.7. Representación del cuerpo humano con 25 articulaciones .....	164
Figura 3.8. Animación del cuerpo con las manos cerradas .....	164
Figura 3.9. Animación del cuerpo con las manos abiertas.....	164
Figura 3.10. Estados de detección de la interfaz con el sensor Kinect v2 .....	165
Figura 3.11. Interfaz humano-máquina de Ludibot v2.....	168
Figura 3.12. Estados de detección de la interfaz con el sensor Kinect v2 .....	169
Figura 3.13. Diagrama del control no lineal basado en el seguimiento de trayectorias.....	170
Figura 3.14. Tránsito del espacio de trabajo del sensor Kinect al espacio de trabajo del robot móvil .....	170
Figura 3.15. Mapeo de un punto del espacio de trabajo del sensor Kinect al espacio de trabajo del robot móvil .....	171
Figura 3.16. Señales de respuesta de Ludibot v1 para la función <i>Posicióname</i> .....	172
Figura 3.17. Diagrama de control de seguimiento de usuario.....	173
Figura 3.18. Diagrama del cuerpo dado por el sensor Kinect y el punto de referencia $P_r$ .....	174
Figura 3.19. Distancia $d_p$ y ángulo $d_p$ para seguir al usuario .....	174
Figura 3.20. Señales de respuesta de la función <i>¡Te sigo!</i> con las ganancias $k_{p_d} = 15$ , $k_{I_d} = 0.5$ y $k_{p_\theta} = 0.5$ , $k_{I_\theta} = 0.01$ para Ludibot v1 .....	175
Figura 3.21. Interfaz con el estado <i>Repasar</i> de Ludibot .....	177
Figura 3.22. Partes del cuerpo humano incluidas en <i>De la tête aux pieds</i> .....	178
Figura 3.23. Interfaz contando el quinto y último acierto en el estado <i>Mover a Ludibot</i> .....	179
Figura 3.24. Señales de respuesta de Ludibot v1 del juego <i>De la tête aux pieds</i> .....	180
Figura A.1. Diagrama del robot móvil .....	201
Figura A.2. Representación de Borenstein del robot móvil .....	202
Figura A.3. Trayectorias del robot móvil sin ajuste de odometría.....	204

Figura A.4. Trayectoria del robot móvil con ajuste de odometría .....	205
Figura A.5. Distancia entre usuario y robot móvil considerando distancia y ángulo .....	205
Figura A.6. Variables de relación distancia $d_p$ y ángulo $\theta_p$ entre el usuario y el robot móvil ...	206
Figura B.1. Atributos de las articulaciones del cuerpo dados por Kinect v2 .....	227
Figura C.1. Articulaciones del cuerpo dadas por Kinect v2.....	230
Figura C.2. Abrir Kinect Studio en Windows.....	233
Figura C.3. Seleccionar el modo Grabar y el botón de conexión en Kinect Studio.....	233
Figura C.4. Ajustes con Kinect Studio.....	234
Figura C.5. Estado conectado y selección de flujos de datos con Kinect Studio.....	234
Figura C.6. Grabar un Clip con Kinect Studio.....	235
Figura C.7. Abrir Visual Gesture Builder en Windows .....	237
Figura C.8. Crear una nueva solución de entrenamiento en Visual Gesture Builder.....	237
Figura C.9. Escribir el nombre de la solución de entrenamiento en Visual Gesture Builder .....	237
Figura C.10. Crear un nuevo proyecto en Visual Gesture Builder .....	238
Figura C.11. Seleccionar el algoritmo AdaBoost para el entrenamiento .....	238
Figura C.12. Seleccionar el algoritmo RFRProgress para el entrenamiento.....	239
Figura C.13. Confirmar el gesto creado para el entrenamiento .....	240
Figura C.14. Agregar un video capturado en Kinect Studio para el entrenamiento en Visual Gesture Builder .....	241
Figura C.15. Agregar el video capturado en Kinect Studio para el entrenamiento en Visual Gesture Builder .....	241
Figura C.16. Seleccionar la opción <i>Bool</i> para el video del entrenamiento en Visual Gesture Builder.....	242
Figura C.17. Seleccionar secciones “falsas” del video de entrenamiento en Visual Gesture Builder.....	243
Figura C.18. Seleccionar secciones “verdaderas” del video de entrenamiento en Visual Gesture Builder.....	242
Figura C.19. Guardar el proyecto en Visual Gesture Builder .....	243
Figura C.20. Construir el proyecto en Visual Gesture Builder .....	244
Figura C.21. Proceso de construcción del proyecto en Visual Gesture Builder .....	245
Figura C.22. Construir el proyecto en Visual Gesture Builder .....	245
Figura C.23. Proyecto completado y guardado con éxito en Visual Gesture Builder.....	245
Figura C.24. Vista previa en vivo del entrenamiento creado en Visual Gesture Builder .....	246

Figura C.25. Abrir el archivo de vista previa en vivo para el entrenamiento creado en Visual Gesture Builder .....	246
Figura C.26. Prueba y validación del entrenamiento en vivo en Visual Gesture Builder.....	247
Figura D.1. Conexión del robot Cozmo con la computadora y el teléfono celular.....	248
Figura D.2. Instalación de Python.....	249
Figura D.3. Instalación de Android Debug Device mediante Android Studio Setup .....	249
Figura D.4. Aplicación de Cozmo para teléfono Android .....	249
Figura D.5. Instalación del SDK de Cozmo desde el símbolo del sistema.....	250
Figura D.6. Instalación de las librerías de Android (ADB) .....	251
Figura D.7. Activación del dispositivo Android desde el símbolo del sistema .....	251
Figura D.8. Conexión de Cozmo desde el teléfono Android .....	252
Figura D.9. Conexión exitosa de Cozmo desde el teléfono Android.....	252
Figura D.10. Pantalla de la aplicación de Cozmo con juegos, aplicaciones, entre otros .....	252
Figura D.11. Habilitar SDK para Cozmo.....	253
Figura D.12. Ventana donde se indica que Cozmo ha sido conectado en el modo SDK.....	253
Figura D.13. Abriendo carpeta desde el CMD.....	254
Figura D.14. Robot Cozmo realizando una animación .....	254
Figura D.15. Ejecución de comandos para el Cozmo Animation Explorer desde el CMD .....	255
Figura D.16. Dirección del archivo Cozmo Animation Explorer desde el CMD .....	255
Figura D.17. Ventana en línea del Cozmo Animation Explorer .....	256
Figura D.18. Animaciones, acciones y comportamientos disponibles desde el Cozmo Animation Explorer.....	256
Figura D.19. Ejemplo de animación de Cozmo: “anim_bored_01” .....	257
Figura D.20. Lista de comportamientos disponibles para Cozmo .....	257

## Índice de tablas

Tabla 2.1. Características genéricas identificadas por Golden.....	111
Tabla 2.2. Características genéricas identificadas por Mc Govern et al. ....	111
Tabla 2.3. Características genéricas identificadas por Gils.....	112
Tabla 2.4. Características genéricas identificadas por el Departamento de Defensa de Estados Unidos .....	112
Tabla 2.5. Características genéricas identificadas por Hatley.....	112
Tabla 2.6. Características genéricas identificadas por Putman .....	113
Tabla 2.7. Características genéricas identificadas por las definiciones de arquitectura de sistemas presentadas .....	114
Tabla 2.8. Elementos que singularizan un objeto en la arquitectura de sistemas .....	116
Tabla 2.9. Aspectos por considerar para determinar las características y propiedades .....	117
Tabla 2.10. Implementaciones dentro de la programación .....	117
Tabla 2.11. Implementaciones dentro de la máquina de estados .....	118
Tabla 2.12. Tipos de acciones multimodales .....	119
Tabla 2.13. Arquitectura de sistemas de un robot ludoeducativo basada en uno o más objetos que funcionan a partir de una máquina de estados .....	121
Tabla 2.14. Arquitectura de sistemas de Ludibot v1 .....	127
Tabla 2.15. Arquitectura de sistemas de Ludibot v2.....	138
Tabla 2.16. Arquitectura de sistemas de Ludibot v1 (sin sombreado) y Ludibot v2 (con sombreado).....	153

## Introducción

El control automático es un área de la ingeniería que tiene como objetivo lograr la adecuada operación de procesos según ciertas especificaciones de funcionamiento, con la menor supervisión humana posible, y a pesar de la existencia de errores y/o de perturbaciones en el sistema. Al ser uno de los pilares fundamentales de la tecnología moderna, el control automático está presente en una gran cantidad de aplicaciones, procesos y sistemas, tales como reactores químicos y biológicos, sistemas de manufactura, sistemas mecatrónicos, procesos agrícolas, sistemas de comunicación, entre otros.

La robótica es una de las áreas del control automático con mayor potencial de aplicación, e incluye desde nanorobots utilizados en complejas operaciones quirúrgicas hasta robots industriales, submarinos y naves espaciales. Los robots han permitido resolver problemas en múltiples áreas, tales como los procesos de fabricación y la automatización, y han tenido un impacto significativo en sectores como la agricultura, la minería, el transporte, la exploración y la rehabilitación, entre otros.

En el ámbito específico de los robots móviles, el control automático ha permitido el desarrollo de aplicaciones mediante sensores capaces de medir las variables de interés –relativas al entorno, al usuario y/o al robot– así como actuadores capaces de poner en movimiento las extremidades, los eslabones o las ruedas del dispositivo, dando así lugar a robots guías [27] [191], robots de apoyo a la rehabilitación y robots sociales [77][78], robots con navegación y el mapeo simultáneo [50][192][214], robots multiagentes [60], robots educativos [147][148] y robots lúdicos [59] [134] [183].

En efecto, en el ámbito educativo y en el universo de los juegos, los robots están también presentes, pues poseen características que ofrecen interesantes perspectivas. Entre sus ventajas destacan su capacidad para resolver problemas abiertos; su flexibilidad de operación; su capacidad para identificar objetos, personas y entornos; además de su capacidad para realizar movimientos específicos, mediante actuadores, sensores e interfaces humano-máquina, que

propician la comunicación, la colaboración y la interacción entre el humano y el robot. Para la interacción humano-robot, se recurre a las interfaces como medio de comunicación con el usuario.

Debido a que la robótica permite resolver tareas específicas y problemas abiertos [166], es fácilmente concebible su aplicación en el ámbito disciplinar de la didáctica de lenguas y culturas (DLC). La DLC es una disciplina científica de aparición reciente, abierta a los aportes de otros campos, aunque arraigada en una tradición de enseñanza/aprendizaje de idiomas dentro de la cual la tecnología ha jugado históricamente un papel importante en la transformación de paradigmas [35] [64]. Así lo muestra el *Marco común europeo de referencia para las lenguas* (MCER) [85], desarrollado por el Consejo de Europa: publicado en 2001 en inglés y francés, y dado a conocer en español un año más tarde [85], el MCER establece pautas para el aprendizaje, la enseñanza y la evaluación en el área de idiomas y promueve un enfoque orientado a la acción en la DLC. Durante las dos últimas décadas, este documento de referencia ha desempeñado una función central en la transformación de las representaciones y las prácticas de enseñanza/aprendizaje, pues postula que el objetivo de aprendizaje ya no es meramente la competencia lingüística o incluso la competencia comunicativa, sino un conjunto de competencias necesarias para la acción social, misma que requiere diversos tipos de interacciones basadas tanto en la comunicación verbal como en la comunicación no verbal, entre otras [85].

El objetivo central de esta tesis consiste en el desarrollo de Ludibot, un robot móvil que busca favorecer la interacción humano-máquina mediante una aplicación ludoeducativa destinada al aula de idiomas, desde la intersección pluridisciplinar de la robótica, la DLC y las ciencias del juego. En efecto, considerando que las herramientas tecnológicas permiten enfrentar los desafíos actuales y futuros en la enseñanza y el aprendizaje, en el caso de una lengua extranjera parece oportuno el desarrollo de nuevas herramientas teóricas y prácticas capaces de generar dinámicas de aprendizaje favorables a la promoción en el aula de las llamadas competencias del siglo XXI [6][207][208][71]. Para ello, es importante esbozar brevemente los antecedentes de la robótica aplicada a la enseñanza/aprendizaje de lenguas desde una óptica lúdica.

## Antecedentes

Las décadas recientes han estado marcadas por el desarrollo de diversas aplicaciones tecnológicas basadas en el uso de robots y enmarcadas en la cuarta revolución industrial (4RI). Actualmente en proceso, la 4RI está caracterizada por una ampliación del impacto de la automatización gracias a los avances de la robótica y la inteligencia artificial, entre otros factores [71]. Como parte de esta revolución, se ha reconsiderado el uso de la tecnología en la educación, dando lugar al diseño de nuevas herramientas destinadas a complementar el aprendizaje y la enseñanza. Ahora bien, la tecnología ha buscado siempre generar instrumentos que modifiquen nuestro entorno con el fin de satisfacer diversas necesidades. Ello implica la necesidad de ponderar el grado de éxito que los productos tecnológicos alcanzan.

En el caso específico de las aplicaciones lúdicas en el ámbito educativo, los trabajos desarrollados con robots adoptan a menudo un enfoque meramente ingenieril –tal es el caso de los robots que juegan a las damas chinas [3] o al ajedrez [184]–, sin incorporar una reflexión específica en torno a las cuestiones pedagógicas o lúdicas. Existen, desde luego, algunos trabajos de investigación enfocados a los juegos que sí estudian el impacto en el aprendizaje [59]. Sin embargo, en su mayoría, los trabajos identificados no suelen problematizar el concepto de juego, además de adoptar casi siempre un enfoque educativo de corte conductista. Un ejemplo de ello es el robot Nao, un robot especialmente popular que ha sido también utilizado para aplicaciones lúdicas. Tal es el caso también del desarrollo propuesto por Veloso *et al.* [200] para realizar juegos de fútbol sin la intervención del usuario.

Entre los robots educativos relacionados con la DLC que recurren al juego se encuentran el robot de Steels [180][181], que combina inteligencia artificial y juegos de lengua a través de secuencias de interacción verbal entre dos agentes situados en un entorno dado. Otras aplicaciones para el aprendizaje de lenguas con robots han sido concebidas mediante la implementación de aplicaciones con el robot Nao [86][148]. Estos trabajos de investigación han sido concebidos mediante aplicaciones de robótica educativa que apuestan por prácticas unidireccionales de aprendizaje y que son apuntaladas por un enfoque conductista, en la medida en que la interacción consiste en repetir palabras después del robot.

Ahora bien, hoy en día, los aprendientes disponen de una gama amplia, diversa y en constante evolución de herramientas de autoaprendizaje. Sumamente atractivas, tales herramientas suelen ser de carácter gratuito y frecuentemente lúdico, y se basan tanto en la comunicación verbal y como en la comunicación no verbal. Este fenómeno de diversificación de la oferta lúdica con fines de aprendizaje de idiomas obedece al auge de la llamada gamificación [46][95] –o, mejor aún, ludicización [167][168][169]–, que consiste en el desarrollo y la implementación de aplicaciones lúdicas con fines distintos al mero entretenimiento. La ludicización ha dado lugar a los *juegos serios* [5][24] que buscan incrementar la motivación en ámbitos usualmente reacios a introducir elementos lúdicos.

En este contexto, la intersección entre DLC y ciencias del juego ha sido ampliamente estudiada [170][171][172][185][186], al igual que la articulación entre juego, educación y tecnología [58][62][63][97][178][182][183] o el sitio de las tecnologías en el aprendizaje de idiomas [177][183], pero sin una atención específica a la robótica. Por lo anterior, que resulta relevante abordar el problema desde una óptica pluridisciplinar que integra tanto la dimensión tecnológica desde la robótica como las consideraciones derivadas de los avances recientes de las ciencias del juego y de la DLC.

## **Problemática**

Para responder desde la robótica y mediante propuestas lúdicas a los desafíos actuales y futuros en el aprendizaje de idiomas, y para aportar elementos teóricos y prácticos destinados al desarrollo en el aula de las llamadas competencias del siglo XXI, es importante acercarse de manera crítica a la intersección hasta ahora poco explorada entre robótica, DLC, y ciencias del juego, desde un enfoque pluricultural, pluridisciplinar y plurilingüe. Las tres principales variables consideradas aquí serán la robótica móvil, el aprendizaje de lenguas y el juego como herramienta de apoyo a dicho aprendizaje.

En efecto, en el campo de la robótica educativa, los robots móviles suelen ser mucho más frecuentes que los robots manipuladores. Eso se debe a la mayor flexibilidad de uso que permiten los primeros, así como a su mayor capacidad de interacción humano-máquina gracias a interfaces



habitualmente más amigables. La arquitectura de control de los robots móviles incluye sensores, cámaras y actuadores que propician una interacción dinámica con el entorno. Tales características hacen de un robot móvil, así como de las aplicaciones a él asociadas, herramientas atractivas para la resolución de problemas abiertos y complejos como aquellos que se plantean durante el proceso de aprendizaje. La especificidad de un robot móvil lo convierte en una opción particularmente prometedora en el caso del aprendizaje de lenguas, ya sea en situaciones formales de aprendizaje como la enseñanza en el aula, ya sea en situaciones semiformales de aprendizaje como el trabajo autónomo en centros de autoacceso y mediatecas. El uso de un robot que interactúa de manera dinámica con el usuario puede incidir en variables afectivas clave, tales como la motivación y la percepción de los papeles respectivos del alumno y del profesor.

Además, la presencia de sensores, cámaras y actuadores permite ir más allá de un trabajo meramente lingüístico para aprovechar la comunicación no verbal y favorecer el desarrollo tanto de la literacidad mediática multimodal [103] como de las llamadas competencias del siglo XXI [6][207][208][71]. Esto concuerda con las orientaciones metodológicas más recientes en la DLC, derivadas del constructivismo; en el enfoque orientado a la acción [43], las competencias discursivas de la lengua van asociadas con competencias generales y comunicativas del aprendiente [41], entendido ya no sólo como un alumno sino como un actor social.

Como complemento de lo anterior, la robótica móvil aplicada al aprendizaje de idiomas puede incluir una dimensión lúdica. Para aprovechar el potencial pedagógico de la gamificación [46][95], también llamada ludicización [167], es importante ir más allá de una visión instrumental del juego y explorar las cuatro regiones metafóricas que cubre esa noción (el material, el contexto, la estructura y la actitud), entretejiéndolas con sus respectivos correlatos dentro de la robótica y la DLC.

## **Contribuciones**

La principal contribución de esta tesis consiste en poner la robótica móvil al servicio del aprendizaje lúdico de idiomas en general y del francés como lengua extranjera en particular. Esto se traduce en dos productos principales: por un lado, una propuesta de marco teórico y conceptual

pluridisciplinar del cual deriva un esquema de análisis de las propiedades de una arquitectura con fines ludoeducativos; por el otro, Ludibot, un prototipo de robot móvil interactivo destinado a ludicizar el aula de idiomas.

En efecto, la reflexión en torno al desarrollo de plataformas como Ludibot ofrece la oportunidad de redefinir el significado y el alcance del ejercicio docente [67], y contribuye a diversificar las representaciones y las prácticas asociadas a la enseñanza y el aprendizaje de idiomas. Gracias a la asociación inédita entre robótica, juego y DLC, es posible desarrollar nuevas herramientas teóricas y prácticas capaces de generar nuevas dinámicas de aprendizaje, favorables a la promoción en el aula de las llamadas competencias del siglo XXI [6][207][208][71].

Por su parte, la propuesta de Ludibot tiene la intención de involucrar a los estudiantes de idiomas en el uso de aplicaciones lúdicas con robots, con el propósito de que desarrollen competencias lingüísticas, discursivas, mediáticas y generales. Asimismo, se busca motivarlos mediante el estímulo de la curiosidad científica, la indagación, la experimentación y la construcción del conocimiento y el saber. En el marco de este proyecto, la interacción humano-máquina se da en situaciones formales y semiformales de aprendizaje del francés como lengua extranjera, con especial énfasis en la competencia plurilingüe y pluricultural.

En cuanto al desarrollo y la implementación, en el segundo capítulo de esta tesis se presenta pormenorizadamente la primera versión de un robot móvil con una interfaz basada en gestos. El control de las tareas del robot se realiza mediante el uso del sensor Kinect v2. Se expone asimismo una segunda versión aún en proceso, desarrollada en el marco del Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica de la Universidad Nacional Autónoma de México, con el apoyo de la Dirección General de Asuntos del Personal Académico. Esta última está basada en dos robots disponibles en el comercio: Loomo, un robot de alto desempeño comercializado por Segway, así como en el robot Cozmo [7], con una amplia gama de animaciones de movimiento. En este trabajo se mencionan la arquitectura, el material, los programas, la programación y el comportamiento para el desarrollo de ambas versiones de Ludibot. Tanto en la primera como en la segunda versión, se ha buscado desarrollar una solución

novedosa de interacción humano/máquina que sea funcional, de fácil manejo y abierta a una personalización de sus mecanismos y sus contenidos en función de contextos variados, con diversos objetivos.

## **Hipótesis**

Este trabajo de tesis parte de cuatro hipótesis principales. La primera hipótesis postula que, hoy en día, existen aún pocos estudios que problematicen adecuadamente la articulación pluridisciplinar entre la robótica, la DLC y las ciencias del juego. Por ende, es necesario establecer un marco teórico y conceptual específico para fundamentar desde la robótica móvil el desarrollo de aplicaciones lúdicas para el aprendizaje de idiomas y, además, proponer un modelo de análisis a partir de la experiencia de desarrollo de las dos versiones de Ludibot.

La segunda hipótesis deriva de la constatación de que los trabajos previos suelen combinar solamente dos de las tres disciplinas consideradas y suelen recurrir además a modelos conductistas de la interacción humano-máquina. Por ello, se busca integrar consideraciones didácticas y lúdicas actualizadas en este proyecto, que tengan en cuenta los nuevos roles asignados a los distintos actores educativos, para promover una interacción humano-máquina atractiva que aproveche no solo la comunicación verbal, sino también la comunicación no verbal, propiciando el desarrollo de la literacidad mediática multimodal [103] y las competencias del siglo XXI [6][207][208][71].

La tercera hipótesis, complementaria de la anterior, plantea que la integración razonada de las tres principales variables de este trabajo (robótica móvil, aprendizaje de lenguas y juego) es posible desarrollar aplicaciones ludoeducativas accesibles y eficaces para incidir en la optimización de las competencias generales y comunicativas, en situaciones formales y semiformales de aprendizaje. Tales aplicaciones vendrían a enriquecer la gama de herramientas actualmente disponibles, ya que en los años recientes ha aumentado de manera significativa la presencia de herramientas y materiales de autoaprendizaje de idiomas basados en juegos.

Finalmente, la cuarta hipótesis considera que, debido a la flexibilidad de la robótica móvil, se pueden resolver problemas abiertos, tales como los que suelen plantear los problemas lúdicos. Se considera así que es factible aprovechar plenamente el potencial pedagógico de la ludicización, gracias a una visión compleja del juego [169].

## **Objetivos**

### **Objetivo general**

El objetivo general de este trabajo consiste en poner a disposición de aprendientes de idiomas, docentes e investigadores en el área de la DLC herramientas teóricas y prácticas concebidas a partir de la robótica, que complementen y faciliten la enseñanza y el aprendizaje lúdicos de lenguas mediante el uso de la tecnología y, en específico, de un robot móvil llamado Ludibot. Se busca apoyar al aprendiente en la construcción del conocimiento a partir de su propia experiencia de juego en el aula de idiomas en general y en el aula de francés como lengua extranjera en particular, desde un enfoque integral favorable a la interacción humano-máquina atractiva, significativa y eficaz.

### **Objetivos de investigación**

En el ámbito de investigación, los objetivos específicos de esta tesis abarcan las disciplinas de la robótica, las ciencias del juego y la DLC:

1. Fortalecer la investigación pluridisciplinar y sinérgica desde la intersección entre robótica, ciencias del juego y DLC.
2. Proponer un marco teórico y conceptual pluridisciplinar para el desarrollo de Ludibot, que integre tanto una visión contemporánea de las teorías del aprendizaje de idiomas como una adecuada problematización de la noción de juego.
3. Proponer un modelo de análisis basado en el marco teórico y conceptual para evaluar hasta donde los dos prototipos desarrollados son robots ludoeducativos.
4. Desarrollar un prototipo de un robot móvil interactivo de fácil manejo, que permita introducir en el aula de idiomas aplicaciones lúdicas y estructuras favorables para el aprendizaje y que aporte elementos para el estudio del impacto de la robótica.

5. Promover el uso del robot para el aprendizaje de idiomas considerando elementos derivados de las cuatro regiones metafóricas del juego (material, estructuras, contexto y actitud) que permitan aprovechar cabalmente el potencial pedagógico de la ludicización.
6. Aprovechar la interacción humano-máquina para obtener una comunicación efectiva entre el usuario y el robot, haciendo uso de la comunicación verbal y no verbal y promoviendo el desarrollo tanto de las competencias generales y comunicativas desde una óptica plurilingüe y pluricultural como de la literacidad mediática multimodal y de las competencias del siglo XXI.
7. Desarrollar aplicaciones lúdicas que propicien un cambio de actitud en situaciones formales y semiformales de aprendizaje, que incrementen la motivación y que contribuyan a modificar las representaciones y las prácticas asociadas al aprendizaje de una lengua.
8. Realizar el pilotaje y la evaluación experimental del robot.

### **Objetivos de desarrollo**

Los objetivos específicos de desarrollo son los siguientes:

1. Diseñar e implementar las dos primeras versiones de Ludibot, la primera basada en un robot libre y la segunda basada en el robot Loomo.
2. Diseñar e implementar la arquitectura general de cada versión de Ludibot.
3. Programar el sensor Kinect v2 para llevar a cabo la adquisición y el reconocimiento de posiciones del cuerpo.
4. Realizar el modelado cinemático e implementar estrategias de control con el robot.
5. Diseñar la etapa de control para cada versión del robot.
6. Diseñar e implementar la etapa de comunicación para cada robot.
7. Diseñar una interfaz gráfica basada en gestos para cada robot mediante el sensor Kinect v2, que permita diferentes tareas y transiciones mediante el uso de la comunicación verbal y no verbal.
8. Programar y configurar la interfaz para conseguir una interacción fácil y sencilla con el usuario.

9. Proponer un lenguaje sencillo de programación basado en el modelo de una máquina de estado que permita la modificación del comportamiento de Ludibot.
10. Diseñar e implementar aplicaciones lúdicas destinadas a la interacción y al aprendizaje de idiomas mediante el robot, según dinámicas atractivas e integradoras que creen entornos de aprendizaje favorables.

## **Estructura de la tesis**

En el capítulo 1, se presenta un estado del arte referente a las disciplinas estudiadas en la tesis: la robótica, las ciencias del juego y la DLC. Dentro de esta revisión se consideran trabajos relativos a la intersección entre robótica y juego, ciencias del juego y didáctica de lenguas, robótica y didáctica de lenguas. Asimismo, se discuten diversos conceptos que enriquecen el marco teórico y conceptual desde un enfoque pluridisciplinar, considerando como principales variables la robótica móvil, el aprendizaje de lenguas y el juego. Finalmente, se presenta la integración de las variables y los procesos que competen a la definición de la interacción, tomando en cuenta diversos parámetros que son evaluados experimentalmente para definir el campo concreto de esta investigación.

En el capítulo 2, con base en el marco teórico y conceptual propuesto en el capítulo anterior, se discuten las nociones de sistema y de arquitectura de sistemas para, posteriormente, identificar las propiedades requeridas por una arquitectura con fines ludoeducativos. A partir de esos elementos se exponen los dos prototipos de Ludibot, y se exploran aquellas propiedades que hacen de ellos robots ludoeducativos.

En el capítulo 3, se alude al desarrollo de la interfaz gráfica y a las tareas llevadas a cabo para Ludibot v1. Las tareas desarrolladas son *Juguemos*, *Posicióname*, *¡Te sigo!*, *Alto* y *Salir*, cada una asociada a los gestos correspondientes. Asimismo, se presentan las pruebas experimentales, se pormenoriza el desarrollo de la interfaz gráfica v1 para las aplicaciones mencionadas y se indican las respectivas pruebas y evaluaciones experimentales. Para Ludibot v2, se presenta cómo está integrada la interfaz correspondiente.

Finalmente, se presentan las conclusiones que se desprenden de la investigación, la implementación y las pruebas experimentales, y se mencionan las perspectivas abiertas por este trabajo. Se agregan tres apéndices para describir el material (*hardware*) de Ludibot v1; el modelado y el control de Ludibot v1; y, finalmente, la programación para la interfaz de Ludibot v2 basada en lenguaje HTML.

## Capítulo 1. Marco teórico y conceptual pluridisciplinar para Ludibot

Hoy en día, existen pocos estudios que problematicen adecuadamente la articulación pluridisciplinar entre la robótica, las ciencias del juego y la DLC. Por ello, es preciso establecer un marco teórico y conceptual específico para desarrollar un producto integrador como Ludibot. En este capítulo, se consideran aportes actuales de las tres disciplinas involucradas, con el fin de desarrollar herramientas teóricas y prácticas pertinentes y eficaces con énfasis en la interacción humano-máquina.

Establecer un estado del arte basado en las tres disciplinas mencionadas implica, primeramente, estudiar el concepto de robot móvil y los elementos esenciales para su funcionamiento. De este modo, se explora cuándo fue empleado por primera vez el término robot y se da una clasificación general que ayuda a tomar una postura acerca del tipo de robot desarrollado en esta tesis; además, se revisa la interacción humano-máquina, que constituye un campo de investigación amplio. En efecto, el concepto de interacción resulta especialmente relevante, pues atañe tanto a la robótica como a las ciencias del juego y a la DLC.

En lo que atañe a la DLC, es necesario tomar en cuenta distintas teorías del aprendizaje en general y del aprendizaje de lenguas en particular. Se recurre también al *Marco común europeo de referencia para las lenguas* (MCER) [85] y, en especial, a la visión del aprendiente como usuario y actor social, así como a las competencias generales y comunicativas. Se complementará este acercamiento con una explicación de lo que se entiende hoy por literacidad mediática multimodal, así como con una discusión acerca de las competencias del siglo XXI [6][207][208][71].

En tercer lugar, en lo referente a las ciencias del juego, se estudia la ludicización y se problematiza el concepto de juego como metáfora heurística, explorando la utilidad de un modelo de la noción de juego basado en cuatro grandes regiones metafóricas: material, estructura, contexto y actitud.



Finalmente, se explora un punto de encuentro entre las tres disciplinas, estudiando la interacción desde los tres enfoques y generando nuevas dinámicas en el aprendizaje considerando la literacidad mediática multimodal. Para terminar, se discute y problematiza qué debe entenderse por robótica ludoeducativa, con el fin de poner al servicio del aprendizaje de una lengua extranjera los recursos de la robótica y del juego.

En efecto, se persigue el objetivo de crear un sistema que posibilite la interacción entre los robots y los humanos de la manera más natural posible, tomando en cuenta la complejidad de dicha interacción e intentando considerar todos sus aspectos. Para lograrlo, es preciso llevar a cabo tareas de investigación, desarrollo, e integración en diversos campos relacionados con la interacción humano-robot, considerando la comunicación verbal y no verbal, así como la multimodalidad, entre otros temas.

Esta tesis busca proponer una respuesta a ese desafío. Con miras a mejorar la interacción humano-robot desde un enfoque pluridisciplinar, se ha puesto atención en que el sistema expuesto funcione de manera robusta y con la menor supervisión humana posible, mediante aplicaciones lúdicas donde el usuario viva nuevas experiencias durante su aprendizaje. Se espera que el sistema de Ludibot constituya una base eficaz para la interacción con robots en pro del aprendizaje de lenguas mediante aplicaciones lúdicas en un salón de clases.

## **1.1 Robótica**

Desde hace siglos, el ser humano se ha interesado por la creación de máquinas autónomas capaces de realizar una función específica. Ese afán ha tenido como resultado el desarrollo de la robótica, disciplina que se desprende de las ingenierías mecánica, electrónica y eléctrica, así como de la teoría del control y las ciencias de la computación, y que estudia el análisis, el diseño, la manufactura y la aplicación de máquinas automáticas con cierto grado de inteligencia, capaces de realizar tareas que pueden reemplazar las actividades de un ser humano.

En nuestros días, la robótica está presente en prácticamente todos los ámbitos de la vida cotidiana y contribuye a resolver problemas abiertos [102][135][166], mediante la realización de tareas

específicas y repetitivas bajo una mínima supervisión. La robótica ha alcanzado un auge importante en aplicaciones industriales, aplicaciones guías, aplicaciones en la rehabilitación, aplicaciones educativas, aplicaciones con juegos, entre otras. A continuación, se presentan varias definiciones dadas al término de robot, que sirven como base para definir el marco teórico y conceptual para el robot móvil Ludibot.

### **1.1.1 Definiciones de “robot” y de “robot móvil”**

El término *robot* fue acuñado por primera vez por el escritor checoslovaco Karel Čapek en 1920, en su obra *Rossum's Universal Robots* [33][34]. Etimológicamente, se deriva de la palabra checa *robota*, cuyo significado es “labor forzada”, “servicio” o “esclavo”. Años más tarde, el escritor y científico ruso-estadounidense Asimov retoma el concepto en su obra *I, Robot* [179], estableciendo diversos principios bajo los cuales se rigen los robots en general.

Existen diversas definiciones de lo que es un robot, formuladas desde diversos enfoques, por lo que es necesario recurrir a la comparación de distintas fuentes para establecer una postura conveniente para el presente trabajo. Se ha elegido aquí partir de cuatro propuestas: la de Spong [179], la del Instituto de Robots de América (RIA) [10], la de Canudas [32][33] y la de Siegwart [166].

Spong [179] utiliza el término robot para referirse a un manipulador industrial controlado por computadora. Este tipo de robot está basado en un brazo mecánico que funciona bajo el control de una computadora, a la cual está integrado por medio de diferentes dispositivos. En este mismo libro, el autor señala que el Instituto de Robots de América (RIA) [179] entiende por robot todo manipulador multifuncional programable diseñado para mover materiales, piezas, herramientas o dispositivos especializados a través de movimientos programados variables para la realización de una pluralidad de tareas. Aunque están lejos de los robots de la ciencia ficción, los manipuladores son sistemas electromecánicos cuya descripción analítica requiere métodos adaptados a sus características no lineales.

Ninguna de estas dos definiciones resulta plenamente pertinente para nuestros fines, dado su carácter orientado hacia la industria. En efecto, en ambas definiciones se habla de un sistema destinado a un entorno industrial en aras de la disminución de los costos de mano de obra, que opera con mayor precisión, productividad y flexibilidad que otras máquinas especializadas y que el trabajo realizado por seres humanos. Así, los robots suelen ser destinados a la realización de tareas aburridas, repetitivas y/o peligrosas. Aunque esta visión de lo que es un robot resulta limitada e insuficiente para los propósitos de Ludibot, destaca la importancia dada al aspecto programable, que sí es clave para el funcionamiento de un robot ludoeducativo.

Por su parte, Canudas [33] y Siegwart [166] definen a los robots como sistemas electromecánicos programables destinados a realizar diferentes tareas repetitivas que requieren un grado elevado de precisión. Ambos autores categorizan los robots según su función, que varía de acuerdo con la actividad a la cual se pretende incorporar un sistema robótico y con la flexibilidad que dicho sistema sea capaz de brindar al proceso en cuestión. Así, en [33] y [166] los autores distinguen dos grandes tipos de robots: robots manipuladores y robots generadores de movimiento o móviles (terrestres, acuáticos y aéreos).

Esta primera categorización resulta relevante en el marco de este trabajo, que versa sobre un robot móvil, así como sobre sus diversos usos y aplicaciones. En [166], los robots móviles son definidos como sistemas aptos para desarrollar tareas en espacios determinados. Los robots móviles se aproximan a la creación de un *agente inteligente* [156], definido como un sistema capaz de procesar la información de su entorno y de establecer un comportamiento específico para lograr un objetivo determinado.

Un robot móvil es pues un vehículo equipado con algún tipo de actuador y algún elemento mecánico como ruedas y/o hélices, capaces de brindar un movimiento autónomo y manejados por una computadora embarcada [33][94][90][96]. Los mecanismos de locomoción le permiten al robot moverse sin límites en todo su entorno o en un determinado espacio de trabajo (nave industrial, laboratorio, hogar, corredor, etc.), [166] ayudando a desarrollar diversas actividades (exploración minera y marítima, inspección y vigilancia, búsqueda y rescate de personas,

limpieza de desechos peligrosos, entre otras) en ambientes múltiples (por ejemplo, en el caso de la asistencia médica, en hogares de la tercera edad, clínicas y hospitales).

A este tipo de robots se les suele conferir detalles estéticos para hacer más atractiva su apariencia. Por estas razones, el robot móvil es ideal y flexible para su utilización en tareas que implican una interacción humano-máquina [166].

Con base en las referencias citadas, es posible afirmar que Ludibot es un sistema programable dotado de movilidad que, lejos de estar destinado a la producción industrial, busca resolver problemas derivados de la interacción humano-máquina en un contexto de juego con intenciones de aprendizaje de idiomas. Ludibot permite ejecutar diferentes aplicaciones lúdicas repetitivas mediante estrategias de control que permiten cambiar su estado y que lo hacen capaz de desplazarse en espacios determinados y de realizar ciertos movimientos predefinidos en áreas firmes según trayectorias deseadas. En efecto, contrariamente a un robot manipulador [108], que estaría fijo en un marco de referencia, Ludibot es capaz de desplazarse en entornos específicos. Ludibot está equipado con cámaras, sensores y actuadores que son controlados por una computadora. Ello permite la interacción mediante una interfaz humano-máquina basada en la comunicación verbal y no verbal, apta para procesar la información de su entorno.

Para brindar al lector un panorama más amplio de los antecedentes de Ludibot dentro de la robótica, en las siguientes secciones se presentan los elementos de los robots móviles interactivos: los medios de locomoción, los sensores, los actuadores y las estrategias de control.

### **1.1.2 Elementos de los robots móviles interactivos**

Entre las aplicaciones con robots móviles interactivos más destacadas en las dos últimas décadas se encuentran los robots guías [27][191]; los robots de apoyo a la rehabilitación [77][78]; los robots que utilizan métodos innovadores para la navegación y el mapeo simultáneo [50][192][214]; los robots multiagentes [60]; así como los robots aplicados a juegos [58][133][182]. Para el desarrollo de estos robots, se requiere elegir el medio de locomoción más adecuado, así como predeterminar las características esperadas de la interacción humano-máquina. Para ello, se acude al uso de sensores capaces de medir las variables de interés, relativas

al entorno, al usuario y/o al robot, así como a actuadores, capaces de poner en movimiento las extremidades o las ruedas del dispositivo. Finalmente, es preciso desarrollar las estrategias de control idóneas. Cada uno de estos cuatro elementos es abordado con más detalle a continuación, con el fin de clarificar sus funciones respectivas y de explicar las decisiones tomadas en el desarrollo de Ludibot.

### ***Medios de locomoción***

De acuerdo con el tipo de locomoción que emplean los robots móviles para desplazarse, dichos robots pueden ser clasificados en tres categorías: de ruedas [1][2][29][105][130][131], de patas [146] [194] y con orugas [69][89]. Aunque la locomoción por patas y la locomoción por orugas han sido ampliamente estudiadas, la mayoría de los robots móviles que se han reportado, construido y evaluado en trabajos de investigación utilizan ruedas para desplazarse. Esto se debe a que los robots móviles de ruedas son más eficientes en energía, puesto que generalmente se desplazan en superficies lisas y firmes; poseen múltiples aplicaciones en la industria; requieren partes menos complejas y en menor número, haciendo más fácil su construcción; suelen ocasionar menor desgaste en la superficie donde se mueven en comparación con las bandas de las orugas. Además, el control de las ruedas es menos complejo que los otros mecanismos de movimiento [83] y los problemas de balance no presentan gran dificultad, ya que el robot permanece en contacto con una superficie.

Un robot móvil de ruedas, en forma general, debe considerar tres aspectos respecto al tipo de rueda empleado: el modelo, el sistema de actuadores y el controlador. El modelo remite a las configuraciones relativas a la disposición de las ruedas en la estructura del robot. El sistema de actuadores permite el movimiento de la estructura cinemática directamente con la rueda. El controlador propicia el accionamiento de los actuadores para mover las ruedas del robot mediante estrategias de control.

Las ruedas que se emplean en el diseño de un robot móvil pueden ser clasificadas en diferentes tipos [130][131][213], entre los que destacan por su frecuencia de uso las ruedas omnidireccionales, las ruedas convencionales, las ruedas tipo castor y las ruedas tipo bola. La posibilidad de combinar estos diversos tipos de ruedas lleva a tener una gran variedad de

modelos de un robot móvil en lo que atañe las ruedas. En el caso de Ludibot, se optó por usar ruedas convencionales.

[25][198][199] estudian la disposición de las ruedas en la estructura de un robot móvil. La disposición omnidireccional permite al robot moverse en cualquier dirección sin necesidad de reorientarse, mientras que las ruedas convencionales, tipo castor y tipo bola aparecen según diferentes configuraciones [202], entre ellas un modelo diferencial, basado en dos ruedas fijas y en una rueda loca tipo castor o bola. Ludibot emplea la configuración diferencial debido a que su construcción es más sencilla de implementar y construir, además de tener una cinemática y una odometría sencillas, propicias al desarrollo de estrategias de control en tiempo real.

En la obtención de modelos cinemáticos de un robot móvil de ruedas, es común introducir suposiciones de diseño y de operación [129][130]. Respecto a la odometría, que es el estudio de estimación de la posición y el desplazamiento de un robot móvil, la configuración diferencial mediante ruedas es una de las más comúnmente empleadas [1][29][130][131]. Sin embargo, durante el movimiento y la navegación [50][204] de un robot con configuración diferencial hay que tomar en cuenta el deslizamiento de las llantas, factor que afecta al robot por dos razones principales [39]: ocasiona errores de posicionamiento en robots móviles de ruedas que utilizan odometría [22] y provoca desperdicio de energía, aspecto fundamental en el desplazamiento del robot móvil. A pesar de esto, se han desarrollado modelos matemáticos, donde se consideran y ajustan los errores por deslizamiento, para el robot móvil de ruedas convencionales en [12][73][149][159][165] y para omnidireccionales en [156][206].

### *Sensores*

Los trabajos de investigación basados en modelos cinemáticos mediante el ajuste de la odometría requieren el uso de diversas clases de sensores, presentadas a continuación.

Los sensores son dispositivos que sirven para medir parámetros del entorno y del sistema. Las mediciones son usadas en el sistema de control para su procesamiento. Los sensores se pueden clasificar en sensores externos y sensores internos. Los sensores externos funcionan para

tomar datos provenientes del entorno del robot, por ejemplo, detectar objetos del medio ambiente. Entre ellos se encuentran los sensores de fin de carrera, sensores ultrasónicos, sensores infrarrojos, sensores de nivel de iluminación, sensores de temperatura, entre otros. Los sensores internos sirven para controlar y medir parámetros para y durante el funcionamiento del robot. Por ejemplo, se tienen los sensores de velocidad, los sensores de posición, los sensores de fuerza, entre otros.

Diversos trabajos de investigación ilustran el uso de los tipos de sensores mencionados. En [16] se usan sensores ultrasónicos para la navegación de un robot móvil en un espacio cerrado; en [109] se presenta la estimación de posición de un robot móvil empleando un escáner de láser y filtros lineales; en [114] se propone un método que permite estimar simultáneamente la configuración de un robot móvil y los errores de odometría sistemáticos y no sistemáticos; en [192] se integran el mapeo enrejado bidimensional y topológico para la navegación de un robot dentro de un ambiente cerrado y estructurado, logrando una mayor precisión y eficiencia en su desplazamiento reduciendo los errores. En el caso específico de Ludibot, se emplean decodificadores ópticos incrementales, que ayudan a medir el número de vueltas que el motor efectúa por segundo considerando la relación de engranaje desde el eje del motor hasta la llanta del mismo robot.

### ***Actuadores***

Los actuadores [19] son elementos de los sistemas de control cuya finalidad consiste en transformar su salida a través de un microprocesador o un controlador, mediante una acción de control de una máquina o un dispositivo. Los actuadores transforman energía; al recibir la orden de un regulador o controlador, generan una nueva orden para activar un elemento final de control, como por ejemplo una válvula, un motor, entre otros. Los actuadores pueden ser neumáticos, hidráulicos y eléctricos, siendo estos últimos los más comunes y usados para los robots móviles.

Los actuadores son elementos que influyen directamente en la señal de salida del proceso automatizado, modificando su magnitud según las instrucciones que reciben de la unidad de

control. La locomoción de Ludibot está basada en dos motores, ambos controlados en modo velocidad con el objetivo de que el robot móvil siga una trayectoria dada. Los actuadores le permiten desplazarse en un espacio determinado mediante la generación de trayectorias. La locomoción está basada en dos llantas acopladas a motores de corriente directa (CD).

El modelado de un motor de CD ha sido estudiado en [93] y [102] como un actuador para diversas aplicaciones. En [18][100][112][145] se menciona que dichos actuadores tienen la característica primordial de ser usados en estudios de robótica mediante el modelado de un sistema lineal, lo que simplifica la identificación de parámetros en su modelo y, por ende, la aplicación de estrategias de control.

Este modelo de motor de CD permite suponer que la única fricción presente es la viscosa, aunque en la práctica existe el efecto de otros tipos de fricción no lineal (que son más apreciables cuando se trabaja con velocidades pequeñas), lo cual se resuelve seleccionando un motor tal que el efecto de cualquier fricción no lineal sea muy pequeño. En el caso específico de Ludibot, esto permite modelar los motores y aplicar estrategias de control de manera más efectiva y sencilla.

### **Estrategias de control**

La robótica requiere de otras disciplinas como el procesamiento de imágenes y señales, la electrónica, la teoría de control y la inteligencia artificial. Durante las últimas décadas, se han diseñado y construido incontables dispositivos robóticos que se pueden clasificar según dos dimensiones principales: autonomía y complejidad del entorno [28][166]. Estas dos dimensiones influyen directamente en el control, la organización, el desarrollo y la estructura de un robot.

La autonomía de un robot depende de sus sensores, así como de la variedad y la estructura de su locomoción en conjunto con sus actuadores, y consecuentemente, de la capacidad de procesamiento que el robot es capaz de percibir y/o interpretar, así como de la cantidad y la estructura de los tipos de acciones que puede realizar para conseguir sus objetivos de manera autónoma. Por su parte, la complejidad del entorno corresponde a la variedad y a la variabilidad de los parámetros del entorno que el robot es capaz de enfrentar e incluso transformar. Por lo



tanto, la complejidad del entorno no remite a la extensión espacial donde los robots pueden funcionar e interactuar, sino a la cantidad de entornos cualitativamente diferentes que el robot es capaz de identificar.

Según Siegwart [166], la estructura de funcionamiento de un robot móvil puede incluir varios elementos esenciales que se presentan en la Figura 1.1. El conocimiento del ambiente es útil para simplificar la construcción de un modelo de dicho ambiente. La percepción utiliza información de los sensores para obtener e interpretar datos que permiten construir un mapa o modelo local del ambiente en donde se mueve el robot. La localización y el mapeo permiten ubicar la posición del robot en un mapa global del ambiente [127]. Una vez que se conoce dónde se encuentra el robot, se puede planificar que siga una trayectoria u ocupe posiciones sucesivas deseadas, empleando información de la tarea que se desea realizar. Finalmente, se ejecutan las trayectorias de movimiento del robot desde la posición inicial hasta la posición deseada, empleando una estrategia de control para los actuadores que mueven al robot.

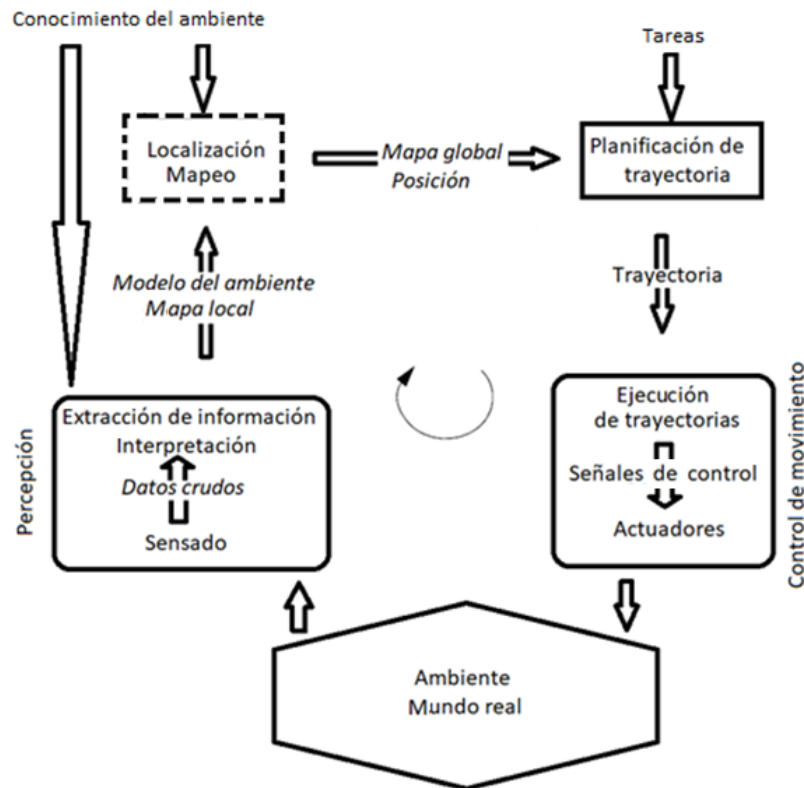


Figura 1.1. Diagrama de funcionamiento de un robot móvil

Tal como lo menciona Siegwart en [166], la robótica móvil no se basa en una arquitectura estandarizada. Por ello, es posible adoptar una estructura funcional que permite organizar la estructura de funcionamiento mediante lenguajes de especificación de tareas, procesos de decisión probabilístico o máquinas de estados [82].

En el caso de Ludibot, las tareas se obtienen a través de una máquina de estados (ME). No se emplea el bloque de localización/mapeo debido a que el robot móvil por el momento sólo efectúa trayectorias que no requieren la localización en un mapa. En esta tesis se propone el uso de un modelo basado en una ME finita, debido a que es una forma directa para programar a Ludibot. Así, el modelo de programación que se emplea para la implementación de aplicaciones ludoeducativas requiere realizar acciones en un estado, cambiar el estado a partir de una entrada—definida como una transición— y realizar acciones durante una transición. La ME basada en comportamientos ofrece dos ventajas interesantes. La primera de ellas es la extensibilidad: es posible añadir nuevos comportamientos correspondientes a nuevos estados y transiciones. La segunda se refiere a la posibilidad de incorporar al planificador de trayectoria la localización/mapeo para navegar en ambientes más complicados.

Formalmente, una ME [82] es definida como una tupla o secuencia ordenada finita de  $n$  objetos:

$$ME = (\Sigma, Q, f, q_0, F) \quad (1.1)$$

donde  $\Sigma$  designa el conjunto de entradas,  $Q$  es el conjunto finito y no vacío de los estados de la máquina de estados,  $f$  es la función de transición que indica en qué situaciones la ME pasa de un estado a otro y se define  $f : Q \times \Sigma \rightarrow Q$ ,  $q_0 \in Q$  es denotado como el estado inicial y  $F \subset Q$  es el conjunto de estados finales de aceptación ( $F \neq \emptyset$ ). Una ME va cambiando de estado dependiendo de la entrada que reciba. La salida está definida por dos valores: *aceptado* y *no aceptado*, que indican si la secuencia de entradas que se ha recibido es o no válida.

Generalmente, existen dos formas de representar una ME: mediante una tabla de transición o mediante un diagrama de transición. En esta tesis, utilizamos un diagrama de transición, en el cual los estados finales aparecen como un óvalo, el estado inicial está indicado por una flecha y las transiciones son dadas igualmente por flechas, que conducen de un estado inicial a un estado final.

A cada aplicación ludoeducativa desarrollada para Ludibot corresponde una ME específica, es decir un conjunto de estados, transiciones y acciones. Cada estado puede ejecutar una o varias acciones, mismas que se repiten mientras la ME se encuentra en ese estado. Se recurre además a diversas transiciones, que corresponden al cambio de un estado a otro y durante las cuales es posible ejecutar una o varias acciones específicas.

Todos los estados  $f$  se incluyen en  $F$ . Las acciones pueden ser multimodales, pues incluyen interacciones verbales y/o no verbales, que serán mencionadas más adelante. En la figura 1.2, se representa un diagrama con cuatro estados usados para Ludibot: *Aprende a controlarme* (estado inicial), *Menú*, *¡Te sigo!* y *Posicióname*. Los estados y las transiciones permiten brindar al robot móvil diferentes modalidades de interacción mediante las transiciones *SalirT*, *Te sigoT* y *PosiciónameT*.

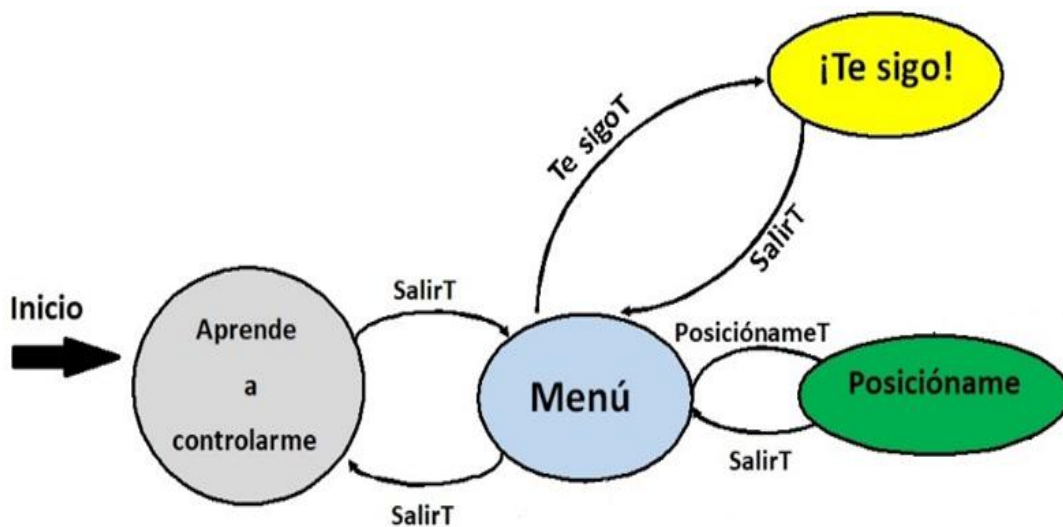


Figura 1.2. Diagrama de transición de la máquina de estados de Ludibot

La planificación de trayectorias, definida como el recorrido que realiza el robot móvil dentro del espacio de trabajo, ha sido resuelta en la literatura de diversas maneras, por ejemplo, mediante un regulador cuadrático lineal [47], donde se presentan resultados experimentales del seguimiento de trayectorias de un robot móvil. En [8] se expone el control de un robot móvil diferencial, recurriendo a un esquema de control de estructura variable. Aprovechando la propiedad de configuración diferencial [56] que satisface el modelo cinemático del robot móvil de ruedas utilizado en [174], se ilustra el diseño de un controlador dinámico para la tarea de seguimiento de trayectoria. En [212] se presenta la linealización del modelo dinámico de un robot móvil de ruedas por medio del método de par calculado. En [109] se realiza el control de seguimiento de trayectoria a través de un control basado en modos deslizantes. En [173], para el modelo cinemático de un robot diferencial se realiza el diseño de control mediante linealización entrada-salida; respecto a sus actuadores, se diseña un controlador que no requiere la medición de variables mecánicas. En [187] se muestra un controlador difuso para el seguimiento de trayectorias de un tractor con cinco tráileres. En [155] se utiliza un compensador neuronal dinámico adaptable. Finalmente, en [13] se realiza un análisis del modelo de la velocidad angular de las ruedas de un robot móvil, lo que deriva directamente en la optimización del tiempo que emplea para desplazarse.

Habiendo considerado múltiples alternativas, en el caso de Ludibot se implementó una estrategia de control para la planificación de trayectorias que permite desplazar al robot móvil en un plano en función de la posición y la orientación definida por el vector  $\bar{X}(t) = (x(t), y(t), \theta(t))$ , como se muestra en la figura 1.3-a. Los movimientos que realiza el robot móvil están dados por la ley de control no lineal realimentada propuesta por Canudas [33].

La estructura del sistema de control de planificación de trayectorias con realimentación de entrada-salida se muestra en la figura 1.3. Dicho sistema de control realimentado también se conoce como “sistema de control en lazo cerrado”, en el cual el controlador utiliza la señal de error –entendida como la diferencia entre la entrada de referencia y la señal de realimentación, medida mediante un sensor– con el fin de reducir el error y llevar la salida del sistema a un valor deseado.

Para que un robot móvil realice movimientos planeados es necesario resolver el problema de seguimiento de trayectorias. Este problema puede formularse y solucionarse mediante estrategias de control que permiten ya sea estabilizar la posición del robot móvil en el plano  $(x, y, \theta)$  respecto a un punto de operación, ya sea hacer que el robot siga una secuencia de puntos en el plano (figura 1.3-b).

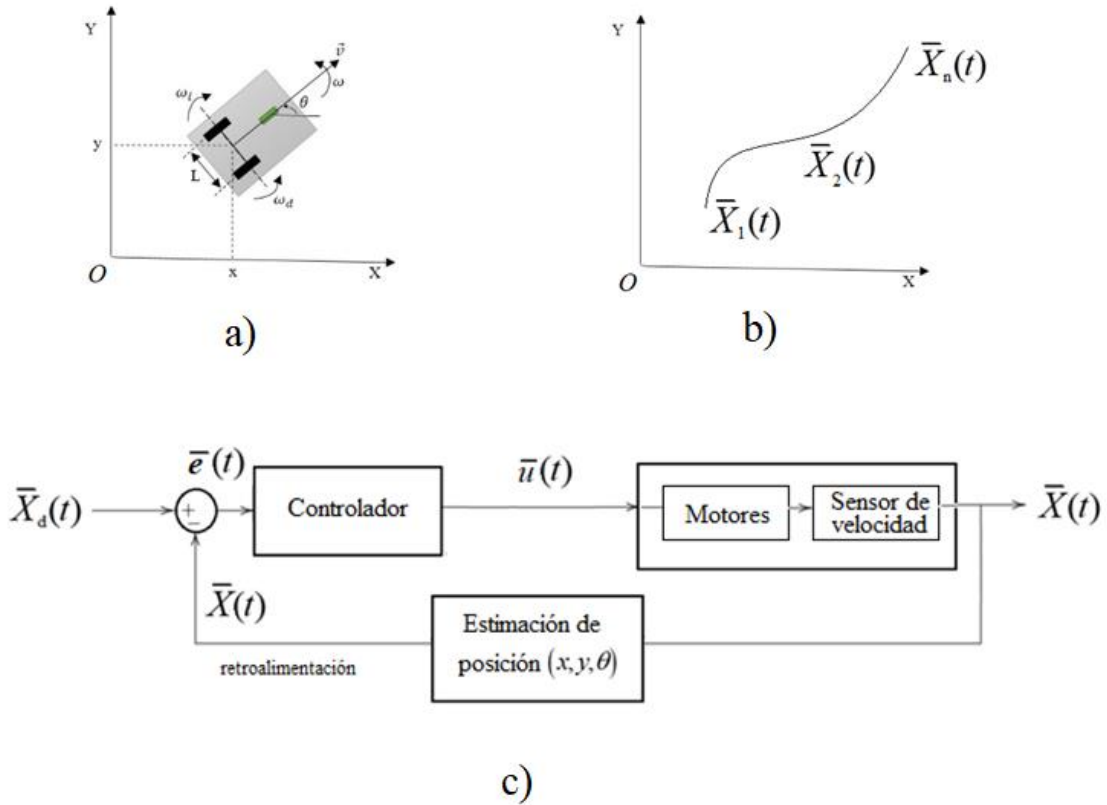


Figura 1.3. Diagrama de control de planificación de trayectorias de un robot móvil

La secuencia de puntos en el plano es definida como  $\bar{X}_d(t) = (x_d(t), y_d(t), \theta_d(t))$ , ya que en caso de que las velocidades fueran nulas, el problema consistiría en alcanzar una posición de referencia fija. El problema para la planificación de trayectorias consiste en diseñar una estrategia capaz de controlar las velocidades de los motores tal que, dada una secuencia de puntos deseados  $\bar{X}_1(t), \bar{X}_2(t), \dots, \bar{X}_n(t)$  (figura 1.3-b), el robot sea capaz de construir una trayectoria y moverse en el plano de manera que la salida del sistema sea  $\bar{X}(t) = (x(t), y(t), \theta(t))$ . En otras palabras, el error

$\bar{e}(t) = \bar{X}(t) - \bar{X}_d(t) = 0$  debe ser cero: el robot debe seguir la trayectoria dada por la secuencia de puntos deseados en el plano, controlando simultáneamente la posición y la orientación.

El control para la planificación de trayectorias está en función de la señal de salida o, dicho de otra manera, el sistema en lazo cerrado emplea la trayectoria de salida. Dicha trayectoria de salida (figura 1.3-c) es comparada con la entrada (trayectoria de referencia) para generar las señales de error de posición y de orientación que recibe el controlador. A partir de tales señales, el controlador ejecuta una acción de control  $\bar{u}(t)$  (en las tareas del robot móvil), con el fin de disminuir el error y, por lo tanto, de llevar la salida del robot móvil a una trayectoria deseada (véase Apéndice B, donde se exponen detalladamente el modelado y las estrategias de control de Ludibot). El modelo es capaz de estimar la posición y la orientación del robot móvil en función de las velocidades de los motores, simultáneamente. Las velocidades son medidas mediante los sensores de velocidad (decodificadores ópticos incrementales) del robot.

Una vez presentado el control que será empleado para llevar a cabo los movimientos del robot móvil en los espacios de aprendizaje, es importante proporcionar elementos teóricos básicos acerca de la interacción lúdica humano-máquina. A continuación, se explora el concepto de interacción desde el enfoque de la robótica.

### **1.1.3 Interacción desde la robótica**

La interacción humano-máquina ha sido un campo de estudios de creciente relevancia en los últimos años en el área de la robótica según Manenat-Thalmann [113]. La interacción humano-robot es una actividad comunicativa dirigida a uno o varios usuarios, mediante un intercambio de acciones y reacciones verbales y no verbales. El modelo de interacción que se desarrolla en las siguientes páginas permite estudiar las interacciones entre humanos y robots y se enfoca particularmente a los robots sociales y de servicio interactivos [139].

La interacción vista desde la robótica busca propiciar una forma natural de interacción humano-máquina por medio de una modalidad dada. La modalidad de interacción se refiere a la manera en que se lleva a cabo la actividad comunicativa. Las modalidades de interacción aquí

consideradas son tres: la modalidad verbal, la modalidad no verbal y la modalidad multimodal [113].

La modalidad verbal incluye los modos oral y escrito. En este caso, la interacción se lleva a cabo mediante el habla y la escritura.

Por su parte, la modalidad no verbal está formada por los modos cinésico, proxémico, visual, sonoro y/o expresivo. El modo cinésico está dado por el significado expresivo, apelativo o comunicativo de los movimientos corporales y de los gestos aprendidos, no orales, de percepción visual, táctil, solos o en relación con la estructura lingüística y paralingüística y con la situación comunicativa. Entre ellos se encuentran los significados gestuales, faciales, posturales y de movimiento. El modo proxémico hace referencia a la manera en que las personas estructuran, utilizan y perciben el espacio en el proceso de interacción, es decir, a la manera en que manejan la proximidad y el alejamiento entre los elementos durante la interacción. El modo visual es dado por la interpretación de lo que percibimos a través de los ojos; permite, entre otros, interpretar las expresiones faciales que proporcionan información social, emocional y anímica. El modo sonoro está constituido por señales acústicas; es un sistema de comunicación específico en el cual, para la transmisión del mensaje, se emplean sonidos peculiares y específicos. Finalmente, el modo expresivo es llevado a cabo mediante la expresión de sensaciones, sentimientos, emociones y pensamientos. Éstos son clasificados en emblemas dados por los gestos empleados en sustitución del lenguaje verbal; en ilustraciones dadas por los gestos que acompañan al lenguaje verbal; y finalmente, en gestos reguladores, que son aquellos que contribuyen favorablemente a la comunicación y la interacción con otros.

Finalmente, la interacción multimodal es dada por la combinación de dos o más modalidades provenientes de las modalidades verbal y no verbal (por ejemplo, texto escrito y sonido; señales orales, sonoras, proxémicas y visuales) ya sea a la entrada, ya sea a la salida, ya sea a la entrada y a la salida.

En Ludibot, se ha elegido trabajar con una interacción multimodal, con énfasis en los modos cinésico, proxémico, visual y sonoro, y con elementos importantes de los modos escrito y expresivo; a futuro, se planea integrar el modo oral.

Independientemente de la modalidad de interacción elegida, un sistema de interacción eficaz debe ser adaptable para el usuario, de uso fácil e intuitivo. La capacidad de adaptación, variable, puede verse reflejada según [4] en los siguientes puntos: el idioma, la distancia de interacción, la familiaridad con el sistema y la respuesta. Se presenta a continuación cada uno de estos aspectos con más detalle.

1. Adaptación al idioma. El sistema es capaz de usar una lengua de fácil manejo para el usuario. La adaptación puede consistir en la capacidad para interactuar en dos o más idiomas (bi o plurilingüismo).

En el caso de Ludibot, se usan por ahora el español y el francés, aunque a futuro podrán ser utilizados otros idiomas. La interfaz de Ludibot permite que el usuario experimente con ella para entrenarse en su uso antes de emplear las aplicaciones lúdicas. Fomenta así la comunicación verbal mediante los sonidos y los mensajes, y la comunicación no verbal mediante el uso de gestos con las manos para identificar las diferentes tareas del robot.

2. Adaptación de la distancia de interacción. El sistema debe ser capaz de determinar cuál es la distancia de interacción a mantener con cada usuario en concreto.

En el caso de Ludibot, la dimensión proxémica está asociada con el sensor Kinect v2, el cual permite detectar y adquirir parámetros del usuario. Entre las funciones de la interfaz, se encuentra el estado de detección del robot, representado mediante iconos. El primero de ellos representa una cara con los ojos abiertos que indica que el sensor Kinect detecta al usuario. El segundo representa una cara con los ojos cerrados para indicar que el sensor Kinect no está detectando a ningún usuario. Por otro lado, la interfaz es capaz de detectar parámetros específicos del cuerpo en cada función, que son útiles para mover al robot, o en su caso, para permitir el seguimiento del usuario en la aplicación *¡Te sigo!*



3. Adaptación a la familiaridad con el sistema. El sistema debe tomar en cuenta la experiencia de uso para ajustar la interacción.

La interfaz de Ludibot está diseñada para que su uso resulte intuitivo, de tal manera que no sea indispensable la presencia de un asesor o profesor y de que los usuarios descubran sobre la marcha el funcionamiento del sistema y las posibilidades que éste les ofrece. En dicha interfaz se incluye la función *Aprende a controlarme*, gracias a la cual el usuario puede entrenarse y aprender los principales gestos basados en las manos. Se agrega una ventana con los gestos disponibles para ayudar al usuario a identificar la serie de gestos disponibles en un instante dado.

4. Adaptación de respuesta. El sistema debe ajustar su respuesta en función de diversos parámetros.

En Ludibot, la interfaz es capaz de identificar y adquirir parámetros del usuario, entre ellos los estados de las manos (abiertas y/o cerradas); la distancia entre el usuario y el robot; la medición de parámetros físicos propios de cada usuario. Esto permite definir estados y transiciones de la interfaz, otorgando al robot movimientos específicos, así como definir estados basados en una función propia para el robot, mediante el uso de la programación.

En resumen, Ludibot posee una serie de características que favorecen una interacción basada en la adaptación, empleando el reconocimiento de gestos específicos y la emisión de sonidos. Se busca aprovechar tanto la comunicación verbal como la no verbal, puesto que esta última –basada en movimientos físicos de cabeza, manos, brazos, cara u otras partes del cuerpo– constituye una forma natural de interacción humana y enriquece la interacción humano-máquina. En efecto, con el avance de la tecnología de sensores y procesadores, ahora es posible contar con sensores a bajo costo y de altas prestaciones, como Kinect, que permiten una implementación lograda de interacción gestual humano-máquina. Se hace así posible controlar máquinas y/o robots gracias a la captación y el procesamiento de los gestos humanos por parte de la máquina.

El reconocimiento de gestos puede ser visto aquí como un sistema adaptable de interacción humano-robot basado en una combinación de sensores, que permiten al robot

entender el significado del lenguaje corporal humano [113], construyendo así una comunicación específica que supera sistemas anteriores, como las interfaces basadas únicamente en texto.

En Ludibot, la adaptación se realiza primordialmente mediante el uso de la máquina de estados: a través de las acciones definidas en los estados y las transiciones se genera la interacción multimodal mediante la comunicación verbal y no verbal. Resulta fundamental entonces considerar los aspectos relativos a la adaptación del idioma. Para ello, resulta útil acercarse a la DLC, con el fin de identificar las variables que, dentro del proceso de enseñanza/aprendizaje de idiomas, son relevantes para la interacción.

## **1.2 Didáctica de lenguas y culturas**

La didáctica de lengua y culturas (DLC) siempre ha buscado la manera de aprovechar los avances científicos y tecnológicos y de implementarlos como herramientas de apoyo en el proceso de enseñanza/aprendizaje, para alcanzar diferentes objetivos, como son, entre otros:

- Complementar una clase mediante actividades y estrategias didácticas innovadoras.
- Diversificar los recursos de enseñanza/aprendizaje.
- Hacer más eficiente el proceso de aprendizaje.
- Motivar a los alumnos a aprender y, en tiempos recientes, también a aprender a aprender.
- Personalizar el tipo de aprendizaje, es decir, permitir que cada alumno aprenda a su propio ritmo y según sus características personales.

Para aprovechar de manera eficaz los avances en la DLC, es importante resaltar el carácter complejo del proceso de aprendizaje. Dicho proceso no sólo depende de la inteligencia y las aptitudes de cada estudiante, sino de otros factores como la motivación y la actitud, así como diversas variables afectivas y psicológicas [43]. El objetivo de este capítulo consiste en presentar los principales fundamentos de las teorías del aprendizaje aplicadas a la enseñanza de lenguas que respaldan esta investigación.

### **1.2.1 Teorías psicológicas del aprendizaje aplicadas a la enseñanza de lenguas**

Cabe recordar que la presente investigación tiene como objetivo el desarrollo de un robot que permita la implementación de aplicaciones lúdicas destinadas al aprendizaje de idiomas, respaldado por fundamentos pluridisciplinarios propicios tanto a un cambio de actitud por parte del aprendiz en situaciones formales y semiformales de aprendizaje como a un incremento de la motivación y a una modificación de ciertas representaciones y ciertas prácticas asociadas al aprendizaje de una lengua, del tal manera que se logre promover en el aula las competencias del siglo XXI. Para desarrollar de manera óptima un robot destinado a aplicaciones lúdicas de aprendizaje de idiomas, es primordial remitir a las teorías psicológicas del aprendizaje, sobre todo aquellas que han influido en el estado actual de la DLC y que pueden brindar información actualizada acerca de cómo se aprende una lengua. En efecto, la cuestión del aprendizaje ha sido objeto de una amplia reflexión en el campo de la psicología, desde donde ha surgido una gama de teorías sistemáticas basadas en evidencias resultantes de la aplicación del método científico.

Al estudiar cómo se desarrolla de manera concreta el aprendizaje de una lengua extranjera, es importante considerar la variabilidad del nivel verbal logrado por los diferentes integrantes de un mismo grupo de enseñanza/aprendizaje. En efecto, el proceso está sujeto a una serie de variables, como son la motivación personal, la motivación del profesor, el entorno en que nos desarrollamos, y en diversos casos, hasta el material que empleamos durante el aprendizaje. Cada uno de estos factores ha sido considerado para el desarrollo de Ludibot, desde la implementación hasta el desarrollo de las aplicaciones lúdicas.

Según Hilgard [81], una teoría del aprendizaje debe ser capaz de responder ciertas preguntas que una persona no especialista en psicología pueda tener sobre el aprendizaje. El autor menciona algunas interrogantes que sirven como guía para estimar la validez de la teoría en relación con el tipo de aprendizaje de interés:

1. Con respecto a la capacidad de aprender y a las diferencias individuales ¿cómo podemos aprender de manera fácil y rápida? ¿Quién logra aprender? ¿Qué limitantes hay en el aprendizaje? ¿Las limitantes están fijadas desde el nacimiento? ¿Todos tenemos las

- mismas capacidades de aprender? ¿De qué depende que alguien aprenda más que otro? ¿Importan factores como el entorno, el profesor, la escuela, el material de apoyo?
2. Con respecto a la práctica ¿es importante hacer uso de la práctica siempre? ¿El mejoramiento de la ejecución depende de la práctica? ¿Cuáles son las circunstancias más favorables para la práctica? ¿La práctica necesita de una teoría? ¿Las experiencias necesarias para aprender diferentes formas de comportamiento son similares?
  3. Con respecto al papel de la cognición ¿qué rol juega el conocimiento en las diversas formas del comportamiento? ¿Por qué aprendemos más fácilmente ciertas cosas cuando sabemos de qué se trata y otras cuando lo hacemos inconscientemente? Este aspecto es especialmente relevante para el aprendizaje verbal, ya que implica diversas variables basadas en la naturaleza del lenguaje humano y del código lingüístico.
  4. Con respecto al papel de las habilidades y de los conocimientos previos ¿aprender una cosa ayuda a aprender otra? ¿En qué circunstancias se lleva a cabo esta transferencia? ¿Cuál es la naturaleza de la transferencia? En el caso de la DLC, es importante analizar cuáles son las interferencias entre las lenguas conocidas por los alumnos.

En el marco del desarrollo de Ludibot, las preguntas anteriores ayudan a identificar las necesidades, a determinar los objetivos, a definir los contenidos de aprendizaje y a diseñar material acorde con los objetivos definidos en los programas de enseñanza y aprendizaje de una lengua. Cabe mencionar que dichas preguntas serán contestadas en la sección dedicada a la robótica ludoeducativa, ya que permiten analizar qué posturas serán contempladas para diseñar, implementar, evaluar y poner Ludibot a disposición de alumnos, profesores e investigadores en salones de clases, laboratorios y mediatecas.

Se ha mencionado con anterioridad que varias de las propuestas desarrolladas en la intersección de robótica, DLC y juego parten de fundamentos conductistas, mismos que expondremos brevemente más adelante. Ahora bien, el conductismo, que tuvo su auge durante la primera mitad del siglo XX, ha sido desplazado por el cognoscitivismo, el constructivismo y el interaccionismo. Estas tres tendencias, que son las que respaldan tanto las llamadas competencias del siglo XXI [6][207][208][71] como la DLC actual y el desarrollo de Ludibot, serán estudiadas a continuación.

## *Conductismo*

Las raíces del conductismo provienen de la psicología experimental cuando, en 1879, Wilhelm Wundt fundó un laboratorio donde estudió las funciones psicológicas desde las perspectivas fisiológica y biológica [43]. En 1913, John B. Watson fundó el conductismo [43], al publicar un artículo donde rechazaba conceptos mentalistas, como conciencia, sensación, voluntad e imagen, y los sustituía por el concepto de conducta, entendida como las actividades observables de un organismo.

Para el conductismo, los estímulos provocan una respuesta. La repetición de la relación entre estímulo y la respuesta implica que ésta se automatice; así, según lo afirma el conductismo, el individuo aprende una nueva conducta o hábito. Uno de los principales teóricos de esta corriente, Skinner [176], planteó que la conducta humana se manifiesta en dos tipos de respuesta:

- a) Las conductas respondientes, que son producidas por los organismos como una acción inmediata, respuesta automática a un estímulo específico. La conducta respondiente está formada por conexiones de estímulo-respuesta denominadas reflejos. El proceso de aprendizaje sigue la pauta del condicionamiento clásico, por lo que es posible adquirir nuevos reflejos o conductas respondientes en situaciones específicas.
- b) Las conductas operantes, emitidas de manera libre, es decir, sin ninguna estimulación específica, como caminar, correr, hablar, etc. La mayor parte de las conductas de los organismos coinciden con las conductas operantes.

Existen dos términos fundamentales en el condicionamiento operante, el concepto de contingencia y el concepto de estímulo reforzador. La contingencia remite a la relación temporal del estímulo con una respuesta de un evento dado y puede programarse explícitamente u ocurrir de modo accidental. Un evento reforzador [43] es contingente a otro cuando ocurre inmediatamente después de él, aunque tal ocurrencia no sea estrictamente obligatoria.

Un reforzador positivo es el estímulo que fortalece la ocurrencia de una respuesta operante, mientras que un reforzador negativo es todo estímulo que, al ser eliminado de una situación, debilita la probabilidad de una respuesta operante. De esta manera, para lograr un mejor aprendizaje, se considera preciso que el psicólogo, el educador o el psicoterapeuta diseñen, seleccionen y lleven a cabo programas de condicionamiento que consistan en varios patrones para la aplicación de reforzadores.

En el caso específico de las aplicaciones para el aprendizaje de una lengua existen trabajos desarrollados mediante Nao [74][86], un robot que ha sido objeto de muchos desarrollos y, entre otros usos, ha sido aplicado para aprender inglés. Sin embargo, en las experiencias llevadas a cabo con dicho robot, se suele emplear un enfoque conductista con resultados de aprendizaje limitados. En efecto, el aprendizaje mediante un programa de contingencias de refuerzos lleva a repetir palabras o series de palabras que no toman en cuenta perfiles individuales de aprendizaje ni situaciones específicas de comunicación.

Mientras que el conductismo se enfoca en reforzadores de respuesta y de repetición para automatizar la interacción, las corrientes metodológicas más recientes insisten sobre el papel activo del sujeto del aprendizaje. En esta tesis se busca que el usuario aprenda de manera efectiva, generando propio conocimiento a partir de la singularidad de una experiencia significativa. Sin dejar de lado la memorización y la repetición como estrategias, se amplía el contexto de la experimentación y se promueve una interacción de otra índole, según lo veremos más adelante.

En el marco del desarrollo de Ludibot, no es pertinente aplicar la teoría conductista, ya que el modelo de aprendizaje correspondiente tiene por objetivo controlar la conducta mediante reforzadores, reduciendo el contexto experimental y condicionando al aprendiente a realizar únicamente actividades definidas con anterioridad. El enfoque pluridisciplinar adoptado en este trabajo no se centra en las actividades observables de un organismo, un objeto o un fenómeno, factores externos del aprendizaje, sino que apuesta por una articulación entre dichos factores externos y el procesamiento interno que de ellos hace el sujeto de aprendizaje. Ése es el postulado central del cognoscitivismo, expuesto a continuación.

## *Cognoscitivismo*

El cognoscitivismo es una corriente teórica dominante en la psicología actual. Está basada en el estudio de los comportamientos humanos complejos y del comportamiento verbal. El cognoscitivismo considera que el ser humano es un ser activo que, con el fin de resolver problemas de aprendizaje, procesa, almacena y recupera la información que recibe del medio ambiente; así, el aprendizaje resulta de los procesos mentales internos del individuo.

El elemento central de esta teoría es la cognición, que se refiere a todos los procesos internos mediante los cuales una información sensorial es transformada, reducida, recuperada y utilizada. A diferencia del conductismo, el cognoscitivismo postula que las acciones del ser humano no están determinadas en sí por las propiedades objetivas de los fenómenos del medio ambiente, sino por la interpretación que el sujeto hace de ellas con base en sus estructuras de conocimiento general, sus expectativas y sus motivaciones.

Esta corriente plantea que la mente es un agente activo responsable de que el conocimiento sea activamente adquirido y no implantado por un agente exterior. Por esta razón, el aprendizaje cognoscitivo es consciente y requiere una participación mental activa de parte del individuo. La cognición es un proceso o conjunto de procesos; es mental y se considera subyacente al comportamiento; es intencional, ya que tiene un propósito; y es voluntaria, ya que está bajo el control del individuo. Entre las aportaciones más destacadas del cognoscitivismo contemporáneo están la teoría de la Gestalt y la teoría de Jean Piaget.

Si bien las propuestas explicativas del cognitivismo actual no son homogéneas [43], tienen en común el hecho de colocar los fenómenos mentales como los agentes causales del comportamiento humano. Además, consideran la existencia de procesos internos mediadores entre la información y/o problema recibido por el sujeto, y la respuesta y/o solución emitida por el mismo sujeto. Diversos autores consideran que las teorías del cognoscitivismo han adquirido mayor legitimidad gracias al auge de la teoría de la información y al surgimiento de las computadoras. Esta tesis parte del cognoscitivismo contemporáneo, desde el enfoque del procesamiento de la información basado en dos analogías [117]:

1. La analogía humano-máquina, en la que el ser humano puede ser comparado con una máquina o computadora compleja.
2. La analogía pensamiento-programa, en la que los procesos de pensamiento utilizados por el ser humano para resolver un problema pueden ser comparados con un programa de computadora.

En el desarrollo de Ludibot, tomando en cuenta la postura del cognoscitivismo, se define al ser humano como un sujeto activo que procesa la información, según un modelo del aprendizaje basado en las analogías mencionadas. El uso de análisis jerárquico para la estructuración, la organización y la secuencia de la información facilita su óptimo procesamiento y crea ambientes de aprendizaje estimulantes, donde los estudiantes pueden establecer conexiones con elementos previamente aprendidos.

El cognoscitivismo ha sido considerado en el desarrollo de Ludibot, ya que toma en cuenta el enfoque un modelo computacional, donde el ser humano es un ser activo que interactúa de manera consciente y voluntaria con el robot. Asimismo, en el desarrollo de las aplicaciones lúdicas se ha tomado en cuenta la existencia de procesos que constan de operaciones básicas para almacenar, recuperar, codificar, comparar y transitar información. La analogía humano-máquina respalda el diseño interactivo de la interfaz humano-robot móvil, desde la planificación de cada una de las tareas del robot hasta la selección de los procesos sensoriales mediante el uso de Kinect v2. En la interfaz de Ludibot, la analogía pensamiento-programa remite a un modelo de máquina de estado finito. Al brindar al usuario la posibilidad de elegir entre diversas funciones, permite realizar transiciones usando como entrada gestos de las manos. Una de las corrientes teóricas a las que ha dado lugar el cognoscitivismo es el constructivismo. Toca ahora el turno de este otro paradigma esencial en la investigación de la educación [30] en general y de la DLC en particular.

### ***Constructivismo***

Como todas las corrientes derivadas del cognoscitivismo, el constructivismo postula que el proceso de aprendizaje no depende exclusivamente de factores externos, sino que obedece al



procesamiento interno que de la información de aprendizaje hace el sujeto. Ahora bien, según el constructivismo [30], el aprendizaje depende de la cantidad y la calidad de las estructuras mentales de organización de una persona. A través de la psicopedagogía, el constructivismo nos ayuda a saber cómo aprenden los alumnos y cómo constituyen los conocimientos.

Desde el punto de vista constructivista, el aprendizaje es definido como un proceso de construcción interno, activo e individual [203]. Para Piaget, el mecanismo básico de aprendizaje de conocimientos consiste en un proceso en el que nuevas informaciones se incorporan a los esquemas o estructuras preexistentes en la mente de las personas, que se modifican, adaptan y reorganizan según un mecanismo de asimilación y acomodación facilitado por la actividad del alumno.

Desde este enfoque, aprender está relacionado con comprender, y para ello es una condición indispensable tener en cuenta lo que el alumno ya sabe sobre aquello que se le quiere enseñar. Por lo tanto, se propone diseñar lo que se llama organizadores previos. En este modelo, el profesor es un guía, un tutor, un facilitador de aprendizajes, capaz de generar en un salón de clases un ambiente de aprendizaje.

Vygotsky [203] subraya la interrelación entre ambiente y sujeto: el conocimiento no es sólo un producto del ambiente ni sólo un resultado de disposiciones internas. La interacción entre, por un lado, los conocimientos previos y la representación que se tenga de la información o de la actividad o tarea por resolver y, por el otro, la actividad externa o interna que el aprendiente realice al respecto permite una construcción propia en constante evolución. Algunos principios educativos esenciales que están asociados con la concepción constructivista del aprendizaje y de la enseñanza, y que fueron tomados en cuenta para el desarrollo de Ludibot, son los siguientes [195]:

1. El aprendizaje implica un proceso constructivo interno, y es considerado subjetivo y personal.
2. El aprendizaje se facilita gracias a la mediación o interacción con los otros, es social y cooperativo.

3. El aprendizaje es un proceso de construcción de saberes culturales.
4. El grado de aprendizaje depende del nivel de desarrollo cognitivo, emocional, social y de la naturaleza del conocimiento.
5. El punto de partida de todo aprendizaje son los conocimientos y las experiencias previas que tiene el alumno.
6. El aprendizaje implica un proceso de organización interno basado en esquemas.
7. El aprendizaje se produce cuando entra en conflicto lo que el alumno ya sabe con lo que debería saber.
8. El aprendizaje tiene un papel afectivo y tiene relación con el autoconocimiento, los motivos y metas personales, la disposición de aprender, las atribuciones de éxito y de fracaso, las expectativas y las representaciones mutuas.
9. El aprendizaje se facilita si existen apoyos que conducen a la construcción de elementos cognitivos entre lo nuevo y lo aprendido, y con los materiales de aprendizaje.

Los principios anteriores ayudan a optimizar la práctica didáctica. Según Freed [57], educar no es simplemente transferir conocimientos sino crear las condiciones para su construcción por los aprendientes. De esta manera, el educando tiene que involucrarse en el proceso de construir nuevos conocimientos en asociación con los conocimientos previos.

Las aplicaciones ludoeducativas desarrolladas y por desarrollar con Ludibot están destinadas al aprendizaje dirigido o semiautónomo del idioma francés. El juego de *De la tête aux pieds* diseñado para Ludibot y presentado en el capítulo 3 de esta tesis tiene por objetivo lingüístico la sensibilización a, la conceptualización y/o la sistematización del vocabulario básico relativo a las partes del cuerpo humano, así como los elementos gramaticales y fonéticos correspondientes. Los usuarios del juego pueden ser tanto aprendientes que se acercan por primera vez a ese vocabulario, como aprendientes que buscan organizar o repasar un conocimiento léxico previo. En ambos casos, se busca que los usuarios apliquen de manera novedosa su conocimiento previo relativo a las partes del cuerpo, de tal manera que se facilite la construcción de elementos cognitivos nuevos.

El proceso constructivo interno, considerado subjetivo y personal, es apuntalado mediante la interacción con los otros, de tal manera que cobra una dimensión social y cooperativa. Además, la interacción con un robot, por su carácter todavía inusual, puede incidir en la percepción afectiva del aprendizaje, aumentar la disposición de aprender, modificar las atribuciones de éxito y de fracaso, así como transformar las expectativas y las representaciones.

En suma, y considerando lo que dice Vygotsky en [203], mediante el juego *De la tête aux pieds* se busca que los usuarios desarrollen conceptualizaciones mediante el estímulo de la curiosidad científica, la indagación y la experimentación.

Ludibot busca que los usuarios desarrollen capacidades y habilidades que les permitan abordar problemas cotidianos relacionados con el uso adecuado de la tecnología. Específicamente, se busca que las aplicaciones lúdicas desarrolladas permitan construir el conocimiento mediante nuevas dinámicas en situaciones formales y semiformales, además de promover nuevas motivaciones y actitudes que contribuyan a mejorar las representaciones y las prácticas asociadas al aprendizaje de una lengua. Sin embargo, es esencial considerar el papel que tiene el entorno. Por esa razón, será necesario considerar también el interaccionismo en el desarrollo de Ludibot.

### ***Interaccionismo***

Para el interaccionismo, el entorno y la persona son dos elementos que se complementan mutuamente. La persona es el principal elemento, considerado activo, y no un mero reflejo de su ambiente. Según Bohannon y Warren-Lenbercker [43], los interaccionistas asumen que diversos factores afectan el desarrollo de las actividades humanas, entre ellos, los sociales, los cognoscitivos, los biológicos y los culturales. Si bien existen diversos enfoques interaccionistas, en este texto se abordará únicamente el interaccionismo social, por ser una aproximación teórica de importancia en la investigación actual sobre el aprendizaje de una lengua.

El enfoque interaccionista social combina diversos aspectos de las posiciones tradicionales conductistas e innatistas. Concuere con el innatismo de tipo lingüístico postulado

por Chomsky, según el cual el lenguaje sigue ciertas reglas que lo hacen ser único frente a los otros tipos de comportamientos humanos. Por otro lado, comparte con los conductistas el énfasis que otorga al papel del ambiente como lugar de producción de la estructura del lenguaje.

El interaccionismo social [43] considera que la estructura del lenguaje humano surge de las funciones comunicativo-sociales que el lenguaje tiene en las relaciones humanas, y plantea que una estructura más madura y experta permitirá formas socialmente variadas y complejas en la comunicación social. Además, para el interaccionismo hay dos consideraciones esenciales, el *input*, que es la información externa de entrada, y *el output*, definido como la información de salida durante la interacción en el proceso de adquisición de una lengua.

Con miras a la adquisición de una lengua, el interaccionismo sugiere cómo debe ofrecerse la experiencia lingüística adecuada para el desarrollo y el aprendizaje de una nueva lengua. Afirma que, para que el individuo aprenda una lengua y se desarrolle de manera adecuada en un ambiente lingüístico, es necesario que desempeñe un papel activo para tener una comunicación social eficiente. Ello derivará en un desarrollo y un progreso efectivo de las habilidades lingüísticas durante el proceso de aprendizaje. De este modo, el aprendizaje de una lengua y el desarrollo en un ambiente lingüístico son vistos como un sistema interactivo.

Para desarrollar el sistema interactivo arriba mencionado, es importante en esta postura reconocer el papel del ambiente. Pese a la importancia dada al ambiente, el interaccionismo difiere del conductismo: en el primero, el aprendiente no es visto como un receptor pasivo de las técnicas de enseñanza, como es el caso de muchas aplicaciones actuales con robots. Además, se aparta del innatismo, porque éste considera al aprendiente como un procesador innato especializado en el lenguaje y la lengua, y como un procesador casi independiente del medio cuya función es explotar la gramática universal.

Para los interaccionistas, las funciones del lenguaje en la comunicación social son de extrema importancia en todos los momentos del desarrollo, pues los seres humanos estamos fisiológicamente especializados para ser usuarios de una lengua, y porque las habilidades lingüísticas requieren de la madurez para manifestarse. De esta manera, aceptan que la

maduración es crítica y que en ocasiones los aprendientes no pueden adquirir un lenguaje hasta que no hayan alcanzado cierto nivel de desarrollo cognoscitivo. A la vez, definen al ambiente como un sistema social interactivo, entendido como el lugar en el cual se debe investigar el surgimiento del lenguaje humano, elemento esencial para el interaccionismo.

Este modelo ayuda a plantear que para lograr la competencia en una lengua no bastan el condicionamiento y la imitación postuladas por el modelo conductista. Las ciencias del lenguaje han demostrado que para “saber” una lengua es necesario “saber hacer” en esa lengua. Es decir, el conocimiento meramente declarativo tiene que ir asociado con un conocimiento procedimental. Este segundo tipo de conocimiento se puede lograr mediante la dinámica de la interacción.

La asociación entre saberes declarativos y procedimentales para una interacción social fue tomada en cuenta en el desarrollo de Ludibot, así como en las características específicas de la interacción humano-máquina elegida. La interfaz de Ludibot está implementada de manera que el usuario puede seleccionar diversas tareas para el robot móvil, con el propósito de fomentar la autonomía. Se incluye asimismo un menú que permite al usuario descubrir de manera independiente el funcionamiento del robot móvil, elegir su nivel de dificultad y descubrir y/o practicar el vocabulario mediante juegos.

El interaccionismo está igualmente presente en el juego *De la tête aux pieds* en lo que atañe al desfase deliberado entre el *input* y el *output*. La regla del juego es la siguiente: cuando el robot enuncia aleatoriamente, una por una, diferentes partes del cuerpo, el jugador debe tocar la parte del cuerpo anunciada, pero no la recién mencionada sino la anterior. En otras palabras, al iniciar el juego, Ludibot enuncia la parte A y el jugador debe permanecer inmóvil. Cuando el robot enuncia la parte B, el jugador toca la parte A. Cuando el robot enuncia la parte C, el jugador toca la parte B, y así sucesivamente. Por ende, no se trata de una actividad durante la cual se siguen al pie de la letra las instrucciones dadas por el robot, sino que el desfase deliberado entre instrucción dada e instrucción realizada exige concentración y memorización a corto plazo. Se trata de una tarea durante la cual el usuario procesa, almacena y recupera el *input* para producir un *output* de manera instantánea.

En suma, Ludibot pretende ser una herramienta que complemente el aprendizaje de una lengua con base en las teorías cognoscitivista, constructivista e interaccionista del aprendizaje. Se aspira a que el aprendiente pueda construir el conocimiento declarativo y procedimental relativo a una lengua, recurriendo a conocimientos previos y activando procesos mentales mediante nuevas dinámicas y aplicaciones lúdicas. Además, se considera al aprendiente como un agente activo de la cognición. Finalmente, se busca propiciar la interacción con el medio ambiente, el usuario y/u otros usuarios mediante el uso de aplicaciones lúdicas.

Una vez estudiadas las teorías en el aprendizaje aplicadas a la enseñanza de una lengua que inciden directamente en el desarrollo de Ludibot y de sus aplicaciones lúdicas, es importante examinar algunas otras nociones clave de la DLC, abordadas a continuación.

### **1.2.2 Nociones clave de la didáctica de lenguas y culturas**

Según Martínez [115], la didáctica es un conjunto de medios, técnicas y procedimientos que concurren en pro de la apropiación por un sujeto dado de elementos nuevos que, en el caso de la didáctica de lenguas, son saberes lingüísticos, saberes procedimentales comunicativos y, finalmente, comportamientos y valores culturales indisociables de la lengua. La DLC, nacida en la intersección de otras disciplinas teóricas (pedagogía, lingüística, ciencias de la educación, cinestésica, psicología, sociología, política, proxémica, ciencias cognitivas, tecnología educativa, pragmática y docimología, entre muchas otras), maneja un abanico muy amplio de conceptos.

Se abordará aquí con más detalle los principales conceptos de la DLC que intervinieron en el desarrollo del proyecto Ludibot, a saber, la diferencia entre lengua y lenguaje, la interacción, la comunicación verbal, las competencias comunicativas, la comunicación no verbal, la aptitud, la inteligencia y las variables afectivas.

#### ***Diferencia entre lengua y lenguaje***

Para muchos hablantes, “lengua” y “lenguaje” son sinónimos, pues ambos remiten a la comunicación, entendida como la transmisión de información. En ciertos idiomas, como el inglés, existe incluso una sola palabra para designarlos: *language*. Ahora bien, en español, en francés y

en muchas otras lenguas, los conceptos de lengua y lenguaje difieren, por lo que es importante aquí explicar la distinción entre ambos, partiendo de lo que al respecto exponen tanto Da Silva y Signoret [43] como Richards [151][152].

En el campo de la lingüística, el término “lengua” es empleado para designar un sistema de comunicación o sistema de signos vocales común a los miembros de una colectividad o región, quienes constituyen una comunidad lingüística [43]. De este modo, el español, el inglés, el francés, el chino, el árabe, el italiano, pero también el esperanto, el náhuatl, el maya, el otomí, etc., son lenguas (también conocidas como idiomas).

Las lenguas pueden ser clasificadas como lenguas vivas o lenguas muertas: las primeras son aquellas que actualmente son empleadas de manera espontánea en diversas comunidades del mundo, en la comunicación tanto oral como escrita; las segundas, son aquellas que ya no poseen hablantes nativos (tal es el caso del latín, del griego antiguo, del sánscrito, entre otras).

La lengua es un fenómeno complejo que se mantiene relativamente estable a través del tiempo pero que cambia y se adapta según la época, el contexto, las necesidades de sus hablantes y los acontecimientos. En esta constante evolución, incorpora formas de expresión nuevas al tiempo que desecha otras. En todos los casos, la lengua es una convención propia de un grupo humano dado, que genera un sentimiento de pertenencia y que es difícilmente separable de la dimensión cultural. De ahí que hoy en día la disciplina anteriormente conocida como didáctica de lenguas reciba más comúnmente el nombre de didáctica de lenguas y culturas.

El término “lenguaje” designa por su parte la facultad innata para comunicar, es decir, la capacidad para expresar pensamientos y sentimientos por medio de la palabra u otro tipo de signos (gestos; símbolos, como en la navegación naval; colores, como en la indumentaria tradicional). Se suele considerar que la facultad de lenguaje es propia del ser humano, aunque también se reconoce que hay otros seres vivos que intercambian información: las abejas, los delfines, etc.

Es indispensable señalar que, en el campo de las ingenierías, suele llamarse “lenguaje de programación” al conjunto de herramientas destinadas a lograr la comunicación entre el usuario y la computadora, permitiendo realizar tareas a través una serie de caracteres que son interpretados por la computadora mediante un *software*. Los lenguajes de programación poseen una estructura y un significado en conjunto con órdenes o comandos que describen una función o algoritmo. Cada lenguaje de programación tiene sus propias instrucciones, que se combinan para formar un programa de cómputo, y permiten construir y diseñar aplicaciones, juegos, procesos o tareas en un sistema dado. Estrictamente hablando, un lenguaje de programación, al ser una convención formal o artificial, pertenece a la categoría de lengua.

En suma, según lo explica Martínez [115], mientras que el lenguaje remite a la facultad innata que, como instrumento de comunicación humana, está ligado a una capacidad del individuo e implica la existencia de un dispositivo neurofisiológico, la lengua es una institución colectiva y un producto social, manifestación concreta de la capacidad de lenguaje; la lengua es un sistema de signos articulados que forman un código admitido por todos.

Durante mucho tiempo, la enseñanza de lenguas estuvo enfocada al aprendizaje de una lengua, mientras que los elementos del lenguaje se daban por adquiridos. Hoy en día, la DLC admite que para una comunicación y una interacción efectivas se requiere manejar tanto la o las lenguas en presencia como las herramientas del lenguaje a nuestra disposición, tales como la comunicación no verbal y las acciones paralingüísticas [85].

Aparece aquí otra oposición importante entre dos nociones que también llegan a ser utilizadas como sinónimo en el habla cotidiana pero que el discurso científico distingue claramente: adquisición y aprendizaje [43]. En la DLC actual, inspirada en las teorías de Krashen [99], se llama “adquisición” de una lengua al proceso “natural” de apropiación, similar a aquel que interviene cuando un niño desarrolla su primera lengua o lengua nativa. El individuo adquiere la lengua de manera intuitiva e inconsciente, sin poner especial atención a las formas lingüísticas, gramaticales o estructurales. En contraste, el aprendizaje es definido como un proceso formal y serio de asimilación de la lengua, durante el cual se modifican y se desarrollan



experiencias, razonamientos, habilidades, destrezas, conductas y valores como resultado de una práctica formal.

Tanto la oposición lengua/lenguaje como la oposición adquisición/aprendizaje resultan relevantes en el marco de este proyecto. En efecto, Ludibot se apoya por ahora en dos lenguas, el español y el francés, aunque a futuro se tiene contemplado que pueda funcionar en cualquier lengua viva o muerta. El español es usado en los textos de la interfaz humano-máquina, tanto en la barra de estados que indica la función seleccionada como en la ventana de los gestos disponibles. El francés es empleado en el juego *De la tête aux pieds*, para los audios gracias a los cuales el robot enuncia aleatoriamente diversas partes del cuerpo.

Por otro lado, en el desarrollo de Ludibot, se requirió hacer uso de varios lenguajes de programación. Para la segunda versión de Ludibot, se propuso un lenguaje propio para programar una máquina de estados, en donde además de la definición de estados, transiciones y gestos (o entradas) se incluyen acciones (visuales, cinésicas, sonoras, etc.) que permiten implementar modalidades de interacción multimodal para el desarrollo la interfaz humano-máquina y permiten diseñar y construir tareas específicas en Ludibot.

Ludibot aprovecha el uso del lenguaje de gestos, que son definidos en la implementación mediante el uso de Kinect v2. Los gestos determinados están dados con posiciones mediante el uso de las manos (arriba, abajo, cerradas y/o abiertas) que permiten definir estados y transiciones para el comportamiento del robot.

Recordemos que ya existen robots desarrollados para el aprendizaje de una lengua, como en la propuesta de Steels [180][181] que incluye también un juego o bien y el robot Nao [86] [148] aprovechado en varios contextos para reforzar el aprendizaje del inglés. Sin embargo, los robots mencionados solamente emplean el lenguaje de programación en el desarrollo de las aplicaciones presentadas y, en lo que atañe al uso de la lengua, ofrecen un desempeño limitado a la percepción y la adquisición de voces para identificar colores y palabras en inglés. La interacción humano-máquina prevé una participación del usuario que se reduce a la repetición del *input* lingüístico. Se deja entonces de lado todo el potencial del lenguaje. Por lo contrario,

Ludibot combina lengua, lenguaje de programación y ademanes, que forman parte del lenguaje, para lograr la interacción humano-máquina. En efecto, se considera que el aprendizaje óptimo de una lengua no radica únicamente en el manejo de los elementos lingüísticos del código correspondiente, sino que requiere desarrollar la capacidad de usar dicha lengua de manera adecuada para comunicar, interactuar y actuar en todos los ámbitos sociales, tal como se explicará con mayor detalle más adelante.

En lo que atañe a la adquisición y al aprendizaje, Ludibot está diseñado para ser usado en situaciones formales, como lo es la enseñanza en el aula, entre otras, y semiformales, como lo son el uso en mediateca o centro de autoacceso. Por ello, combina el aprendizaje dirigido, bajo la guía del profesor, con la posibilidad de adquirir la lengua de manera más autónoma y, por ende, más cercana a las condiciones consideradas naturales: en las mediatecas y en los centros de autoacceso, los alumnos ya no son guiados por un profesor, sino que eligen por sí mismos los temas, los materiales, las actividades, la duración de la sesión, entre otros parámetros de la situación, con la ayuda eventual de un asesor. No se trata pues de situaciones espontáneas o improvisadas, ni tampoco de situaciones donde el aprendiente es 100% autónomo en sus decisiones, pero sí crece el grado de participación de dicho aprendiente en la toma de decisiones relativas a las condiciones en las que se desarrolla su proceso de aprendizaje.

En la mayoría de las situaciones formales y semiformales de aprendizaje de lenguas extranjeras, suele excluirse o minimizarse la dimensión del comportamiento no verbal y de las acciones paralingüísticas; a través de una propuesta como Ludibot, se intenta reintroducir en la interacción ademanes y gestos, para tomar mejor en cuenta el papel que desempeña el cuerpo en el uso adecuado del lenguaje. Dado que en la implementación de la interfaz humano-máquina de Ludibot se recurre entonces tanto a la lengua como al lenguaje para lograr la interacción, es necesario aquí ampliar el concepto de interacción, previamente discutido en este trabajo a propósito de la robótica, tomando en cuenta los aportes de la DLC.

### ***Interacción desde la didáctica de lenguas y culturas***

Una vez definidas tanto la diferencia entre lengua y lenguaje como la diferencia entre aprendizaje y adquisición, es primordial definir la interacción desde la DLC y, específicamente, la interacción

verbal. Ésta consiste en un tipo de interacción realizada por dos o más participantes que se comunican mutuamente, en un intercambio de acciones verbales y no verbales. Una conversación cotidiana, una reunión de trabajo, una clase y/o un saludo, son ejemplos de interacciones verbales [43]. En efecto, el uso de la lengua se da en el marco de las relaciones sociales, generalmente con el objetivo de comunicar o intercambiar ideas. En cualquier acto de interacción se usan signos verbales y no verbales, y elementos correlativos, como son los gestos, la mímica, los sonidos expresivos, siempre y cuando expresen o comuniquen algo [193][205].

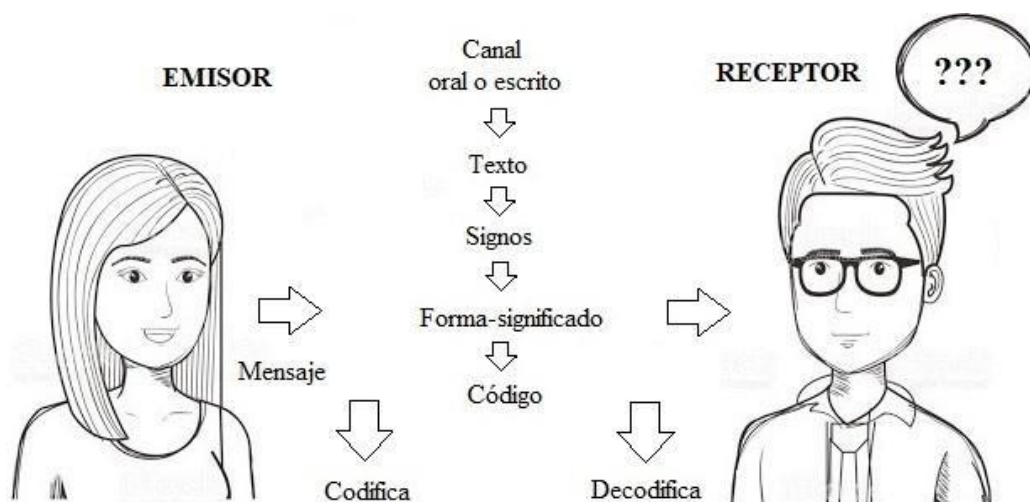
Durante la utilización de Ludibot aparecen dos tipos de interacción verbal. Está, por un lado, aquella que se da entre el jugador y el robot mediante la interfaz humano-máquina, basada en el uso del sensor Kinect v2: el robot emite mensajes verbales escritos y orales, mientras que los usuarios recurren a ademanes o signos no verbales. Los signos no verbales están también presentes en el uso de iconos para simbolizar el estado de detección del sensor y el estado de la batería, así como en los sonidos que indican los aciertos, los errores y el final del juego. Desde luego, se trata de una interacción verbal un tanto peculiar, pues si bien cumple con la definición, uno de los participantes en el intercambio es una máquina. Ahora bien, el uso de Ludibot promueve también una interacción verbal humano-humano, pues la plataforma ha sido diseñada para un juego colectivo, durante el cual los participantes pueden hacer uso de los recursos lingüísticos y del lenguaje para comentar la partida, expresar emociones, hablar de su experiencia lúdica, compartir estrategias, etc. Para ello, recurrirán no sólo a la lengua (francés y/o español) sino que emplearán los recursos no verbales y paralingüísticos a su disposición. Para comprender mejor la articulación entre lengua y lenguaje durante el proceso de interacción verbal, es primordial tener claridad en torno al papel de la comunicación verbal y no verbal en dicho proceso.

### ***Comunicación verbal y no verbal***

La comunicación verbal es aquella donde el mensaje es verbalizado, es decir, aquella en la que se recurre a la lengua pues se emplean palabras, ya sea de manera oral y/o escrita. En la comunicación verbal, según el esquema clásico de Jakobson [91], intervienen seis elementos fundamentales: a) un *remitente* o emisor que envía b) un *mensaje* a c) un *destinatario* o receptor a

través de d) un *canal*. El mensaje es construido mediante e) un *código* compartido, gracias al cual el emisor y el receptor se refieren a f), un *contexto*.

La figura 1.4 ilustra el acto de comunicación verbal, en el cual se establece un vínculo vivo entre por lo menos dos personas mediante el intercambio de mensajes verbalizados. Idealmente, el interlocutor que habla se vale de un sistema de signos articulados para representar todos los detalles de los significados del mensaje que se desea transmitir. Por otro lado, el interlocutor que recibe el mensaje debe compartir la misma capacidad del uso del código, así como ciertas experiencias y ciertos conceptos, para comprender el mensaje.



**Figura 1.4. Esquema del acto de comunicación verbal humana y sus componentes básicos**

Es importante mencionar que durante la comunicación verbal se pueden utilizar otros canales y códigos, propios de la comunicación no verbal. Si bien la comunicación verbal suele permitir una eficiencia mayor, pues las lenguas son sistemas organizados preestablecidos para transmitir información, la comunicación no verbal desempeña un papel importante en el éxito o el fracaso de la interacción.

En efecto, la comunicación no verbal [85] es un proceso de comunicación mediante el cual se transmite un mensaje mediante el uso de gestos y/o acciones, que sirven para señalar, demostrar o dar mayor claridad a una acción. Así, los movimientos de las manos, la dirección de la mirada o el grado de inclinación de la cabeza, por dar tan sólo un ejemplo, pueden reforzar o

incluso contradecir el sentido de un mensaje verbal durante la interacción. La asociación entre comunicación verbal y no verbal es particularmente visible en el caso de la deixis. Se entiende por deixis la función de algunos elementos lingüísticos que permite señalar una persona, un lugar o un tiempo exteriores al discurso: “aquí”, “hoy”, “ayer”, “éste”, “aquél”. Durante una interacción verbal presencial, la frase “yo tomo esto de aquí y lo pongo allá” es difícil de entender sin recurrir a la comunicación no verbal: el enunciado no puede interpretarse adecuadamente a menos de que se perciba la acción.

Ahora bien, además de tales gestos y acciones, la comunicación no verbal incluye acciones paralingüísticas [85]. Éstas se distinguen de los gestos y las acciones arriba mencionados por el hecho de que sus significados se han hecho convencionales y pueden variar de una cultura a otra. Incluyen el lenguaje corporal (gestos, expresiones faciales, posturas, contacto visual, contacto corporal y proxémica, siempre y cuando se usen según códigos preestablecidos), sonidos extralingüísticos en el habla (por ejemplo, “ay” para expresar dolor) y cualidades prosódicas (cualidad de voz, tono, volumen, duración).

Finalmente, caben también dentro de la comunicación no verbal las características paratextuales: ilustraciones (fotografías, dibujos, etc.); gráficos, tablas, diagramas, figuras, etc.; y características tipográficas (tipos de letras, intensidad, espaciado, subrayado, diseño).

En el caso de Ludibot, se emplea la comunicación verbal mediante los textos de la interfaz y los audios en el juego *De la tête aux pieds*. También se usa la comunicación no verbal, mediante ademanes y gestos que definen estados y transiciones. Éstos permiten al aprendiente transitar de un estado a otro, mediante el uso del sensor Kinect v2. Por su parte, la interfaz del robot se apoya en acciones paralingüísticas, gracias a tres elementos: gestos para definir estados preestablecidos; posturas, dadas por el diagrama esquemático que representa el cuerpo y la detección; y proxémica, presente en la distancia de detección del usuario.

La comunicación no verbal está asimismo presente mediante una serie de sonidos específicos que le permiten al usuario reconocer errores y aciertos e identificar cuándo ha ganado o perdido el juego. Las características paratextuales están presentes en la interfaz humano-máquina mediante ilustraciones para representar cada estado y función. Los gráficos, las figuras y

los diagramas ayudan a determinar el estado de la batería, el estado de detección y el diagrama representativo del cuerpo humano. Las características tipográficas están dadas por el uso de diversos tipos de letra, el subrayado, el espaciado, el diseño y el marcado en negritas en los diversos textos de cada función, transición y estado de detección de Kinect v2. Cada uno de estos elementos es representado mediante figuras e iconos implementados con caracteres específicos.

En la comunicación verbal y no verbal están implícitas las intenciones comunicativas. Los usuarios de una lengua ejercen competencias generales en conjunto con competencias comunicativas relacionadas con la lengua. Las competencias generales y las competencias comunicativas son expuestas a continuación.

### ***Competencias generales***

Con el fin de realizar las tareas y las actividades que se requieren para abordar las situaciones comunicativas, los usuarios utilizan diversas competencias desarrolladas en su experiencia previa. La participación en acontecimientos comunicativos da como resultado un mayor desarrollo de las competencias, tanto en el uso inmediato como en el uso a largo plazo, denotadas como competencias generales [85].

La capacidad comunicativa del usuario contribuye de una u otra manera a desarrollar competencias generales que, a su vez, contribuyen a mejorar la capacidad comunicativa del aprendiente, en un proceso iterativo. Las competencias generales son aquellas que ayudan a construir el conocimiento declarativo (saber) para generar destrezas y habilidades (saber hacer), conduciendo a tener una competencia existencial (saber ser) y una capacidad de aprender (saber aprender). Cada uno de estos tipos de conocimiento es explicado a continuación.

#### ***Conocimiento declarativo (saber)***

Las características fundamentales del conocimiento declarativo se desarrollan totalmente durante la primera infancia, pero el modelo continúa desarrollándose por medio de la educación y de la experiencia durante la adolescencia y también durante la vida adulta. La pregunta “¿qué es eso?” puede referirse al nombre de un fenómeno observado por primera vez o al significado (referente)

de una palabra nueva. El conocimiento declarativo, según el *Marco común europeo de referencia para las lenguas* (MCER) [85], está constituido a su vez por el conocimiento sociocultural y la consciencia intercultural.

El conocimiento sociocultural es un aspecto del conocimiento del mundo: remite a la sociedad y a la cultura de la comunidad o comunidades en las que se habla el idioma. Este conocimiento merece la atención del aprendiente pues es probable que algunos de sus elementos no se encuentren en la experiencia previa del individuo, y puede que estén distorsionados por los estereotipos. El conocimiento sociocultural puede darse respecto a las características distintivas de una sociedad en específico y de su cultura: la forma de vida, las condiciones de vida, las relaciones personales, los valores, las creencias, las actitudes, el lenguaje corporal, las convenciones sociales y el comportamiento ritual.

La consciencia intercultural relaciona el conocimiento, la percepción y la comprensión del mundo de origen con los del mundo de la comunidad objeto de estudio (similitudes y diferencias distintivas). Tales aspectos producen una consciencia intercultural. Esta conciencia incluye la diversidad regional y social, que se enriquece con la conciencia de una serie más amplia de culturas respecto de las que conllevan la lengua materna y la segunda lengua, lo cual contribuye a ubicar ambas en su contexto.

### *Destrezas y habilidades (saber hacer)*

Las destrezas y las habilidades, que remiten al saber hacer, están clasificadas según dos grandes tipos: destrezas y habilidades prácticas, destrezas y habilidades interculturales [85]. A su vez, las destrezas y las habilidades prácticas están integradas por las destrezas sociales, las destrezas de la vida, las destrezas profesionales y las destrezas de ocio.

Las destrezas sociales son definidas como la capacidad de actuar de acuerdo con los tipos de las características distintivas de una sociedad y de su cultura y de comportarse de la forma esperada, siempre que los foráneos, y en especial los extranjeros, lo consideren apropiado. Las destrezas de la vida remiten a la capacidad de llevar a cabo con eficacia las acciones rutinarias que se requieren en la vida diaria (bañarse, vestirse, pasear, cocinar, comer, etc.).

Las destrezas profesionales están conceptualizadas como la capacidad de realizar acciones especializadas (mentales y físicas) que se necesitan para realizar los deberes del (auto)empleo. Finalmente, las destrezas de ocio son dadas como la capacidad de realizar con eficacia las acciones necesarias para las actividades de ocio o entretenimiento (artes, trabajos manuales, deportes, aficiones, entre otras).

Mientras tanto, las destrezas y las habilidades interculturales incluyen la capacidad de relacionar entre sí la cultura de origen y la cultura extranjera; la sensibilidad cultural y la capacidad de identificar y utilizar una variedad de estrategias para establecer contacto con personas de otras culturas; la capacidad de cumplir el papel de intermediario cultural entre la cultura propia y la cultura extranjera,; la capacidad de abordar con eficacia los malentendidos interculturales y las situaciones conflictivas; y, finalmente, la capacidad de superar relaciones estereotipadas.

#### *Competencia existencial (saber ser)*

La actividad comunicativa de los usuarios no sólo se ve afectada por sus conocimientos, su comprensión y sus destrezas [85], sino también por factores individuales relacionados con la personalidad de cada individuo, tales como las actitudes, las motivaciones, los valores, las creencias, los estilos cognitivos y los tipos de personalidad que contribuyen a la identidad personal.

Las actitudes se refieren tanto al grado de apertura hacia nuevas experiencias, personas, ideas, pueblos, sociedades y culturas, como al interés hacia esos elementos. Incluyen también la voluntad de relativizar la propia perspectiva cultural y el propio sistema de valores culturales, al igual que la voluntad y la capacidad de distanciarse de las actitudes convencionales en cuanto a la diferencia cultural.

Las motivaciones tomadas en cuenta en [85] son cinco: intrínsecas y extrínsecas; instrumentales e integradoras; aquellas ligadas al impulso comunicativo mismo. Los valores considerados son los éticos y morales. Las creencias pueden ser religiosas, ideológicas y filosóficas. Los estilos cognitivos pueden ser el convergente o el divergente; el holístico, el



analítico o el sintético. Finalmente, los factores de personalidad pueden estar constituidos por el espíritu emprendedor, la indecisión, el optimismo, el pesimismo, la introversión, la extraversión, la mentalidad abierta, la mentalidad cerrada, la espontaneidad, el autocontrol, el grado de inteligencia, entre otros factores.

### *Capacidad de aprender (saber aprender)*

En su sentido más general, saber aprender exige la capacidad de observar y de participar en nuevas experiencias, y de incorporar conocimientos nuevos a los conocimientos existentes, modificando éstos cuando sea necesario [85]. Las capacidades de aprendizaje de lenguas se desarrollan en el transcurso de la experiencia de aprendizaje. Permiten al aprendiente abordar con mayor eficacia e independencia los nuevos desafíos del aprendizaje de la lengua para ver qué opciones existen y hacer un mejor uso de las oportunidades. La capacidad de aprender un idioma tiene varios componentes, entre ellos, la reflexión sobre el sistema de la lengua y la comunicación, las destrezas fonéticas generales, las destrezas de estudio y las destrezas de descubrimiento y de análisis.

La reflexión sobre el sistema de la lengua y la comunicación considera al conocimiento y a la comprensión como fundamentales para el aprendizaje de una lengua, pues permiten que se asimile la nueva experiencia en un marco ordenado y que se acepte como un enriquecimiento. De ese modo, la nueva lengua se puede aprender y utilizar con mayor facilidad, y no se la considera una amenaza para el sistema lingüístico ya establecido del aprendiente.

La reflexión sobre el sistema fonético y las destrezas correspondientes está presente cuando los aprendientes, sobre todo los de edad madura, ejercen su capacidad de pronunciación de las nuevas lenguas mediante la capacidad de distinguir y de producir sonidos corrientes y estructuras prosódicas; la capacidad de percibir y de concatenar secuencias desconocidas de sonidos; la capacidad, como oyente, de convertir una corriente continua de sonido en una cadena significativa y estructurada de elementos fonológicos; y, finalmente, la comprensión o dominio de los procesos de percepción y producción de sonido aplicables al aprendizaje de una nueva lengua.

Las destrezas de estudio incluyen el reconocimiento de las cualidades y las carencias propias como aprendiente: la capacidad de hacer un uso eficaz de las oportunidades de aprendizaje creadas por las situaciones de enseñanza; la capacidad de utilizar materiales disponibles para el aprendizaje independiente; la capacidad de organizar y utilizar materiales para el aprendizaje autodirigido; la capacidad de aprender con eficacia partiendo de la observación directa de los acontecimientos de comunicación y de la participación mediante el desarrollo de destrezas perceptivas, de descubrimiento y de análisis; la capacidad de identificar las necesidades y las metas propias; y, finalmente, la capacidad de organizar las estrategias y los procedimientos propios para conseguir estos fines de acuerdo con las características y los recursos que cada uno tiene.

Las destrezas de descubrimiento y de análisis incluyen la capacidad que tiene el aprendiente de adaptarse a la nueva experiencia y de ejercer otras competencias en la situación específica de aprendizaje; la capacidad que tiene el aprendiente de encontrar, comprender y, si es necesario, transmitir nueva información; y, finalmente, la capacidad de utilizar nuevas tecnologías.

Durante el desarrollo de Ludibot, se consideraron tanto las competencias generales como los tipos de conocimiento que de ellas derivan. Así, en cuanto a las competencias generales, se pretende que el robot contribuya a mediano plazo al desarrollo del conocimiento sociocultural, pues se tiene previsto implementar aplicaciones en torno a la francofonía que permitan abordar aspectos de la sociedad y de la cultura de las comunidades en las que se habla francés.

En cuanto a las destrezas y habilidades, que remiten al saber, están presentes en Ludibot mediante el uso de la interfaz y las aplicaciones lúdicas. Específicamente, se busca generar destrezas sociales mediante el uso de los gestos; destrezas de la vida, mediante la consolidación de un vocabulario útil en diversas situaciones de comunicación; destrezas de ocio, pues se propone la práctica lingüística mediante juegos.

En lo relativo a la competencia existencial, mediante la interacción humano-máquina se pretende incidir en las actitudes, las motivaciones, los valores, las creencias, los estilos cognitivos y los tipos de personalidad.

Finalmente, el saber aprender está relacionado con el objetivo general de Ludibot, que consiste en poner a disposición herramientas teóricas y prácticas para la enseñanza y el aprendizaje de lengua fortaleciendo la investigación pluridisciplinar y sinérgica desde la robótica, las ciencias del juego y la DLC. Ludibot busca promover el aprendizaje de idiomas considerando una dimensión lúdica, que permita aprovechar el potencial pedagógico de la gamificación, ludicización y explorar las cuatro regiones metafóricas, articulándolas con la robótica y la DLC.

### ***Competencia comunicativa***

La competencia comunicativa es aquella usada para la realización de las intenciones comunicativas. Según el *Marco común europeo de referencia para las lenguas* (MCER) [85], la competencia comunicativa abarca los siguientes componentes: las competencias lingüísticas, las competencias sociolingüísticas y las competencias pragmáticas. Cada una de estas categorías de competencias incluye a su vez otras competencias específicas, como se verá a continuación.

### ***Competencias lingüísticas***

Los sistemas lingüísticos son complejos y, por ende, la lengua de una sociedad amplia, diversificada y avanzada nunca llega a ser dominada por ninguno de sus usuarios, pues cada lengua sufre una evolución continua como respuesta a las exigencias de su uso en la comunicación [85]. Los lingüistas descriptivos definen la competencia lingüística como el conocimiento de los recursos formales de la lengua y la capacidad para utilizarlos. A partir de estos componentes, se pueden articular y formular mensajes bien formados y significativos. La competencia lingüística se desglosa en cinco competencias directamente relacionadas con el dominio del código lingüístico: la competencia léxica, la competencia gramatical, la competencia semántica, la competencia fonológica, la competencia ortográfica y la competencia ortoépica.

La competencia léxica [85] se refiere al conocimiento del vocabulario de una lengua y a la capacidad para utilizarlo. Además de palabras, los elementos léxicos incluyen expresiones hechas (fórmulas fijas, modismos, estructuras fijas, etc.). Entra también en esta competencia el adecuado manejo de la polisemia, es decir, el uso de una misma palabra con diversos significados.

La competencia gramatical [85] es definida como el conocimiento de los recursos gramaticales de una lengua y la capacidad de utilizarlos. Formalmente, la gramática de una lengua se puede considerar como un conjunto de principios que rigen el ensamblaje de elementos en oraciones con significado, elementos clasificados y relacionados entre sí. La competencia gramatical es la capacidad de comprender y expresar significados expresando y reconociendo frases y oraciones bien formadas de acuerdo con estos principios.

La competencia semántica [85] comprende la conciencia y el control del significado. Esta competencia incluye tres tipos de semántica: la semántica léxica, que trata asuntos afines con el significado de las palabras; la semántica gramatical, que trata el significado de los elementos, las categorías, las estructuras y los procesos gramaticales; y, finalmente, la semántica pragmática, que se ocupa de relaciones lógicas, como la vinculación, la presuposición, la implicación, entre otras.

La competencia fonológica [85] considera el conocimiento y la destreza en la percepción y la producción de las unidades de sonido de la lengua, llamados fonemas, y su realización en contextos concretos. Los rasgos fonéticos distinguen elementos como sonoridad, nasalidad, oclusión, labialidad. La composición fonética de las palabras está dada por la estructura silábica y la secuencia acentual de las palabras. La fonética de las oraciones, denominada prosodia, incluye el acento y el ritmo de las oraciones. Por otro lado, la reducción fonética abarca la reducción vocal, las formas fuertes y débiles, la asimilación y la elisión.

La competencia ortográfica [85] supone el conocimiento y la destreza en la percepción y la producción de los símbolos de que se componen los textos escritos. Su dominio incluye elementos como las formas de las letras, la correcta ortografía de las palabras, los signos de puntuación y sus normas de uso, las convenciones tipográficas y las variedades de tipos de letra, los signos no alfabetizables de uso común (por ejemplo: @, &, \$, etc.).

La competencia ortoépica [85] se requiere para leer en voz alta un texto o utilizar en el habla palabras que los usuarios han aprendido en su forma escrita. Ellos necesitan saber articular una pronunciación correcta partiendo de la forma escrita. La competencia ortoépica incluye el conocimiento de las convenciones ortográficas, la capacidad de consultar un diccionario y el

conocimiento de las convenciones utilizadas en los diccionarios para la representación de la pronunciación.

En cuanto a las competencias lingüísticas, Ludibot emplea específicamente la competencia gramatical, ya que permite consolidar el uso de los artículos determinados en francés (*l'*, *la* y *le*). La competencia semántica está presente al exigir reconocer el significado de cada parte del cuerpo para lograr un juego exitoso. Se busca reforzar la competencia fonológica durante el juego *De la tête aux pieds* mediante la percepción y la producción de sonidos de la lengua. Además, se trabaja con la pronunciación de las partes del cuerpo. En cuanto a la competencia ortográfica, se hace uso de símbolos, formas de las letras, uso correcto de las palabras, convenciones tipográficas y variedades de tipos de letra el diseño de la interfaz. Finalmente, se aborda la competencia ortoépica, al asociar la palabra hablada con la correspondiente palabra escrita.

#### *Competencias sociolingüísticas*

Las competencias sociolingüísticas [85] remiten al conocimiento y a las destrezas requeridos para abordar la dimensión social del uso de la lengua. En ella, se consideran los marcadores lingüísticos de relaciones sociales, las normas de cortesía, las expresiones de la sabiduría popular, las diferencias de registro, el dialecto y el acento.

Aunque las competencias sociolingüísticas no están explícitamente presentes por ahora en Ludibot, sí se contempla a futuro el desarrollo de aplicaciones que permitan abordar los aspectos sociolingüísticos en el aula. En efecto, si bien el robot está diseñado para ser usado específicamente en francés, este idioma está en uso en Francia, pero también en otros países de Europa, África y América, e incluso en Asia [31]. El francés como lengua de comunicación internacional está presente en los cinco continentes y su proceso de enseñanza-aprendizaje está inscrito en una perspectiva de diversificación lingüística y en una didáctica del plurilingüismo. Ludibot podrá servir por ejemplo para ilustrar la diversidad cultural de la francofonía en general y de la francofonía de las Américas en particular.

### *Competencias pragmáticas*

Las competencias pragmáticas [85] son tres y se refieren al conocimiento que posee el usuario a través del cual organiza, estructura y ordena su discurso (competencia discursiva), realiza funciones comunicativas (competencia funcional) y secuencia la información según esquemas de interacción y de transacción (competencia organizativa).

La competencia discursiva [85] es la capacidad que posee el usuario o el aprendiente de ordenar oraciones en secuencias para producir fragmentos coherentes de lengua. Comprende también la capacidad de controlar esa ordenación en función de los temas y las perspectivas, la secuencia natural, las relaciones causa y efecto, la capacidad de estructurar y la organización del texto. La competencia funcional [85] supone que el uso del discurso hablado y de los textos escritos en la comunicación persigue fines funcionales concretos. Finalmente, la competencia organizativa [85] comprende el conocimiento de modelos de interacción social que se derivan de la comunicación, tales como los modelos de intercambio verbal, y la capacidad de utilizarlos.

Las competencias pragmáticas están presentes en el uso y el desarrollo de Ludibot, ya que está diseñado para usuarios principiantes que deben organizar, estructurar y ordenar los elementos discursivos en juego (competencia discursiva). Además, el juego *De la tête aux pieds* requiere tener en cuenta una parte del cuerpo dada y una parte del cuerpo nueva, cuyo desfase exige que el usuario use su memoria de trabajo. Por ende, las palabras aprendidas serán utilizadas para realizar funciones comunicativas (competencia funcional). Todo ello ocurre mediante esquemas de interacción y de transacción específicos (competencia organizativa), dictados por la interfaz humano-máquina, pero durante los cuales el usuario debe tomar decisiones.

Se ha visto hasta aquí de qué manera diversos aspectos de la lengua y del lenguaje fueron tomados en cuenta para el desarrollo de Ludibot. Ahora bien, es importante subrayar que esta plataforma busca centrarse en el diseño y el desarrollo de aplicaciones lúdicas que complementen y faciliten la enseñanza y el aprendizaje de lenguas mediante el uso de la tecnología.

Así, es necesario estudiar algunas de las principales variables implícitas durante el aprendizaje, tales como la aptitud, la inteligencia, la motivación, el modelo del salón de clases y el modelo didáctico. Todas ellas son variables afectivas y se explican a continuación.

### *Variables afectivas en el aprendizaje*

Existen diversos factores que permiten analizar la implicación del estudiante en el aprendizaje de una lengua extranjera. Según los expertos [36], las dos variables más representativas son la aptitud y la inteligencia. La aptitud hace referencia a las características cognoscitivas que controlan la velocidad, la facilidad y el éxito en el estudiante, es decir, el estado inicial de rapidez y de capacidad del individuo para aprender una lengua extranjera. Por su parte, la inteligencia es un factor general hipotético que sustenta nuestra capacidad de dominar y utilizar una gama de habilidades académicas [51]. La inteligencia también es definida como la capacidad de saber, y de acuerdo con [110], está constituida de diversos factores: verbal, psicomotor, lógico, espacial, numérico, perceptivo y analógico.

Ahora bien, el resultado de la enseñanza de una lengua no sólo depende de la inteligencia y la aptitud, sino también de otros factores psicológicos, afectivos, sociales o motivacionales. El modelo socioeducacional y el modelo de la aculturación [43], desde la psicología social, plantean que el éxito de la segunda lengua no depende sólo de la capacidad intelectual y de la aptitud verbal del estudiante, sino también de variables socioafectivas como las actitudes hacia un grupo etnolingüístico, el interés del uso de una lengua con fines prácticos, entre otros.

Para estudiar estos otros factores es posible recurrir al modelo de la motivación, al modelo didáctico y al modelo del salón de clase. Dichos modelos son primordiales en el aprendizaje, pues el entorno de aprendizaje influye significativamente en el proceso. Cada uno de estos tres modelos es abordado en su relación con Ludibot en las líneas siguientes.

### *Modelo de la motivación*

El modelo de la motivación [43] consiste en un conjunto de procesos implicados en la activación, la dirección y la persistencia de la conducta que se relaciona directamente con el desarrollo y éxito del ser humano. Si consideramos la esfera escolar, las diversas aptitudes, las actitudes, las

percepciones y las expectativas del estudiante constituyen factores que orientan su conducta. Usualmente, la motivación no es un fenómeno meramente personal, sino que también depende de la interacción dentro de un grupo o comunidad. Ello implica la importancia de proveer a los estudiantes un entorno favorable, en el que puedan cumplir sus objetivos y metas. Ausubel [11] clasifica la motivación en dos grandes tipos: la motivación intrínseca y la motivación extrínseca que, en la práctica, suelen estar combinadas.

La motivación intrínseca es generada por motivos internos del individuo. Se lleva a cabo de manera constante y sin ningún tipo de contingencia externa, en forma de actitudes como la curiosidad por explorar nuevos campos; el deseo de saber, de entender, de formular y de resolver problemas; el afán de entender, analizar, aprender cosas nuevas. El incentivo es intrínseco a la actividad misma, es decir, la realización de la conducta en sí misma impulsa al individuo. Por ejemplo, está presente en la práctica de un pasatiempo. En otras palabras, la motivación intrínseca está sustentada por la pulsión cognoscitiva.

Por su parte, en la motivación extrínseca, los motivos que impulsan la acción realizada son ajenos a la misma. Es decir, la disposición favorable es estimulada por impulsos como incentivos o reforzadores negativos o positivos externos al propio sujeto y a su actividad. Entre algunos ejemplos, se tiene lograr excelentes calificaciones, granjearse el reconocimiento de una persona, competir con otros, o alcanzar una recompensa. La motivación extrínseca incluye a su vez tres tipos [11]: la motivación de mejoramiento del yo, la motivación afiliativa y la motivación aversiva:

1. La motivación de mejoramiento del yo conduce al individuo a lograr el aprovechamiento académico con el fin de obtener prestigio, reconocimiento, estatus y autoestima.
2. La motivación afiliativa es aquella que posee un individuo cuando logra un buen rendimiento académico para satisfacer las expectativas u obtener la aprobación de una persona o de un grupo de personas, de los cuales depende afectivamente (los padres, los hermanos, los amigos, el profesor, entre otros).
3. La motivación aversiva está relacionada con la amenaza de ser castigado por haber fracasado. En este caso, el objetivo no es tener éxito sino evitar el fracaso.



La relación entre la motivación y el aprendizaje debe ser recíproca y estrecha, dado que la motivación es tanto un efecto como una causa del aprendizaje del alumno. Por tal razón, a través del desarrollo de Ludibot, se busca brindar a los usuarios una herramienta que permita, gracias a aplicaciones lúdicas, incrementar tanto la motivación intrínseca como la extrínseca durante el proceso de aprendizaje y enseñanza. En efecto, según se ha visto con anterioridad, los robots son hasta ahora poco usuales en el aula; su uso aparece como una propuesta didáctica innovadora, potencialmente atractiva para los usuarios. La novedad de la robótica, asociada a las dinámicas lúdicas, puede contribuir a la motivación intrínseca. Por otro lado, el juego introduce una dinámica de emulación y competencia, relacionada con la motivación extrínseca.

Los especialistas recomiendan no esperar a que la motivación del alumno se desarrolle por sí sola, sino brindar elementos para complementar dicha motivación en el salón de clases, con el fin de obtener un aprendizaje óptimo. Para ello, resulta útil tomar en cuenta el modelo del salón de clases.

#### *Modelo del salón de clases*

Uno de los factores más importantes del proceso educativo formal es el salón de clases. En efecto, el aula suele ser el lugar donde el estudiante pasa la mayor parte del tiempo dedicado al aprendizaje. El salón de clases es dado como un sistema, cuyos componentes son el maestro, los compañeros de clase, los materiales, las actividades, las herramientas complementarias y los métodos de aprendizaje. Todos ellos funcionan como fuentes de actitudes y motivaciones que impactan positiva o negativamente tanto en la adquisición como en el aprendizaje.

En el caso de la enseñanza/aprendizaje de una lengua, los componentes del salón de clases pueden generar actitudes positivas hacia la situación de aprendizaje y pueden hacer surgir una motivación intrínseca que beneficiará el desarrollo de la segunda lengua. El modelo del salón de clases remite a que se puede adquirir un idioma a partir de la motivación integrativa, entendida como el deseo de adquirir determinada lengua para volverse como los miembros prestigiosos de esa comunidad de lingüística, adoptando ciertos rasgos de la conducta propios de ese grupo. Este proceso de aprendizaje de una segunda lengua también incluye por ende un proceso de aculturación: al mismo tiempo que conserva sus valores, el aprendiente tiene mayor o menor

disposición a adoptar rasgos sociales y psicológicos de la comunidad de la lengua. El grado en el que un aprendiente se acultura al grupo de la lengua meta incide en el grado en el que adquiere la lengua.

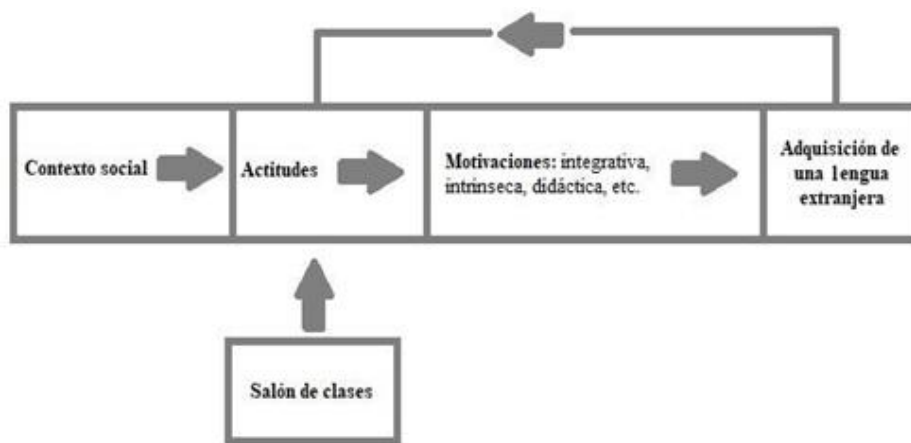
Así, en el aula de idiomas, es recomendable que el profesor promueva en el alumno la motivación intrínseca, la motivación integrativa y el deseo de aculturarse. Con ese fin, el profesor puede implementar actividades propicias al aprendizaje mediante recursos auténticos, que presenten aspectos llamativos, nuevos e innovadores de la cultura de la lengua extranjera. El profesor también puede recurrir a actividades de comunicación significativas en el mundo real, favorables al análisis y a la comparación de las culturas en presencia [190].

A través de la propuesta metodológica y tecnológica de esta tesis, se busca generar en el alumno la motivación mediante actividades y dinámicas lúdicas. Ludibot es un robot que usa la tecnología para generar una interacción humano-máquina basada en la lengua, el lenguaje y el movimiento. Presenta aspectos llamativos, nuevos e innovadores para los usuarios en el proceso de aprendizaje de una lengua. Pretende contribuir al proceso de aculturación desde un contexto social, es decir, el contexto del aula. Sin embargo, según Gardner [151], la motivación integrativa y la aculturación no siempre causan los resultados esperados: en ocasiones, se requiere la motivación instrumental, aquella que guía a los seres humanos a perseguir determinadas metas.

La motivación instrumental desempeña un papel importante en el aprendizaje de lenguas extranjeras. En efecto, la mayoría de los aprendientes suele tener como meta lograr el dominio de la segunda lengua con fines prácticos, y elige la lengua que considera más rentable. Tal es el caso del inglés, idioma predominante hoy en el ámbito de la educación, en la medida en que es visto como una lengua internacional de amplia presencia en los cinco continentes. Si bien este enfoque instrumental es muy eficaz, Filipinas, Gardner y Lambert [151] consideran que los estudiantes de lenguas extranjeras sacan mucho más provecho de la motivación integrativa. En lugar de apostar por el aprendizaje de una sola lengua extranjera dominante, en lo que se conoce como un enfoque de monolingüismo selectivo, resulta productivo adoptar un enfoque plurilingüe y pluricultural: no se trata de dar prioridad a un idioma sobre otro, sino de aprovechar el potencial de un enfoque que reconoce la pluralidad.

En este contexto, el aprendizaje del francés puede ser abordado tanto desde una óptica instrumental como desde una óptica integrativa. En efecto, la lengua francesa es la cuarta lengua más hablada del mundo y la única, junto con el inglés, que se habla en los cinco continentes [31], pero la DLC francófona se apoya constantemente en postulados plurales tales como los enunciados en [85]. Esta doble dimensión apunala el interés de desarrollar una herramienta didáctica basada en un robot capaz de resolver aplicaciones lúdicas definidas como tareas específicas y problemas abiertos [166].

En efecto, el salón de clases puede ser por sí mismo una fuente de motivación, generando motivación didáctica [106], que nace del curso mismo. Esta motivación se ve reflejada cuando el estudiante asiste al curso y se siente satisfecho. El diagrama de la figura 1.5, basado en el modelo de Gardner [151], engloba de manera general las variables que influyen en el aprendizaje dentro del salón de clases, incorporando los tres modelos estudiados aquí: el de la motivación, el del salón de clases y el didáctico.



**Figura 1.5. Diagrama del salón de clase y la adquisición de una lengua extranjera**

*Modelo didáctico*

El modelo didáctico [43] busca estimular las funciones de los sentidos para acceder de manera fácil a la adquisición y el aprendizaje de conceptos, habilidades, actitudes o destrezas considerando el material, los programas y las actividades de un tema. Todas estas variables

permiten una mejor relación entre el aprendiente y lo que desea aprender, entre el alumno y el profesor, e incluso entre el objeto del aprendizaje y la realidad del entorno.

Actualmente, el papel del profesor no puede considerarse como única fuente de información. El modelo de enseñanza basado en el libro de texto se ha superado y el docente ya no es el único miembro del grupo que puede presentar propuestas o iniciativas, puesto que la enseñanza está más centrada en las necesidades, los intereses e incluso las preferencias del alumno. Por ello, es importante buscar herramientas para complementar el proceso educativo en un salón de clases, considerando las aportaciones potenciales de todos los miembros del grupo. Hoy en día, se postula que el profesor es un guía para el alumno. Su función consiste en coordinar, asesorar y guiar a un grupo para alcanzar los objetivos previstos.

Por lo anterior, en esta tesis, Ludibot es propuesto como una herramienta para complementar el proceso de aprendizaje de una lengua, con el fin de brindar al aprendiente situaciones formales y semiformales de aprendizaje propicias tanto a la autonomía como a experiencias motivantes.

Una vez estudiadas las variables afectivas que inciden directamente en el desarrollo de Ludibot, se ha concluido el subcapítulo dedicado a las nociones clave de la DLC. Ahora bien, esta disciplina se inscribe dentro de las ciencias de la educación, dentro cuyo campo se han desarrollado en fechas recientes importantes reflexiones que transforman la manera de percibir el aprendizaje. El siguiente subcapítulo aborda dos aportes recientes en las ciencias de la educación que tienen singular relevancia para este proyecto, a saber, la literacidad mediática multimodal y las competencias del siglo XXI [6][207][208][71].

### **1.2.3 Nuevos aportes de las ciencias de la educación**

En pleno siglo XXI, los estudiantes necesitan más que un aprendizaje tradicional, más allá del modelo de un salón de clases. En efecto, numerosos expertos, entre ellos [71], coinciden en que es necesario desarrollar otras habilidades tales como la colaboración, la experimentación, la comunicación y la resolución de problemas, que son algunas de las habilidades ligadas al aprendizaje social y emocional.

Considerando el glosario de la Comisión Europea [53], una habilidad se define como la destreza para realizar tareas y resolver problemas, mientras que la competencia es la capacidad de aplicación de los resultados del aprendizaje específicamente en un contexto definido (educación, trabajo, desarrollo personal o profesional no limitado a elementos cognitivos, que implica el uso de teoría, conceptos o conocimiento tácito). La competencia también abarca aspectos funcionales que involucran habilidades técnicas, así como atributos interpersonales, habilidades sociales u organizativas y valores éticos.

El panorama antes descrito ha dado lugar a una serie de iniciativas nuevas sobre la enseñanza y la evaluación de los maestros tras la constatación del requerimiento de un conjunto de habilidades y competencias muy diferentes al propuesto por la educación tradicional. De manera general, una competencia [207][208] va más allá del conocimiento del contenido o del dominio de una habilidad, pues implica la capacidad de satisfacer demandas complejas, aprovechando y movilizandorecursos psicosociales (incluyendo habilidades y actitudes) en un contexto particular. Por su parte, una cualidad [207][208] es entendida como la característica que distingue y define a una persona, determinada por rasgos, componentes permanentes, diferenciados, peculiares y distintivos. En 2015, el Foro Económico Mundial propuso nuevas formas de abordar las habilidades mediante el uso de la tecnología [207][208]. Entre los aportes más relevantes de las ciencias de la educación están las llamadas competencias del siglo XXI, mencionadas a continuación.

### ***Competencias del siglo XXI***

Según se mencionó ya, la competencia es un concepto amplio que comprende tanto conocimientos como habilidades y actitudes. Las competencias del siglo XXI integran las cualidades que los jóvenes deberán tener para aprender de manera eficaz y lograr ser profesionistas efectivos en la sociedad del conocimiento presente y futura [6][207][208][71]. Las competencias del siglo XXI remiten a diferentes tipos de literacidades, competencias y cualidades, tal como lo ilustra la figura 1.6.



**Figura 1.6. Competencias del siglo XXI**

Para el desarrollo de estas competencias, la tecnología debe complementar los enfoques pedagógicos existentes. En efecto, la adopción de herramientas tecnológicas no da como resultado automático el beneficio esperado para el aprendizaje. La tecnología educativa debe ser siempre adaptada a los contextos específicos donde busca ser aplicada, y replantear aspectos tales como las modalidades y los contenidos de la formación, el papel de aprendientes y docentes en el proceso, el proceso mismo de aprendizaje y autoaprendizaje. Por estas razones, es indispensable alinear el uso de la tecnología con objetivos de aprendizaje claramente formulados.

El énfasis que las competencias del siglo XXI en la tecnología evidencian que el uso de robots en la educación ha dejado de pertenecer al campo de la ciencia ficción. Desde el primer robot industrial que fue diseñado por George Devol en 1954 [173] hasta el estado actual de la robótica educativa, ha habido un progreso enorme en la inteligencia artificial: hoy en día, los robots son capaces de responder a los cambios en su entorno en lugar de simplemente repetir la misma tarea. Ahora bien, en la práctica, queda mucho camino por recorrer. Para ser plenamente aprovechadas, las nuevas posibilidades brindadas por robots cada vez más sofisticados deben ir de la mano con una reflexión en torno a las competencias, las literacidades y las cualidades por desarrollar en el proceso de aprendizaje.

El uso de la robótica en el ámbito educativo puede contribuir al desarrollo de las competencias del siglo XXI. En el caso específico de Ludibot, se trata de una herramienta gracias a la cual el aprendiente es invitado a construir el conocimiento a partir de su propia experiencia en el aula de idiomas en general y en el aula de francés como lengua extranjera en particular, desde un enfoque integral favorable a la interacción humano-máquina. Busca contribuir al desarrollo del pensamiento crítico, la intuición, la creatividad, la comunicación y la colaboración. Se requiere asimismo que el usuario eche mano de sus cualidades: la curiosidad, la iniciativa y la adaptabilidad permiten indagar el uso de este robot con aplicaciones lúdicas. Finalmente, tanto en su desarrollo como en su uso, apuesta por el ejercicio de diversas literacidades, por ejemplo, la científica y la cultural: la científica está presente en la programación, el modelado matemático y la aplicación de estrategias de control, de las tecnologías de la información, mientras que la cultural tiene que ver con la aculturación propiciada por el aprendizaje de una segunda lengua. Se apoya asimismo en la literacidad mediática multimodal que se expondrá a continuación.

### ***Literacidad mediática multimodal***

El término “literacidad”, derivado del inglés *literacy*, era considerado hasta hace poco en español como un sinónimo de “alfabetización” que, según la Real Academia Española, designa tanto la acción como el efecto de alfabetizar, verbo a su vez definido como “enseñar a alguien a leer y a escribir” [150][151]. Hoy en día, el concepto se ha ampliado y se usa en plural. Las literacidades remiten a la posibilidad de recibir, procesar y crear sentido mediante distintos instrumentos de comunicación y según modalidades diversas de recepción, intercambio y producción de mensajes. Se habla así de literacidad científica para remitir a la capacidad de manejar adecuadamente los códigos de recepción, interpretación, intercambio y producción de mensajes científicos; de literacidad financiera, de literacidad cultural e incluso de literacidad lúdica, para enfatizar que cada uno de estos ámbitos exige un manejo distinto de la información, tanto en su producción como en su recepción, su intercambio y su interpretación.

Si bien la literacidad remitía en un principio al campo de lo escrito, ahora busca tomar en cuenta una realidad que tiende a la complejidad comunicativa, siendo cada vez más accesible y común la generación de mensajes que enriquezcan o reemplacen la escritura con uno o más sistemas de significación distintos. Más allá de la alfabetización, que no suele considerar el rol

del receptor en la lectura, el concepto de literacidad [132] [150] toma en consideración, para el estudio de la recepción y la decodificación de los significados de un texto en sentido amplio, aspectos que condicionan la interpretación, tales como el contexto sociocultural.

Es así como el concepto de literacidad empieza a dar lugar a reflexiones específicas, entre ellas la que conduce al concepto de literacidad mediática multimodal. La multimodalidad [150] es entendida aquí como la articulación entre diferentes modos de recepción y producción de sentido (icónico, lingüístico y/o auditivo) en un mismo sistema. Una secuencia de video, por ejemplo, comprende imágenes y sonidos, los cuales son entregados, recibidos e interpretados de manera conjunta.

De acuerdo con Lebrun, Lacelle y Boutin [107], la literacidad mediática multimodal puede definirse como la habilidad de acceder a textos multimodales, de analizarlos, de evaluarlos o de crearlos a lo largo de una variedad de contextos. La definición considera tanto la recepción activa como la importancia del análisis y la evaluación, además de mostrar la capacidad del receptor de convertirse en emisor.

En el caso concreto de nuestro proyecto de investigación, cuya finalidad es desarrollar e implementar un robot móvil para apoyar de manera lúdica el aprendizaje de una lengua extranjera, la capacidad de moverse y de reconocer gestos le permiten a Ludibot funcionar como una herramienta educativa polivalente, que recurre a al menos tres modos distintos: el gestual, el interactivo y el lingüístico, que generan un significado unificado, no equivalente a la suma mecánica de cada una de las partes de manera independiente.

Ludibot busca fomentar el desarrollo de la literacidad mediática multimodal al privilegiar la articulación entre diversos modos: figuras, sonidos, audios, palabras, gestos, ademanes y símbolos específicos. Asimismo, se otorga al sujeto del aprendizaje un rol activo que no se limita a la recepción y a la repetición de un *input*, sino que exige producir, interpretar y evaluar la información recibida a través de los diferentes modos ya mencionados. Ese rol activo es indispensable para interactuar con el robot y participar en el o los juegos.



Ludibot se inscribe de esta manera en una tendencia que tiene como fin aprovechar todas las ventajas de la robótica móvil para promover diversos tipos de aprendizajes, trátense de robots destinados para juegos, robots en el aula de clases, robots sociales en museos, etc.

Se ha considerado hasta ahora una amplia selección de trabajos relativos tanto a la evolución histórica de la enseñanza en general [67] y del aprendizaje de idiomas en particular [35][43][64][75][115][182] como a la tecnología educativa [37][58][91][95][97][178], en particular en lo que atañe a la DLC [168][177]. Ahora bien, Ludibot abarca tres campos disciplinares: la robótica, la DLC y las ciencias del juego. Ha llegado el momento de explorar el interés específico de aplicar el juego a la enseñanza y el aprendizaje de idiomas mediante el uso de la robótica móvil. Se empezará por problematizar la definición de juego en referencia al fenómeno de gamificación, para explorar después el concepto de juego como metáfora heurística y, finalmente, el papel de la interacción lúdica en el proyecto.

### **1.3 Ciencias del juego**

El juego como práctica social recurre casi siempre a la lengua y al lenguaje [169] y ofrece un modelo productivo para la enseñanza, que puede transformar las experiencias de aprendizaje del lenguaje. Gee [62] menciona que la capacidad de jugar, aprender y comprender a través de juegos puede considerarse una nueva forma de alfabetizarse gracias a la tecnología, en lo que él llama “literacidad lúdica”. Este tipo de literacidad tiende a ocupar un lugar creciente y cada vez más significativo en las sociedades contemporáneas.

Para sacar mejor provecho de las herramientas lúdicas, es importante problematizar la o las nociones de juego en presencia, desde las ciencias del juego, de reciente desarrollo. En efecto, no hay que perder de vista que la noción de juego, capital para el presente trabajo, se presta difícilmente a una sola definición, ante la pluralidad de fenómenos heterogéneos e incluso heteróclitos que engloba. El juego es un fenómeno que ha acompañado al ser humano a lo largo de toda su historia. Basta con observar la existencia del juego como actividad libre y apartada de la realidad funcional en los animales, para evidenciar su origen primitivo. En la última década, sin embargo, ha aumentado el interés por el juego y su potencial heurístico, en un fenómeno conocido como “gamificación”.

### 1.3.1 Gamificación, ludificación y ludicización

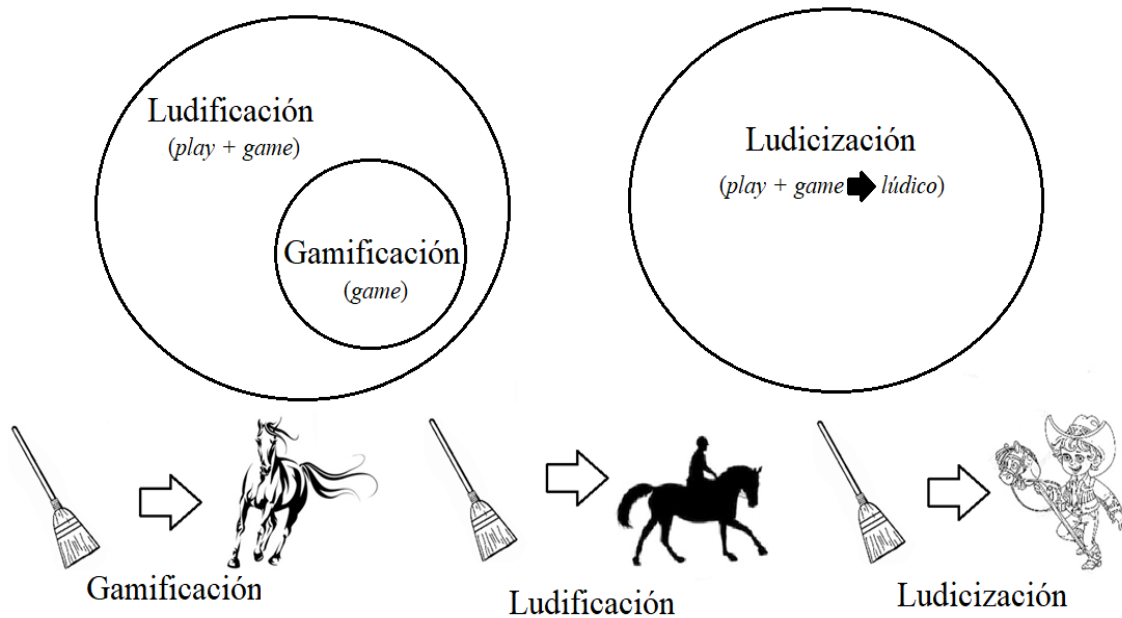
La palabra “gamificación”, de acuerdo con [26][46][63], tiene dos grandes acepciones: la primera se refiere a la adopción, a la institucionalización y a la omnipresencia crecientes de los (video)juegos en la vida de todos los días. La segunda acepción, más específica, atañe al uso de mecanismos de juego con la finalidad de motivar a los usuarios a involucrarse con mayor intensidad y mayor duración, incluso en actividades que no necesariamente son juegos. Esto puede verse reflejado en las aplicaciones sociales que otorgan una serie de estímulos propios de los videojuegos (medallas, insignias, puntos, niveles, rangos) a los usuarios más activos, como ocurre en *Facebook*, *Google Maps*, *Waze*, entre otras. El proceso de gamificación busca aprovechar los elementos que propician la actitud lúdica para aplicarlos a dinámicas de consumo o de acción y así volverlas más amigables para los receptores, por lo que puede definirse, según las palabras de [46] como “el uso de elementos de *game design* en contextos no lúdicos”. Aunque la gamificación se extiende a muy diversos ámbitos, recurrir a elementos lúdicos para suscitar más placer y compromiso es una estrategia de mercadotecnia para promover un mayor consumo de productos y servicios. Una de las críticas que suele hacerse a la gamificación tiene entonces que ver con la mercantilización.

La palabra misma de “gamificación” resulta controversial, pues distintas culturas han señalado realidades muy diversas mediante vocablos a veces unificados y a veces diferentes que se insertan en el campo semántico del juego, como lo demuestra el estudio que realiza Huizinga [84]. Así, algunas lenguas poseen voces más específicas que otras para nombrar los fenómenos lúdicos, mientras otras carecen de las generalizaciones que los abarcan a todos potencialmente, revelando lo difícil que es encontrar los límites de la esfera del juego. Esto puede observarse, por ejemplo, en un caso contemporáneo, si se compara el español con el inglés. En esta segunda lengua, dos términos distintos, *play* y *game*, se refieren a dos procesos que en español existen unidos en la palabra *juego*. *Play* denota específicamente la actividad libre de recreación y esparcimiento, mientras que *game* hace referencia a una estructura organizada en torno a una serie de convenciones y reglas. Elegir hablar de gamificación tiene a privilegiar el sentido asociado a *game* y a dejar de lado los elementos asociados al *play*.

Para evitar el uso de “gamificación”, hay quienes recurren a la palabra “ludificación” (*ludification* en inglés y francés) [20][26][63][167], término igualmente utilizado para designar la difusión del juego en la cultura [20][46][63][144]. El proceso de ludificación de la cultura implica la construcción de una red de referentes (que afectan el contexto y el código, según los términos que analizamos en la sección sobre didáctica de lenguas) que tienen su origen en juegos. Sin embargo, el término de “ludificación” ha sido igualmente criticado. Si bien gamificación sigue siendo el vocablo más popular, con 2’060,000 resultados en un motor de búsqueda configurado en español en noviembre de 2020, contra 142,000 resultados para “ludificación”, ciertos autores, como Genvo [63] y Silva [167], subrayan que el sufijo –ficación tiende a hacer creer en un proceso de transformación automática, independiente de los actores en presencia, por lo que sugieren recurrir a la palabra “ludicización” para enfatizar el papel activo de los sujetos que intervienen en la transformación. Sin embargo, esta última palabra arroja apenas 74 resultados en un motor de búsqueda en español, y 11,500 para *ludicisation* en francés.

La ludicización se refiere al hecho de que ciertos objetos que no solían ser considerados como juegos son progresivamente calificados de ese modo. En resumen, allí donde la gamificación tiende a reducir la visión del juego a una estructura, sin tomar en cuenta la actividad de los jugadores, la ludificación pretende integrar la doble dimensión de actividad y estructura. Sin embargo, tanto gamificación como ludificación remiten a la transformación de un fenómeno en juego, mientras que la ludicización remite a la transformación de la percepción de ese mismo fenómeno, a partir de ciertos filtros introducidos de manera deliberada y basados en uno o varios de los múltiples sentidos atribuidos al juego.

Por dar un ejemplo concreto, postular que la gamificación de una actividad de aprendizaje la transforma en juego equivale a postular que un palo de escoba utilizado para jugar se convierte en un caballo, mientras que hablar de ludicización del palo de escoba indica que se recurrió a añadir ciertos elementos materiales al palo de escoba y se promovió cierto uso en un contexto dado, de tal manera que el palo de escoba puede ser percibido o no dentro de los límites del juego como un caballo (véase figura 1.7).



**Figura 1.7. Diferencia entre gamificación, ludificación y ludificación**

Por lo anterior, se considera que Ludibot aplica un proceso de ludificación: si bien los robots son dispositivos que originalmente no fueron considerados para el juego o juegos, es posible utilizarlos para modificar la percepción que de ellos se tiene, así como de la o las actividades que con ellos se realizan, acercándolas al universo de juego [5][24]. Ludibot no busca transformar mediante la robótica el aprendizaje de lengua en un juego, sino aprovechar en el ámbito educativo el potencial de los robots para incrementar la posibilidad de que una experiencia de aprendizaje de idiomas sea percibida como lúdica. Para ello, se recurre a la propuesta de Henriot [79][80], posteriormente desarrollada por Silva [169] para aplicarla a la DLC. Yendo más allá de aquellos trabajos que asocian robótica y juego sin problematizar el segundo término, se ha elegido en este proyecto ampliar la reflexión gracias al modelo de paradigmas y niveles desarrollado por Silva [168][169] y expuesto a continuación.

### **1.3.2 Regiones metafóricas de juego**

Según Silva [168][169], el significado atribuido a la noción de juego es un constructo social determinado por factores históricos y culturales. Si bien el conjunto de significados es muy amplio y diverso, puede ser estudiado mediante la aplicación de un modelo en el que se distinguen cuatro grandes niveles semánticos. Para referirse a esos niveles semánticos, la autora

usa también la expresión de “regiones metafóricas” para subrayar, tal como lo hace Henriot [73] [74], que el objeto de estudio no es un hipotético referente fenomenológico único sino más bien la red analógica del discurso que se construye en torno a aquello que en un momento y en un sitio dado se denomina juego, recurriendo a un mecanismo metafórico.

La teoría de las regiones metafóricas nos permite problematizar el concepto de juego y llevarlo a la práctica para el desarrollo de aplicaciones lúdicas mediante Ludibot. Estas cuatro regiones metafóricas son [169]: el material lúdico, el contexto lúdico, la estructura lúdica y la actitud lúdica.

El material lúdico se refiere a aquello con lo que se juega, hecho o no explícitamente para la actividad lúdica. Esta región de la metáfora considera los elementos que son parte del juego, es decir, se pueden tener algunos que son usados comúnmente en juegos, como dados, cartas, etc.; y, eventualmente, elementos que no fueron elaborados para jugar.

La estructura lúdica es definida como un sistema de reglas que rigen la actividad de juego. Estas reglas, propias de cada juego, ponen en marcha un cierto número de mecanismos y de principios precisos. Existen cuatro clases de reglas: las reglas específicas del juego, esenciales para su desarrollo; las disposiciones técnicas para jugar bien, aplicadas por los buenos jugadores; las reglas ligadas a la estrategia individual; y aquellas asociadas a la cultura lúdica, estrechamente relacionadas con la literacidad lúdica.

El contexto lúdico es comprendido como el conjunto de elementos objetivos externos al juego. Entre los factores que incluye están la determinación histórica, la determinación ideológica y las condiciones concretas materiales y espaciotemporales.

Finalmente, la actitud lúdica es determinada como la disposición de ánimo del jugador, su convicción interna del jugador respecto al sentido de sus acciones.

En esta tesis, el robot es el principal dispositivo, elemento y/o herramienta que permitirá al aprendiz o a los aprendices el uso de las aplicaciones lúdicas. La plataforma es pues el principal material lúdico, sin que ello implique la incompatibilidad con otros materiales. De

hecho, todas las aplicaciones tendrán como material la lengua y el lenguaje; mientras que el juego *De la tête aux pieds* incluye dentro de aquello con lo que se juega el cuerpo de los participantes. En efecto, como todo robot móvil, Ludibot puede desempeñar tareas específicas y/o resolver problemas abiertos [166], pues está diseñado para ser programado y su arquitectura permite desplazarse mediante estrategias de control en aula de idiomas, mediatecas y/o centros de autoacceso.

Silva [168][169] subraya cómo, a pesar de su importancia, el material resulta a menudo insuficiente para transformar la percepción de una actividad y hacerla entrar dentro de la categoría de juego. Por más que el material sea atractivo y novedoso, debe ir acompañado por una estructura lúdica, es decir por un conjunto de reglas específicas.

En el caso de Ludibot, la o las estructuras de juego están dadas por las reglas definidas para cada una de las aplicaciones lúdicas. Estas reglas definen las tareas y las acciones pertinentes del robot durante la interacción humano-máquina. Una vez que se haya podido realizar el pilotaje, se añadirán a las reglas específicas del juego nuevas reglas identificadas por los propios jugadores como aquellas que permiten jugar mejor; cada jugador desarrollará además sus propias estrategias, con base en su mayor o menor conocimiento del patrimonio lúdico colectivo y en su mayor o menor grado de literacidad lúdica. En la medida de lo posible, para aprovechar de manera óptima el potencial de la plataforma, tales reglas incorporarán consideraciones relativas a la identificación de movimientos, gestos, distancias, posiciones y producciones lingüísticas.

El desarrollo de aplicaciones lúdicas permite diseñar e implementar tanto juegos ya existentes –es decir, los juegos tradicionalmente entendidos como tales– como nuevas dinámicas diseñadas con fines lúdicos y eventualmente educativos. Es decir, algo que previamente no se consideraba como perteneciente al universo lúdico puede ser asociado a éste siempre y cuando se implementen reglas lúdicas de uso, como puede ocurrir con la robótica, concebida originalmente para tareas industriales. Dentro de este proyecto, se eligió desarrollar un primer juego, *De la tête aux pieds*, inspirado en un juego previamente existente y sin fines explícitos de aprendizaje. Se aprovechó así un conjunto de reglas proveniente del patrimonio lúdico, gracias al cual es posible generar una interacción propicia a la práctica de elementos lingüísticos y del lenguaje. A futuro,

se prevé tanto adaptar la misma estructura a diferentes lenguas como ampliar el abanico de estructuras disponibles para los usuarios de Ludibot.

En cuanto al contexto lúdico, éste resulta de la suma compleja de numerosos factores. Las determinaciones histórica e ideológica remiten al contexto histórico, cultural y social: Ludibot y los juegos que de él derivan son productos propios del siglo XXI, por lo que obedecen a una lógica propia de nuestra época, con sus prácticas, sus creencias y sus representaciones. Son asimismo productos que, si bien están destinados a un uso amplio, tienen origen geográficamente en un país, una ciudad, instituciones específicas: para bien y para mal, no serán percibidos de la misma manera que si hubieran sido diseñados en otros lugares. Los postulados para el desarrollo de Ludibot, que se han ido presentado a lo largo de estas páginas, contribuyen también a la definición de un contexto lúdico específico. Tal es el caso, por ejemplo, del proceso de aculturación que se espera propiciar.

Respecto a las condiciones concretas materiales y espacio-temporales, Ludibot ha sido diseñado para ser utilizado en espacios preferentemente cerrados, con iluminación uniforme. Su desplazamiento es posible sobre superficies uniformes, a lo largo y ancho de espacios libres de por lo menos 9m<sup>2</sup>. Tales condiciones suelen estar reunidas en los espacios a los que está destinado Ludibot, a saber, aulas escolares, mediatecas y centros de autoacceso. De no ser así, el robot puede ser reprogramado para adecuarse a un contexto específico. El robot permite aplicaciones individuales o grupales, de preferencia en grupos reducidos, y busca promover no sólo una interacción humano-máquina atractiva y fácil de aprender sino también la interacción entre usuarios.

Todos los elementos anteriormente citados tienen como objetivo fomentar la aparición de la actitud lúdica, de tal manera que los participantes, al estar convencidos de estar jugando, adopten hacia la actividad una disposición favorable, asociada con la implicación y la motivación. Así, se pretende que la introducción del robot en una clase de idiomas contribuya al proceso de aprendizaje, ayudando a incrementar el compromiso por aprender una segunda lengua a través de aplicaciones y dinámicas lúdicas específicas.

Antes de desarrollar la reflexión en torno a la manera en que robótica, DLC y ciencias del juego contribuyen a generar un campo de intersección disciplinar específico, es preciso abordar otra vez la cuestión de la interacción, vista esta vez desde el ángulo del juego, con el fin de delimitar mejor el o los tipos de interacción considerados para Ludibot.

### **1.3.3 Interacción desde el juego**

Para entender el rol de la interacción desde el juego, se considera aquí una perspectiva social, siguiendo a Halliday y Kumaravadivelu [185][186], quienes estudian la interacción durante juego desde un enfoque ideacional, interpersonal y textual. Por su parte, Salen y Zimmerman [185] han planteado la existencia de tres tipos de interacción lúdica: la interacción formal con el sistema de juego, la interacción social con otros jugadores y la interacción cultural con un conjunto de comunidades y culturas.

En el ámbito de juego, la interacción es el resultado del diseño interactivo desde el cual es posible crear condiciones para la interacción. Sin embargo, no se puede controlar la interacción por completo, pues el jugador elegirá en cualquier momento interactuar o no con el juego según las modalidades previstas. Así, existen en el juego una interacción ideal y una interacción real, que no siempre coincidirán; el diseño de las actividades de juego debe tomar en cuenta ese desfase.

Sykes y Reinhardt [185], quienes han estudiado el juego digital aplicado a la enseñanza/aprendizaje de idiomas, mencionan que la interactividad, en muchos casos, es un elemento esencial en los juegos. Esta característica lo distingue de otros tipos de soportes de entretenimiento, como libros o películas, que no suelen ser explícitamente participativos. Según Juul, Salen, Schell y Zimmerman [185][186], el juego pone en marcha cuatro tipos de interacción: cognitiva, funcional, explícita y cultural.

La interacción cognitiva ocurre en la mente del jugador, al igual que en la mente del lector, del oyente de música o del espectador. Un juego cognitivamente interactivo busca atraer a los jugadores al sumergirlos en una experiencia visual y/o auditiva e inclusive táctil. Al igual que los libros, las canciones o las películas, los juegos incorporan textos, imágenes, palabras,



narraciones, entre otros elementos; ahora bien, a diferencia de lo que ocurre con otros medios, los diseñadores de juegos deben considerar que los jugadores podrán interactuar activamente con los elementos mencionados e incluso modificarlos o modificar sus reglas de uso.

La interacción funcional se refiere a la interfaz de un juego y a la manera en que ésta interactúa físicamente con el jugador. Lo anterior puede darse mediante el uso de botones, menús y otras estructuras, tangibles o no. El diseño de la interfaz es una fase primordial para asegurar el éxito de un juego, ya que debe la interfaz debe ser fácil de usar, intuitiva y ergonómica.

La interacción explícita se refiere a las estructuras diseñadas en el juego en sí, que no forman parte de la interfaz sino se basan en la elección y la toma de decisiones por parte del jugador. En otras palabras, el jugador elige ciertos aspectos del juego y decide si el juego es motivador, si el juego es de interés y atractivo, si el juego es fácil, entre otros aspectos.

La interacción cultural se refiere a la creación de o a la participación en un juego afín a una cultura y una comunidad en particular. Los autores [185][186] remiten al ejemplo de los juegos multijugadores en línea y presenciales, que fomentan prácticas comunicativas: en los juegos de videos basados en civilizaciones, como *Age of Empires* o *Age of Mythology*, en los juegos basados en guerras, como *Halo* o *Gears of Wars*, y en algunos juegos de mesa como *Monopoly*, los jugadores comparten y/o intercambian conocimientos y consejos,

Se dice que la interacción cognitiva y explícita permiten la interacción cultural. Por ejemplo, un juego basado en contextos artísticos puede dar paso a prácticas socioculturales basadas en nuevas prácticas, o simplemente un juego multijugador puede generar comunidades con prácticas de interacción únicas que se aprenden y negocian tanto dentro como fuera del juego.

Estas diferentes interacciones aparecen en el caso de Ludibot: la interacción cognitiva incita al usuario a utilizar la mente y el cuerpo, mediante una experiencia visual, auditiva y táctil. La interacción funcional está asociada con las aplicaciones lúdicas y con una interfaz humano-máquina basada en un modelo de estados y transiciones que incentiva la comunicación verbal y no verbal. Se busca que el usuario desarrolle la interacción explícita, ya que mientras juega se

espera que la actividad lo motive y que desarrolle un compromiso propio para usar el robot con un objetivo de aprendizaje. Además, la interacción cultural está presente debido a que las aplicaciones lúdicas desarrolladas mediante el robot están enfocadas a generar nuevas dinámicas de aprendizaje propias de un público adolescente familiarizado con cierto patrimonio lúdico común.

Durante el diseño de las aplicaciones lúdicas, se busca asimismo generar nuevas dinámicas destinadas a comunidad de estudiantes en proceso de aprendizaje de una segunda lengua, durante las cuales les sea posible intercambiar impresiones, consejos, experiencias y conocimientos. Tales prácticas de interacción tienen por objetivo permitir a los usuarios aprender una segunda lengua a través del juego, dentro o fuera de un salón de clases, promoviendo el desarrollo de habilidades y competencias diferentes a las consideradas por la educación tradicional. Se trata, por ende, de poner al servicio del aprendizaje de una lengua extranjera los recursos de la robótica y del juego, para aprovechar el potencial de la robótica ludoeducativa. Tal será el tema del quinto y último subcapítulo de este marco teórico y conceptual.

## **1.4 Estado del arte de la robótica, aprendizaje de lenguas y juego**

Una vez establecido el marco teórico y conceptual para el desarrollo de Ludibot, con base en la robótica, la DLC y las ciencias del juego, es importante presentar un estado del arte de los trabajos de investigación que asocian las tres disciplinas mencionadas. En este subcapítulo, se brinda un balance de los trabajos ya desarrollados, con una doble intención: identificar los principales problemas que no han sido resueltos y contar con más herramientas para mejorar el desarrollo y la implementación de Ludibot, tanto desde una óptica teórica y conceptual como desde una perspectiva práctica apuntalada por un enfoque pluridisciplinar, pluricultural y plurilingüe.

A continuación, se presentan aquellos trabajos de investigación identificados durante la revisión de la literatura, según tres categorías: aquellos que exploran la articulación entre la robótica y el aprendizaje de lenguas, aquellos que combinan robótica y juego y, finalmente, aquellos que asocian el juego y el aprendizaje de lenguas. Cabe destacar que, según los resultados de la exploración realizada, ninguno de los trabajos existentes combina plenamente la robótica, la

DLC y las ciencias del juego. Se consideran los aportes más actuales de las tres disciplinas involucradas, con el fin de diseñar y desarrollar herramientas teóricas y prácticas pertinentes y eficaces considerando las tres principales variables de esta tesis: la robótica móvil, el aprendizaje de lenguas y el juego como herramienta de apoyo a dicho aprendizaje para el desarrollo de Ludibot.

### **1.4.1 Robótica y aprendizaje de lenguas**

En cuanto a la robótica aplicada al aprendizaje de lenguas, se han encontrado trabajos de robótica educativa que permite concebir, diseñar y desarrollar robots que apoyen el proceso de enseñanza/aprendizaje [182][183]. Tienen como objetivo despertar el interés de los estudiantes mediante la implementación de herramientas tecnológicas que transformen las asignaturas tradicionales gracias a dinámicas atractivas e integradoras destinadas a crear entornos de aprendizaje favorables. Sin embargo, en muchos casos, tales desarrollos tecnológicos no están respaldados por una adecuada problematización de los aspectos didácticos y pedagógicos, y abarcan generalmente áreas no humanísticas, tales como matemáticas, ciencias naturales y tecnología.

En el ámbito específico de la DLC, se han diseñado algunos robots educativos tales como el de Steels [180][181], que combina inteligencia artificial y juegos de lengua a través de secuencias de interacción verbal entre dos agentes situados en un entorno dado. Sin embargo, no se trata de implementar dinámicas lúdicas para el aprendizaje, ya que la acepción de “juegos de lengua” remite más a la actualización de elementos lingüísticos que a una interacción lúdica. En realidad, la mayoría de las aplicaciones para el aprendizaje de lenguas con robots han sido concebidas para su implementación mediante el robot Nao [86][148] (véase figura 1.8). Nao ha sido ampliamente utilizado en diferentes disciplinas; ejemplo de ello es la propuesta de Veloso *et al.* [200], donde ha sido empleado para un juego de fútbol sin la intervención del usuario.



**Figura 1.8. Robots Nao como asistentes para el aprendizaje del idioma inglés**

En la mayor parte de los casos, como se había mencionado con respecto a la robótica y su relación con los juegos serios, se opta por un enfoque conductista. Los enfoques elegidos se diferencian del enfoque desarrollado para Ludibot por el hecho de que este último pone énfasis en la interacción humano-máquina y humano-humano, con el fin de promover el aprendizaje en un entorno donde el robot es parte de un sistema más amplio.

Además, la mayoría de las aplicaciones identificadas han sido desarrolladas en el idioma inglés, dada su importancia en el contexto sociocultural actual, dejando de lado otras lenguas. Por su parte, Ludibot pretende funcionar para un aula de idiomas en general, partiendo del idioma francés, pero sin cerrarse a la posibilidad de ser adaptado a futuro para la inclusión de otras lenguas, abriendo así la robótica a un enfoque plurilingüe y pluricultural.

Ludibot busca aprovechar los recursos tecnológicos para brindar herramientas atractivas, interactivas y de fácil uso para el aprendizaje de lenguas. Por esta razón, se ha optado por una perspectiva crítica hacia lo que Genvo sugiere llamar ludicización antes que gamificación [63], integrando además los trabajos realizados por Alvarez-Chaumette [5], Bouko-Alvarez [24] y Deterding *et al.* [46], tema abordado en ciencias del juego.

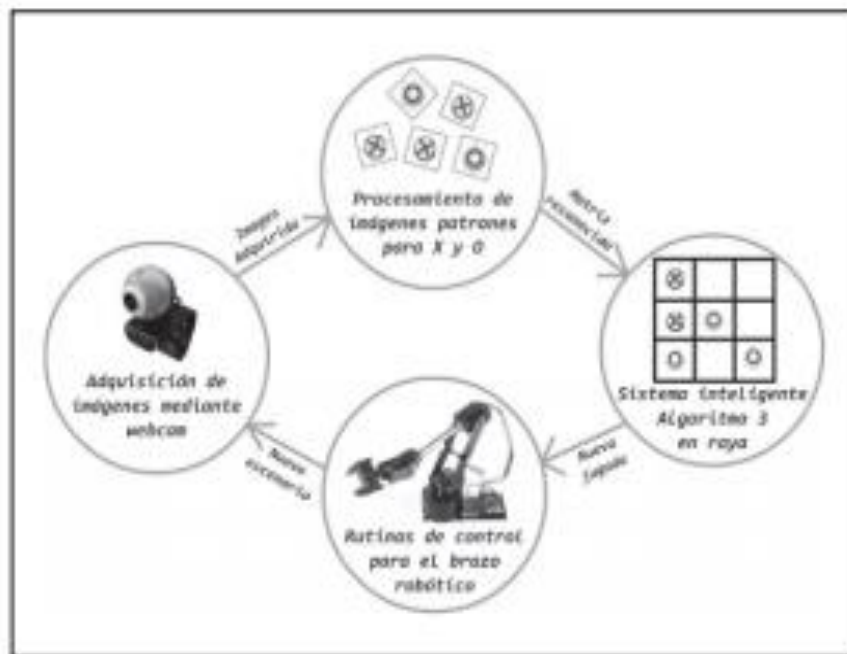
### **1.4.2 Robótica y juego**

La combinación de la robótica con el juego ha permitido desarrollar numerosas aplicaciones de robótica lúdica, que se ha convertido en una rama de la robótica que explora la interacción

humano-máquina a través de juegos. La robótica lúdica no está condicionada a la invención de juegos nuevos; de hecho, en muchos casos, sólo se hace uso de la tecnología para adaptar juegos ya existentes. Tampoco se requiere la presencia de un aprendizaje fuera de la lógica de la propia actividad, puntos esenciales que distinguen esta área de los objetivos de Ludibot.

Dentro de los trabajos de investigación en la intersección entre robótica y juego, es posible distinguir algunos proyectos que involucran robots destinados a juegos de estrategia con base en la interacción con un usuario. Entre ellos, destacan:

1. El robot diseñado para jugar gato contra un humano desarrollado por Galán *et al.* [59]. En este caso, los autores emplean una webcam para identificar las marcas “x” y “o” en un tablero de 3x3 mediante un brazo robótico (véase figura 1.9).



**Figura 1.9. Esquema del brazo robótico aplicado al juego de gato con un humano**

2. El juego de damas implementado por Aliane *et al.* [3] mediante la intervención de un robot manipulador. Está integrado por un control del robot, una comunicación con el controlador del robot y un subsistema de visión (véase figura 1.10).



**Figura 1.10. Robot manipulador diseñado para un juego de damas mediante sistema de visión**

3. Los robots móviles programados para jugar ajedrez chino, descritos en Su *et al.* [184]. Poseen una interfaz de comunicación gracias a la cual se procesan y se envían instrucciones que indican las posiciones deseadas según los requerimientos del juego (véase figura 1.11).



**Figura 1.11. Robots móviles usados para un juego de ajedrez chino**

Estos ejemplos evidencian el resultado de un proceso de ludicización aplicado a la robótica [144][148]. En ellos, se busca renovar juegos clásicos de estrategia mediante el uso de robots. A pesar de esto, en pocos casos logran un cambio de actitud en el usuario, es decir, motivarlo a

jugar de manera divertida y afectiva, puesto que no presentan una interacción significativa humano-máquina y no se problematiza el concepto de juego.

En los ejemplos mencionados, los robots correspondientes sólo actúan ante cambios procesados y detectados por la cámara y conservan una función de computador. En otras palabras, la máquina sólo es usada para calcular combinaciones propicias al juego y reta al humano sólo por la velocidad y la amplitud del procesamiento de datos. Por lo tanto, se anula en ocasiones la actitud lúdica nacida de la incertidumbre, tal como ocurre en el modelo de Galán *et al.* [59], cuyo robot es infalible e incapaz de ser vencido en su juego.

En el caso de Ludibot, se toma distancia de este tipo de diseño, pues es un robot que interactúa con el usuario a través de sus movimientos y no únicamente mediante el reconocimiento de posiciones de juego en un espacio de trabajo. En un nivel más profundo de interacción con el usuario destacan las propuestas siguientes:

1. En Faust *et al.* [55], se utilizan las tecnologías de diseño usuales en los juegos de video en un entorno dentro del cual los jugadores experimentan, cooperan y se enfrentan entre sí para resolver problemas y retos mediante equipos de robots móviles (véase figura 1.12).



**Figura 1.12. Robots móviles desarrollados por Faust *et al.***

2. En Kose *et al.* [98], se usa el robot Nao (véase figura 1.13) como asistente para un juego basado en signos de lenguaje mediante el uso de la interacción y la imitación. Dicho lenguaje está basado en gestos con las manos.



**Figura 1.13. Robots Nao empleados como asistentes en un juego basado en signos de lenguaje**

3. En Hansen *et al.* [77][78], los autores presentan un robot destinado a la rehabilitación de personas de la tercera edad a partir la utilización de juegos de persecución. En este contexto, el robot es empleado para una actividad cuyo fin es lograr mantener la atención de los usuarios mediante expresiones (véase figura 1.14).



**Figura 1.14. Robot destinado a la rehabilitación de personas de la tercera edad**

4. En Negrete *et al.* [133] se usa un robot de servicio, llamado Justina, diseñado y programado para la navegación en un laboratorio, logrando la interacción del robot con el entorno (véase figura 1.15)

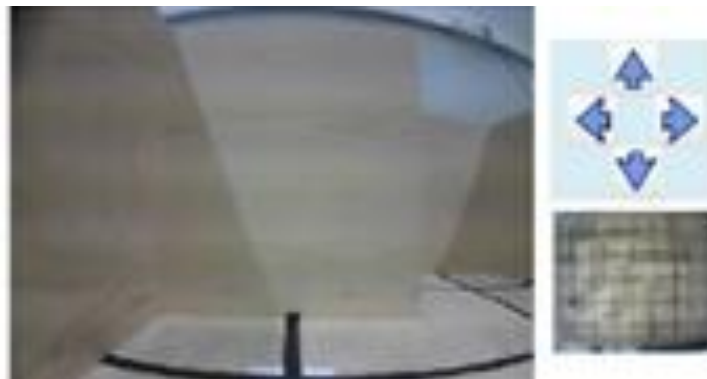




**Figura 1.15. Robot Justina y laboratorio donde es usado para la navegación**

En cuanto a la asociación entre robótica y juegos más cercana de este proyecto de tesis, existen tres precedentes:

1. La plataforma abierta diseñada por Iturrate *et al.* [88], que permite controlar a distancia un robot móvil dentro de un laberinto mediante preguntas con fin pedagógico (véase figura 1.16).



**Figura 1.16. Plataforma de un robot móvil dentro de un laberinto**

2. La arquitectura orientada al servicio para juegos serios concebida por Carvalho *et al.* [37], destinada al aprendizaje de la teoría de probabilidad utilizando un esquema de preguntas y respuestas (véase figura 1.17).

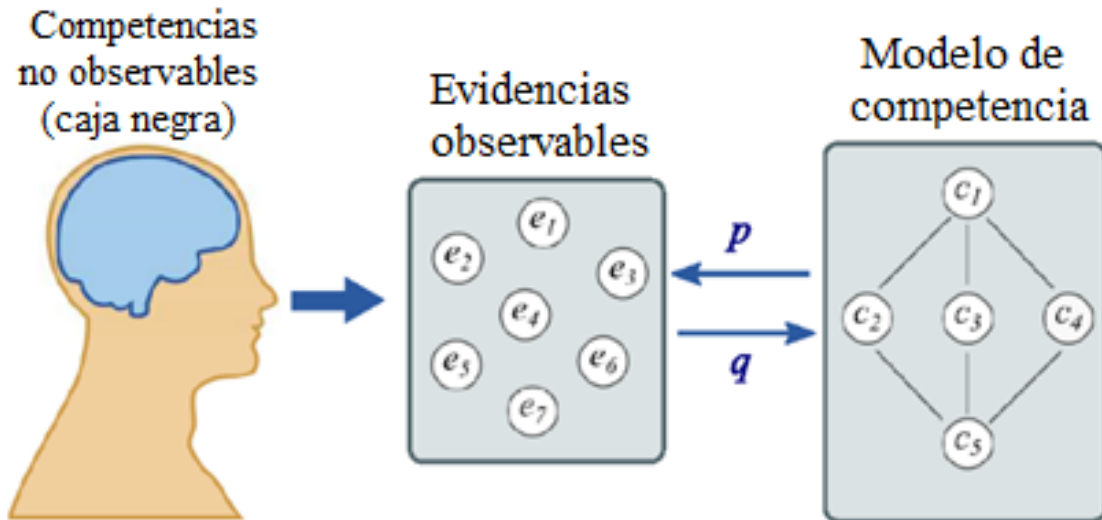


Figura 1.17. Arquitectura orientada al servicio para juegos serios

3. La propuesta de Qingtang *et al.* [142], quienes diseñaron un juego en el cual es preciso controlar un avatar con movimientos del cuerpo mediante el uso de Kinect (véase figura 1.18).



Figura 1.18. Juego basado en movimientos del cuerpo con el sensor Kinect

Estos ejemplos, a pesar de expresar cierta preocupación por la enseñanza a través de los juegos, están basados en una dinámica de acción-reacción mecánica, aplicando técnicas de corte conductista. Por lo tanto, el proceso de ludicización no se lleva a cabo de manera idónea y se mantiene en el nivel de la mera gamificación. El modelo de aprendizaje sigue siendo tradicional, pues contempla pregunta y respuesta, y solamente se agrega una acción del robot.

Es importante subrayar que los desarrollos tecnológicos arriba mencionados no siempre están respaldados por una adecuada problematización de los aspectos didácticos, pedagógicos y lúdicos, y que ninguno pertenece específicamente al área de la DLC. Pocos autores toman en cuenta la complejidad de la red semántica asociada a la noción de juego como metáfora heurística [79][80][169], por lo que la mayor parte de los trabajos omiten referirse propiamente a aplicaciones lúdicas destinadas a favorecer el aprendizaje de un idioma.

### **1.4.3 Juego y aprendizaje de lenguas**

En el cruce del juego y el aprendizaje aparece la didáctica lúdica, enfocada a la aplicación de juegos a la didáctica. Mediante técnicas participativas de enseñanza encaminadas a estimular el aprendizaje, no sólo se busca propiciar el uso del conocimiento y el desarrollo de habilidades, sino que en muchos casos se contribuye al logro de la motivación extrínseca e intrínseca.

Respecto al contacto entre el juego y el aprendizaje de lenguas, resulta de gran importancia el trabajo de Yanes y Bououd [211], que realiza el análisis de una taxonomía basada en el uso de la gamificación y juegos serios para el aprendizaje del idioma inglés, presentando fortalezas, oportunidades, debilidades y amenazas. Sin embargo, en él no se exponen aplicaciones ni desarrollos experimentales de la clasificación propuesta, dejando una laguna práctica que se presenta como un área de oportunidad.

A propósito de las aplicaciones destinadas a la DLC mediante juegos, destacan los siguientes antecedentes:

1. Wu *et al.* [209] desarrollan un juego interactivo para el aprendizaje de una segunda lengua (véase figura 1.19). El juego está basado en la interacción humano-máquina al incluir animaciones e imágenes interactivas para niños de preescolar valiéndose del uso del Kinect. No se utiliza como tal un robot pese a lo que enuncia el título del trabajo.



**Figura 1.19. Juego interactivo para niños de preescolar basado en Kinect**

2. Kitichaiwat *et al.* [97] implementan un juego basado en los juegos de videos para el aprendizaje del idioma inglés mediante sonidos. Sin embargo, no se refieren propiamente a actividades lúdicas destinadas a favorecer el aprendizaje de una lengua (véase figura 1.20). Como puede apreciarse, nuevamente en esta área destaca el inglés como lengua dominante y se omite la problematización del concepto de juego.



**Figura 1.20. Juego para el aprendizaje del idioma inglés basado en videojuegos**

Tras la revisión bibliográfica anteriormente expuesta, se puede apreciar el indiscutible interés por aplicar la tecnología en los procesos de enseñanza y de aprendizaje a través del juego. No obstante, los productos mencionados no suelen estar respaldados por una adecuada

problematización de los aspectos didácticos, pedagógicos y lúdicos. Por añadidura, pocos están enfocados a la DLC y aún menos a la problematización de juego [168][169], aspectos centrales en el desarrollo de Ludibot. Lo anterior fundamenta la relevancia de integrar de manera equilibrada la robótica móvil, el aprendizaje de lenguas y el juego.

En efecto, los trabajos mencionados en este subcapítulo han tomado como antecedente principal la tecnología: pese a estar presentes, el juego y el aprendizaje de lenguas pocas veces son objeto de una reflexión específica. Por ello, no es posible hablar de una visión pluridisciplinar e integradora. El balance realizado confirma la hipótesis de Silva [169], quien considera que el punto de encuentro entre estas tres áreas no ha sido explorado suficientemente. Con el objetivo de llenar este vacío teórico, en esta tesis se consideran las tres disciplinas en el marco teórico y conceptual para Ludibot.

Cabe asimismo subrayar el impacto actual de las tecnologías, presentes en la mayor parte de las actividades que una persona realiza de manera cotidiana, incluso en áreas donde tradicionalmente poseían una injerencia menor. Tal es el caso del campo de la educación, donde uno de los indicadores más claros del impacto de las tecnologías ha sido la gran demanda de dispositivos electrónicos, tales como computadoras, tabletas y teléfonos inteligentes. Esto revela la diversificación creciente de herramientas tecnológicas con el fin de facilitar la enseñanza y el aprendizaje. Este fenómeno representa hoy en día numerosos desafíos, entre ellos, el replanteamiento de las modalidades y los contenidos de la formación docente, una nueva visión del papel que deben desempeñar aprendientes y docentes, y una noción renovada y más compleja del proceso de aprendizaje, que incluye, por ejemplo, el reforzamiento de la literacidad mediática multimodal [103][132][151].

En este capítulo, se ha propuesto una revisión de los principales conceptos que aporta cada una de las disciplinas consideradas y se ha explorado de qué manera los trabajos anteriores han explorado la intersección pluridisciplinar desde la cual se desarrolla Ludibot. Todo ello permite ahora proponer una reflexión específica en torno a los conceptos de robótica ludoeducativa y de robot ludoeducativo, como corolario del marco teórico y conceptual de esta investigación.

## 1.5. Robótica ludoeducativa

Hasta hoy, según se ha expuesto más arriba, existen pocos estudios que problematicen adecuadamente la articulación pluridisciplinar entre la robótica, la DLC y las ciencias del juego. Los diversos trabajos previos suelen combinar sólo dos de las tres disciplinas consideradas, realizando estudios sobre robótica educativa, robótica lúdica [136] y/o didáctica lúdica. La robótica ludoeducativa es por lo tanto un campo poco estudiado, que requiere ser abordado aquí de manera más detallada.

### 1.5.1 Definición de la robótica ludoeducativa

Para definir la robótica ludoeducativa, es preciso partir de los tres elementos conceptuales que la componen: robótica, ludo- y educativa. La robótica [151] [179] es dada como la disciplina que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia. Por su parte, el prefijo ludo- remite al campo de lo lúdico [151], derivado etimológicamente del sustantivo latino *ludus* (juego); así designa, todo lo relativo a la metáfora de juego, trátase de materiales, estructuras, contextos y/o actitudes. Finalmente, “educativa” [151] remite a aquello que sirve para educar. Etimológicamente, la palabra proviene del latín *educare* [38], criar, formar, instruir en alguna actividad.

Así, la robótica ludoeducativa engloba los robots con fines educativos que recurren a aplicaciones lúdicas con fines de enseñanza y aprendizaje. Ahora bien, esta denominación no hace evidente un aspecto básico de la robótica ludoeducativa, a saber, la interacción. En efecto, los robots arriba mencionados requieren una interfaz humano-máquina capaz de procesar la información de su entorno, detectar al usuario y establecer movimientos específicos. Dicha interfaz debe estar equipada con cámaras, sensores y actuadores controlados por una computadora, un procesador, un microprocesador o un sistema que le permita ser programado.

Por ende, podemos reformular y completar la definición anterior para postular que la robótica ludoeducativa es aquella rama de la robótica que busca aprovechar la interacción humano-máquina integrando consideraciones didácticas y lúdicas.

La robótica ludoeducativa aparece entonces como una disciplina nueva y específica, constituida a partir de la sinergia razonada de la robótica, la didáctica y las ciencias del juego para promover una interacción eficaz. Consta de diferentes subáreas, entre las cuales está aquella que está en el meollo de este trabajo: la robótica ludoeducativa aplicada a la enseñanza/aprendizaje de lenguas extranjeras.

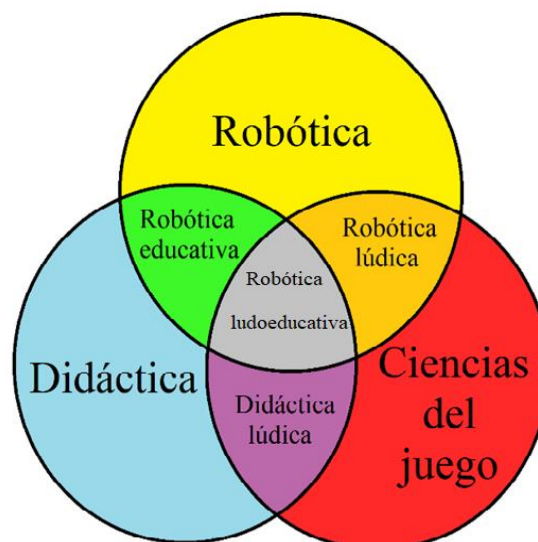
En otras palabras, la robótica ludoeducativa busca concebir, diseñar y desarrollar aplicaciones lúdicas que promuevan el aprendizaje mediante el diseño y construcción de robots capaces de desempeñar tareas específicas, mediante el uso de sistemas de control y la programación. Contribuye a complementar el proceso de aprendizaje/enseñanza gracias al uso de las tecnologías aplicadas en situaciones formales, semiformales o informales de aprendizaje, mediante la introducción de nuevas dinámicas y nuevas actitudes, recurriendo a y promoviendo el uso de la literacidad mediática multimodal [103][132][151] y de las competencias del siglo XXI [6][207][208][71].

Durante el desarrollo del marco teórico y conceptual, se puso especial énfasis en el concepto de interacción, cuya importancia se ha subrayado también en la segunda definición de robótica ludoeducativa. Desde la robótica ludoeducativa según ha sido postulada aquí, la interacción ludoeducativa puede ser definida como un sistema de intercambio comunicativo bi o multidireccional: la interacción se da ya sea entre el usuario y el robot ludoeducativo, ya sea entre los usuarios entre sí, ya sea entre los usuarios y el robot, sin que una de estas opciones excluya a la(s) otra(s). Las partes que intervienen en la interacción se influyen mutuamente, en un intercambio de acciones basadas en la aplicación ludoeducativa, mediante el uso de la comunicación verbal y no verbal. Esta última comprende ademanes, gestos, movimientos, sonidos expresivos, símbolos, figuras, entre otros, siempre y cuando expresen algo [193][195]. Físicamente, la interfaz puede estar integrada mediante botones, menús y/u otros elementos tangibles o no.

En resumen, la robótica ludoeducativa tiene por objetivo mejorar la calidad del aprendizaje mediante el uso activo de robots como material lúdico, promoviendo nuevas experiencias basadas en una interacción comunicativo bi- o multidireccional propicia a la

aplicación de nuevos enfoques de aprendizaje, tales como la literacidad mediática multimodal y las competencias del siglo XXI.

En la figura 1.21 se presenta un diagrama representativo de la sinergia de cada una de las disciplinas considerando en la intersección entre ellas a la robótica ludoeducativa. Es importante destacar que, contrariamente al diagrama aquí ilustrado, en los robots concebidos desde la robótica ludoeducativa no necesariamente deben aparecer representadas las tres disciplinas de manera proporcional. El grado de presencia de cada una dependerá del objetivo de las aplicaciones lúdicas, del tipo de interacción, del tipo de interfaz, de los elementos de literacidad mediática multimodal utilizados, del tipo de robot desarrollado, y por supuesto, del área disciplinar de aprendizaje en la cual será utilizado el robot y de la o las teorías del aprendizaje que sustenten el conjunto.



**Figura 1.21. La intersección disciplinar de la robótica ludoeducativa**

### **1.5.2 Robot ludoeducativo**

Definir el concepto de robótica ludoeducativa permite a su vez definir el concepto de robot ludoeducativo. Se propone aquí entender por robot ludoeducativo toda máquina programable destinada al desarrollo de aplicaciones lúdicas con fines de aprendizaje y constituida por una computadora, un procesador, un microprocesador y/o un microcontrolador, además de sensores, actuadores y/o cámaras, así como un sistema de control. Dicha máquina emplea una interfaz



humano-máquina que permite una comunicación específica con el usuario y puede hacer uso de sonidos expresivos, símbolos, figuras, entre otros. Los robots ludoeducativos son capaces de percibir tanto las características del entorno como los movimientos realizados por el usuario.

El propósito principal de un robot ludoeducativo debe pues consistir en promover un aprendizaje más efectivo, al tomar en consideración variables tales como el entorno de aprendizaje, la participación en grupo, la interacción frecuente, la retroalimentación y las conexiones con el contexto del mundo real [154] mediante aplicaciones ludoeducativas.

### **1.5.3 Aplicación ludoeducativa**

Una aplicación ludoeducativa es dada como un programa o conjunto de programas informáticos respaldados por una estructura lúdica, con fines de aprendizaje. El sistema de reglas propio de cada aplicación ludoeducativa pone en marcha un cierto número de mecanismos y de principios precisos. La aplicación ludoeducativa tiene un contexto lúdico dado por un conjunto de elementos objetivos externos al juego. Finalmente, la aplicación ludoeducativa busca promover una actitud lúdica dada como la disposición de ánimo del jugador, vista como la convicción interna del jugador respecto al sentido de sus acciones con el juego.

Al combinar el juego con las herramientas tecnológicas, se busca que los estudiantes tengan la oportunidad de ampliar su experiencia gracias a un aumento en su motivación. Cuando su uso se da en una situación formal o semiformal, es posible lograr que el aula tradicional se convierta en un espacio renovado y atractivo, en donde tengan lugar actividades significativas, de carácter colaborativo y creativo, destinadas a afianzar el aprendizaje. En ellas, el significado es construido de tal manera que el individuo interactúa de forma significativa con el mundo que lo rodea. Esto implica dar la prioridad a las actividades grupales, de manera que exista una participación colaborativa.

Hasta fechas recientes, la enseñanza se había visto condicionada en gran parte por las herramientas educativas más usuales que se encontraban disponibles, tales como lápiz, papel y pizarrón. Las reflexiones sobre la educación del futuro [207][208] destacan cómo los sistemas

tecnológicos, adecuadamente configurados y utilizados, pueden resultar más poderosos, accesibles, motivantes y significativos.

Según los términos de la definición propuesta más arriba, Ludibot es un robot ludoeducativo, que tiene como fin promover el aprendizaje activo de una lengua a partir de la experiencia propia de cada usuario. Su diseño integral, basado en la robótica, la DLC y las ciencias del juego, favorece la interacción humano-máquina mediante el uso de aplicaciones que favorecen la comunicación verbal y no verbal, para el proceso de aprendizaje de idiomas en situaciones formales y semiformales.

Para asegurar una óptima integración de la robótica en el ámbito educativo, es necesario indagar las necesidades del personal encargado de los procesos de enseñanza. De acuerdo con las necesidades de Ludibot, el prototipo robótico debe contar con las siguientes características: una estructura mecánica; una programación fácil propicia al diseño de nuevas aplicaciones lúdicas para el aprendizaje; la posibilidad de desplazarse en aula de idiomas, mediatecas y/o centros de autoacceso con el objetivo de realizar diferentes trayectorias específicas.

En cuanto a teorías del aprendizaje, Ludibot está basado en los principios del cognoscitivismo, el constructivismo y el interaccionismo. Del cognoscitivismo se retoma la idea según la cual el aprendizaje resulta de los procesos mentales internos del individuo, pues las acciones del ser humano no están determinadas por las propiedades de los fenómenos del medio ambiente, sino por la interpretación que el sujeto hace de ellas. Del constructivismo, se adoptan los postulados según los cuales el aprendizaje es dependiente de la cantidad y la calidad de las estructuras de organización en una persona, y el ambiente de aprendizaje puede tener múltiples efectos en la constitución de la realidad [207][208] y del conocimiento. Finalmente, del interaccionismo [43] se aprovecha la idea de no establecer relaciones de determinismo mecánico, sino de considerar al individuo como un sujeto activo: el entorno y la persona interactúan y se complementan mutuamente. Las tres teorías están centradas en un sujeto activo que construye su propio conocimiento, enfocándose en tareas que suelen tener relevancia y utilidad en el mundo real. Se vinculan así con la literacidad mediática multimodal y las competencias del siglo XXI [6][207][208][71], que también insisten en ese papel activo del sujeto de aprendizaje. En efecto, la literacidad mediática multimodal subraya la importancia de proveer a los aprendientes

herramientas para la recepción, pero también para la interpretación y la creación de significado, vehiculado hoy según modos cada día más diversos y complejos, mientras que el desarrollo de las competencias del siglo XXI exige actividades significativas de aprendizaje basadas en el uso de tecnologías recientes para expandir la capacidad de crear, compartir, aprender y dominar el conocimiento [158].

Con base en la propuesta de Hilgard [81] y las preguntas hechas anteriormente en este capítulo sobre el aprendizaje, Ludibot toma en cuenta la capacidad de aprender y las diferencias individuales. Parte de la idea de que las herramientas lúdicas para el aprendizaje pueden ayudar a motivar el aprendizaje, siempre y cuando se tenga presente la complejidad tanto del proceso de aprendizaje como de la metáfora lúdica: no todo lo que se llama juego produce los efectos esperados de un juego.

Ludibot busca brindar un material, unas estructuras y un contexto propicios para que sus usuarios aprendan de manera fácil, atractiva y duradera, mediante un aprendizaje significativo. Si bien los usuarios principales son aprendientes de idiomas en situaciones formales o semiformales, cualquier persona que lo desee puede utilizarlo para aprender. Así, Ludibot será designado como un material de apoyo que pretende fortalecer en el aprendiente motivaciones intrínsecas y extrínsecas.

El tipo de aprendizaje promovido por Ludibot es activo y cognitivo, gracias a actividades lúdicas que requieren una interacción bi y multidireccional, en lugar de una observación pasiva o una secuencia de explicación-repetición, tal como ocurre con los usos identificados del robot Nao, inspirados en un enfoque conductista [86]. En suma, Ludibot se basa en dos principios educativos generales: el aprendizaje como un proceso activo y el aprendizaje completo, auténtico y real según Piaget [138]. Por lo tanto, desde el enfoque orientado a la acción en la enseñanza de lenguas [85], incluye tanto el desarrollo de competencias propiamente lingüísticas como el desarrollo de competencias generales y de comunicación.

Por añadidura, un enfoque constructivista apoyado por el interaccionismo invita a considerar al profesor como un guía y mentor, otorgándole al alumno la libertad necesaria para explorar el ambiente tecnológico y construir su conocimiento, brindándole apoyo en caso de que

tenga dudas o se enfrente a algún problema. Desde el enfoque del interaccionismo, la interacción de los estudiantes juega un papel muy importante, por supuesto, máxime que las tecnologías recientes han generalizado el uso de sensores capaces de medir diversas variables de interés, relativas tanto al entorno como al usuario y/o al robot.

Por su parte, las interfaces humano-máquina son capaces de procesar informaciones relativas al usuario, permitiendo una comunicación específica entre el usuario y la máquina [52][175]. Entre los dispositivos destinados a medir las variables mencionadas y a controlar un robot móvil y sus acciones se encuentran los joysticks [101], los dispositivos hápticos [210], las cámaras [209], los teléfonos inteligentes [190][215], los relojes inteligentes [201], por sólo citar los más comunes.

En el caso particular de Ludibot, la plataforma usa una interfaz humano-robot móvil basada en gestos y ademanes. El reconocimiento opera mediante Kinect v2, y permite realizar el control de movimientos y de acciones bajo el esquema de una máquina de estados.

En el siguiente capítulo, se aborda la arquitectura de sistemas propia de un robot ludoeducativo, con el fin de presentar con detalle los dos prototipos de Ludibot. Se presenta asimismo un modelo de análisis, basado en el marco teórico y conceptual pluridisciplinar propuesto en este capítulo, con miras a determinar si los prototipos son o no ludoeducativos.

## **Capítulo 2. Ludibot: arquitectura y carácter ludoeducativo**

En este capítulo se presentan las características de los dos prototipos de Ludibot para determinar, con base en el marco teórico y conceptual propuesto en el capítulo 1, su carácter ludoeducativo. En primer lugar, se presentan las definiciones de sistema y de arquitectura de sistemas, con el fin de esbozar la definición de la arquitectura específica para un robot ludoeducativo. Posteriormente, se establecen las características de la arquitectura de sistemas y el comportamiento del robot necesarios para que se pueda concretar el carácter ludoeducativo. Más adelante, se utiliza la propuesta de arquitectura de un robot ludoeducativo para analizar las dos versiones de Ludibot. Finalmente, se señalan pistas de reflexión para enriquecer el desarrollo en curso de la segunda versión de la plataforma, Ludibot v2.

### **2.1 Arquitectura de sistemas para un robot ludoeducativo**

Para establecer la arquitectura de sistemas propia de un robot ludoeducativo, comenzaremos por dos definiciones: la de sistema y la de la arquitectura de sistemas.

#### **2.1.1 Sistema**

En nuestro entorno existen conjuntos de cosas que, debido a las relaciones ordenadas que establecen entre sí, contribuyen a constituir un objeto determinado, que desempeña cierta función dada. Tal es el caso de una computadora, un automóvil, una bacteria, una familia o una economía de mercado, entre muchos otros. Según Ogata “un sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado. Un sistema no está necesariamente limitado a los sistemas físicos. El concepto de sistema se puede aplicar a fenómenos abstractos y dinámicos, como los que se encuentran en la economía. Por tanto, la palabra sistema debe interpretarse en un sentido amplio que comprenda sistemas físicos, biológicos, económicos y similares” [135].

De esta manera, puede hablarse de diversas clases de sistemas. Hay, por ejemplo, sistemas conceptuales –también llamados sistemas formales o sistemas ideales–, que remiten a constructos compuestos por conceptos de cuatro tipos: individuos, predicados, conjuntos y operadores. Por su

parte, un sistema de ecuaciones está formado por un conjunto de ecuaciones con más de una incógnita que conforman un problema matemático. Un sistema biológico incluye una red de entidades biológicas, por ejemplo, los cinco reinos *Fungi*, *Plantae*, *Protista*, *Monera* y *Animalia*. En suma, un sistema debe ser entendido como una unidad cuyos elementos interactúan con miras a un objetivo común. Se trata de una entidad que se distingue de lo que la rodea, y que es capaz de mantener esa identidad a lo largo del tiempo y bajo una dinámica cambiante [48].

El estado de un sistema dinámico es el conjunto más pequeño de variables (llamadas variables de estado), conjunto tal que el conocimiento de esas variables en un determinado instante  $t_0$  junto con el conocimiento de los valores de la señal de entrada para los instantes  $t \geq t_0$ , permite determinar el comportamiento y la evolución del sistema para cualquier instante de tiempo  $t \geq t_0$  [49]. La dinámica de un sistema se puede describir en función del valor del vector de estados y de la señal de entrada [54]. La dinámica de un sistema es aquello que interviene para cambiar el estado de un sistema. Comúnmente, los sistemas tienen una dinámica que evoluciona con el tiempo [9]. La dinámica describe cómo cambia el sistema de un estado llamado actual al estado siguiente.

Así, adoptamos aquí una definición de sistema como un conjunto estructurado de componentes diferenciados que actúan juntos para realizar una función determinada, y que poseen además un estado y una dinámica que determinan su comportamiento. Los sistemas pueden tener entradas y salidas con el fin de procesar energía o información. Tales elementos tienen su origen o su destino en otros sistemas: dicho de otra manera, los sistemas interactúan con otros sistemas, dentro de un metasistema, también conocido como arquitectura de sistemas.

### **2.1.2 Arquitectura de sistemas**

La arquitectura de sistemas [45][160], también conocida como “arquitectura sistema” por la traducción literal de su nombre en inglés, es una disciplina que se enfoca al estudio de las propiedades, la estructura y el comportamiento de objetos específicos, los sistemas [128]. En otras palabras, la arquitectura de sistemas estudia las características, las relaciones y los comportamientos de un metasistema o conjunto de sistemas que incluye varios sistemas o subsistemas.

Puesto que la arquitectura de sistemas estudia tanto los metasisistemas como los subsistemas, lo cual puede producir confusión, de ahora en adelante utilizaremos los términos “arquitectura” o “sistema” para referirnos al metasisistema y el término de “objeto” para referirnos a los sistemas o subsistemas que conforman dicha arquitectura.

En la literatura especializada, existen múltiples definiciones de la arquitectura de sistemas. Presentamos y discutimos a continuación siete definiciones [65] [68] [76] [111] [118] [141] [196] a partir de las cuales construimos una definición operatoria para este trabajo. La primera ofrece la base general de discusión; las seis siguientes permiten explorar diferentes aspectos relevantes para nuestra definición.

Según Golden [68], la arquitectura de sistemas es aquella disciplina que permite el manejo de sistemas a partir de las propiedades estructurales de tales objetos. Por ende, la arquitectura de un sistema remite al modelo global de un sistema real, mismo que abarca:

- Una estructura basada en subsistemas, lo que aquí hemos llamado objetos.
- Las propiedades de los objetos involucrados.
- Las relaciones entre los objetos involucrados.
- El comportamiento y la dinámica del sistema.
- Múltiples vistas de los objetos, complementarias y consistentes.

Este primer enfoque destaca cuatro características genéricas que permiten caracterizar los elementos de una arquitectura de sistemas, a saber, los objetos, sus propiedades, sus relaciones y su comportamiento. Tales características reaparecen de diversas maneras en las seis definiciones siguientes y nos permitirán construir nuestra propia definición, razón por la cual serán discutidas en cada ocasión.

Para Luckham [111], una arquitectura es una especificación de los componentes de un sistema y de la comunicación entre ellos. Los sistemas están obligados a ajustarse a una arquitectura. Una arquitectura debe garantizar ciertas propiedades de comportamiento de un sistema, es decir, los componentes están configurados de acuerdo con la arquitectura. Una arquitectura también debería ser útil de varias formas durante el proceso de construcción de un sistema. Una arquitectura es una especificación de una clase de objetos en algún nivel de

abstracción. Una arquitectura consta de interfaces, conexiones y restricciones. Las interfaces especifican los componentes del sistema, mientras que las conexiones y las restricciones definen cómo pueden interactuar los objetos o componentes. La principal relación entre una arquitectura y un objeto es la conformidad. La conformidad define si un objeto se ajusta o no a una arquitectura.

En esta primera definición, las cuatro características genéricas identificadas por Golden (objetos, propiedades, relaciones, comportamiento) pueden ser descritas como sigue: los objetos son los componentes de un sistema; las propiedades son las interfaces, las restricciones y la conformidad; las relaciones remiten a la interacción de objetos; mientras que el comportamiento corresponde a las propiedades de comportamiento.

1.	Objetos	Propiedades	Relaciones	Comportamiento
Luckham [111]	Componentes de un sistema	Interfaces Restricciones Conformidad	Interacción de objetos	Propiedades de comportamiento

**Tabla 2.1. Características genéricas identificadas por Golden**

Por su parte, Mc Govern *et al.* [118] consideran que la arquitectura de sistemas es a la vez un proceso y una disciplina para producir sistemas de información eficientes. Los autores definen un sistema como un conjunto de máquinas, aplicaciones y recursos de red. La arquitectura de sistemas unifica el conjunto imponiendo estructura en el sistema y alinea la funcionalidad del sistema con los objetivos.

En esta definición, los objetos son los sistemas, entendidos como el conjunto de máquinas, aplicaciones y recursos de red. No se definen sus propiedades. En cuanto a las relaciones, se dice que sirven para unificar el conjunto, imponiendo estructura en el sistema. El comportamiento consiste en alinear la funcionalidad del sistema con los objetivos.

2.	Objetos	Propiedades	Relaciones	Comportamiento
Mc Govern et al. [118]	Máquinas, Aplicaciones Recursos de red	NE	Unifican Estructuran	Alinea la funcionalidad del sistema con los objetivos

**Tabla 2.2. Características genéricas identificadas por Mc Govern et al.**

Gils [65] postula que una arquitectura de sistemas es la organización de un sistema integrada por sus componentes, las relaciones de los componentes entre sí y con el medio



ambiente, y los principios que guían su diseño y su evolución. La propuesta puede ser sintetizada como sigue:

3.	Objetos	Propiedades	Relaciones	Comportamiento
Gils [65]	Componentes		Relaciones de los componentes entre sí Relaciones con el medio ambiente	Regido por los principios de diseño y evolución

**Tabla 2.3. Características genéricas identificadas por Gils**

A partir de la definición propuesta por el Departamento de Defensa de Estados Unidos [196], una arquitectura de sistemas designa la estructura de los componentes en un programa o un sistema, en función de sus interrelaciones; en conjunto brindan principios y directrices que rigen su diseño y su evolución en el tiempo. Las cuatro categorías consideradas aquí para el análisis de las definiciones aparecen como sigue:

4.	Objetos	Propiedades	Relaciones	Comportamiento
Departamento de Defensa de Estados Unidos [196]	Componentes en un programa o un sistema	NE	Interrelaciones	Regido por principios y directrices de diseño y evolución

**Tabla 2.4. Características genéricas identificadas por el Departamento de Defensa de Estados Unidos**

Hatley [76] propone un modelo de arquitectura de sistema para clasificar los sistemas o subsistemas, sus componentes y sus actividades en cinco categorías: funciones principales del subsistema, interacción con el usuario, información de entrada proveniente de otros subsistemas, información de salida para alimentar otros subsistemas, funciones de soporte para mantener la operación de la arquitectura. Hatley aporta elementos relevantes en cuanto al comportamiento:

5.	Objetos	Propiedades	Relaciones	Comportamiento
Hatley [76]	Sistemas o subsistemas	NE	Interacción con el usuario Información de entrada y de salida	Funciones principales Funciones de entrada Funciones de salida Funciones de soporte

**Tabla 2.5. Características genéricas identificadas por Hatley**

Finalmente, Putman [141] aborda con más detalle la arquitectura de sistemas, a partir de los elementos estructurales (entidades de software) y de aquellos elementos que soportan el interfuncionamiento de los objetos que componen un sistema. Según este autor, el comportamiento de las relaciones de interfuncionamiento entre las entidades de software también

forma parte de la especificación de la arquitectura. Por ende, debe tomarse en cuenta el comportamiento de todas las entidades involucradas: tanto los componentes y los conectores como su interfuncionamiento.

Así, para concebir una arquitectura se debe justificar el sistema que se está diseñando; definir el alcance de dicho sistema; usar terminología precisa, centrada en sistemas de procesamiento distribuido; indicar las reglas de especificación y las reglas de estructuración; identificar el modelado; mencionar las razones que sustentan las decisiones de diseño; reutilizar las lecciones aprendidas; dar prioridad a las soluciones de costo asequible, menor requerimiento de tiempo y mejor desempeño técnico.

La propuesta de Putman puede resumirse como sigue:

6.	Objetos	Propiedades	Relaciones	Comportamiento
Putman [141]	Elementos estructurales (entidades de software) Elementos que soportan el interfuncionamiento	Especificación de la arquitectura Justificación, Alcances Terminología específica Reglas de especificación Reglas de estructuración y modelado	Interfuncionamiento entre componentes y conectores.	Comportamiento de las relaciones de interfuncionamiento Patrones de razonamiento Reutilización de lecciones aprendidas Costo asequible Requerimiento de tiempo Desempeño

**Tabla 2.6. Características genéricas identificadas por Putman**

El siguiente cuadro retoma los elementos principales de las seis definiciones anteriores, a partir de los cuales construiremos nuestra definición operatoria:

Autor(es)	Objetos	Propiedades	Relaciones	Comportamiento
Luckham [111]	Componentes de un sistema	Interfaces Restricciones Conformidad	Interacción de objetos	Propiedades de comportamiento
Mc Govern <i>et al.</i> [118]	Máquinas, Aplicaciones Recursos de red	NE	Unifican Estructuran	Alinea funcionalidad del sistema y objetivos
Gils [65]	Componentes		Relaciones de los componentes entre sí y con el medio ambiente	Regido por los principios de diseño y evolución
Departamento de Defensa de Estados Unidos [196]	Componentes en un programa o un sistema	Principios de diseño y evolución.	Interrelaciones	Regido por principios y directrices de diseño y evolución
Hatley [76]	Sistemas o subsistemas	NE	Interacción con el usuario Información de entrada y de salida	Funciones principales Funciones de entrada Funciones de salida Funciones de soporte
Putman [141]	Elementos estructurales (entidades de software) Elementos que soportan el interfuncionamiento	Especificación de la arquitectura Justificación Alcances Terminología específica Reglas de especificación Reglas de estructuración y modelado	Interfuncionamiento entre componentes y conectores.	Comportamiento de las relaciones de interfuncionamiento Patrones de razonamiento Reutilización de lecciones Costo asequible Requerimiento de tiempo Desempeño

**Tabla 2.7. Características genéricas identificadas por las definiciones de arquitectura de sistemas presentadas**

A partir de las consideraciones anteriores, entenderemos aquí por arquitectura de sistemas aquel conjunto de objetos o subsistemas diferenciados con propiedades y características propias, relacionados entre sí y con un comportamiento individual específico.

Una arquitectura de sistemas busca describir el sistema en objetos o subsistemas gracias a un modelo conceptual que, además de hacer visible la estructura del conjunto, considera las características de los elementos que lo integran. Los objetos de la arquitectura, también denominados sistemas, subsistemas o componentes, funcionan en conjunto para implementar un sistema más amplio o metasistema. Son elementos estructurales que pueden referirse tanto a máquinas y a aplicaciones como a recursos de red.

Las propiedades o características de cada objeto permiten diferenciar y describir los medios que dicho objeto requiere para poder realizar su comportamiento. Gracias a tales propiedades, que deben ser definidas según una terminología precisa, es posible justificar el

sistema y definir su alcance. Las propiedades incluyen las reglas de especificación y las reglas de estructuración y los principios de modelado.

En cuanto a las relaciones de interacción que se establecen tanto entre los objetos de la arquitectura como entre la arquitectura y su entorno, pueden consistir en la transmisión de información de entrada y de salida o en la transmisión de energía. Sirven para unificar y estructurar el conjunto, así como para configurar el interfuncionamiento entre componentes y conectores. Identificar las relaciones entre elementos dentro de una arquitectura permite identificar la estructura del metasistema.

El comportamiento de cada objeto es único y, como tal, permite estructurar, organizar y dividir el funcionamiento global de la arquitectura en comportamientos más sencillos. Está regido por principios razonados de diseño y evolución que definen diferentes tipos de funciones (principales, de entrada, de salida, de soporte). Un comportamiento adecuado –basado en la reutilización de lecciones aprendidas y capaz de satisfacer requerimientos de costo, tiempo y desempeño– ayuda a alinear la funcionalidad del sistema con los objetivos del mismo.

Una arquitectura de sistemas busca describir el sistema en subsistemas definido por un modelo conceptual que representa su estructura y, además, considera las características de los elementos que lo integran. La estructura se refiere a la manera en que el sistema está interconectado y describe la relación entre todos sus elementos. En conjunto, cada subsistema de la arquitectura de sistemas funciona en conjunto para implementar un sistema en general.

### **2.1.3 Arquitectura de un robot ludoeducativo**

En el capítulo 1 se propuso definir un robot ludoeducativo como una máquina programable destinada al desarrollo de aplicaciones lúdicas con fines de aprendizaje. Se dijo asimismo que un robot ludoeducativo suele estar constituido por una computadora, un procesador, un microprocesador y/o un microcontrolador, sensores, actuadores y/o cámaras, así como un sistema de control.

Ahora bien, se postula aquí que un robot ludoeducativo debe emplear una interfaz humano-máquina basada en una máquina de estados. Dicha máquina permite una comunicación

específica con el usuario mediante el uso de sonidos expresivos, símbolos, figuras, entre otros. El robot ludoeducativo puede asociarse a una arquitectura de sistemas, que contiene objetos o sistemas dados por materiales. Dichos materiales tienen características y propiedades específicas que son relacionados e interconectados con el propósito de obtener aplicaciones ludoeducativas. Estas aplicaciones aprovechan así funciones y comportamientos específicos tanto del robot como del usuario, de la interfaz y del control de bajo nivel. A partir de estos elementos, se presenta a continuación una propuesta de definición específica de arquitectura de un robot ludoeducativo.

### ***A. Análisis de la arquitectura de un robot ludoeducativo***

Partiremos aquí de la definición de arquitectura de sistemas arriba mencionada, según la cual se trata de un conjunto de objetos o subsistemas diferenciados con propiedades y características propias, relacionados entre sí y con un comportamiento individual específico. Tomaremos asimismo en cuenta el marco teórico y conceptual expuesto en el capítulo 1.

Definir la arquitectura de un robot ludoeducativo requiere considerar los siguientes cuatro elementos: los objetos o sistemas de la arquitectura, sus propiedades y características, sus relaciones y sus comportamientos.

Los objetos o sistemas propios de la arquitectura de un robot ludoeducativo se singularizan a partir de los otros tres elementos mencionados (características y propiedades, relaciones y comportamientos) que transforman al conjunto de objetos en una arquitectura y a cada objeto en un subsistema. Por ende, la construcción de la arquitectura de un robot ludoeducativo es un proceso iterativo que requiere de varias etapas de refinamiento, según podrá apreciarse a continuación.

<b>Objeto</b>		
Características y propiedades	Relaciones	Comportamientos

**Tabla 2.8. Elementos que singularizan un objeto en la arquitectura de sistemas**

Por su parte, las características y propiedades de un robot ludoeducativo dependen de tres aspectos: el material, los programas y la programación. El material corresponde a los componentes físicos como son sensores, actuadores, estructura mecánica y computadoras. Los

programas corresponden a las herramientas de software que permiten realizar la programación, como son sistemas operativos, compiladores, núcleos de tiempo real o software de cálculo, entre otros. La programación, que permite darle la funcionalidad necesaria al subsistema de la arquitectura, es la implementación de las funciones y el comportamiento del sistema de la arquitectura, como son la detección de un gesto, una ley de control o una estimación de estado.

Objeto				
Características y propiedades			Relaciones	Comportamientos
Material	Programas	Programación		

**Tabla 2.9. Aspectos por considerar para determinar las características y propiedades**

Se postula aquí que, para adquirir un carácter ludoeducativo, la arquitectura de un robot debe poseer al menos un subsistema que incluya dentro de sus características y propiedades, en la categoría de programación, cuatro implementaciones fundamentales, a saber, la implementación de una máquina de estados finita, la implementación de acciones multimodales —tanto para el robot como para el usuario—, la implementación de una interfaz gráfica, la implementación de una estrategia de control.

Objeto				
Características y propiedades			Relaciones	Comportamientos
<i>Material</i>	<i>Programas</i>	<i>Programación</i>		
		Implementación		
		Máquina de estados	Acciones multimodales	Interfaz gráfica
				Estrategia de control

**Tabla 2.10. Implementaciones dentro de la programación**

La implementación de la máquina de estados requiere a su vez la implementación de la detección de entradas, para cambiar el estado del robot; la implementación o definición de estados específicos; la implementación de transiciones de estado, según la cual una entrada activa la transición a un estado siguiente predeterminado; la implementación de acciones multimodales en un estado o en una transición. Cabe señalar que las acciones multimodales asignadas a un estado

se repiten mientras el robot se encuentre en un estado; las acciones multimodales asignadas a una transición se ejecutan una sola vez.

Objeto					
Características y propiedades				Relaciones	Comportamientos
<i>Material</i>	<i>Programas</i>	<i>Programación</i>			
Implementación					
Máquina de estados		Acciones multi-modales	Interfaz gráfica	Estrategia de control	
Detección de entradas	Definición de estados	Transiciones de estado	Acciones multimodales (estado o transición)		

**Tabla 2.11. Implementaciones dentro de la máquina de estados**

Así pues, la máquina de estados ocupa un sitio fundamental en el desarrollo de un robot ludoeducativo, pues tiene por objetivo brindar un comportamiento específico al robot según la aplicación que se desarrolla. Optar por implementar una máquina de estados ofrece dos ventajas interesantes, relativas a la extensibilidad y a la incorporación de algoritmos. La extensibilidad permite añadir de manera relativamente sencilla nuevos comportamientos, mediante la incorporación de los estados, las transiciones y las acciones multimodales relativos a esos comportamientos. La incorporación de algoritmos específicos a las acciones de control facilita, entre otros, la navegación en ambientes dinámicos.

La implementación de acciones multimodales (véase Tabla 2.12) remite, por su parte, al conjunto de acciones llevadas a cabo por el robot y caracterizadas por la combinación de dos o más modos (verbales y/o no verbales). Según se explicó en el capítulo 1, las acciones verbales pueden ser llevadas a cabo tanto por el robot como por el usuario, mediante la comunicación oral y escrita. Las acciones de comunicación oral incluyen toda clase de expresiones orales que comuniquen algo. Las acciones de comunicación escrita pueden manifestarse mediante textos,

oraciones, palabras y/o letras. Las acciones de comunicación no verbal son acciones cinésicas, proxémicas, visuales, sonoras y expresivas.

<b>Objeto</b>										
<b>Características y propiedades</b>										<b>Relaciones</b>
<b>Programación</b>										<b>Comportamientos</b>
<i>Material</i>	<i>Programas</i>									
<b>Implementación</b>										
Máquina de estados				Acciones multimodales					Interfaz gráfica	Estrategia de control
Detección de entradas	Definición de estados	Transiciones de estado	Acciones multimodales (estado o transición)	Comunicación verbal		Comunicación no verbal				
				Oral	Escrita	Cinésicas	Proxémicas	Visuales	Sonoras	Expresivas

**Tabla 2.12. Tipos de acciones multimodales**

La tercera implementación requerida dentro de la programación de al menos un subsistema (véase Tabla 2.12) corresponde a la interfaz gráfica: dicha interfaz permite al robot interactuar con el usuario a través de imágenes. Su principal utilidad consiste en proporcionar un entorno visual amigable y sencillo, que refuerza la comunicación entre el robot y el usuario.

Finalmente, la implementación de una estrategia de control (véase Tabla 2.12) tiene por objetivo controlar el movimiento del robot y de los dispositivos a él conectados. Las acciones de



control elegidas obedecen a un algoritmo que permite mantener las variables de interés dentro de los rangos predeterminados.

Se mencionó más arriba que una arquitectura de sistemas es la representación de varios subsistemas y objetos, cada uno de los cuales tiene características y propiedades dadas por el material, los programas y la programación. Los subsistemas establecen relaciones y tienen un comportamiento dado por funciones, determinadas tanto por los alcances del material, los programas y la programación como por la relación de la integración del programa y de la programación con la integración del material, para llevar a cabo una interacción. Partiendo de tales elementos, se postula aquí que la arquitectura de un robot ludoeducativo es una arquitectura de sistemas, integrada por uno o más objetos o subsistemas que cumplen con una función específica y que pueden o no funcionar de manera simultánea. Al menos uno de tales objetos debe corresponder a las características del modelo que se ha venido desarrollando en este capítulo y que se retoma en la tabla 2.6, más abajo, contemplando la presencia de uno o más objetos.

Las propiedades y características de los objetos en cuestión están dadas por el material, los programas y la programación del robot ludoeducativo. Las relaciones describen las conexiones entre los subsistemas que conforman el robot ludoeducativo. Los comportamientos remiten a la manera de funcionar del robot ludoeducativo.

		<b>Objeto 1</b>										<b>Objeto 2</b>	<b>Objeto n</b>
		<b>Características y propiedades</b>										<b>Relaciones</b>	<b>Comportamientos</b>
<i>Material</i>	<i>Programas</i>	<i>Programación</i>											
		<b>Implementación</b>											
		<b>Máquina de estados</b>				<b>Acciones multimodales</b>				<b>Interfaz gráfica</b>	<b>Estrategia de control</b>		
		<b>Detección de entradas</b>		<b>Definición de estados</b>		<b>Transiciones de estado</b>		<b>Acciones multimodales (estado o transición)</b>		<b>Comunicación verbal</b>		<b>Comunicación no verbal</b>	
										<b>Oral</b>		<b>Escrita</b>	
										<b>Cinésicas</b>		<b>Proxémicas</b>	
										<b>Visuales</b>		<b>Sonoras</b>	
										<b>Expresivas</b>			

**Tabla 2.13. Arquitectura de sistemas de un robot ludoeducativo basada en uno o más objetos que funcionan a partir de una máquina de estados**

Una vez presentada la arquitectura de sistemas de un robot ludoeducativo, es necesario considerar su articulación con el marco teórico y conceptual del capítulo 1, integrando los componentes lúdicos y didácticos que se presentaron de manera pormenorizada en ese capítulo.

### ***B. Consideraciones didácticas y lúdicas para conferir a un robot un carácter ludoeducativo***

Se parte aquí del postulado según el cual los robots diseñados con fines de aprendizaje deben propiciar una interacción multimodal a través de su interfaz humano-máquina, con el fin de establecer una comunicación específica con el usuario, con el fin de promover un aprendizaje más efectivo, al tomar en consideración variables tales como el entorno de aprendizaje, la participación en grupo, la interacción frecuente, la retroalimentación y las conexiones con el contexto del mundo real [154].

En suma, el carácter educativo de un robot depende de la manera en que se integran a su desarrollo las teorías del aprendizaje, incluyendo los enfoques más recientes, con énfasis en la motivación y la actitud, la interacción, las competencias por desarrollar y la literacidad mediática multimodal. Se trata en efecto de favorecer la exploración, la resolución de problemas y la indagación, ya que el aprendizaje es entendido como un proceso cognitivo, constructivo e interactivo, durante el cual el usuario-aprendiente debe participar activamente en la construcción del conocimiento. La asociación de la tecnología a situaciones formales o semiformales de aprendizaje debe evitar caer en visiones conductistas hoy seriamente cuestionadas y apoyarse en los aportes más recientes del cognoscitivismo, el constructivismo y el interaccionismo, incluyendo además consideraciones en torno a la literacidad mediática multimodal y a las competencias del siglo XXI. Resultan especialmente relevantes los elementos destinados, dentro de la arquitectura de sistemas, a promover la motivación, la interacción y el desarrollo de competencias específicas.

En efecto, mediante el uso del robot ludoeducativo, se busca fomentar diferentes tipos de motivación, con especial énfasis en la motivación intrínseca, tanto a través de las representaciones favorables relativas a la robótica como a través del uso de un robot de apariencia atractiva. En efecto, los robots siguen siendo relativamente inusuales en el aula, y más aún los robots que permiten una interacción multimodal humano-máquina: su uso tiene altas probabilidades de aparecer como una propuesta didáctica innovadora, potencialmente llamativa para los usuarios. Cada uno de los subsistemas (mecánico, electrónico, robótico y de control) puede incidir en la motivación de manera específica.

La motivación está asimismo ligada a las aplicaciones, que deben resultar propicias a la interacción, tanto la interacción humano-máquina como la interacción entre usuarios. La interacción es, de hecho, uno de los conceptos clave de nuestro marco teórico y conceptual, en la medida en que desempeña un papel crucial en las tres disciplinas consideradas y más aún en la intersección de tales disciplinas.

Las modalidades previstas de interacción humano-máquina deben ser amigables con el usuario, de tal manera que éste se concentre en la tarea por realizar y no en las dificultades ligadas al uso de la tecnología. Por ende, la interacción debe satisfacer las expectativas del usuario y seguir la lógica de las actividades de aprendizaje. Es recomendable que la interfaz promueva una interacción multimodal, gracias a la cual el robot lleva a cabo la adquisición y el reconocimiento de la información de su entorno y del cuerpo del aprendiente, promoviendo la comunicación verbal y no verbal. En efecto, la interacción multimodal fomenta el intercambio comunicativo bi- o multidireccional entre el usuario y el robot ludoeducativo, entre diversos usuarios y el robot e incluso usuarios entre sí, sin que ninguna de estas opciones excluya a la(s) otra(s). La interacción puede funcionar mediante sensores, cámaras y/o cualquier otro dispositivo programado para llevar a cabo la adquisición de datos desde los sonidos y las imágenes predeterminados en la interfaz hasta los movimientos específicos dados por el usuario y el robot. Es necesario pues, mediante los programas y la programación adecuados, prever la adquisición y el reconocimiento de ademanes, de gestos y de movimientos. La interfaz puede estar integrada por dispositivos y menús, tangibles o no. Una máquina de estados ayuda a estructurar la interfaz, capaz tanto de identificar acciones a partir de una entrada definida como una transición como de realizar acciones durante dicha transición con miras a obtener un estado final.

Gracias a las características mencionadas, se espera que el robot fomente el desarrollo de competencias específicas. Dichas competencias pueden ser generales, discursivas o lingüísticas, con énfasis en las competencias del siglo XXI y en una amplia gama de literacidades. En efecto, los robots con fines de aprendizaje tienen como propósito apoyar el desarrollo de distintas habilidades (lógico-matemáticas, visuales y espaciales, emocionales; generales, comunicativas, sociales y de investigación), incrementar la capacidad de atención, memoria y concentración;

fomentar la diversificación de los modos de organización (trabajo individual y en equipo), la exploración y/o el autoaprendizaje, entre otros.

Mediante la integración dada por la arquitectura, se busca asimismo fomentar el desarrollo de competencias generales y comunicativas. En lo que atañe a las competencias generales, se articula así el conocimiento declarativo (saber) con las destrezas y las habilidades (saber hacer), la competencia existencial (saber ser) y la capacidad de aprender (saber aprender), dados mediante el uso y manejo del robot. En cuanto a las competencias comunicativas, son usualmente requeridas para la realización de las intenciones comunicativas requeridas por la tarea, e incluyen tanto las competencias lingüísticas y las discursivas como las pragmáticas e incluso las sociolingüísticas.

En cuanto a la noción de literacidad, las decisiones en torno a la arquitectura de un robot educativo integran elementos susceptibles de contribuir al desarrollo por parte del usuario del pensamiento crítico, la intuición, la creatividad, la comunicación y la colaboración. Para ello, el usuario debe echar mano de la curiosidad, la iniciativa y la adaptabilidad y poner a la obra diversas literacidades, entre ellas la científica, mediante el uso autónomo o semidirigido de las tecnologías de la información. La literacidad mediática multimodal puede ser promovida mediante el uso de diversos dispositivos que propician la comunicación verbal y no verbal, articulando diversos modos.

Hemos expuesto hasta ahora aquellos elementos que pueden favorecer el carácter educativo de un robot. En el caso de un robot ludoeducativo, los elementos didácticos deben articularse con elementos lúdicos.

Desde las ciencias del juego, cabe destacar que el material lúdico requerido por el robot ludoeducativo es usado como un elemento activo. Hecho o no explícitamente para la actividad lúdica, adquiere su interés lúdico y educativo en función de las reglas según las cuales se usa, del contexto en el que se usa y, sobre todo, de la actitud que logra generar en el usuario. En efecto, la actitud lúdica, entendida como la disposición de ánimo del jugador respecto de sus acciones, es la

que da sentido de juego lúdico: ya no se trata de una denominación impuesta desde fuera, a partir de un material, una estructura o un contexto, sino de una convicción íntima del actor del juego.

Por ende, el enfoque constructivista, cognoscitivista e interaccionista debe verse apuntalado por un contexto propicio a un aprendizaje activo, donde el material lúdico es utilizado para dinámicas específicas, apuntaladas por la presencia de una estructura lúdica capaz de cautivar y mantener la atención durante el proceso de aprendizaje. Debidamente planeado, el juego puede introducir una dinámica de emulación y competencia, relacionada con la motivación extrínseca. Se genera entonces una disposición de ánimo peculiar. El aprendiente no se limita a recibir y emitir un *input*, sino que analiza, interpreta y evalúa la información recibida a través de acciones multimodales. Recurrir a una máquina de estados permite una gran flexibilidad que deja atrás los enfoques conductistas para centrarse en interacciones singulares, brindando al usuario la libertad necesaria para explorar el ambiente tecnológico y construir el conocimiento a partir de su propia experiencia. Además, es necesario prever una interacción que no sólo ocurra entre el usuario y la máquina, sino también entre usuarios, con el apoyo de la interfaz gráfica en un contexto multimodal propicio a la comunicación verbal y no verbal entre usuarios.

Así, un robot ludoeducativo puede ser considerado como parte central más no exclusiva del material lúdico, ya que puede ir acompañado por otros elementos de juego, entre ellos las palabras y el cuerpo. A través de la arquitectura de sistemas, se busca que el robot sea manejado mediante el uso de entradas, salidas y transiciones (una máquina de estados). Asimismo, se intenta que las estrategias de control brinden al robot la posibilidad de moverse en un entorno dinámico.

Habiendo explorado de qué manera la asociación entre la arquitectura de sistemas y las consideraciones didácticas y lúdicas contribuye a fortalecer el carácter ludoeducativo de un robot, es momento de presentar y discutir las arquitecturas de los dos prototipos desarrollados en el marco de la investigación doctoral: Ludibot v1 y Ludibot v2.

## **2.2 Arquitectura de Ludibot v1 y Ludibot v2**

En el marco de la investigación doctoral, fue posible desarrollar, entre 2016 y 2020, dos prototipos del robot ludoeducativo Ludibot. Se presenta a continuación la arquitectura de cada

uno de ellos, discutiendo sus ventajas y sus desventajas respectivas, e identificando los ajustes deseables.



**Figura 2.1. Ludibot v1**

### **2.2.1 Arquitectura de Ludibot v1**

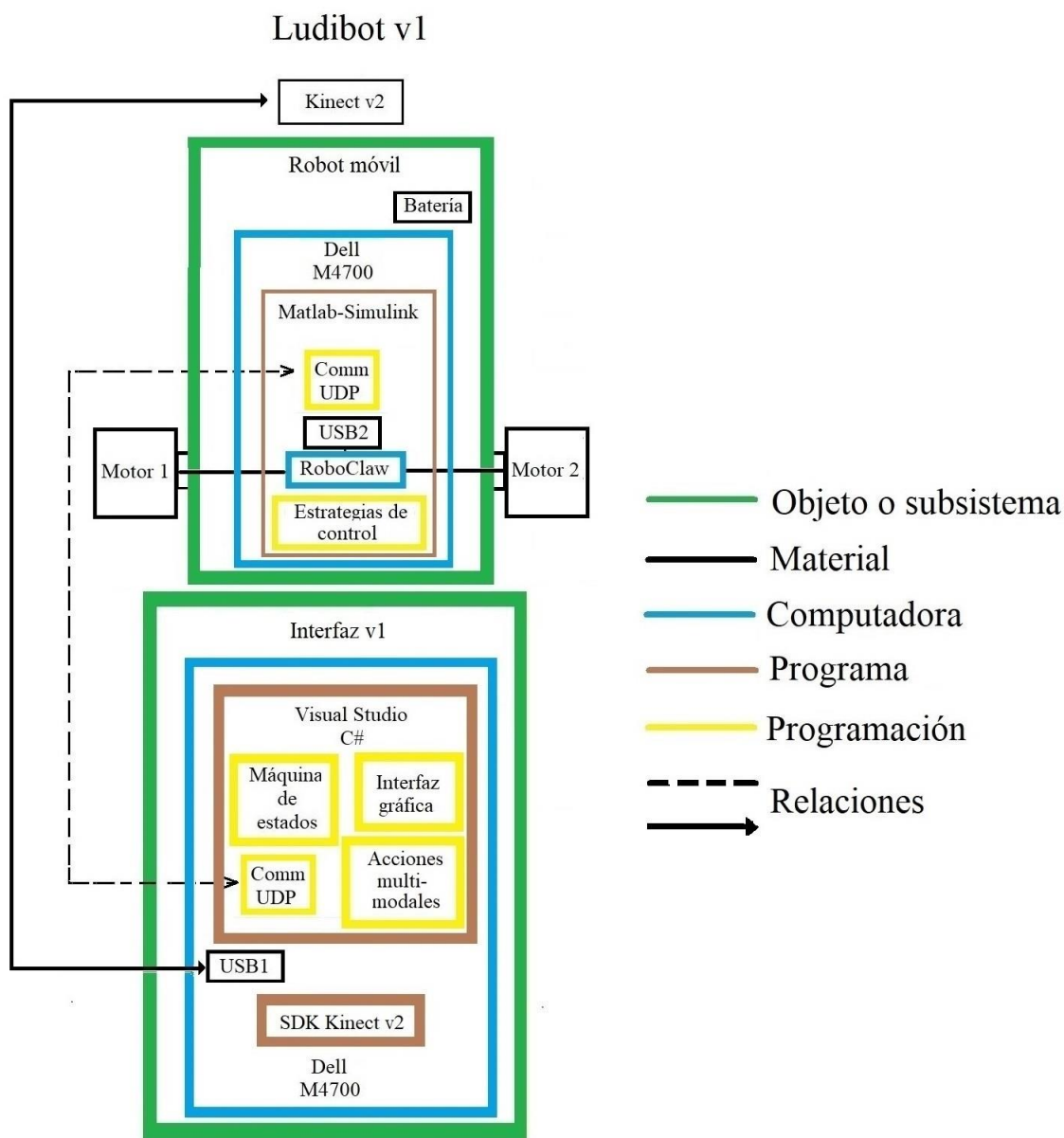
La arquitectura para la primera versión de Ludibot, Ludibot v1 (véase Figura 2.1), se presenta en la tabla Tabla 2.14, siguiendo el esquema definido más arriba en la sección relativa a la arquitectura de sistemas. Esta arquitectura consta de dos subsistemas u objetos: el robot móvil y la interfaz v1. El primero abarca los elementos necesarios que permiten al robot móvil controlar los movimientos, mientras que el segundo implementa la máquina de estados y la interfaz gráfica.

En esta arquitectura, se utiliza una sola computadora para los dos subsistemas. La Figura 2.2 presenta un diagrama a bloques de su arquitectura.

Objetos	Características y propiedades			Relaciones	Comportamientos
	Material	Programas	Programación		
<b>Robot móvil</b>	Computadora M4700 Amplificador de potencia Motores Decodificadores ópticos incrementales Batería Estructura mecánica Ruedas	Matlab-Simulink QuaRC Sistema operativo	<b>Implementación</b>	Recibe de la interfaz v1 la posición deseada vía la comunicación UDP	Utiliza la posición deseada para mover al robot móvil mediante las estrategias de control
			Estrategia de control Comunicación UDP		
<b>Interfaz v1</b>	Computadora M4700 Pantalla Kinect v2	Visual Studio (C#) SDK de Kinect V2 Sistema operativo	Interfaz gráfica Máquina de estados (detección de entradas, definición de estados, transiciones de estado) Acciones multimodales (oral escrita, sonoras, expresivas, cinésicas, proxémicas y visuales) Comunicación UDP	Envía al robot móvil la posición deseada vía la comunicación UDP	Detecta los gestos como entradas a la máquina de estados Realiza transiciones de que cambian el estado Repite las acciones multimodales asociadas a un estado Ejecuta acciones multimodales definidas en las transiciones Lleva a cabo las aplicaciones lúdicas usando la máquina de estado

**Tabla 2.14. Arquitectura de sistemas de Ludibot v1**





**Figura 2.2. Diagrama de la arquitectura de Ludibot v1**

### *Características y propiedades del subsistema Robot móvil*

#### *Material del subsistema Robot móvil*

El material del subsistema Robot móvil consta de una computadora, dos motores de corriente directa, dos decodificadores ópticos incrementales para medir las posiciones angulares de las

llantas, dos llantas, una batería que energiza al sistema, una estructura mecánica que soporta al robot móvil y un amplificador de potencia para mover los motores.

La computadora empleada es una Dell M4700 con un procesador i7 de 2.9Ghz y 16 Gb de memoria. Los motores utilizados [153] son de corriente directa con engrane planetario, con 313 revoluciones por minuto en el eje de salida del motor, 20 A de corriente máxima, 30kgf-cm de par y voltaje de operación de 12V. Con el fin de obtener una mejor estimación de la odometría, los decodificadores ópticos originales se sustituyeron por decodificadores con mayor resolución. Los decodificadores empleados para medir la posición angular del eje del motor son el modelo E2 de Us Digital [196] de 5000 pulsos por revolución. Las llantas son de tipo neumático 200x50 con cámara. La batería es de litio polímero de 4500mAh a 11v de 3 elementos. La estructura mecánica es de aluminio.

El amplificador de potencia utiliza el controlador de motores de corriente directa RoboClaw [140], de dos canales, con capacidad de 30 A por canal. Se configuró el controlador RoboClaw en modo velocidad. Así, el controlador interno de velocidad proporcional, integral y derivativo (PID) se sintonizó para regular las velocidades de las llantas y alcanzar el valor de referencia deseado. Para poder establecer los valores de referencia de velocidad y obtener la velocidad de las ruedas, se configuró el modo de comunicación vía el bus universal en serie (USB, por sus siglas en inglés: *Universal Serial Bus*).

### *Programas*

Los programas que el subsistema Robot móvil utiliza son el sistema operativo Windows 10 [126], el entorno Matlab-Simulink [189] y el núcleo de tiempo real QuaRC [143]. Matlab-Simulink y QuaRC se ejecutan dentro Windows 10. El código generado por Matlab-Simulink se ejecuta dentro del núcleo de tiempo real QuaRC. Para generar el código en el ambiente Matlab-Simulink, se utilizaron las cajas de herramientas de Matlab Coder, Simulink Coder y Quanser Real-Time control.

### *Programación del subsistema Robot móvil*

La programación del subsistema Robot móvil implementa la estrategia de control y la comunicación de datos entre dicho objeto y el objeto Interfaz v1.

La estrategia de control emplea ley de control no lineal, que proporciona la velocidad de referencia deseada a la llanta derecha ( $u_d$ ) y a la llanta izquierda ( $u_i$ ). La ley de control requiere de las velocidades angulares de la llanta derecha ( $\omega_d$ ) y la velocidad de la llanta izquierda ( $\omega_i$ ) para estimar la posición y orientación del robot  $[x, y, \theta]$  en el plano.

Los valores de la velocidad angular de las llantas  $\omega_d$  y  $\omega_i$  se obtienen del controlador RoboClaw. Los valores de la velocidad de referencia  $u_d$  y  $u_i$  que se proporcionan al controlador RoboClaw se obtienen de la ley de control no lineal.

En el apéndice B se presenta el modelado y control de Ludibot v1 de manera detallada. La programación de la ley de control se implementó en el ambiente Matlab-Simulink utilizando diagramas de bloque de Simulink. Se desarrolló un bloque S-Function, que utiliza un puerto serial virtual USB para la comunicación de datos con el controlador RoboClaw.

Para la comunicación de datos entre el subsistema Robot móvil y el subsistema Interfaz v1 se utilizó el protocolo de transmisión de datos (*Transmission Data Protocol*, TCP). Se emplearon asimismo los bloques Simulink Stream Call, Stream Write y Stream Read, configurados con el protocolo TCP.

#### *Relaciones y comportamiento del subsistema Robot Móvil*

La relación que tiene el subsistema Robot móvil con el subsistema Interfaz v1 consiste en la comunicación de datos. De esa manera, el robot móvil recibe una posición deseada en el plano  $[x_{ref}, y_{ref}, \theta_{ref}]$  de la interfaz v1.

Se implementó un generador de trayectoria que primero orienta al robot al ángulo  $\theta_{ref}$  y luego se mueve al punto  $[x_{ref}, y_{ref}]$ . En cuanto al comportamiento del subsistema Robot móvil, éste utiliza la posición deseada proporcionada por el subsistema Interfaz V1 para mover el robot, utilizando la ley de control no lineal.

## Características y propiedades del subsistema Interfaz v1

### Material del subsistema Interfaz v1

El material del subsistema interfaz v1 consta de una computadora m4700, una pantalla y un sensor Kinect V2 [120]. La pantalla, a colores, integrada a la computadora portátil m4700, tiene una resolución de 1920x1080 pixeles y una tasa de refresco de 70hz. Como se dijo antes, se trata de la misma computadora que la del subsistema Robot móvil.

El sensor Kinect v2 permite la detección de la posición (x,y,z) de 26 puntos del cuerpo humano, como se muestra en la figura 2.3.

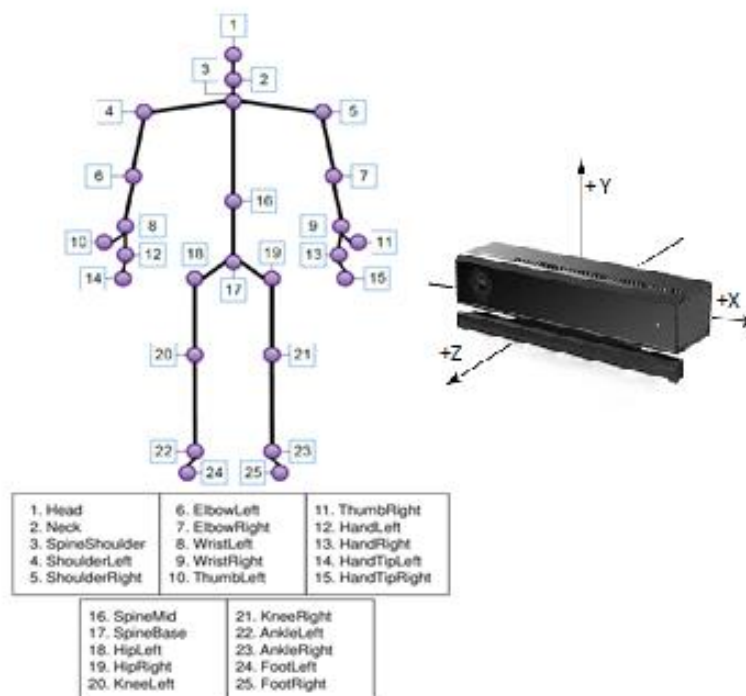


Figura 2.3. Puntos de las articulaciones del cuerpo detectados por Kinect y ejes de referencia



**Figura 2.4. Componentes del sensor Kinect V2**

El sensor tiene una cámara a color, un proyector infrarrojo, una cámara infrarroja y un arreglo de micrófonos (véase figura 2.4).

La cámara a color tiene una resolución de 1920x1080 píxeles con capacidad de 30 cuadros por segundo. La cámara infrarroja tiene una resolución de 512x424 píxeles, con capacidad de 30 cuadros por minuto. La medición de la profundidad se basa en el tiempo de vuelo de señales infrarrojas. El sensor tiene un arreglo de 4 micrófonos. El rango de medición es de 0.4 a 4.5m y puede seguir las posiciones de hasta 6 cuerpos humanos simultáneamente.

#### *Programas del subsistema Interfaz v1*

Los programas utilizados en el subsistema Interfaz v1 son tres: el sistema operativo Windows 10 [126], el entorno de desarrollo de programas Visual Studio [125] para lenguaje C# y, finalmente, el kit de desarrollo de programas SDK [124] (*Software Development Kit* en inglés) Kinect 2.0 para Windows. Visual Studio se ejecuta dentro de Windows 10. El SDK permite a Visual Studio acceder a las clases para programar el sensor Kinect V2.

#### *Programación del subsistema Interfaz v1*

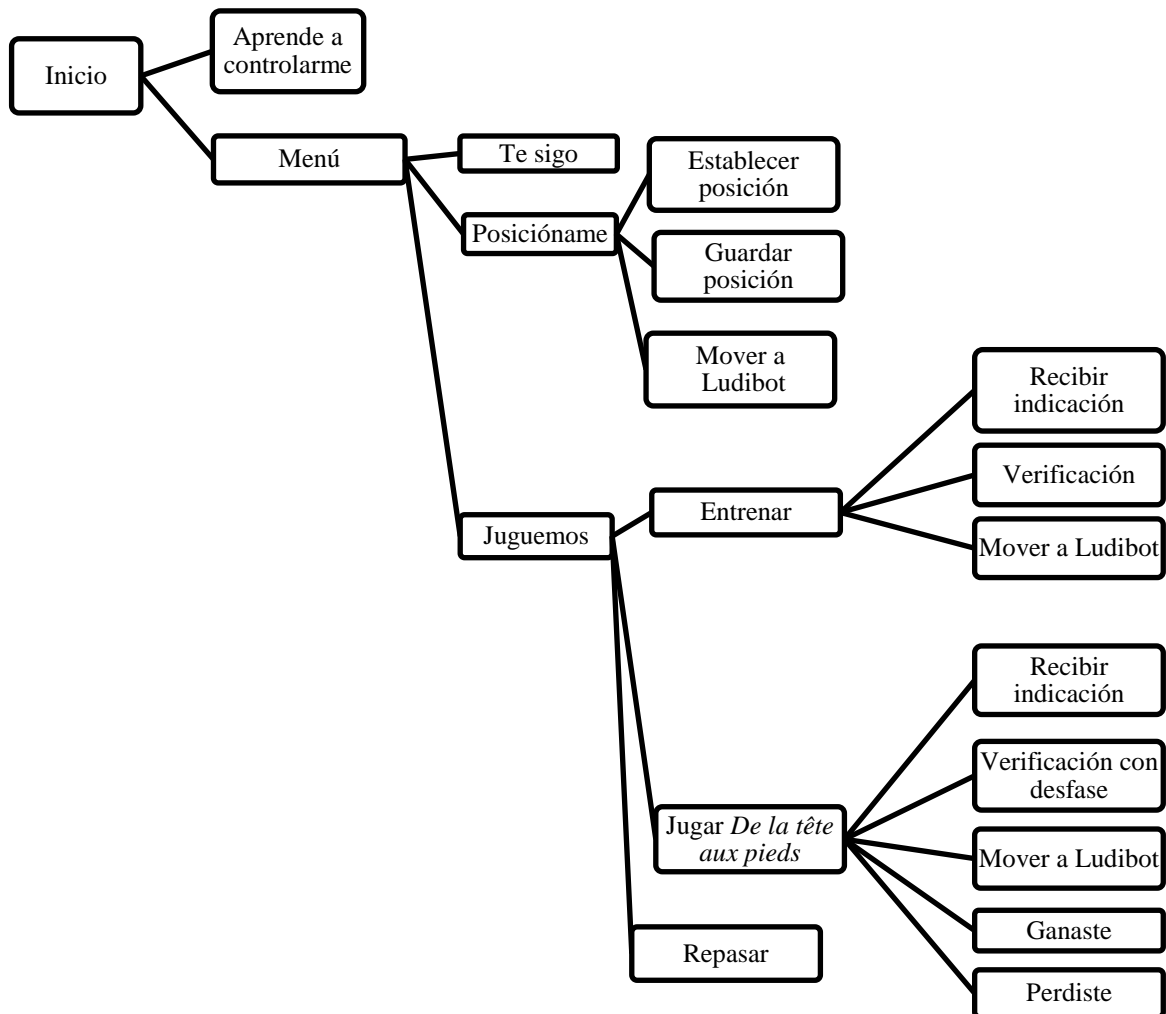
La programación del subsistema interfaz v1 determina la interfaz gráfica, la máquina de estados, las acciones multimodales y la comunicación de datos UDP con el subsistema Robot móvil.

La interfaz gráfica utiliza la pantalla para presentar información del funcionamiento de la máquina de estados. La interfaz está dividida en seis espacios:

- La barra de estado indica el nombre del estado actual.
- La ventana de animación presenta la animación del esqueleto del usuario detectado, utilizando los puntos de las articulaciones del cuerpo definidas por Kinect.
- La ventana del gesto detectado brinda la imagen del gesto que se detectó a partir de la información de Kinect.
- La ventana de gestos disponibles incluye los gestos que se pueden detectar a partir del estado actual.
- El estado de detección de Kinect señala si el sensor Kinect está o no activo.
- La barra de estado de la batería informa el nivel de la batería.

La implementación de la máquina de estados requiere de la detección de entradas así como de la definición de estados y de las transiciones de estado. En este caso, las entradas son gestos formados a partir de la combinación de dos elementos: el primero de ellos es la posición relativa de las manos (arriba o abajo) respecto de la parte superior de la columna vertebral (punto *spine shoulder*); el segundo es la situación de las manos (abierta o cerrada). En la ventana de animación del esqueleto, la situación de las manos se representa mediante un círculo verde cuando la mano está abierta y con un círculo rojo cuando la mano se encuentra cerrada.

Los estados que se definen para Ludibot v1 son 20, según se aprecia en la Figura 2.5. Las transiciones que se han definido son: *SalirT*, *TesigoT*, *PosiciónameT*, *Guardar PosiciónameT*, *EstablecerPosiciónT*, *GuardarPosiciónT*, *Mover a Ludibot*, *AltoT*, *JuguemosT*, *EnternarT*, *RecibirT*, *VerificaciónT*, *AciertoT*, *PerdisteT* y *RepasarT*. Las transiciones se ejecutan según los gestos definidos para realizar la transición en un estado determinado. Estados y transiciones serán presentados en detalle en el capítulo siguiente.



**Figura 2.5. Estados de Ludibot v1**

Ludibot v1 puede realizar acciones multimodales tanto de comunicación verbal como de comunicación no verbal. La comunicación oral está presente en los audios para el juego en torno las partes del cuerpo. La comunicación escrita y visual es utilizada para la barra de estados, la ventana del gesto detectado y la ventana de los gestos disponibles como los gestos detectados que permiten presentar las partes del cuerpo por medio de imágenes.

Las acciones no verbales que puede realizar Ludibot V1 son cinésicas, proxémicas, sonoras y expresivas:

- Acción cinésica: detección de gestos de las manos como entradas.
- Acción proxémica: seguimiento del usuario por el robot guardando una relación espacial definida.
- Acciones sonoras: emisión de señales para indicar tanto errores y aciertos durante el juego como el resultado final de la partida (“Error”, “Acierto”, “Ganaste”, “Perdiste”).
- Acciones expresivas: asociación de imágenes y sonidos a la situación de juego, mediante el uso de emoticones de alegría cuando el usuario acierta o se gana y de tristeza cuando comete un error o pierde.

Para la comunicación de datos entre el subsistema Interfaz v1 y el subsistema Robot móvil se utilizó el protocolo de transmisión de control (TCP). Se implementó un cliente (TCP Client) y un servidor (TCP Server) para lograr la comunicación con el robot móvil.

#### *Relaciones y comportamiento del subsistema Interfaz v1*

La relación que tiene el subsistema interfaz v1 con el subsistema Robot móvil consiste en la comunicación de datos. La interfaz v1 envía una posición deseada en el plano  $[x_{ref}, y_{ref}, \theta_{ref}]$  para el robot móvil.

El comportamiento de la interfaz v1 está determinado por su máquina de estados, que permite la detección de gestos de las manos del usuario, el cambio de estado por medio de una transición cuando se detecta un gesto particular, la ejecución de acciones multimodales definidas en la transición (que se ejecutan una vez) o definidas en el estado (que se repiten mientras el estado no cambie).

### **2.2.2 Arquitectura de Ludibot v2**

Tras el desarrollo y el pilotaje de la primera versión de Ludibot, y con el apoyo de la Dirección General de Asuntos del Personal Académico de la Universidad Nacional Autónoma de México, en el marco del proyecto PAPIIT IT400119 (2019-2021), los participantes en dicho proyecto nos dimos a la tarea de diseñar una nueva versión del robot. Más adelante se discutirán las principales semejanzas y diferencias entre ambas versiones y las razones que fundamentaron los principales



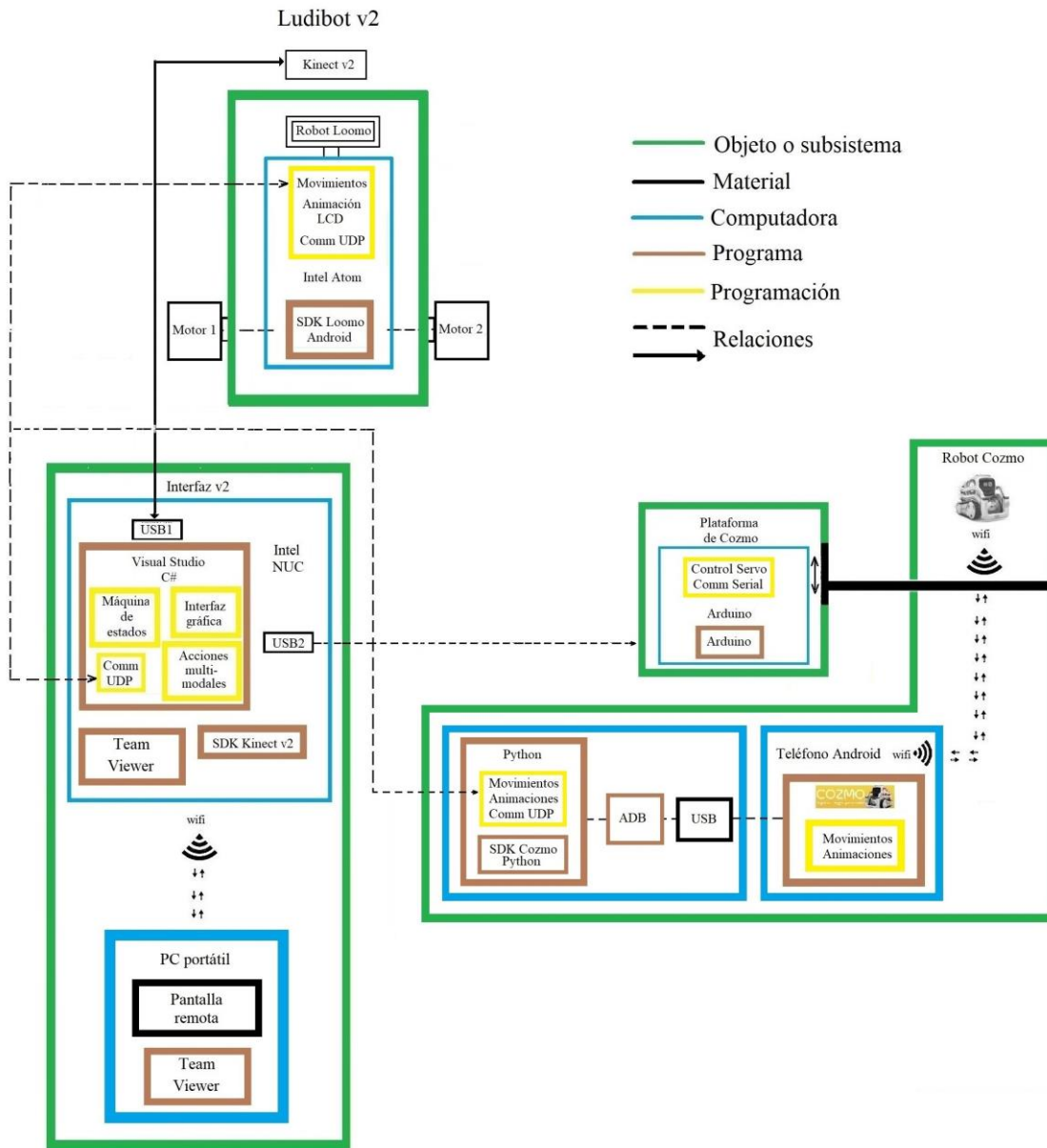
cambios. En esta sección, describiremos la arquitectura de la más reciente versión del robot, Ludibot v2 (véase Figura 2.6).



**Figura 2.6. Ludibot v2**

La arquitectura de Ludibot v2 consta de cuatro subsistemas: el robot Loomo, la interfaz v2, el robot Cozmo y la plataforma de este último. La Figura 2.7 presenta el diagrama de la arquitectura, mientras que la Tabla 2.15 detalla cada objeto dentro del metasisistema (características y propiedades, relaciones y comportamientos) según el modelo de arquitectura de sistemas desarrollado aquí con anterioridad.

El primer subsistema integra los elementos de un robot móvil con una cabeza con pantalla y el material necesario para su movimiento. El segundo implementa la máquina de estado y la interfaz gráfica. El tercero consta de un minirobot, que se monta sobre el cuarto subsistema, constituido por una plataforma de ascenso y descenso.



**Figura 2.7. Diagrama de la arquitectura de Ludibot v2**

Objetos	Características y propiedades			Relaciones	Comportamientos
	Material	Programas	Programación		
<b>Loomo</b>	Robot móvil Loomo (dos ruedas) autobalanceado Pantalla LCD de la cabeza (2gdl)	SDK Loomo Sistema operativo Android	<b>Implementación</b> Movimiento del robot Animación en la pantalla LCD Comunicación UDP	Recibe posición y animaciones de la interfaz v2 vía la comunicación UDP	Utiliza la posición deseada para mover al robot móvil Realiza animaciones con la cabeza y la pantalla LCD
<b>Interfaz v2</b>	Kinect v2 Pantalla remota vía TeamViewer Mini-PC Intel NUC PC portátil	Visual Studio (C#) SDK de Kinect V2 TeamViewer	Lenguaje propio de programación de marcado (tipo HTML); incluye variables y transiciones por variable booleana y por gestos Interfaz gráfica Máquina de estados Acciones multimodales Comunicación UDP	Envía al robot Loomo movimiento deseado y ejecución de animaciones vía comunicación UDP Envía al robot Cozmo posición la deseada de movimiento y la ejecución de animación vía comunicación UDP Sube o baja la plataforma de Cozmo	Detecta los gestos para efectuar transiciones Efectúa transiciones por variable booleana Ejecuta transiciones de estado y acciones multimodales en la transición Ejecuta cambios de estado y acciones multimodales de estado Manda acciones a Cozmo, a Loomo y a la plataforma de Cozmo
<b>Cozmo</b>	Minirobot móvil Cozmo Teléfono Android	Aplicación Cozmo Sistema operativo Android Python Android Debug Device (ADB)	Comunicación UDP con la Interfaz V2 Ejecución de movimientos y animaciones	Recibe movimientos y ejecución de animaciones de la interfaz vía comunicación UDP	Utiliza la posición deseada para mover al robot Realiza animaciones
<b>Plataforma</b>	Arduino Mega Servomotor Estructura mecánica	Arduino IDE	Control del servomotor Comunicación vía puerto virtual serial USB	Recibe de la interfaz v2 la indicación de bajar o subir su plataforma	Controla el movimiento del servomotor para subir o bajar el robot Cozmo

**Tabla 2.15. Arquitectura de sistemas de Ludibot v2**

Cada uno de los cuatro subsistemas que constituyen a Ludibot v2 es presentado a continuación.

### ***Características y propiedades del subsistema Loomo***

#### ***Material***

El material del subsistema Loomo consta de un robot comercial producido por Segway y conocido como Loomo [161][162]. Este robot posee un procesador Intel Atom Z8750, una

memoria RAM 4GB, un almacenamiento 64GB, un puerto USB 3.0 tipo C, un puerto USB 2.0, dos motores de corriente directa, dos decodificadores tipo Hall que permiten medir las posiciones angulares de las llantas, dos llantas neumáticas, una batería que energiza al sistema, una estructura mecánica que soporta al robot móvil, una pantalla LCD, una cabeza, una cámara a color HD de 1030p a 30Hz, una cámara RealSense ZR300, un arreglo de cinco micrófonos, un sensor ultrasónico, un arreglo de sensores tácticos en la cabeza y una batería que energiza al robot de 329 W.

La estructura del robot Loomo mide 650x310x70mm (alto x profundidad x ancho). El peso neto del robot es 19kg (42 lbs). Su velocidad máxima es de 8km/h. El rango de rotación de la cabeza es de  $\pm 150$  grados, mientras que el rango de cabeceo es de -90 a 180 grados. La pantalla LCD es de 4.3 pulgadas, con una resolución de 480 x 800 pixeles. La capacidad de carga útil es de hasta 100 kg.

### *Programas*

Los programas que utiliza el subsistema Loomo son tres: el sistema operativo Android, el SDK Loomo y Android Studio para programar las aplicaciones Android.

### *Programación*

La programación del subsistema Loomo permite mover al robot a una posición deseada, realizar una animación utilizando tanto la cabeza como la pantalla LCD, e implementar la comunicación de datos con el subsistema Interfaz v2.

El kit de desarrollo de software de Loomo (Software Development Kit por sus siglas en inglés, SDK) se divide en Visión, Habla, Locomoción, Emoji e Interacción. Si bien en la aplicación desarrollada que se presenta en el capítulo 3 se utilizan únicamente el servicio de Locomoción y los Emoji, las características de este robot permiten considerar aplicaciones más complejas que aprovechen en mayor medida el potencial de este robot comercial.

Para mover a Loomo a una posición deseada se emplea el modo de navegación, agregando puntos de control o *checkpoints*. Loomo se moverá a estos puntos sucesivamente, con una exactitud de 0.25m. Para la cabeza se utiliza el modo emoticonos. En este modo, existen 23

animaciones o comportamientos, por ejemplo “mirar abajo”, “mirar arriba” o “ver alrededor”. Actualmente se sigue trabajando para agregar las acciones dentro del lenguaje propio, tipo HTML, con el fin de mover a Loomo y ejecutar las animaciones vía la función emoticonos del SDK.

Para la comunicación UDP se utiliza un hilo que, haciendo recibido la información de la interfaz v2, puede efectuar movimientos o animaciones. En el caso de los movimientos, se regresa información cuando el robot se encuentra en la posición deseada.

#### *Relaciones y comportamiento del subsistema Loomo*

La relación que tiene el subsistema Loomo con el subsistema Interfaz v2 se basa en la comunicación de datos. El robot Loomo recibe información de la posición deseada que debe alcanzar respecto de su posición actual. En cuanto a la animación o comportamiento, la información recibida consiste en el nombre de la animación por realizar.

#### *Características y propiedades del subsistema Interfaz v2*

##### *Material*

El material del subsistema Interfaz v2 consta de una minicomputadora Intel NUC, una pantalla de presentación remota vía una computadora portátil y un sensor Kinect v2. La minicomputadora Intel NUC se encuentra embarcada en el robot. Una pila de litio-polímero energiza tanto la minicomputadora como el sensor Kinect v2. La pila litio-polímero es independiente de la pila que utiliza el robot Loomo.

No se ha determinado aún el tipo de pantalla que se conectará a la computadora portátil, pues ello dependerá de las opciones disponibles al inicio del pilotaje con alumnos de bachillerato. Se requiere utilizar ya sea un proyector, ya sea una pantalla grande de televisión, pues las imágenes de la interfaz deben ser visibles a distancia (de uno a seis metros).

##### *Programas*

Los programas que el subsistema Interfaz v2 utiliza son cinco: el sistema operativo Windows 10, el entorno Visual Studio mediante lengua C#, el SDK de Kinect v2, Kinect Studio [121], Visual

Gesture Builder [123] y TeamViewer [188]. Todos estos programas se ejecutan dentro de Windows 10.

Visual Studio y el SDK de Kinect sirven para programar la máquina de estados, la interfaz gráfica, las acciones multimodales, la comunicación UDP y la comunicación serial por puerto virtual USB. Kinect Studio y Visual Gesture Builder se utilizan para grabar información de las posiciones del esqueleto de Kinect V2 y para construir bases de datos con miras a la detección de gestos. El programa TeamViewer se emplea para conectarse vía remota desde la computadora portátil a la computadora Intel NUC, con el propósito de visualizar la interfaz gráfica mediante una pantalla conectada a la computadora portátil.

### *Programación*

La programación del subsistema Interfaz v2 difiere de la programación de la Interfaz v1. Los tres cambios fundamentales son los siguientes:

- Lenguaje de programación propio, tipo HTML. La versión v1 tenía un número fijo de estados y acciones una vez compilado el programa. Agregar estados o acciones consumía mucho tiempo pues requería de un ciclo completo de programación, prueba y depuración. Introducir un lenguaje propio permite desarrollar un analizador léxico-sintáctico, un creador de objetos y un ejecutor de los objetos. Los objetos son estados, transiciones y acciones multimodales. En el apéndice D se presenta el lenguaje con mayor detalle.
- Detección de gestos arbitrarios usando aprendizaje-máquina. Mientras que en la primera versión v1 la detección de gestos está limitada a una combinación de dos elementos (manos arriba o abajo y mano abierta o cerrada), en esta versión se realiza un entrenamiento para la clasificación de gestos a partir de la información relativa al esqueleto, marcando cuando el gesto es válido. Así, es posible clasificar gestos estáticos y dinámicos. El clasificador se entrena usando los algoritmos de aprendizaje AdaBoost [119] para gestos estáticos y Random Forest Regression (RFR) [122] para gestos continuos. Se utilizan las herramientas Visual Studio y Visual Gesture Builder con el fin de obtener las bases de datos necesarias para la clasificación de gestos. En el apéndice E se presenta con mayor detalle el uso de estas herramientas.

- Introducción de variables booleanas, vectores de posición (x,y,z) y variables reales. Gracias a estos nuevos elementos, cada transición puede efectuarse ya sea mediante una variable booleana, ya sea mediante un gesto de entrada. Se definieron operaciones sobre las variables, tales como suma, resta, mayor que, menor que, etc. Así, un usuario puede programar el robot utilizando el modelo de una máquina de estado (estados, transiciones, acciones) y un lenguaje con sintaxis sencilla (<Etiqueta Atributo= “Valor del atributo” /Etiqueta>).

Las acciones multimodales para Ludibot v2 son similares a las que puede efectuar Ludibot V1, pero ahora se agregan animaciones tanto del robot Cozmo (574 animaciones) como del robot Loomo (16 animaciones). La interfaz es similar y se reorganizó tanto en la parte de presentación como en las clases que la manejan, incluyendo programación multihilos para la máquina de estados.

### *Relaciones y comportamiento*

El subsistema Interfaz v2 se relaciona con los otros subsistemas por medio de una comunicación de datos. Para los robots Loomo y Cozmo se envía una posición deseada o una animación y los robots envían de regreso una señal, ya sea cuando terminan las acciones, ya sea cuando detectan un error. Adicionalmente, la interfaz v2 manda una señal a la plataforma de Cozmo, para que ésta baje o suba.

### *Características y propiedades del subsistema Cozmo*

#### *Material*

El material del subsistema Cozmo consta del robot Cozmo [7] (véase Figura 2.8) y de un teléfono Android. El robot Cozmo incluye un procesador NXP Kinetis ARM Cortex M4, un procesador Expressif ESP8266, tres motores de corriente directa, decodificadores magnéticos, pantalla, dos orugas de caucho como medio de locomoción, batería de litio y una estructura mecánica. El teléfono celular se utiliza para ejecutar el SDK en la aplicación Cozmo.



**Figura 2.8. Robot Cozmo**

### *Programas*

Los programas que utiliza este subsistema son cuatro: la aplicación Cozmo, el sistema operativo Android, el lenguaje de programación Python y el programa Android Debug Device (ADB). La aplicación Cozmo se ejecuta bajo el sistema operativo Android en el teléfono celular. El SDK de Cozmo está escrito en el lenguaje Python. El programa ADB se usa para establecer la comunicación entre la aplicación Cozmo y el lenguaje Python.

### *Programación*

Debido a que el SDK de Cozmo funciona con Python, se integró a la interfaz v2 usando una comunicación UDP. Se implementó un servidor UDP que recibe la información tanto sobre la posición deseada que debe alcanzar Cozmo como sobre la animación por realizar según la interfaz v2. También se incluyó la ejecución de un script en Python desde el lenguaje de programación de la máquina de estado.

### *Relaciones y comportamiento*

El subsistema robot Cozmo recibe información de la interfaz v2 para realizar movimientos o animaciones.



## ***Características y propiedades del subsistema Plataforma***

### *Material*

El material de la plataforma del robot Cozmo está integrado por una estructura tipo elevador que permite a Cozmo bajar y subir. Se utiliza para ello un servomotor. así como una placa Arduino para controlar el servomotor.

### *Programa*

Este subsistema utiliza el entorno de desarrollo Arduino para programar la placa.

### *Programación*

La programación implementa una comunicación serial, utilizando un puerto virtual serial USB de la computadora NUC.

### *Relaciones y comportamiento*

Este subsistema reciba una señal para bajar o subir la plataforma, coordinando los robots Loomo y Cozmo.

## **2.2.4 De Ludibot v1 a Ludibot v2: principales semejanzas y diferencias**

Se presenta a continuación una comparación entre ambas versiones de Ludibot, empezando por sus semejanzas y discutiendo enseguida sus principales diferencias, para terminar con algunas consideraciones acerca de las perspectivas de desarrollo de este robot ludoeducativo.

### ***Semejanzas entre Ludibot v1 y Ludibot v2***

Las principales semejanzas entre Ludibot v1 y Ludibot V2 tienen que ver con su naturaleza, su propósito ludoeducativo y, por ende, con los fundamentos para su desarrollo.

### *Naturaleza*

Ludibot v1 y Ludibot v2 son plataformas programables destinada al desarrollo de aplicaciones lúdicas con fines de aprendizaje. Ambas están constituidas por una computadora, un procesador, un microprocesador y/o un microcontrolador, además de sensores, actuadores y/o cámaras, así como un sistema de control. Emplean una interfaz humano-máquina que permite una

comunicación específica con el usuario y son capaces de percibir tanto las características del entorno como los movimientos realizados por el usuario. Sus características las hacen capaces de desempeñar tareas específicas y/o resolver problemas abiertos, tomando en consideración variables tales como el entorno de aprendizaje, la participación en grupo, la interacción frecuente, la retroalimentación y las conexiones con el contexto del mundo real

Se trata pues de robots móviles interactivos con fines ludoeducativos, basados tanto en la comunicación verbal como en la no verbal. Si bien ambas plataformas están destinadas al aprendizaje dirigido o semiautónomo de idiomas, en aulas, laboratorios y/o mediatecas, es factible considerar el desarrollo de aplicaciones que operen en otros entornos y relacionadas con otros campos del conocimiento.

Los dos robots cuentan con un sensor Kinect que registra y da seguimiento al registro de la posición  $x,y,z$  de hasta 26 puntos del esqueleto del usuario cada 33 milisegundos. La interfaz gráfica incluye imágenes y animaciones. Ello permite implementar acciones proxémicas, cinésicas, sonoras y/o visuales, confiriendo al robot una capacidad de interacción multimodal poco frecuente hasta ahora.

Ludibot v1 y Ludibot v2 están respaldados por una máquina de estados, misma que permite definir los comportamientos esperados del robot a través de tres elementos: los estados, las transiciones y las acciones multimodales. En ambos casos es posible agregar nuevos estados, transiciones y acciones multimodales para mejorar el comportamiento como material lúdico; ahora bien, en Ludibot v2 se ha simplificado el proceso correspondiente: ello permite ampliar el número de usuarios que pueden agregar elementos y ahorrar tiempo en la programación de nuevas estructuras de juego. La máquina de estados es en ambos casos el componente fundamental para que el robot sea percibido como un material lúdico y, por ende, contribuya a fomentar una actitud lúdica propicia a la motivación y al aprendizaje a través de un material novedoso.

### *Propósito ludoeducativo*

Se ha dicho más arriba que la robótica ludoeducativa tiene como propósito concebir, diseñar y desarrollar aplicaciones lúdicas que promuevan el aprendizaje mediante el diseño y construcción

de robots capaces de desempeñar tareas específicas, mediante el uso de sistemas de control y la programación. La robótica ludoeducativa tiene asimismo por objetivo promover experiencias basadas en una interacción comunicativo bi- o multidireccional propicia a la aplicación de nuevos enfoques de aprendizaje.

Ludibot v1 y Ludibot v2 tienen ese fin, pues buscan contribuir a complementar el proceso de aprendizaje/enseñanza gracias al uso de las tecnologías aplicadas en situaciones formales, semiformales o informales de aprendizaje, mediante la introducción de nuevas dinámicas y nuevas actitudes, recurriendo a y promoviendo el uso de la literacidad mediática multimodal. Se trata, en ambos casos, de fomentar tanto el pensamiento crítico, la intuición, la creatividad, la comunicación y la colaboración, como la curiosidad, la iniciativa y la adaptabilidad. Se otorga así al usuario, que es aquí entendido como sujeto del aprendizaje, un rol activo que no se limita a la recepción y a la repetición de un input, sino que exige producir, interpretar y evaluar la información recibida a través de diferentes modos.

Para alcanzar los fines ludoeducativos planteados, ambas versiones de Ludibot recurren a la ludicización, pues buscan modificar la percepción de los robots, acercándolos al universo de juego, sin pretender transformar el aprendizaje de la lengua en un juego. Se postula así que lo más importante es que la experiencia de aprendizaje de idiomas propiciada por el o los robots sea percibida como lúdica.

Este propósito ludoeducativo está respaldado por el marco teórico y conceptual presentado en el capítulo anterior, cuyos principales fundamentos resumimos a continuación.

### *Fundamentos*

Para la concepción, el diseño y el desarrollo de Ludibot v1 y V2, se tomaron en cuenta las preguntas planteadas por Hilgard [81] respecto del aprendizaje (relativas a la capacidad de aprender, a las diferencias individuales, a la práctica, al papel de la cognición y al papel de las habilidades y de los conocimientos previos), con el fin de identificar las necesidades de los usuarios, determinar los objetivos, definir los contenidos de aprendizaje y diseñar material acorde con los objetivos. En ambos casos se prescinde deliberadamente de la visión meramente conductista del aprendizaje, en aras de lograr una articulación entre los factores externos y el procesamiento interno que de ellos hace el sujeto de aprendizaje.

Así, se integra el postulado central del cognoscitvismo y del constructivismo, corrientes que define al ser humano como un sujeto activo que procesa la información. Se trata de fomentar la estructuración, la organización y la secuencia de la información con miras a un óptimo procesamiento y a la creación de ambientes de aprendizaje estimulantes, donde los estudiantes dispongan de elementos para establecer conexiones con elementos previamente aprendidos. La analogía humano-máquina respalda aquí el diseño interactivo de la interfaz humano-robot móvil.

Por añadidura, Ludibot V1 y Ludibot v2 parten de una visión pluridisciplinar de la noción de interacción, que combina consideraciones ligadas a la robótica, la DLC y las ciencias del juego. Así, apuestan por con una interacción humano-máquina multimodal, que por el momento incluye los modos cinésico, proxémico, visual y sonoro, así como elementos de los modos escrito y expresivo; a futuro, se planea integrar el modo oral. La interacción considerada se da tanto entre el jugador y el robot, mediante la interfaz humano-máquina, como entre usuarios, en el marco de un uso colectivo de la plataforma.

De esta manera, el proceso de aprendizaje interno, considerado subjetivo y personal, es apuntalado mediante la interacción cognitiva, funcional y cultural con los otros, de tal manera que cobra una dimensión social y cooperativa. Además, se postula que la interacción con un robot, por su carácter todavía relativamente inusual en el ámbito educativo, sobre todo en el bachillerato público mexicano, puede incidir en la percepción afectiva del aprendizaje, aumentar la disposición de aprender, modificar las atribuciones de éxito y de fracaso, así como transformar las expectativas y las representaciones.

Para el desarrollo de ambas versiones, se tuvo en cuenta una cuádruple capacidad de adaptación: al idioma (por el momento, español y francés; otros idiomas, a futuro, desde una óptica plurilingüe que integra tanto el binomio lengua/lenguaje como la oposición adquisición/aprendizaje); a la distancia de interacción, gracias al sensor Kinect v2; a la familiaridad con el sistema, gracias a las funciones incluidas en la interfaz de juego que propician un uso intuitivo en autonomía; y, finalmente, a la respuesta, gracias al uso de gestos que permite elegir temas, acciones, duración de la actividad y niveles de dificultad gracias a diferentes estados y transiciones. Esta capacidad de adaptación se ve reforzada por el uso de la máquina de estados.

En ambas versiones de Ludibot se consideraron tanto las competencias generales como las competencias discursivas y lingüísticas, las competencias del siglo XXI y la literacidad mediática multimodal, con especial énfasis en la capacidad de las plataformas de favorecer tanto la motivación intrínseca como la extrínseca durante el proceso de aprendizaje.

Si bien los componentes del material de juego varían entre Ludibot v1 y Ludibot v2, en ambos casos se sigue viendo el material como una parte importante más no suficiente para suscitar la actitud lúdica. Por ello, se busca, por un lado, diversificar las estructuras de juego que, además de definir las tareas y las acciones pertinentes del robot durante la interacción humano-máquina, contribuyen al surgimiento de una disposición de ánimo propicia a una experiencia óptima de juego y aprendizaje; por el otro, se considera la posibilidad de utilizar Ludibot en contextos variados.

### ***Diferencias entre Ludibot v1 y Ludibot v2***

Si bien Ludibot v1 y Ludibot v2 comparten una naturaleza, un propósito y numerosas características en común, también presentan entre sí numerosas diferencias. Esta sección discute las diferencias significativas entre ambas versiones.

Ludibot v1 y Ludibot v2 divergen en cuanto a su apariencia, a su arquitectura general y a su arquitectura de sistemas. Todo ello conduce a un cambio en el carácter ludoeducativo de cada versión, tema que se abordará una vez recapitulados los tres puntos anteriores.

## *Apariencia*



**Figura 2.9. Ludibot v1 y Ludibot V2**

La apariencia es sin duda alguna la principal y más visible diferencia entre el subsistema Robot móvil y Loomo. Según puede observarse en la Figura 2.9, la apariencia de Ludibot v1 es menos atractiva que la de Ludibot v2, por diversas razones: al estar basado en un robot personal de uso comercial, Ludibot v2 corresponde mejor a la representación usual de un robot.

Resulta natural que el primer prototipo, desarrollado desde cero en laboratorio con recursos relativamente modestos, tenga un aspecto más rudimentario e inacabado. Es difícil poder realizar diseños industriales como el que posee Loomo en robots construidos en entornos académicos. Las proporciones de los elementos, la cabeza con pantalla integrada, el cuerpo con dos extensiones que asemejan brazos, los ángulos redondeados, la iluminación integrada, la minicomputadora embarcada que sustituye a la computadora personal, los materiales de fabricación, entre otros elementos, brindan un aspecto más llamativo a Ludibot v2. Además, se incluye un robot más pequeño.

La relevancia del cambio de material va más allá de consideraciones meramente estéticas. En efecto, el subsistema Robot móvil de Ludibot V1 es un robot de tres ruedas mientras que el robot Loomo utilizado en Ludibot v2 es un robot auto balanceado. Al controlar la inclinación y brindar un movimiento más suave al sensor Kinect, Loomo evita que la detección adecuada de los usuarios se vea afectada y reduce por ende el riesgo de interrumpir involuntariamente la partida.

### Arquitectura general

Los cambios en la apariencia obedecen también a razones arquitecturales. La arquitectura general de Ludibot v2 es más compleja que la de Ludibot v2, según es posible verificarlo al comparar los diagramas de ambas versiones (véase Figura 2.10).

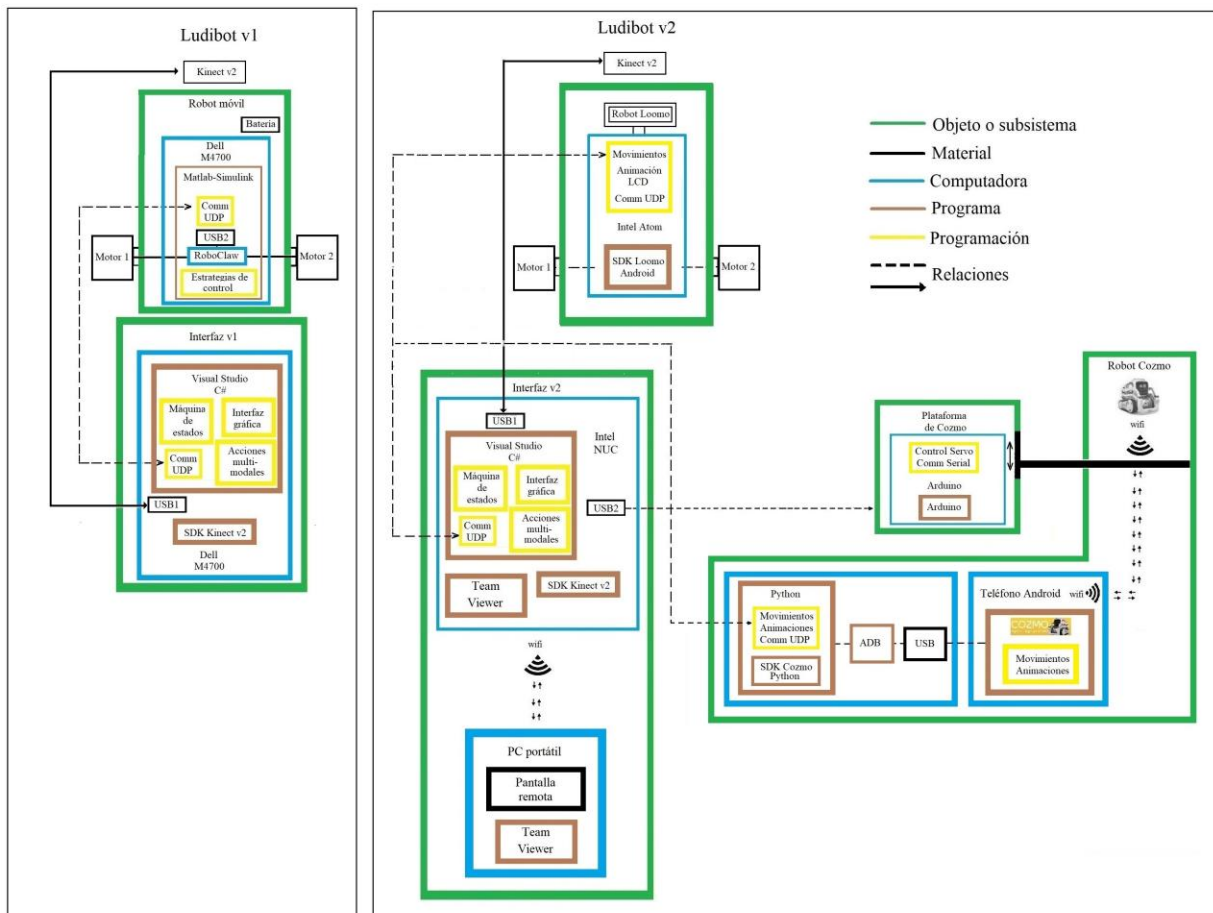


Figura 2.10. Diagramas de la arquitectura de Ludibot v1 y Ludibot v2

Los cambios introducidos no sólo obedecen a un mayor número de subsistemas (dos en Ludibot v1; cuatro en Ludibot v2), pues incluso los sistemas comunes en ambas versiones (robot móvil principal y robot Loomo, interfaz) se vieron modificados.

En efecto, allí donde en Ludibot v1 la arquitectura incluía un solo robot móvil manufacturado para los fines específicos del proyecto, correspondiente a un subsistema, en Ludibot v2 se incluyen dos robots comerciales (Loomo, el robot principal, y Cozmo, un robot auxiliar), correspondientes a dos subsistemas.

Ahora bien, aunque tanto el robot móvil de Ludibot 1 como Loomo están equipados con motores y un sensor Kinect, la computadora Dell fue sustituida por una minicomputadora Intel NUC más compacta, que ya no es visible por el usuario. La arquitectura de la segunda versión de Ludibot emplea también más puertos USB para conectar el subsistema Cozmo. Se introdujo asimismo el programa TeamViewer, con el fin de establecer comunicación inalámbrica entre la interfaz que se ejecuta en la computadora Intel NUC y la pantalla remota, conectada a su vez a una computadora portátil o a un proyector, lo cual permite visualizar a distancia las imágenes mostradas por la interfaz. Por añadidura, la computadora NUC es manipulada a distancia gracias a una terminal remota, via TeamViewer.

Ludibot v2 cuenta con dos nuevos subsistemas propios Cozmo y Plataforma. Ambos trabajan en conjunto, pues el segundo está al servicio del primero. La programación de la máquina de estados a partir de las acciones de Loomo y Cozmo permite a ambos robots interactuar entre sí y con los usuarios para hacer los juegos más atractivos.

La plataforma de Cozmo sirve de base para transportar al minirobot, bajarlo al piso durante el juego y permitirle volver a su posición inicial. Por el momento, el desplazamiento de Cozmo se basa únicamente en su odometría. Ahora bien, debido a la acumulación de errores, Cozmo no siempre logra regresar a su base para subir por la plataforma. Se tiene previsto desarrollar un control visual para que la cámara de Loomo localice al robot Cozmo, facilitando el regreso a la base. La localización en 3D del robot Cozmo mediante un esquema visual de cámara en mano, asociada a una máquina de estados, permitirá tomar el control de Cozmo utilizando un gesto y capturar una posición de una mano detectada por Kinect para guiar a Cozmo a su base, manualmente y a distancia.



Una vez definido el lenguaje de programación de Ludibot v2 para incluir acciones sobre el robot Cozmo, será posible moverlo a una posición deseada o realizar una animación. Esta opción ampliará el potencial de uso del robot, pues algunas de las aplicaciones ludoeducativas podrían estar basadas solamente el robot Cozmo, significativamente más compacto que Loomo, más fácil de transportar, de almacenar e incluso de adquirir.

Las aplicaciones ludoeducativas basadas en Cozmo ya no requerirían un espacio tan amplio como Loomo, pues funcionarían incluso en un escritorio. El material requerido consistiría en el robot Cozmo, el sensor Kinect y una computadora portátil. Ello abriría la posibilidad de explorar el uso de los robots ludoeducativos en situaciones formales e informales de aprendizaje caracterizadas por contextos difíciles. Tal sería el caso, por ejemplo, de una aplicación que permitiese familiarizar a migrantes no hispanohablantes con el *input* lingüístico y discursivo propio de la valoración médica con fines de diagnóstico. Se respondería así a la necesidad real de aquellos migrantes provenientes de África o del Caribe francófono que hoy se agolpan en los albergues del Norte de la República mexicana. Tal sería asimismo el caso de comunidades rurales de difícil acceso donde podría promoverse una visión plurilingüe que equilibre el valor dado a la lengua materna y a la lengua vehicular dominante.

#### *Arquitectura de sistemas*

Se ha establecido ya una comparación pormenorizada entre la arquitectura de ambas versiones de Ludibot. Baste aquí, por ende, presentar de manera conjunta ambas versiones a manera de tabla (véase Tabla 2.16), desde la óptica de sus respectivas arquitecturas de sistema. Los objetos correspondientes a Ludibot v2 aparecen en las filas sombreadas.

Objetos	Características y propiedades			Relaciones	Comportamientos
	Material	Programas	Programación		
<b>Robot móvil</b>	Computadora M4700 Amplificador de potencia Motores Decodificadores ópticos incrementales Batería Estructura mecánica Ruedas	Matlab-Simulink QuaRC Sistema operativo	<b>Implementación</b>  Estrategia de control Comunicación UDP	Recibe de la interfaz v1 la posición deseada vía la comunicación UDP	Utiliza la posición deseada para mover al robot móvil mediante las estrategias de control
<b>Loomo</b>	Robot móvil Loomo (dos ruedas) autobalanceado Pantalla LCD de la cabeza (2gdl)	SDK Loomo Sistema operativo Android	Movimiento del robot Animación en la pantalla LCD Comunicación UDP	Recibe posición y animaciones de la interfaz v2 vía la comunicación UDP	Utiliza la posición deseada para mover al robot móvil Realiza animaciones con la cabeza y la pantalla LCD
<b>Interfaz v1</b>	Computadora M4700 Pantalla Kinect v2	Visual Studio (C#) SDK de Kinect V2 Sistema operativo	Interfaz gráfica Máquina de estados (detección de entradas, definición de estados, transiciones de estado) Acciones multimodales Comunicación UDP	Envía al robot móvil la posición deseada vía la comunicación UDP	Detecta los gestos como entradas a la máquina de estados Realiza transiciones de que cambian el estado Repite las acciones multimodales asociadas a un estado Ejecuta acciones multimodales definidas en las transiciones Lleva a cabo las aplicaciones lúdicas usando la máquina de estado
<b>Interfaz v2</b>	Kinect v2 Pantalla remota vía TeamViewer Mini-PC Intel NUC PC portátil	Visual Studio (C#) SDK de Kinect V2 TeamViewer	Lenguaje propio de programación de marcado (tipo HTML); incluye variables y transiciones por variable booleana y por gestos Interfaz gráfica Máquina de estados Acciones multimodales Comunicación UDP	Envía al robot Loomo movimiento deseado y ejecución de animaciones vía comunicación UDP Envía al robot Cozmo posición la deseada de movimiento y la ejecución de animación vía comunicación UDP Sube o baja la plataforma de Cozmo	Detecta los gestos para efectuar transiciones Efectúa transiciones por variable booleana Ejecuta transiciones de estado y acciones multimodales en la transición Ejecuta cambios de estado y acciones multimodales de estado Manda acciones a Cozmo, a Loomo y a la plataforma de Cozmo
<b>Cozmo</b>	Minirobot móvil Cozmo Teléfono Android	Aplicación Cozmo Sistema operativo Android Python Android Debug Device (ADB)	Comunicación UDP con la Interfaz V2 Ejecución de movimientos y animaciones	Recibe movimientos y ejecución de animaciones de la interfaz vía comunicación UDP	Utiliza la posición deseada para mover al robot Realiza animaciones
<b>Plataforma</b>	Arduino Mega Servomotor Estructura mecánica	Arduino IDE	Control del servomotor Comunicación vía puerto virtual serial USB	Recibe de la interfaz v2 la indicación de bajar o subir su plataforma	Controla el movimiento del servomotor para subir o bajar el robot Cozmo

**Tabla 2.16. Arquitectura de sistemas de Ludibot v1 (sin sombreado) y Ludibot v2 (con sombreado)**

## *Perspectivas*

Según puede apreciarse en las líneas anteriores, el desarrollo de ambas versiones de Ludibot ha permitido alcanzar varias de las metas inicialmente planteadas en el proyecto. Los objetivos alcanzados y por alcanzar se discuten en la conclusión del presente trabajo. Sin embargo, cabe destacar aquí, tras haber comparado las dos plataformas desarrolladas, algunas de las perspectivas abiertas por las labores asociadas al desarrollo de Ludibot v1 y Ludibot v2.

En un horizonte cercano, será importante poner a prueba la más reciente versión de Ludibot dentro de un contexto de interacción con los usuarios. El pilotaje permitirá tanto comprobar las ventajas de las modificaciones introducidas como identificar los ajustes por realizar. La contingencia sanitaria mundial iniciada en enero de 2020 ha obligado a posponer esta etapa.

Una vez realizado el pilotaje, será posible introducir los ajustes identificados, pero también se podrá comenzar a trabajar en la concepción, el diseño y la implementación de nuevas aplicaciones lúdicas, en diferentes idiomas, con mayor número de modos –entre ellos el oral, tan importante en el aprendizaje de una lengua–, asociadas a diversas competencias, en contextos con objetivos de aprendizaje diversos.

Por ahora, la plataforma Ludibot cuenta con un juego completo, *De la tête aux pieds*. En el siguiente capítulo, tras una descripción detallada de la interfaz de Ludibot v1 y una exposición general de la interfaz de Ludibot v2, se presentan tres aplicaciones: Te sigo, Posicióname y el juego *De la tête aux pieds*.

## Capítulo 3. Interfaces y aplicaciones de Ludibot

Habiendo presentado en el capítulo 2 la arquitectura de sistema de ambas versiones de Ludibot, y subrayado las características propias de un robot ludoeducativo, en este tercer y último capítulo se expondrá el desarrollo de ambas interfaces y otras aplicaciones diseñadas para cada versión. Las aplicaciones pertenecen a dos niveles semánticos de la metáfora lúdica: el material y la estructura. En efecto, si bien forman parte del sistema tangible y pueden ser consideradas entonces como aquello con lo que se juega, también inciden en la manera en que dicho material es utilizado, remitiendo entonces a aquello a lo que se juega.

Incluimos el análisis de las dos interfaces dentro del análisis de las aplicaciones ya que, estrictamente hablando, la primera aplicación desarrollada para cada versión de Ludibot, con fines de comunicación, fue su interfaz. La interfaz permite tanto la conexión funcional entre los diferentes subsistemas como la interacción humano-máquina. En otras palabras, la interfaz puede ser entendida como un sistema de interacción basado en una combinación de sensores, que permiten al robot establecer una comunicación específica.

Algunos trabajos actuales de investigación involucran el reconocimiento de movimientos corporales como la emoción facial, movimientos con el cuerpo y el reconocimiento de gestos, específicamente con la cara, las manos y el cuerpo. La interfaz permite así interpretar el significado del lenguaje corporal humano [113]. En ambas versiones del robot, se desarrolló una interfaz cuya interacción humano-máquina está basada en gestos. Se trata de un área de estudio con creciente relevancia en los últimos años [113]. En efecto, los gestos son considerados como una forma natural de interacción humana que ocurre por medio de movimientos físicos de la cabeza, manos, brazos, cara u otras partes del cuerpo. En el área de robótica, el reconocimiento de gestos tiene como objetivo interpretar gestos humanos a través de algoritmos matemáticos y/o computacionales.

Entre los sensores usados para el reconocimiento de gestos se encuentra el Kinect v2 como una herramienta basada en cámaras y sensores infrarrojos que permiten identificar gestos y posturas en tiempo real. Diversos autores han trabajado con este sensor, logrando capturar y

procesar datos para mover robots móviles y articulares [40]. También ha sido utilizado para el reconocimiento de gestos y posturas en las artes marciales y otros deportes específicos [87], en el baile [116] y en aplicaciones musicales [137].

Entre los trabajos más destacados se encuentra el de Baron *et al.* [15], cuyos autores presentan un sistema de reconocimiento de gestos para el desarrollo de una interfaz humano-robot para un brazo robótico. Por su parte, Grazia *et al.* [70] proponen un clasificador para reconocer posturas estáticas del cuerpo humano y gestos corporales en tiempo real usando un método denominado lenguaje de descripción de gestos. Dardan *et al.* [44] realizan una interfaz para reconocimiento de gestos de niños mientras bailan; Cueva *et al.* [42] usan una interfaz de las librerías del SDK para un robot industrial de siete grados de libertad, sin introducir cambios en la interfaz proveniente del SD. Finalmente, Benabdallah *et al.* [17] implementan una interfaz basada en LabVIEW para un robot industrial.

Partiendo del estado del arte de las interfaces basadas en Kinect v2, para el desarrollo del Ludibot se buscó implementar con base en ese mismo sensor una interfaz humano-máquina de fácil uso, descriptiva, interactiva, atractiva para los aprendientes y docentes de cualquier área. La interfaz debe ser programable para el uso de aplicaciones, actividades y juegos destinados al aprendizaje. Por ende, debe permitir diferentes movimientos y trayectorias.

La interfaz de Ludibot v1, descrita a continuación, tiene como propósito central la interacción con los estudiantes mediante una interfaz humano-robot móvil que facilite al usuario controlarlo mediante gestos con las manos, mientras que el robot maneja sonidos, imágenes y palabras.

### **3.1 Interfaz de Ludibot v1**

La interfaz de Ludibot v1 está basada en una interacción humano-máquina. Está implementada de manera que el usuario puede seleccionar diversas funciones para el robot móvil, y tiene como objetivo aprovechar la comunicación no verbal para un aprendizaje activo y significativo de idiomas. Con miras a fomentar la autonomía, incluye un menú que permite al usuario descubrir de manera independiente el funcionamiento del robot ludoeducativo, practicar el vocabulario por adquirir y, finalmente, jugar.

La interfaz está inspirada en un modelo de máquina de estado finito. Al brindar al usuario la posibilidad de elegir entre diversas funciones, permite realizar transiciones usando como entrada gestos y ademanes de las manos. Estos gestos y ademanes son definidos según dos posiciones (arriba/abajo) y dos estados (mano abierta/mano cerrada). El usuario logra que el robot pase de un estado a otro mediante la adquisición y el reconocimiento de los parámetros del cuerpo humano. Para la interacción con el usuario, la interfaz recurre además a sonidos expresivos, símbolos, figuras, entre otros.

### **3.1.1 Estados principales de la interfaz de Ludibot v1**

La Figura 3.1 muestra el diagrama de estados de la interfaz de Ludibot v1. En la interfaz se representan los estados con óvalos y las transiciones con una flecha. Las flechas de transición señalan el paso de un estado de origen a un estado de destino. A cada transición se asocia una condición que la desencadena. Las acciones asociadas a una transición se ejecutan sólo una vez en el estado de destino. Las acciones asociadas a un estado se ejecutan cuando el robot se encuentra en ese estado.

*Aprende a controlarme* es el estado inicial. A partir de ese estado, el usuario puede experimentar por primera vez el uso del robot y descubrir los gestos básicos para controlarlo. Una vez que el usuario ha aprendido los gestos necesario, el gesto *Salir* le permite ejecutar la función *Menú*.

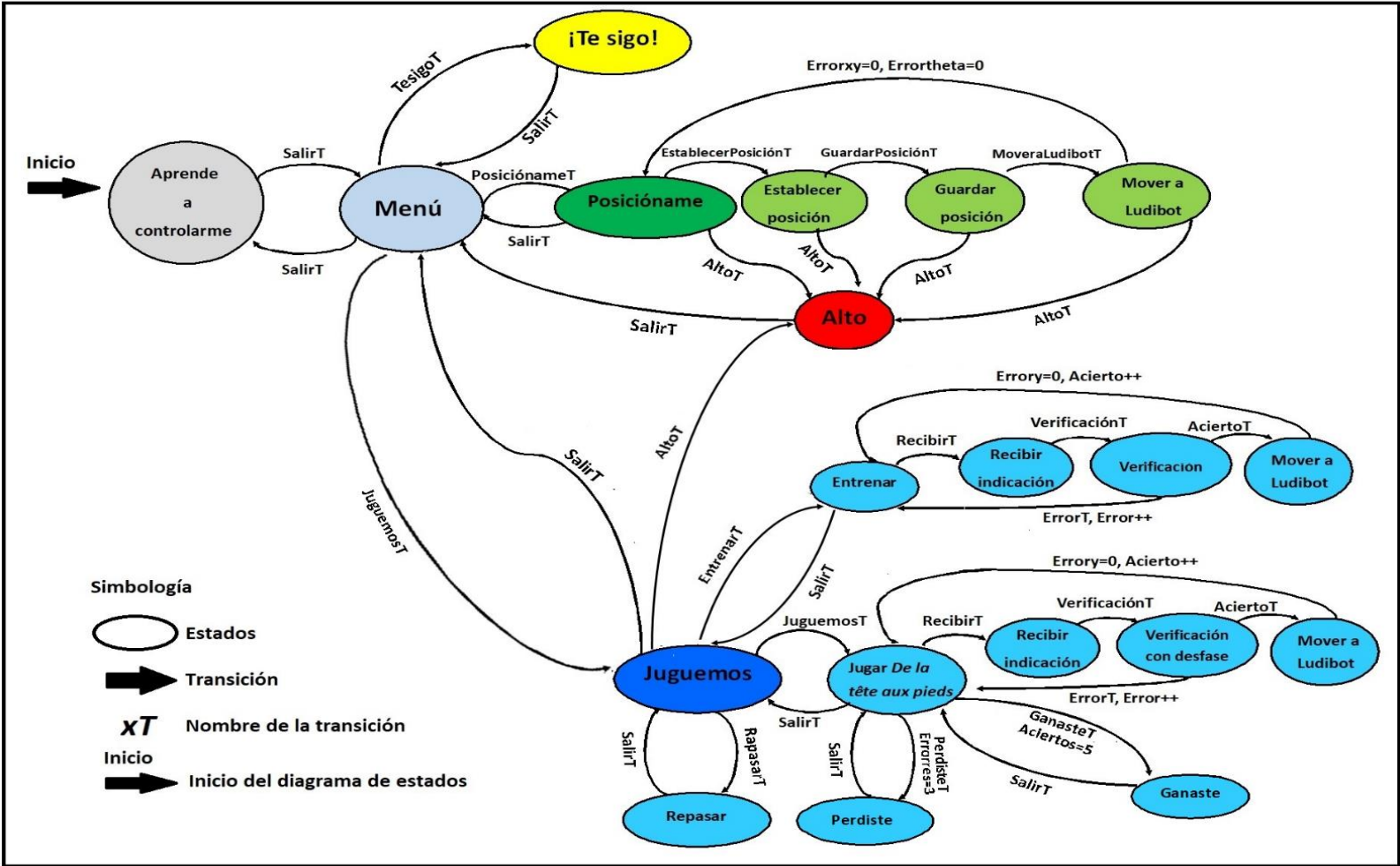
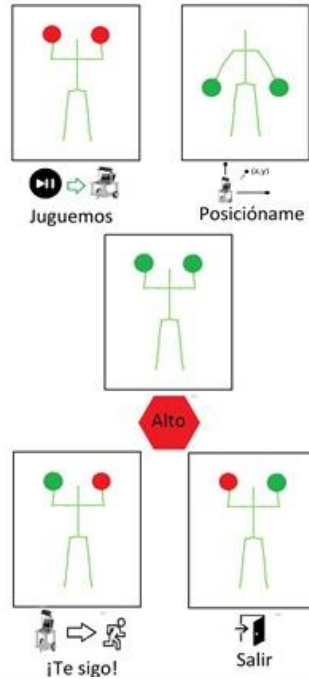


Figura 3.1. Diagrama de estados de la interfaz de Ludibot V1

Una vez en la función *Menú*, el usuario puede elegir entre cinco funciones: *Posicióname*, *¡Te sigo!*, *Juguemos*, *Alto* y *Salir*. Para ello, deberá realizar los gestos correspondientes ala función elegida (véase Figura 3.2).

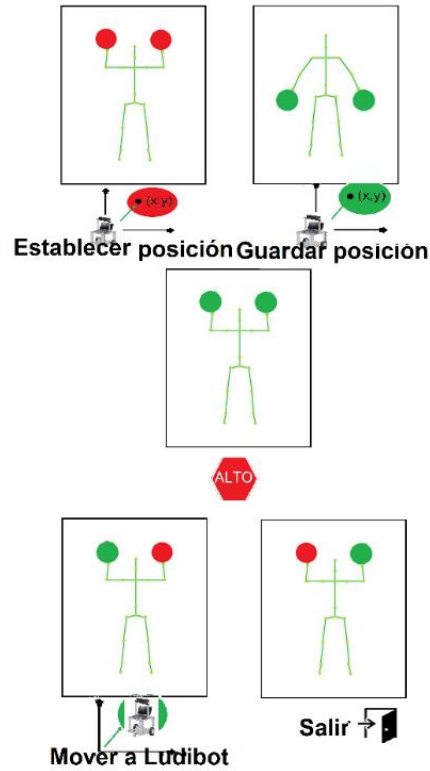


**Figura 3.2. Gestos disponibles en la función *Menú* de la interfaz**

Dentro de la función *Posicióname*, el usuario puede escoger los gestos *Establecer posición*, *Guardar Posición*, *Mover a posición*, *Alto* y *Salir* (véase Figura 3.3). La función *Establecer posición* permite al usuario establecer una posición deseada. *Guardar posición* indica al robot que debe permanecer en la posición definida con las manos. *Mover a posición* ejecuta el movimiento a la posición deseada. La función *Alto* permite detener el robot ante cualquier emergencia. El gesto *Salir* regresa al estado inicial *Aprende a controlarme*.

La aplicación *Posicióname* es presentada con más detalle más adelante en este trabajo, en la sección específica destinada a esa aplicación.

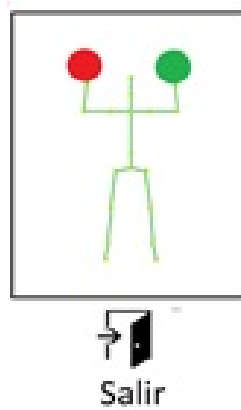




**Figura 3.3. Gestos disponibles en la función *Posiciónname* de la interfaz**

La función *¡Te sigo!* (véase Figura 3.4) permite al usuario activar el seguimiento del robot: el robot se despalza en función del usuario, a una distancia específica y según el ángulo deseado. En este caso, también es posible elegir el gesto *Salir* para regresar al *Menú*.

La aplicación *¡Te sigo!* es presentada con más detalle más adelante en este trabajo, en la sección específica destinada a esa aplicación.



**Figura 3.4. Gestos disponibles en la función *¡Te sigo!* de la interfaz**

La función principal desde el punto de vista lúdico es *Juguemos* (véase Figura 3.5), ya que permite seleccionar el juego *De la tête aux pieds*. Se trata de un juego donde el jugador debe seguir indicaciones desfasadas. El robot anuncia mediante un audio una parte del cuerpo. La primera vez, el jugador debe permanecer inmóvil. A partir de la segunda propuesta, el jugador tiene que tocar la parte del juego sugerida en la indicación anterior para acumular puntos.

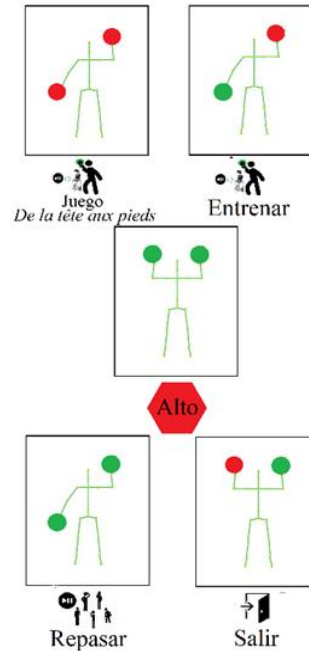
Por ejemplo, el robot empieza diciendo “la cabeza” y el jugador debe permanecer inmóvil; de lo contrario, el robot emitirá una señal de error y contabilizará un error. El robot dice enseguida “la rodilla”: el jugador debe tocarse la cabeza para obtener un acierto. El robot menciona en tercer lugar “la cintura”: el jugador debe tocarse la cintura. Al cabo de cinco aciertos, el jugador gana. Si el jugador comete más de tres errores, pierde.

La función *Juguemos* permite al usuario elegir entre cinco opciones: *Juego partes del cuerpo*, *Repasar*, *Entrenar*, *Alto* y *Salir*.

La función *Repasar* permite al usuario revisar el vocabulario y experimentar la detección de las partes del cuerpo, como una preparación al juego. Cuando el usuario hace un gesto, el sensor detecta la parte del cuerpo que es tocada y la muestra en la ventana correspondiente.

La función *Entrenar* permite jugar sin contabilizar puntos. En tal caso, el usuario experimenta la dinámica de juego, pero no puede ni ganar ni perder.

Como en los casos anteriores, *Alto* permite detener la interacción, mientras que *Salir* permite regresar al menú principal.



**Figura 3.5. Gestos disponibles en la función *Jugamos* de la interfaz**

La aplicación *Jugamos* es presentada con más detalle más adelante en este trabajo, en la sección específica destinada a esa aplicación.

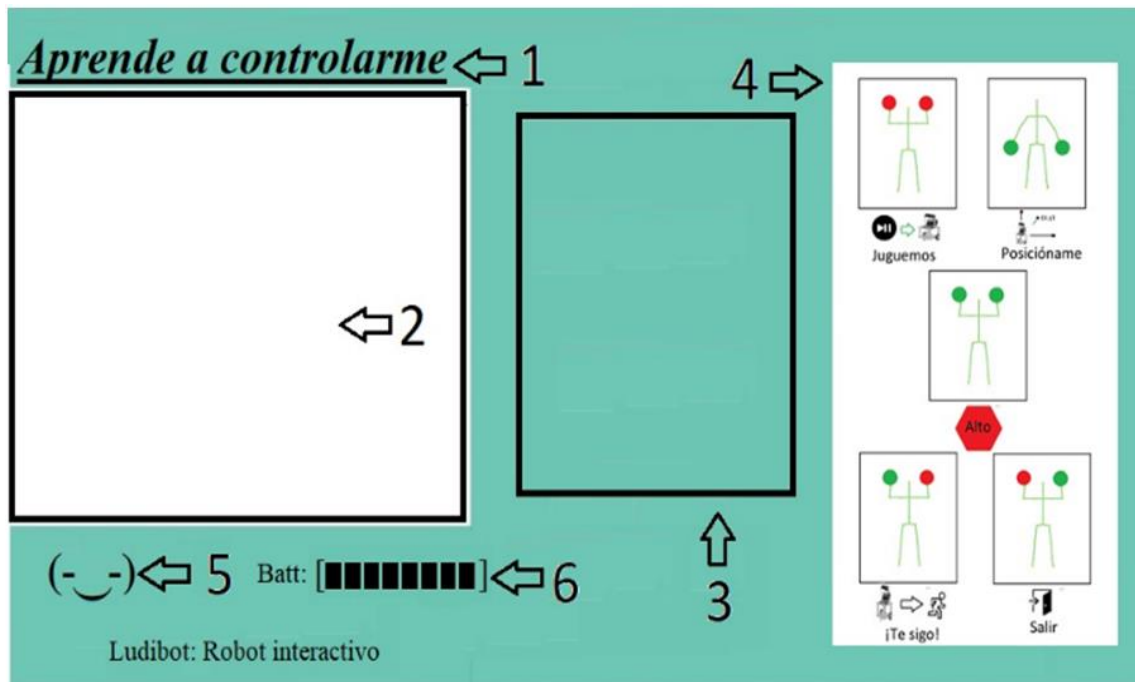
### 3.1.2 Elementos de la interfaz humano-máquina de Ludibot v1

La ventana principal de la interfaz humano-máquina de Ludibot v1 es mostrada en la Figura 3.6. Está compuesta por los siguientes elementos:

1. La barra de estados.
2. La ventana de animación.
3. La ventana del gesto detectado.
4. La ventana de los gestos disponibles.
5. El estado de detección de Kinect.
6. La barra de estado de la batería.

*Aprende a controlarme* es el estado inicial de la interfaz. En él, el usuario puede descubrir cómo funciona Ludibot y aprender los gestos preestablecidos para la interacción. Una vez que el usuario ha entendido y memorizado los gestos, el gesto correspondiente a *Salir* le permite transitar hacia el menú. En el estado *Menú*, el usuario tiene la posibilidad de elegir entre cuatro estados: *¡Te sigo!*, *Posicióname*, *Jugamos* y *Salir*. A cada uno de ellos corresponde un

gesto específico. *¡Te sigo!* permite que Ludibot siga al usuario en movimiento, según la distancia y el ángulo predefinidos en el diagrama de control. *Posicióname* permite mover el robot a la posición deseada, recurriendo a los datos relativos a las magnitudes y al ángulo del brazo derecho del usuario. *Salir* permite regresar al menú o, en el caso de estar ya en *Menú*, regresar al estado inicial *Aprende a controlarme*.



**Figura 3.6. Interfaz humano-máquina de Ludibot v1**

La barra de estados (1) describe la función (o estado) actual del robot: *Aprende a controlarme*, *¡Te sigo!*, *Posicióname*, *Juguemos*.

La ventana de animación (2) muestra una representación del cuerpo humano obtenida a partir del sensor Kinect. Se basa en 25 articulaciones (véase Figura 3.7), gracias a las cuales es posible medir los parámetros del cuerpo, la distancia a la que se encuentra el usuario y el estado de las manos que sirve para controlar el robot por medio de gestos. La interfaz codifica las manos cerradas representándolas mediante un círculo rojo (véase Figura 3.8), mientras que las manos abiertas se codifican mediante círculos verdes (véase Figura 3.9). Para las aplicaciones desarrolladas hasta ahora, se emplean las posiciones de las manos y el punto superior del pecho a la altura del hombro, mostrado en la Figura 3.7 con el número 21 (*SpineShoulder*)

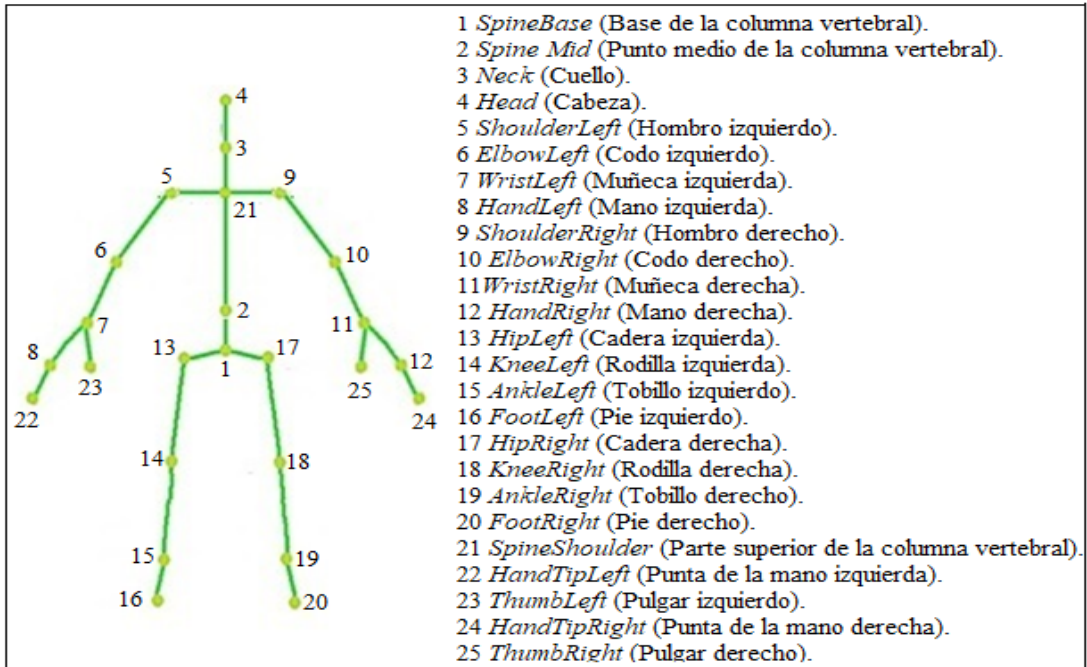


Figura 3.7. Representación del cuerpo humano con 25 articulaciones

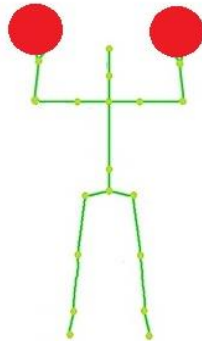


Figura 3.8. Animación del cuerpo con las manos cerradas

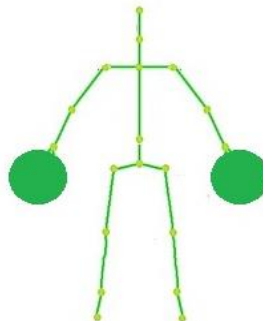
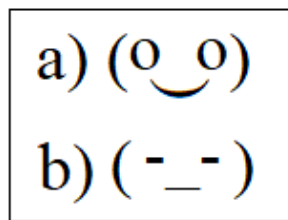


Figura 3.9. Animación del cuerpo con las manos abiertas

La ventana del gesto que se detecta (3) permite al usuario saber cuál es el gesto identificado en ese momento por el sensor. Por su parte, la ventana de gestos disponibles (4) indica los gestos que están permitidos en esa función, cuya lista varía.

El estado de detección de Kinect (5) se representa mediante emoticonos muy sencillos, implementados con caracteres ASCII. El primer emoticón representa una cara sonriente con los ojos abiertos (véase Figura 3.10a) e indica que el sensor Kinect detecta al usuario. El segundo representa una cara de expresión neutra con los ojos cerrados (véase Figura 3.10b) e indica que el sensor Kinect está inactivo pues no está detectando ningún usuario.



**Figura 3.10. Estados de detección de la interfaz con el sensor Kinect v2**

Finalmente, la barra de estado de la batería (6) muestra el nivel de la batería que energiza Ludibot.

## **3.2 Interfaz de Ludibot v2**

### **3.2.2 Descripción general de la interfaz de Ludibot v2**

La interfaz de Ludibot v2 está desarrollada para el procesador Intel NUC en Visual Studio. Dicha interfaz permite determinar el comportamiento de Ludibot v2 y está programada en lenguaje HTML en C#. Esta interfaz usa gestos y ademanes que permiten diferentes funciones y transiciones durante la interacción humano-máquina.

La interfaz de Ludibot v1, programada mediante diversas clases, presenta una estructura demasiado amplia y compleja. La extensión del código dificulta su configuración, su desarrollo y su comprensión. La segunda interfaz de Ludibot tiene como objetivo ofrecer al usuario una opción más sencilla, concreta y fácil de su implementar en cuanto a su programación y su configuración. Se trata de que el usuario de Ludibot v2, sin ser necesariamente especialista en computación y/o robótica, sea capaz de comprender la estructura de la interfaz y, por ende, de participar en la configuración de la interfaz e incluso en su modificación, con miras ya sea a la

contextualización de las aplicaciones ludoeducativas y su adaptación a objetivos de aprendizaje específicos, ya sea al desarrollo de nuevas aplicaciones.

La interfaz de Ludibot v2 está basada en un modelo que realiza cálculos en forma automática a partir de una entrada, para producir una salida determinada. La entrada depende de los movimientos del usuario, adquiridos mediante Kinect v2, y seleccionados para definir la salida, a saber, el comportamiento esperado tanto del robot Loomo como del robot Cozmo.

La interfaz está estructurada con base en un modelo de una máquina de estado finito, dotada de un conjunto de estados, un conjunto de entradas, un conjunto de variables, un conjunto de transiciones, un estado inicial, un conjunto de acciones y un conjunto de gestos y ademanes. Es necesario asignar a los estados un nombre característico de la función, por ejemplo “Inicio” o “Menú”, con el fin de establecer temporalmente la situación, la entrada o el modo de Ludibot v2.

Por lo general, las entradas son definidas mediante un gesto o un ademán específico. Sin embargo, también es posible especificarlas mediante el valor de una variable booleana (verdadero o falso).

Las variables usadas en la interfaz pueden ser de tres tipos: booleanas, enteras y reales. Es posible comparar los tipos de variables ( $>$ ,  $<$ ,  $<=$ ,  $>=$  o  $==$ ), así como incrementar, decrementar o mantener un valor fijo previamente asignado. Las comparaciones se especifican con tres variables: el resultado y dos variables por comparar. Entre las variables predeterminadas pueden estar, por ejemplo, la posición deseada de Ludibot o la posición deseada de la cabeza de Ludibot.

Las transiciones son determinadas mediante un estado de inicio, una entrada y un estado siguiente. Así, Ludibot v2 puede cambiar su estado ya sea a partir de un gesto y/o un ademán, ya sea a partir del valor de una variable.

Todos los estados definidos se consideran como estados finales, de tal manera que Ludibot v2 permanezca en uno de esos estados definidos. Un estado definido puede incluir acciones tanto para una transición como para un estado. Las acciones pueden ayudar a pausar, posicionar o capturar la posición de una parte del esqueleto detectado por Kinect, por ejemplo, la posición de la mano derecha.

La lista de gestos y ademanes que puede detectar el sensor Kinect v2 se define gracias a una base de datos generada con un constructor de gestos (*Kinect Visual Gesture Builder*). Es posible detectar hasta 6 gestos de manera simultánea dentro de un mismo estado.

El lenguaje de marcado es utilizado para seleccionar el comportamiento deseado de Ludibot. Para ello, Ludibot v2 dispone de un analizador del marcaje que permite la construcción de una máquina de estados. Dicho analizador es capaz de reconocer un conjunto cerrado de etiquetas con cierto tipo de atributos. El lenguaje de programación correspondiente es presentado en el “Apéndice B. Lenguaje de programación para implementar la máquina de estados de Ludibot v2 mediante un lenguaje de marcado tipo HTML”, que tiene como objetivo adentrar al usuario y/o desarrollador de aplicaciones ludoeducativas en la estructura de la programación para la interfaz.

En dicho apéndice se detalla cómo utilizar tanto los elementos siguientes como las operaciones pertinentes con cada uno ellos:

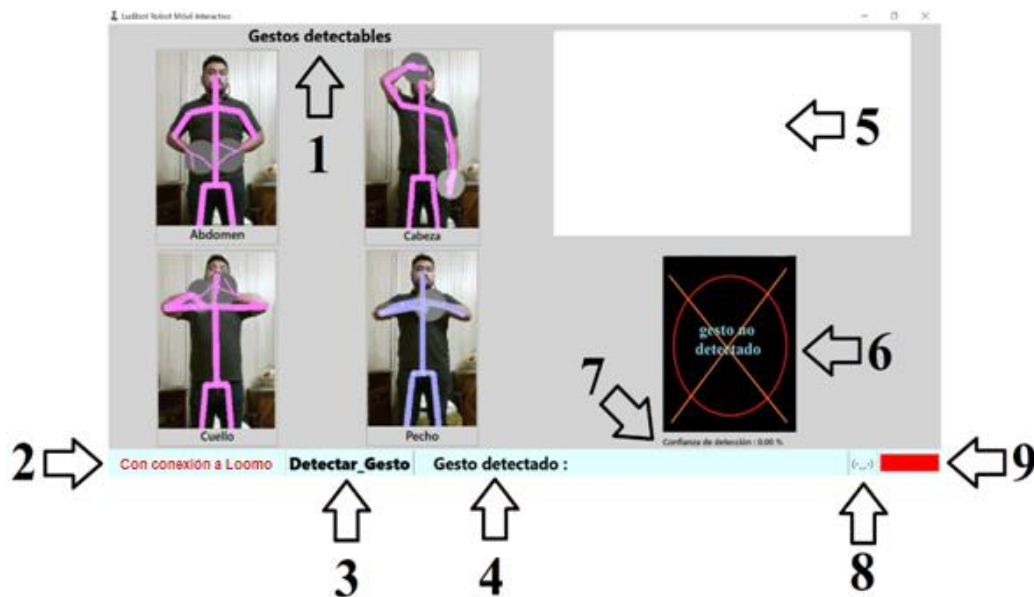
- Archivos.
- Subdirectorios.
- Estructura de la máquina de estados.
- Tipos de variables.
- Estados.
- Gestos.
- Acciones.
- Acciones con variables.
- Impresión de textos.
- Pausa.
- Reproducción de audios.
- Captura del esqueleto con Kinect v2.

### **3.2.2 Elementos de la interfaz humano-máquina de Ludibot v2**

La interfaz de Ludibot v2 fue diseñada tomando en cuenta las dificultades de implementación y de programación identificadas durante el desarrollo de la primera versión, cuyo código de programación era muy extenso. Por ende, la interfaz de Ludibot v2 va acompañada por un manual de referencia de la programación, con el objetivo de que cualquier usuario de Ludibot v2 disponga de las herramientas necesarias para configurarla.



La interfaz de Ludibot v2 usa, por un lado, acciones paralingüísticas dadas por gestos y ademanes, con el propósito para definir estados preestablecidos; por el otro, recurre a las posturas dadas por el diagrama esquemático obtenido a partir de las partes del cuerpo susceptibles de ser detectadas y de sus diferentes combinaciones posibles. Utiliza asimismo la proxémica, expresada por la distancia de detección del usuario.



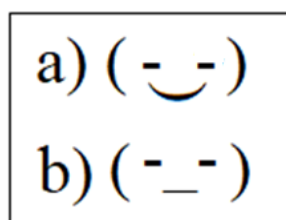
**Figura 3.11. Interfaz humano-máquina de Ludibot v2**

En lo que atañe a la comunicación no verbal, la interfaz usa sonidos específicos que permiten al usuario reconocer errores y aciertos durante la identificación de las partes de cuerpo. La interfaz de Ludibot v2 tiene características paratextuales, expresadas mediante ilustraciones para representar cada estado y cada gesto detectables. Usa gráficos, figuras y diagramas, complementadas con características tipográficas tales como el uso de diversos tipos de letra, el espaciado, el diseño y el marcado en negritas en los textos.

La interfaz humano-máquina contiene los siguientes elementos (véase Figura 3.11):

1. Gestos detectables por la interfaz. Esta sección muestra los gestos y los ademanes disponibles en ese momento en la interfaz. En efecto, los gestos detectables dependen de la función actual de la interfaz, aquí se muestran los gestos y ademanes disponibles en dicha función.
2. Texto de conexión con Loomo. Éste indica si el robot está activo y conectado con la interfaz, o no, en un momento dado.

3. Función actual de la interfaz. Describe la aplicación disponible en el momento.
4. Texto del gesto detectado. Éste informa acerca del gesto detectado de entre los gestos disponibles en (1). Aparece cuando el usuario selecciona el gesto que le permite transitar a una función específica.
5. Ventana de animación. Muestra el esqueleto del cuerpo con sus 25 articulaciones. gracias al cual, en asociación con el estado de las manos (abiertas o cerradas) es posible medir los parámetros del cuerpo y la profundidad a la que se encuentra el usuario.
6. Ventana del gesto detectado. Muestra gráficamente el gesto detectado en un momento dado.
7. Nivel de confianza de detección de la interfaz. Se indica en un rango de porcentaje que va de 0 (nivel nada confiable) a 100 % (nivel altamente confiable). Está basado en Kinect Visual Gesture Builder, que usa el algoritmo AdaBoost. Es particularmente útil para determinar el rendimiento en el aprendizaje de gestos.
8. Estado de detección de Kinect v2. Indica si el sensor está detectando el cuerpo o no. Utiliza íconos compuestos de caracteres ASCII: la cara que sonríe (véase Figura 3.18a), significa que Kinect v2 detecta al usuario; la cara triste (véase Figura 3.18b) significa que Kinect v2 no está detectando a ningún usuario.
9. Barra de estado de la batería. Muestra el nivel de batería disponible.



**Figura 3.12. Estados de detección de la interfaz con el sensor Kinect v2**

A continuación, se presentan tres aplicaciones desarrolladas para Ludibot v1, que tienen en común el uso de la interfaz arriba descrita. Las dos primeras aplicaciones, *Posicióname* y *¡Te sigo!* no son lúdicas, pero sí pueden ser usadas con fines educativos, en la medida en que permiten promover la literacidad digital. Además, son útiles para aprender a usar el robot. La tercera, *De la tête aux pieds*, es hasta ahora la primera y única aplicación propiamente lúdica, e incluye tanto el juego en sí como una fase de descubrimiento como otra de entrenamiento.

### 3.3 Aplicación *Posicióname*

Esta aplicación mueve a Ludibot v1 a una posición deseada, indicada gracias a la posición las manos del usuario. La posición de las manos se mapea en el marco de referencia de Kinect para establecer la posición deseada en el marco de referencia del robot. Una vez establecida dicha posición, se emplea el control no lineal mencionado en [32][33]; el esquema de la implementación del control se muestra en el diagrama de la Figura 3.13.

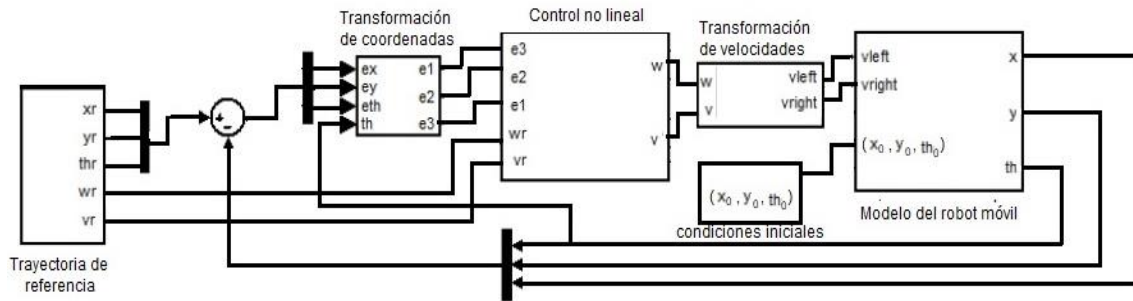


Figura 3.13. Diagrama del control no lineal basado en el seguimiento de trayectorias

Las articulaciones detectadas por el sensor Kinect v2 se miden con respecto al origen  $O_K$  del marco de referencia del sensor Kinect. La posición de la mano derecha, correspondiente a  $O_K$  en el marco de referencia del sensor, es identificada dentro del marco de referencia del robot, convirtiéndose en  $O_r$ . Aplicando una rotación  $R_x = 90^\circ$ , se obtiene el marco de referencia resultante para el robot, tal como se muestra en la Figura 3.14.

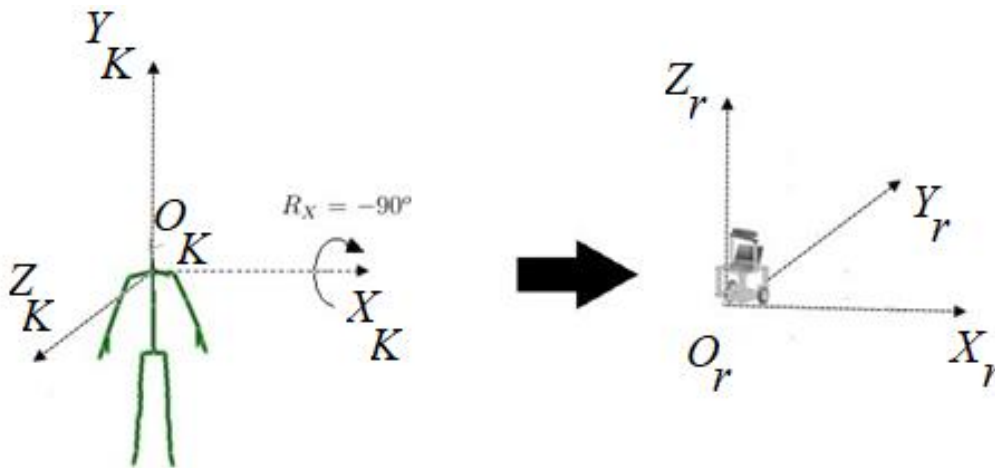
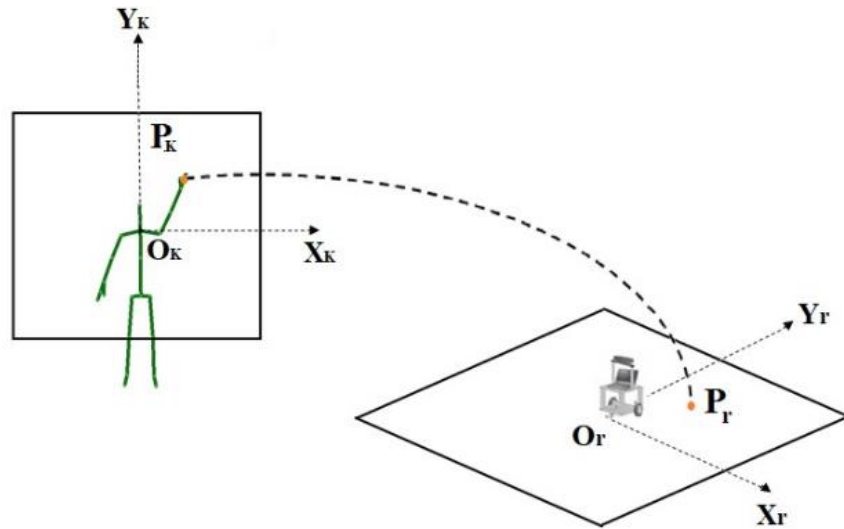


Figura 3.14. Tránsito del espacio de trabajo del sensor Kinect al espacio de trabajo del robot móvil

El punto  $P_K(x_K, y_K)$  se obtiene de la posición medida de la mano derecha. Se mapea a un punto  $P_r(x_r, y_r)$  en el espacio de trabajo del robot, como se muestra en la Figura 3.15. Se emplea la distancia del brazo y antebrazo hasta la mano, para obtener  $P_r(x_r, y_r) = P_K(x_K, y_K)$  y así poder escalar la posición deseada del robot, mediante un factor  $z$  que representa la distancia de profundidad del usuario.

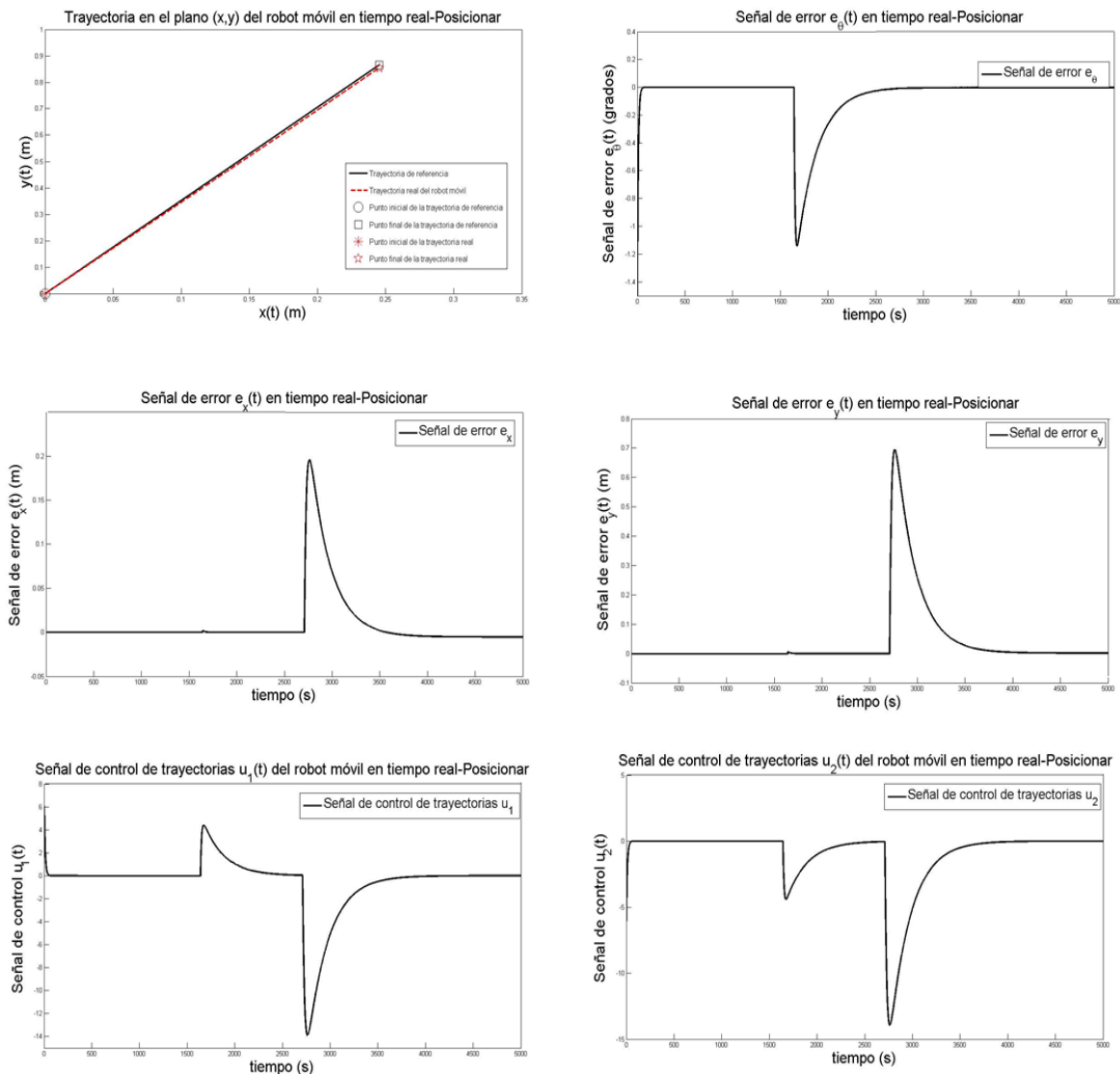


**Figura 3.15.** Mapeo de un punto del espacio de trabajo del sensor Kinect al espacio de trabajo del robot móvil

La interfaz emplea tres funciones, activadas por los gestos correspondientes, para definir una posición deseada y mover el robot a esa posición:

- *Establecer posición.* Gracias a esta función se adquiere la posición de la mano derecha, con la mano derecha abajo, cerrada, y la mano izquierda arriba, cerrada. La posición de la mano depende de las longitudes del antebrazo y brazo de cada persona. El ángulo es determinado como el ángulo de apertura medido desde el eje  $X_K$ , y definido en el rango de  $0^\circ \leq \theta \leq 180^\circ$ . Para escalar los valores, el usuario avanza (para incrementar  $z$ ) o retrocede (para decrementar  $z$ ).
- *Guarda posición.* Cuando el usuario abre la mano izquierda, se guarda la posición deseada.
- *Mover a Ludibot.* Cuando el usuario abre la mano derecha, el robot se mueve a la posición deseada.

Para moverse a la posición deseada, el robot primero se desplaza  $\theta$  hasta que el error de posicionamiento angular es menor a un valor cercano a cero. Enseguida, efectúa el movimiento en línea recta, hasta que el error de posicionamiento en  $x_r, y_r$  es menor a un valor cercano a cero. En la Figura 3.16, que muestra los resultados del control para posicionar Ludibot, se puede apreciar cómo primero se realiza la orientación y, luego, la traslación.



**Figura 3.16. Señales de respuesta de Ludibot v1 para la función *Posicioname***

La calibración de los parámetros fisiológicos de cada persona se lleva a cabo mediante las herramientas del kit de desarrollo de *software* (SDK) de Kinect v2 [66], que permite obtener los puntos de las articulaciones del cuerpo y sus posiciones en el espacio 3D.

El reconocimiento del cuerpo se realiza mediante las funciones que ofrece el SDK (*BodyTracking*, *BodyFrameReady* y *KinectBodyFrameReady*), que permiten obtener los parámetros específicos de las partes del cuerpo: longitudes, ángulos, estados de las manos, entre otros.

### 3.4 Aplicación ¡Te sigo!

Esta implementación con Ludibot v1 está basada en el control de seguimiento mediante el cálculo de la distancia y la posición angular del usuario. El robot usa la información del sensor Kinect v2 para realimentar dichas variables. En el control se utilizan dos controladores PI (proporcional integral), uno para cada grado de libertad. El sensor Kinect utiliza el tiempo de vuelo de una nube de puntos que se proyectan sobre el usuario para determinar la profundidad detectar las partes del cuerpo.

En la Figura 3.17 se representa el diagrama general del control de seguimiento de usuario, compuesto por el control de distancia y de orientación, donde los datos del sensor son empleados para cerrar el lazo de control.

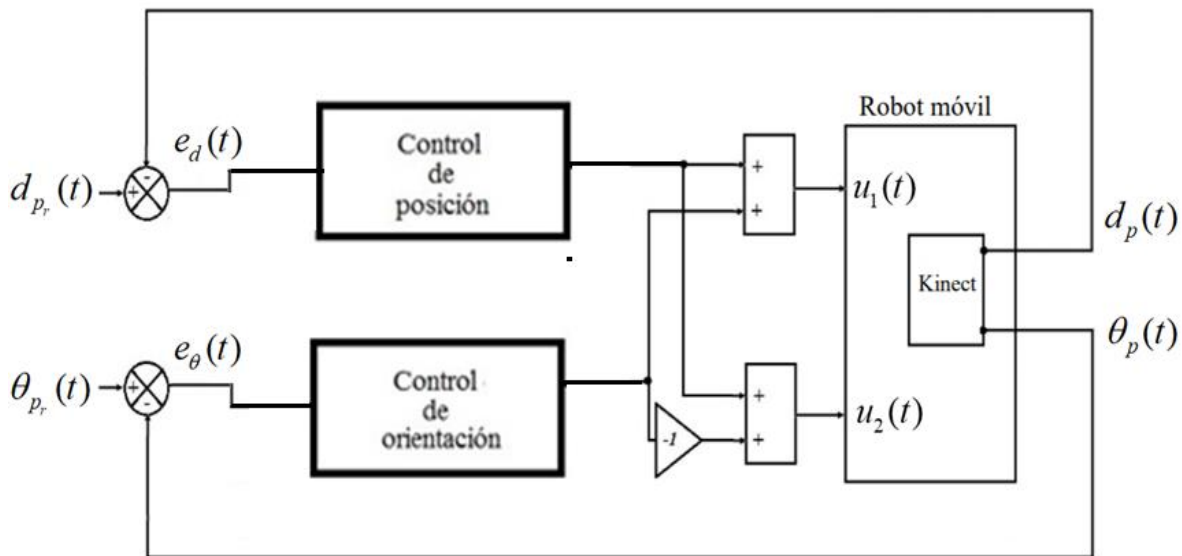


Figura 3.17. Diagrama de control de seguimiento de usuario

A partir de las posiciones de las partes del cuerpo detectadas por el sensor Kinect, se obtiene la distancia al usuario, tomando como referencia el punto superior de la espina dorsal  $P_r(x, y, z)$  (véase Figura 3.18).

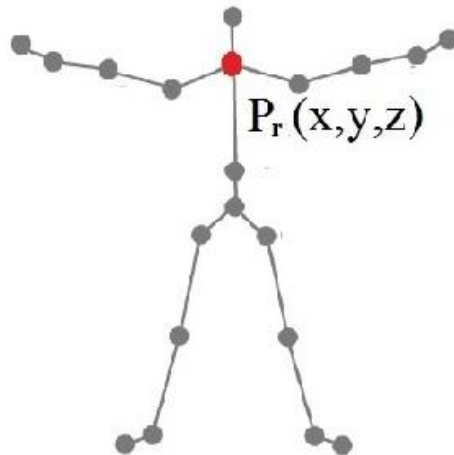


Figura 3.18. Diagrama del cuerpo dado por el sensor Kinect y el punto de referencia  $P_r$

La orientación del usuario con respecto al robot en el plano  $X_K, Z_K$ , se calcula como el ángulo  $\theta$  entre el eje  $Z_K$  y la recta que va del origen  $O_K$  al punto  $P_r(x, y, z)$ , detectado por Kinect como se muestra en la Figura 3.19.

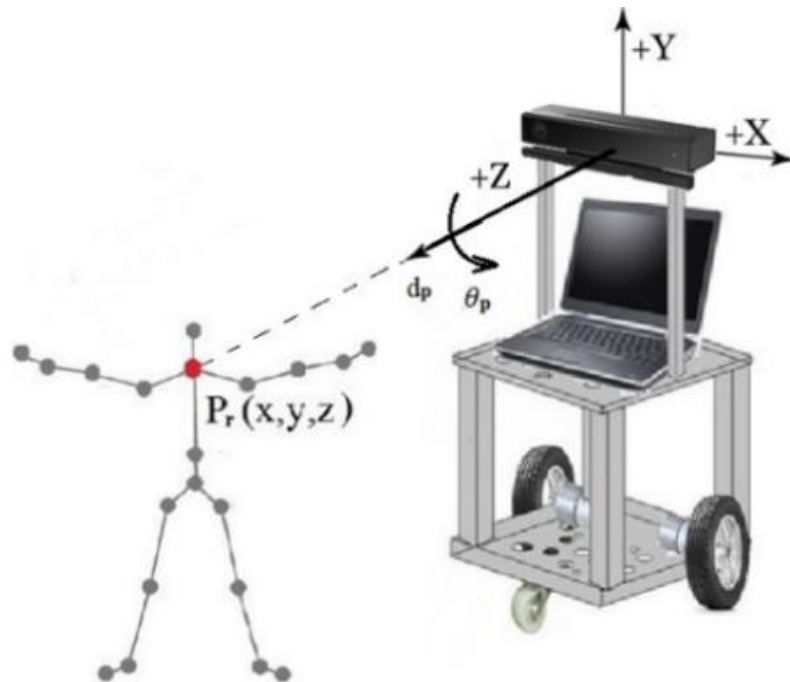
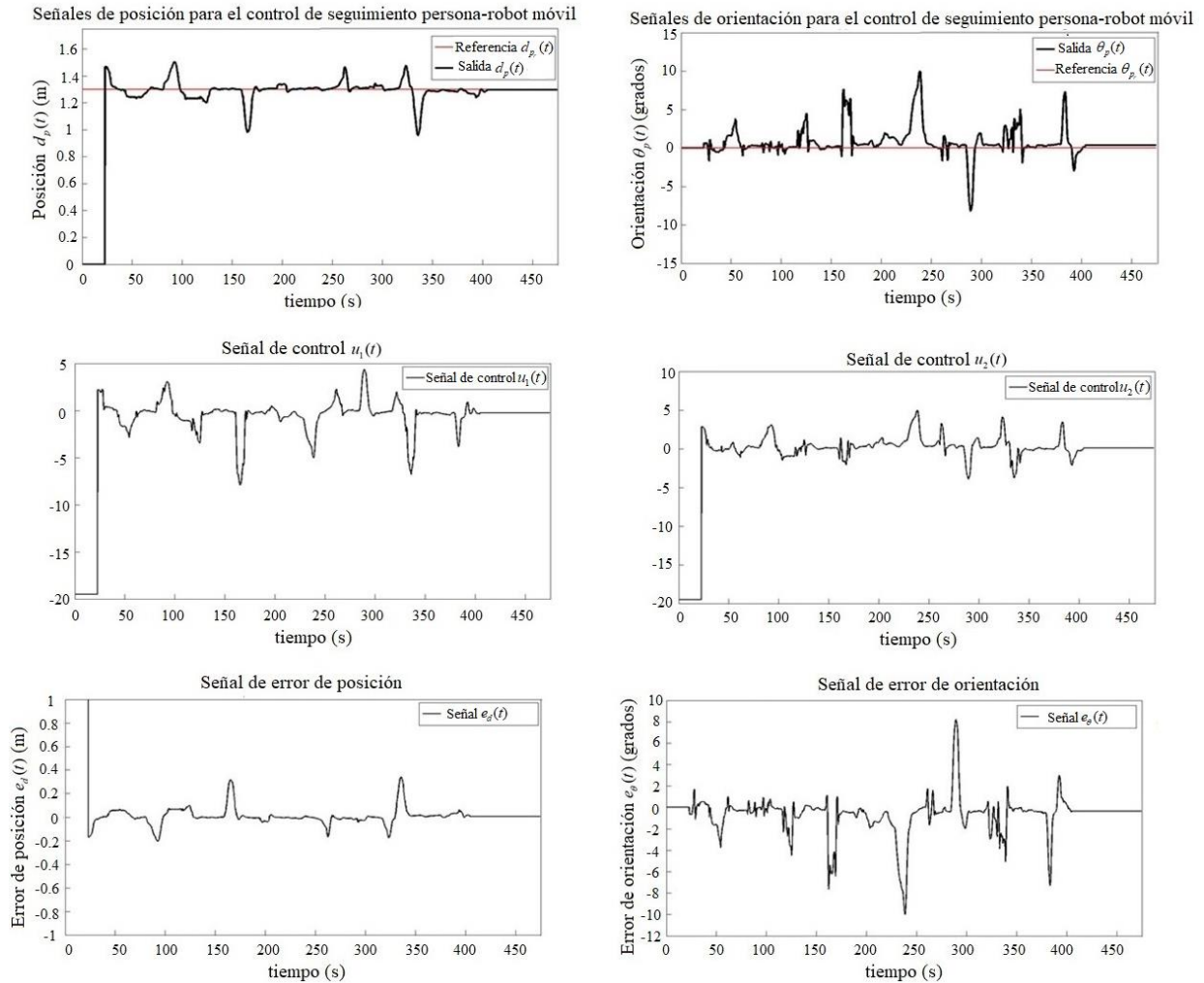


Figura 3.19. Distancia  $d_p$  y ángulo  $\theta_p$  para seguir al usuario

La distancia  $d_p$  (medida en metros) y el ángulo  $\theta_p$  entre el punto detectado  $P_r$  y el origen

$$O_K \text{ son } d_p = \sqrt{Z_k^2 + X_k^2} \text{ y } \theta_p = \sin^{-1}\left(\frac{X_k}{\sqrt{Z_k^2 + X_k^2}}\right).$$

En la Figura 3.20 se muestran los resultados experimentales para seguir al usuario. Las referencias utilizadas son  $d_p = 1.3\text{ m}$  y  $\theta_r = 0^\circ$ .



**Figura 3.20. Señales de respuesta de la función ¡Te sigo!**  
con las ganancias  $k_{P_d} = 15$ ,  $k_{I_d} = 0.5$  y  $k_{P_\theta} = 0.5$   $k_{I_\theta} = 0.01$  para Ludibot v1

Es importante señalar que, cuando se sigue al usuario, en las gráficas aparecen discontinuidades, debido a que la interfaz diseñada realiza el envío y la recepción de los datos  $d_p$  y  $\theta_p$  solamente cuando se activa la función ¡Te sigo!



### 3.5 Juego *De la tête aux pieds*

El desarrollo del juego *De la tête aux pieds* requirió el diseño de una arquitectura específica, que incluye un robot y una interfaz interactiva, presentados en las secciones anteriores de este trabajo. El protocolo de comunicación permite la interacción humano-máquina y establece la conexión para realizar el envío y la recepción de datos, mientras que la estrategia de control permite al robot realizar los movimientos necesarios.

La primera aplicación lúdica desarrollada con Ludibot v1 está destinada al aprendizaje dirigido o semiautónomo del idioma francés, teniendo por objetivo lingüístico la sensibilización, la conceptualización y/o la sistematización del vocabulario básico relativo a las partes del cuerpo humano, así como los elementos gramaticales y fonéticos correspondientes. Los usuarios pueden ser tanto aprendientes que se acercan por primera vez a ese vocabulario, como aprendientes que buscan organizar o repasar un conocimiento léxico previo.

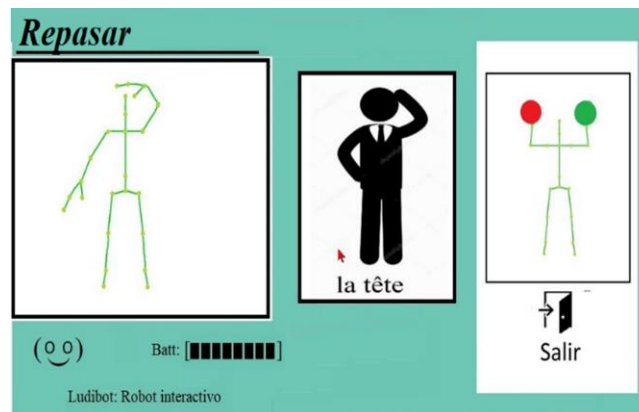
A partir del material lúdico constituido por el robot y por el cuerpo del usuario, se sigue una regla específica que se expondrá a continuación, correspondiente a la estructura. El contexto de uso es variable y puede ser tanto una situación formal de aprendizaje en el aula como una situación semiformal en mediateca o en centro de autoacceso. El objetivo es propiciar que el usuario se acerque con una actitud lúdica al idioma meta.

En la primera versión de Ludibot, la aplicación *Juguemos* permite acceder al juego *De la tête aux pieds*. El objetivo del jugador es acumular cinco aciertos sin cometer más de dos errores. Por cada acierto, el robot avanzará una distancia determinada.

Durante el desarrollo del juego, el robot enuncia aleatoriamente, una por una y en francés, diferentes partes del cuerpo por tocar, precedidas por el artículo definido correspondiente: *la tête* (la cabeza), por ejemplo. Ahora bien, el jugador no debe tocar la parte del cuerpo recién anunciada sino la anterior. En otras palabras, al iniciar el juego, Ludibot enuncia la parte A y el jugador debe permanecer inmóvil. Cuando el robot enuncia la parte B, el jugador toca la parte A; cuando el robot enuncia la parte C, el jugador toca la parte B, y así sucesivamente. Por ende, no se trata de una actividad durante la cual se siguen al pie de la letra las instrucciones dadas por el robot, sino de una tarea en la que el desfase deliberado entre instrucción dada e instrucción realizada exige concentración y memorización a corto plazo.

Cada vez que el usuario obtiene un acierto, el robot emite una señal sonora específica, muestra en pantalla un ícono de aplauso, y avanza una distancia preestablecida. Por el contrario, si el usuario se equivoca o tarda más de cinco segundos en dar su respuesta, el robot emite una señal sonora de error y no avanza. Cuando el jugador obtiene cinco aciertos, gana. El robot ludoeducativo emite una señal sonora de fanfarria para felicitarlo. Para reiniciar el juego, el usuario debe emplear el estado *Salir*. Cuando el usuario llega a acumular tres errores, se reproduce una señal sonora distinta, que indica que el jugador ha perdido el juego. En tal caso, es necesario reiniciar el juego con el estado *Salir*. A futuro, las distintas señales sonoras mencionadas serán sustituidas por mensajes vocales.

El propósito es permitir al usuario iniciar con un proceso de aprendizaje intuitivo, escuchando las palabras y tocando la parte del cuerpo correspondiente. De esta manera, pretende motivar al usuario a descubrir o repasar elementos léxicos (nombre de la parte del cuerpo), gramaticales (género de la palabra) y fonéticos (pronunciación), identificando a qué partes de su propio cuerpo alude Ludibot como se muestra en la Figura 3.21.



**Figura 3.21. Interfaz con el estado *Repasar* de Ludibot**

La pantalla incluye tres grandes secciones: a la izquierda aparece el título de la modalidad de juego, seguida por un recuadro con la representación del cuerpo del usuario. En el caso de la Figura 3.21, vemos al jugador tocando su propia cabeza, en respuesta al audio previamente emitido (*la tête* [la tèt]). Más abajo figuran el estado de detección de Kinect y el estado de la batería. Al centro, se muestra el modelo de figura humana que permite identificar la posición, el nombre y la ortografía de la parte del cuerpo mencionada. A la derecha, se ofrece al usuario la opción *Salir* y se le recuerda el gesto correspondiente.

Esta primera dinámica brinda un acercamiento sonoro, visual y escrito al corpus de vocabulario. En efecto, la interfaz enuncia el nombre de cada parte del cuerpo y, simultáneamente, muestra en pantalla la imagen correspondiente, acompañada del nombre en cuestión, siempre precedido del artículo definido (*le, la o l'*) que, en los dos primeros casos, permite determinar el género de la palabra.

La Figura 3.22 ilustra las partes del cuerpo humano incluidas hasta ahora en el repertorio de Ludibot. A futuro, se tiene contemplado ampliar la lista de vocabulario disponible.



**Figura 3.22. Partes del cuerpo humano incluidas en *De la tête aux pieds***

Antes de iniciar el juego, el usuario puede elegir dos modalidades adicionales de utilización, de menor complejidad que el juego en sí: *Repasar* y *Entrenar*. *Juguemos* incluye asimismo la posibilidad de pasar al estado *Alto* (para interrumpir anticipadamente el juego y dar por terminada la partida en curso) y la de recurrir a la transición *Salir* (para regresar al menú).

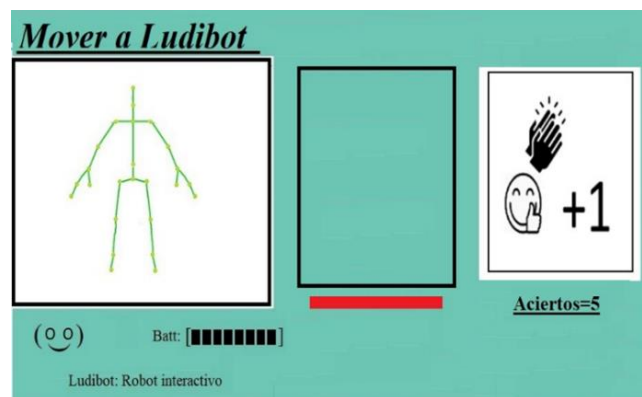
*Repasar* permite revisar (o incluso descubrir por vez primera) el vocabulario en francés de las partes del cuerpo utilizado en *De la tête aux pieds*. En esta modalidad, cuando el usuario toca una parte de su cuerpo, Ludibot permanece fijo, detecta la parte del cuerpo correspondiente y formula el nombre de la misma en voz alta. Si bien el idioma predeterminado es el francés, el dispositivo es fácilmente adaptable a otros idiomas.

Por su parte, *Entrenar* ayuda a practicar la interacción con el robot sin el reto adicional del desfase entre instrucciones que introduce *De la tête aux pieds*. En esta modalidad, el usuario solicita al robot que le indique una parte del cuerpo por tocar, misma que será elegida aleatoriamente dentro del repertorio previsto. Si el usuario sigue adecuadamente la instrucción, el

robot emite una señal sonora específica, contabiliza el acierto y avanza. Si el usuario se equivoca, el robot emite una señal de error. Si el usuario acumula tres errores antes de alcanzar cinco aciertos, es invitado a comenzar de nuevo.

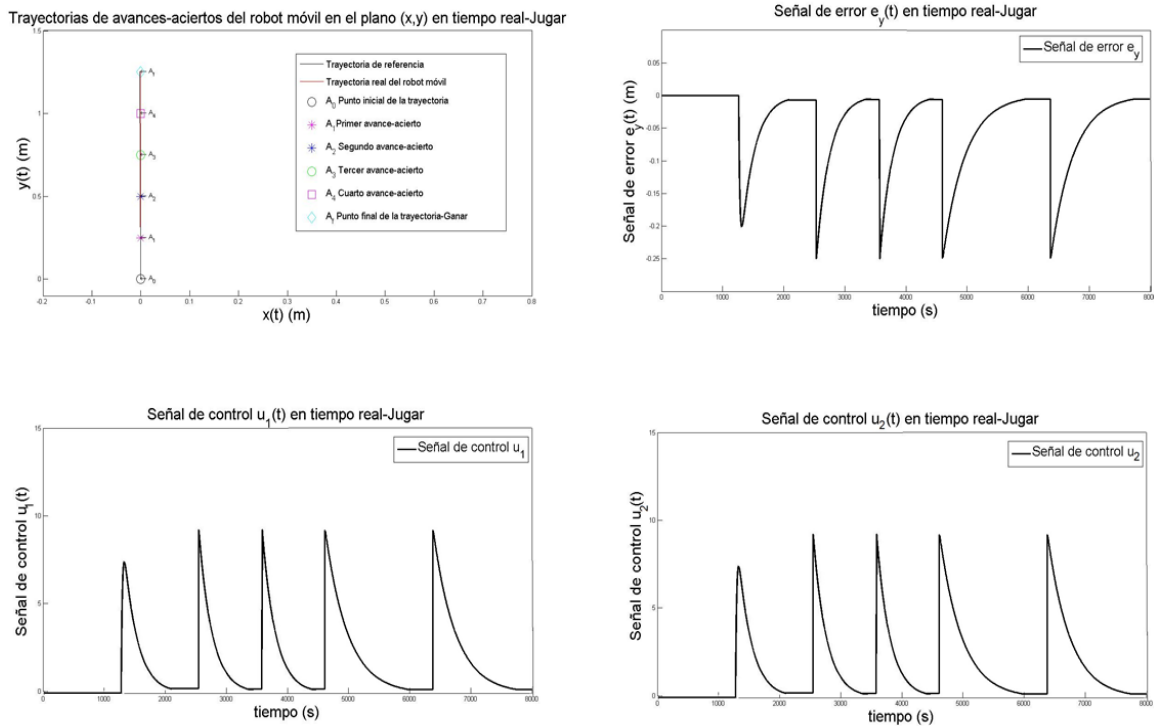
Lo que distingue el estado *Entrenar* del estado *Jugar* De la tête aux pieds es, como se mencionó más arriba, la introducción de una dificultad adicional durante el juego, provocada por el desfase entre la parte enunciada y la parte por tocar.

Ludibot v1 lleva explícitamente a cabo el conteo de aciertos (véase Figura 3.23), según dos modalidades, gráfica y textual. La barra central, de color rojo, va aumentando su longitud conforme el robot avanza en el espacio respecto del punto de partida; además, en la esquina inferior derecha de la pantalla, se indica el número de aciertos acumulados. Con cada acierto, el robot móvil avanza hacia el jugador.



**Figura 3.23. Interfaz contando el quinto y último acierto en el estado *Mover a Ludibot***

La Figura 3.24 muestra el comportamiento del robot cuando realiza cinco avances, pues el usuario acertó al identificar todas y cada una de las partes del cuerpo siguiendo las reglas del juego. Se presentan en dicha figura las trayectorias de referencia en partes y la trayectoria del robot de cada segmento recorrido, así mismo como las marcas de los avances correspondientes a cada acierto, y el punto final de la trayectoria que corresponde al punto cuando el usuario ha ganado el juego.



**Figura 3.24. Señales de respuesta de Ludibot v1 del juego *De la tête aux pieds***

El juego aquí presentado es fácilmente adaptable a cualquier idioma, pues basta con sustituir los archivos de los audios y las imágenes con los archivos correspondientes en el idioma deseado. También puede ajustarse el nivel de lengua, enriqueciendo el repertorio léxico o remplazándolo por una lista de nivel más avanzado.

Es posible asimismo programar adaptaciones a otros campos léxicos, siempre y cuando a cada unidad léxica corresponda una posición específica en el espacio. Es factible, por ejemplo, crear juegos basados en la identificación de figuras proyectadas en una pantalla o en la adecuada ubicación de sitios en un mapa. Los ajustes introducidos en la interfaz permitirán llevar a cabo de manera más eficiente el diseño, el desarrollo y la implementación de nuevas aplicaciones ludoeducativas.

## Conclusiones

El proceso de investigación y desarrollo que respalda esta tesis ha permitido estudiar la pertinencia de un acercamiento pluridisciplinar al aprendizaje de idiomas en general y del francés en particular desde la robótica, la DLC y culturas y las ciencias del juego. Es posible considerar que el objetivo general planteado en un inicio fue alcanzado, pues se ha puesto a disposición de aprendientes de idiomas, docentes e investigadores herramientas teóricas y prácticas concebidas a partir de la robótica, que complementen y faciliten el aprendizaje de lenguas mediante el uso de la tecnología y, en específico, de Ludibot.

El marco teórico y conceptual presentado en el primer capítulo, que apuntaló el desarrollo de Ludibot, integra tanto una visión contemporánea de las teorías del aprendizaje de idiomas como una problematización de la noción de juego. A partir de dicho marco, fue posible proponer un concepto específico de robótica ludoeducativa, con base en el cognoscitivismo, el constructivismo y el interaccionismo, así como en una ludicización que toma en cuenta las cuatro regiones metafóricas de la noción de juego.

En una época marcada por la complejidad, con retos constantemente renovados, se decidió además no adoptar posturas unidisciplinarias en el área de la DLC, ni asumir un bilingüismo selectivo. Se consideró más productivo adoptar un enfoque plurilingüe y pluricultural, para no dar prioridad a un idioma sobre otros, pues se trata de aprovechar el potencial de un enfoque que reconoce la pluralidad.

Se dio especial énfasis a la participación del aprendiente en la construcción del conocimiento, desde un enfoque integral favorable a una interacción humano-máquina atractiva, significativa y eficaz. Dicha interacción ocurre en el marco de un proceso de enseñanza/aprendizaje de idiomas en situaciones formales y semiformales, desde una óptica que no parece haber sido estudiada aún en otros trabajos de investigación dentro de ninguna de las áreas disciplinares consideradas.

En cuanto a los objetivos de desarrollo, se logró el diseño y la implementación de las dos versiones de Ludibot con base en la arquitectura de un robot ludoeducativo, considerando dicha

arquitectura como un metasistema dentro del cual es preciso definir para cada objeto o subsistema sus características (material, programas, programación) así como la relación con los demás objetos y el comportamiento de cada uno, en lugar de evaluar los componentes del sistema y sus propiedades por separado. Este modelo funciona para examinar las diferentes particularidades de un robot ludoeducativo.

Ludibot v1 está basado en una arquitectura libre que incluye una computadora, los materiales y los programas, mientras que Ludibot v2 está integrado por un robot Loomo, un robot Cozmo, un Arduino Mega y una plataforma para Cozmo como principales componentes. Ludibot v1 y –en un futuro próximo– Ludibot v2, podrán ofrecer a los alumnos del bachillerato universitario y a otros usuarios herramientas novedosas para la construcción del conocimiento, mediante aplicaciones lúdicas que propicien un cambio de actitud en situaciones formales y semiformales de aprendizaje, incrementen la motivación y contribuyan a modificar las representaciones y las prácticas asociadas al aprendizaje de una lengua. Cabe subrayar que Ludibot aprovecha tanto la interacción humano-máquina como la comunicación verbal y no verbal para promover no sólo el aprendizaje de un idioma sino también las competencias generales y comunicativas desde una óptica plurilingüe y pluricultural, así como el desarrollo de la literacidad mediática multimodal y las llamadas competencias del siglo XXI.

En lo que atañe a la interfaz, se llevó a cabo el diseño y la implementación de la arquitectura general de cada versión de Ludibot. Ambas interfaces son capaces de llevar a cabo la interacción humano-máquina mediante la adquisición y el reconocimiento de posiciones del cuerpo humano, gracias a los cuales es posible asignar tareas y determinar transiciones para manejar cada robot mediante gestos y ademanes detectados por el sensor Kinect v2.

Para Ludibot v2 se ha desarrollado una versión de la interfaz humano-robot basada en la comunicación verbal y no verbal que aprovecha el uso de sonidos, gestos y ademanes. Esta segunda versión tiene la característica de ser programada en lenguaje HTML de una manera más sencilla. Además, ofrece un apéndice donde se apoya al usuario para un adecuado uso de audios, datos, imágenes, entre otros. Allí se presentan elementos para crear una interfaz específica, con base en una máquina de estados, los tipos requeridos de variables, las posiciones deseadas de los robots, las variables predeterminadas, el uso previsto de los gestos y las acciones, entre otros. Cabe mencionarse que las pruebas de Ludibot han sido llevadas a cabo en el laboratorio

con resultados satisfactorios, lo cual permite predecir un buen aprovechamiento en situaciones formales e informales de aprendizaje, una vez que las condiciones de la contingencia sanitaria lo permitan.

Las aplicaciones lúdicas desarrolladas con el objetivo de promover el aprendizaje consideran variables como el entorno de aprendizaje, la participación en grupo, la interacción frecuente, la retroalimentación y las conexiones con el contexto del mundo real. De tal manera, es posible lograr apostar por transformar el aula tradicional en un espacio dinámico y atractivo, en donde tienen lugar actividades significativas, de carácter colaborativo y creativo. En ellas, el significado debe ser construido de tal manera que el individuo interactúe tanto con el robot como con sus pares, recurriendo a la literacidad mediática multimodal y a un amplio rango de competencias.

Hasta el momento, se han implementado tres aplicaciones específicas para Ludibot v1: la primera es la aplicación *¡Te sigo!*, está basada en el control de seguimiento robot-humano. La segunda aplicación es *Posicióname*, que permite mapear posiciones con la mano y el brazo derechos, del espacio de trabajo del usuario al espacio de trabajo del robot móvil. La aplicación *Juguemos* permite acceder al juego *De la tête aux pieds* cuyo objetivo es promover el aprendizaje del vocabulario relativo a las partes del cuerpo. El juego se basa en acumular cinco aciertos sin cometer más de dos errores.

Los videos que ilustran los experimentos de las aplicaciones con Ludibot v1 están disponibles en línea mediante el acceso a la cuenta de correo [robotinteractivoludibot@gmail.com](mailto:robotinteractivoludibot@gmail.com) con la contraseña: *claveludibot*; allí, el usuario encontrará un mensaje con las ligas correspondientes. También es posible acceder desde la cuenta indicada al sitio de acceso reservado: [sites.google.com/site/ludibotrobotinteractivo](https://sites.google.com/site/ludibotrobotinteractivo). Un nuevo sitio está en desarrollo en [ludibot.mx](http://ludibot.mx).

A mediano y largo plazo, Ludibot ofrece aún numerosas pistas de exploración, entre las cuales están:

- Mejorar la arquitectura, la apariencia y la integración de la plataforma de Ludibot v1, así como el desarrollo de otras aplicaciones ludoeducativas para el aprendizaje de lenguas.



- Realizar otras aplicaciones con Ludibot v2, basado en Loomo y Cozmo. Se ha considerado la programación de animaciones con la cabeza del primero, acompañadas de animaciones independientes por parte del segundo en respuesta a ciertas acciones del usuario.
- Realizar los experimentos y los pilotajes de las aplicaciones lúdicas con aprendientes de francés de nivel medio superior y nivel superior en el aula de idiomas, la mediateca y el centro de autoacceso, con ambas versiones, para evaluar mediante cuestionarios y entrevistas el nivel de desempeño, satisfacción y motivación.
- Desarrollar la programación con Kinect v2 para el desarrollo de aplicaciones mediante la voz y el control vocal.
- Implementar y desarrollar otros recursos ludoeducativos destinados a favorecer el aprendizaje de idiomas, incluyendo los diferentes tipos de competencias generales y lingüísticas.
- Extender las aplicaciones de robótica ludoeducativa a otras áreas de aprendizaje.

En suma, la conclusión de este proceso de investigación, lejos de agotar el tema, conduce a abrir nuevas perspectivas.

## Bibliografía

- [1] AILON, A., LOZANO, R., GIL, M. I., “Iterative Regulation of an Electrically Driven Flexible-Joint Robot with Model Uncertainty”, *IEEE Trans. Robot. Automat.*, vol. 16, pp. 863-870, 2000.
- [2] ALEXANDER, J. C., MADDOCKS, J. H., “On the Kinematics of Wheeled Mobile Robots”, *International Journal Robotics Res.*, vol. 8, pp. 15-27, 1989.
- [3] ALIANE, N., BEMPOSTA, S., “A Checkers Playing Robot: A Didactic Project”, *IEEE Latin America Transactions*, vol. 9, núm. 5, pp. 821-826, 2011.
- [4] ALONSO, M., F., *Sistema de Interacción Humano-Robot basado en diálogos Multimodales y Adaptables*. Tesis doctoral, Universidad Carlos III de Madrid, 2014. [En línea] <https://core.ac.uk/download/pdf/29406358.pdf>
- [5] ALVAREZ, J., CHAUMETTE, P., “Présentation d’un modèle dédié à l’évaluation d’activités ludopédagogiques et retours d’expériences”, *Jeux et langues dans l’enseignement supérieur*, vol. 36, núm. 2, 2017.
- [6] ANANIADOU, K., CLARO, M., “21st Century Skills and Competences for New Millennium Learners in OECD Countries”, *OECD Education Working Papers*, núm. 41, 2009.
- [7] ANKI INC., “Meet Cozmo”, *Cozmo*. [En línea] <https://www.anki.com/en-us/cozmo>
- [8] ARANDA-BRICAIRE, E., SALGADO-JIMÉNEZ, T., VELASCO-VILLA, M., “Control no lineal discontinuo de un robot móvil”, *Computación y Sistemas, número especial*, pp. 42-49, 2002.
- [9] ARACIL J., GORDILLO, F., *Dinámica de sistemas*, Alianza Editorial, Madrid, 1997.
- [10] ASIMOV, I., *I, Robot*, Gnome Press, New York, 1950.
- [11] AUSUBEL, D., *Psicología educativa. Un punto de vista cognoscitivo*, Trillas, México, 1976.
- [12] BALAKRISHNA, R., GHOSAL, A., “Modeling of Slip for Wheeled Mobile Robots”, *IEEE Trans. Robot. Automat.*, vol. 11, pp. 126-132, 1995.
- [13] BALKCOM, D. J., MASON, M. T., “Time Optimal Trajectories for Bounded Velocity Differential Robots”, *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2499-2504, San Francisco, April 2000.
- [14] BAÑÓ AZCÓN, A., “Análisis y diseño del control de posición de un robot móvil con tracción diferencial”. Tesis profesional, L’Escola Técnica Superior d’Enginyeria (ETSE) de la Universitat de València, 2003.

- [15] BARON, G., CZEKALSKI, P., GOLENIA, M., TOKARZ, K., “Gesture and Voice Driven Tribot Mobile Robot Using Kinect Sensor”, *International Symposium on Electrodynamical and Mechatronic Systems (SELM)*, pp. 33-34, 2013.
- [16] BAUZIL, G., BRIOT, M., RIBES, P., “A Navigation Sub-System Using Ultrasonic Sensors for the Mobile Robot Hilare”, *Proc. of the First International Conference on Robot Vision and Sensory Controls*, pp. 47-58, Stratford-upon-Avon, April 1981.
- [17] BENABDALLAH, I., BOUTERAA, Y., BOUCETTA, R., REKIK, C., “Kinect-based Computed Torque Control for Motion Robotic Arm”, *International Conference on Modelling, Identification and Control (ICMIC)*, Sousse (Túnez), pp. 1-6, 2015.
- [18] BOLDEA, I., NASAR, S. A., *Electric Drives*, 2nd ed., CRC Press, Boca Raton, 2005.
- [19] BOLTON, W., *Mecatrónica, sistemas de control electrónico en la ingeniería mecánica y eléctrica*, Alfaomega, Quinta Edición, México, 2013.
- [20] BONENFANT, M., GENVO, S., “Une approche située et critique du concept de gamification”, *Sciences du jeu*, núm. 2, 2014.
- [21] BORENSTEIN, J., “The Nursing Robot System”. Tesis doctoral, Technion, Haifa, Israel, 1987.
- [22] BORENSTEIN, J., EVERETT, H. R., FENG, L., *Where I am? Sensors and Methods for Mobile Robot Positioning*, University of Michigan, Ann Arbor, 1996.
- [23] BORENSTEIN, J., FENG, L., “Measurement and Correction of Systematic Odometry Errors in Mobile Robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 869-880, 1996.
- [24] BOUKO, C., ALVAREZ, J., “Serious Gaming, Serious Modding, a Serious Diverting: Are you Serious?”, *Mapping the Digital: Cultures and Territories of Play*, Joyce, L., Quinn, B. (eds). Inter-Disciplinary Press, Oxford, 2016.
- [25] BRÄUNL, T., *Embedded Robotics: Mobile Robot Design and Applications With Embedded Systems*, Springer-Verlag, New York, 2006.
- [26] BROUGÈRE, G., *Jouer/apprendre*, Economica, París, 2005.
- [27] BURGARD, W., CREMERS, A. B., FOX, D., HANEL, D., LAKEMEYER, G., SCHULZ, D., STEINER, W., THRUN, S., “The interactive museum tour-guide robot”, *AAAI-98 Proceedings*, 1998.
- [28] CADENAT, V., SWAIN, R., SOUÉRES, P., DEVY, M., “A Controller to Perform a Visually Guided Tracking Task in a Cluttered Environment”, *IEEE/RSJ international Conference on Intelligent Robots and Systems*, pp. 775-780, 1999.

- [29] CAMPION, G., BASTIN G., D'ANDRÉA-NOVEL, B., "Structural Properties And Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots", *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 47-62, 1996.
- [30] CAMPOS, M., A., GASPAS, S., *La construcción del constructivismo en investigación cognoscitiva*, Siglo XXI, México, 1996.
- [31] CANSIGNO G., Y., "El francés, lengua de comunicación internacional", *Reencuentro*, núm. 47, pp. 23-28, diciembre de 2006.
- [32] CANUDAS DE WIT, C., KHENNOUF, H., SAMSON, C., SORDALEN, O. J., "Nonlinear Control Design for Mobile Robots", *World Scientific Series in Robotics and Intelligent Systems. Recent Trends in Mobile Robots*, pp. 121-156, 1994.
- [33] CANUDAS DE WIT, C., SILICIANO, B., BASTIN, G., *Theory of Robot Control*, Springer, New York, 1996.
- [34] ČAPEK, K., *R.U.R. (Rossum's Universal Robots)*, Penguin Classics, London, 2004.
- [35] CARAVOLAS, J. A., *Le point sur l'histoire de l'enseignement des langues*, Centre éducatif et culturel, Anjou, 1995.
- [36] CARROLL, J., B., "Twenty-Five Years of Research on Foreign Language Aptitude", Ed. K. C. DILLER, *Individual Differences and Universals in Language Learning Aptitude*, Newbury House Publishers, Rowley, Massachusetts, pp. 83-118, 1981.
- [37] CARVALHO, M., BELLOTTI, F., BERTA, R., DE GLORIA, A., GAZZARATA, G., HU, J., KICKMEIER-RUST, M., "A Case Study on Service-Oriented Architecture for Serious Games", *Entertainment Computing*, vol. 6, pp. 1-10, 2015.
- [38] CASTELLANO, L. A., MÁRSICO, C. T., *Diccionario etimológico de términos usuales en la praxis docente*, Altamira, Buenos Aires, 1995.
- [39] CHAKRABORTY, N., GHOSAL, A., "Dynamic Modeling and Simulation of a Wheeled Mobile Robot for Traversing Uneven Terrain Without Slip", Department of Mechanical Engineering, Indian Institute of Science, Bangalore, 2003.
- [40] CICIRELLI, G., ATTOLICO, C., GUARAGNELLA, C., D'ORAZIO, T., "A Kinect-Based Gesture Recognition Approach for Natural Human Robot Interface", *International Journal of Advanced Robotic Systems*, vol. 12, núm. 3, pp. 1-11, 2014.
- [41] CONSEIL DE L'EUROPE, *Cadre européen commun de référence pour les langues*, Estrasburgo, Conseil de l'Europe/Didier, 2001.
- [42] CUEVA, W., TORRES, H., KERN, J., "7 DOF Industrial Robot Controlled by Hand Gestures Using Kinect v2", 18-20 de octubre, Cartagena (Colombia), pp. 1-6, 2017.
- [43] DA SILVA, G. C., M. H., SIGNORET D., A., *Temas sobre la adquisición de una segunda lengua*, Trillas, México, 2005.

- [44] DARDAN, M., MARAJ, A., HAJZERAJ, A., “Application Interface for Gesture Recognition with Kinect Sensor”, *International Conference on Knowledge Engineering and Application*, pp. 98-102, 2016.
- [45] DEREK J. H., HRUSCHKA, P., PIRBHAI, I. *Process for System Architecture and Requirements Engineering*, Dorset House Pub., New York, 2000.
- [46] DETERDING, S., DIXON, D., KHALED, R., NACKE, L., “Du game design au gamefulness: définir la gamification”, *Sciences du jeu*, núm. 2, 2014.
- [47] DIVELBISS, A. W., WEN, J. T., “Trajectory Tracking Control of a Car-Trailer System”, *IEEE Trans. Control Syst. Technol.*, vol. 5, núm. 3, pp. 269-278, 1997.
- [48] DONADO CAMPOS, J. DE M., DORMIDO CANTO, S., MORILLA GARCÍA, F., *Fundamentos de la dinámica de sistemas*, Madrid, 2005. [En línea] [http://www.proyectosame.com/ds\\_documentos/manual\\_dinamica\\_sistemas.pdf](http://www.proyectosame.com/ds_documentos/manual_dinamica_sistemas.pdf)
- [49] DORF, R. C., BISHOP, R. H., *Modern Control Systems*, 9th ed. Upper Saddle River, NJ: Prentice Hall, 2001.
- [50] ELFES, A., “A Sonar-Based Mapping and Navigation System”, *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1151-1156, 1986.
- [51] ELLIS, R., *Understanding Second Language Acquisition*, Oxford University Press, Oxford, 1985.
- [52] ESCOLANO, C., MINGUEZ, J., “Sistema de teleoperación multi-robot basado en interfaz cerebro-computador”, *Revista Iberoamericana de Automática e Informática Industrial*, vol. 8, núm. 2, pp. 16-23, 2011.
- [53] EUROPEAN CENTRE FOR THE DEVELOPMENT OF VOCATIONAL TRAINING, *Terminology of European Education and Training Policy. A Selection of 100 Key Terms*, Office for Official Publications of the European Communities, Luxembourg, 2008.
- [54] EVANS, W. R., *Control System Dynamics*, McGraw-Hill Book Company, New York, 1954.
- [55] FAUST, J., SIMON, C., SMART, D., “A Video Game-Based Mobile Robot Simulation Environment”, *International Conference on Intelligent Robots and Systems*, pp. 3749-3754, 2006.
- [56] FLIESS, M., LÉVINE, J., MARTIN, P., ROUCHON, P., “Flatness and Defect of Non-Linear Systems: Introductory Theory and Examples”, *International Journal of Control*, vol. 61, núm. 6, pp. 1327-1361, 1995.
- [57] FREED, B., *La educación. Autocrítica de Paulo Freire-Ivan Illich*, Galerna-Búsqueda de Ayllú, Buenos Aires, 2002.

- [58] FREITAS, S., MAHARG, P., *Digital Games and Learning*, Continuum, Nueva York, 2011.
- [59] GALÁN, J., ROBLES, L., MORENO, H., “La robótica aplicada a la lúdica”, *Tecnura*, vol. 15, núm. 30, pp. 52-63, 2011.
- [60] GARCÍA, C., SALTARÉN, R., LÓPEZ, J., ARACIL, R., “Desarrollo de una interfaz de usuario para el sistema robótico multiagente SMART”, *Revista Iberoamericana de Automática e Informática Industrial*, vol. 7, núm. 4, pp. 7-27, 2010.
- [61] GARDNER, R. C., CLÉMENT, R., SMYTHE, P. C., SMYTHE, C. L., *Attitude and Motivation Test Battery. Revisal Manual*, University of Western Ontario, London (Ontario), 1979.
- [62] GEE, J., P., *What Video Games Have to Teach Us about Learning and Literacy*, Palgrave Macmillan, Nueva York, 2007.
- [63] GENVO, S., “Jeux vidéo”, *Communications*, vol. 88, núm. 1, pp. 93-101, 2011.
- [64] GERMAIN, C., *Évolution de l’enseignement des langues: 5000 ans d’histoire*, CLE International, París, 1990.
- [65] GILS, B. V., DIJK, S. V., *The Practice of Enterprise Architecture. Experiences, Techniques and Best Practices*, Bizzdesign, Enschede, 2013.
- [66] GIORIO, C., FASCINARI, M., *Kinect in Motion-Audio and Visual Tracking by Example*, Packt Publishing, Birmingham (UK), 2013.
- [67] GITOMER, D., COURTNEY, A., (eds), *Handbook of Research on Teaching*, 5th ed., American Educational Research Association, Washington, D. C., 2016.
- [68] GOLDEN, B., *A Unified Formalism for Complex Systems Architecture*. Tesis doctoral, École Polytechnique, 2013.
- [69] GRANOSIK, G., BORENSTEIN, J., “Integrated Joint Actuator for Serpentine Robots”, *IEEE/ASME Trans. on Mechatronics*, vol. 10, pp. 473-481, 2005.
- [70] GRAZIA, C., CAMERLA, A., CATALDO, G., D’ORAZIO, T., “A Kinect-based Gesture Recognition Approach for Natural Human Robot Interface”, *International Journal of Advanced Robotic Systems*, pp. 1-11, 2014.
- [71] GROUPE MÉDIA TFO, *Une urgence pour le monde de l’éducation: s’adapter à la révolution numérique*, Groupe Média TFO, Ottawa, 2017.
- [72] GULDNER, J., UTKIN, V. I., “Sliding Mode Control for Gradient Tracking and Robot Navigation Using Artificial Potential Fields”, *IEEE Transactions on Robotics and Automation*, vol. 11, núm. 2, pp. 247-254, 1995.
- [73] HAMDY, A., BADREDDIN, E., “Dynamic Modeling of a Wheeled Mobile Robot for Identification, Navigation and Control”, *Proc. of the IMACS Conf. Modeling and Control of Technol. Syst.*, pp. 119-128, 1992.

- [74] HASHIMOTO, S., “Humanoid Robots in Waseda University Hadalay-2 and WABIAN”, *IARP First International Workshop on Humanoid and Human Friendly Robotics*, Tsukuba, October 1998.
- [75] HATCH, E., *Psycholinguistics: A Second Language Perspective*, Newbury House Publishers, Rowley (MA), 1983.
- [76] HATLEY, D., HRUSCHKA, P., PIRBHAI, I., *Process for Systems Architecture and Requirements Engineering*, Dorset House Publishing Co. Inc., New York, 2000.
- [77] HANSEN, T., BAK, T., RISAGER, C., “An Adaptive Game Algorithm for an Autonomous, Mobile Robot - A Real World Study with Ederly Users”, *International Symposium on Robot and Human Interactive Communication*, pp. 125-130, 2012.
- [78] HANSEN, T., RASMUSSEN, M., BAK T., “Field Study of a Physical Game for Older Adults Based on an Autonomous Mobile Robot”, *International Conference on Collaboration Technologies and Systems*, vol. 3, pp. 125-130, 2012.
- [79] HENRIOT, J., *Le jeu*, Synonyme, Paris, 1983.
- [80] HENRIOT, J., *Sous couleur de jouer. La métaphore ludique*, José Corti, Paris, 1989.
- [81] HILGARD, E., BOWER, G., *Teorías del aprendizaje*, Trillas, México, 1976.
- [82] HINRICHSSEN, D., PRITCHARD, A. J., *Mathematical Systems Theory I, Modeling, State Space Analysis, Stability and Robustness*, Springer, Berlin, 2005.
- [83] HOLLAND, J. M., *Basic Robotics Concepts*. Howard W. Sams & Co. Inc., Indianapolis, IN, 1983.
- [84] HUIZINGA, J., *Homo ludens*, Alianza, Madrid, 1990.
- [85] INSTITUTO CERVANTES, *Marco común europeo de referencia para las lenguas*, Ministerio de Educación, Cultura y Deporte, Madrid, 2002.
- [86] ISHIDA M., KHALIFA A., KATO, T., YAKAMOTO, S., “Features of Learner Corpus Collected with Joining-in Type Robot Assisted Language Learning System”, *Conference of the Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Technique*, pp. 128-131, 2016.
- [87] ISLAM, M. U., MAHMUD, H., BIN, A., HOSSAIN, I., HASAN, M. K., “Yoga Posture Recognition by Detecting Human Joint Points in Real Time using Microsoft Kinect”, *Humanitarian Technology Conference (R10-HTC)*, pp. 668-673, 2017.
- [88] ITURRATE, I., MARTÍN, G., GARCÍA-ZUBIA, J., ANGULO, I., DZIABENKO, O., ORDUÑA, P., ALVES, G., FIDALGO, A., “A Mobile Robot Platform for Open Learning Based on Serious Games and Remote Laboratories”, *1st International Conference of the Portuguese Society for Engineering Education*, pp. 1-7, 2013.

- [89] IWAMOTO, T., YAMAMOTO, H., HONMA, K., “Transformable Crawler Mechanism with Adaptability to Terrain Variations”, *Proc. of the International Conference on Advanced Robotics*, pp. 285-292, 1983.
- [90] JACKOBI, N., “Minimal Simulations for Evolutionary Robotics”. Tesis doctoral, University of Sussex, May 1998.
- [91] JAKOBSON, R., “Functions of Language”, P. B. Allen & S. P. Corder, *Reading for Applied Linguistics*, OUP, Oxford, vol. 1, pp. 53-58, 1973.
- [92] JIMÉNEZ, L. M., PUERTO, R., PAYÁ, L., *Sistemas distribuidos. Arquitectura y aplicaciones*, Universidad Miguel Hernández, Elche, 2017.
- [93] JONES, J. L., SEIGER, B. A., FLYNN, A. M., *Mobile Robots: Inspiration to Implementation*, 2nd ed., A. K. Peters/CRC Press, Natick, MA, 1998.
- [94] KASPER, M., FRICKE, G., STEUERNAGEL, K., VON PUTTKAMER, E., “A Behavior-Based Mobile Robot Architecture for Learning from Demonstration”, *Robotics and Autonomous Systems*, vol. 34, núm. 2, pp. 153-164, 2001.
- [95] KAPP, K., *The Gamification of Learning and Instruction*, Pfeiffer, San Francisco, 2012.
- [96] KELLY, A., *Mobile robots, Mathematics, Models and Methods*, Cambridge University Press, Cambridge, 2013.
- [97] KITICHAIWAT P., THONGSUK, M., NGAMSURIYAROJ, S., “Melody Touch: A Game for Learning English from Songs”, *IEEE Third ICT International Student Project Conference*, pp. 13-16, 2014.
- [98] KOSE, H., YORGANCI, R., ITAUMA, I., “Humanoid Robot Assisted Interactive Sign Language Tutoring Game”, *Proc. International Conference on Robotics and Biomimetics*, Phuket, pp. 2247-2248, 2011.
- [99] KRASHEN, S., *Second Language Acquisition and Second Language Learning*, Pergamon Press, Oxford, 1981.
- [100] KRAUSE, P. C., WASYN CZUK, O., SUDHOFF, S. D., *Analysis of Electric Machinery and Drive Systems*, 2nd ed., Wiley–Blackwell, Hoboken, NJ, 2002.
- [101] KRUZIC, S., MUSIC, J., STANCIC, I., “Influence of Human-Computer Interface Elements on Performance of Teleoperated Mobile Robot”, *Proc. MIPRO 2017*, pp. 1015-1020, 2017.
- [102] KUO, B. C., GOLNARAGHI, F., *Automatic Control Systems*, 8th ed., John Wiley & Sons, Hoboken, NJ, 2003.
- [103] LACELLE, N., BOUTIN, J.-F., LEBRUN, M., *La littératie médiatique multimodale appliquée en contexte numérique. LMM@: outils conceptuels et didactiques*, PUQ, Montreal, 2017.



- [104] LAGARIAS, J., REEDS, J., WRIGHT, M., WRIGHT, P., “Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions”, *SIAM J. Optimization*, vol. 9, núm. 1, pp. 112-147, 1998.
- [105] LAUWERS, T. B., KANTOR, G. A., HOLLIS, R. L., “A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive”, *Proc. of the IEEE International Conference on Robotics and Automation*, May 2006.
- [106] LECOMTE, P., “Pourquoi votre fille est muette... ou réflexions sur la motivation en didactique des langues”, *Les langues modernes*, vol. 79, núm. 5, pp. 19-27, 1997.
- [107] LEBRUN, M., LACELLE, N., BOUTIN, J., “Genèse et essor du concept de littératie médiatique multimodale”, *Mémoires du livre*, vol. 3, núm. 2, pp. 1-19, 2017.
- [108] LEWIS, F. L., ABDALLAH, C. T., DAWSON, D. M., *Control of Robot Manipulators*, Macmillan, New York, 1993.
- [109] LINGEMANN, K., SURMANN, H., NÜCHTER, A., HERTZBERG, J., “Indoor and Outdoor Localization for Fast-Mobile Robots”, *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2185-2190, 2004.
- [110] LIUNGMAN, C. G., *El mito de la inteligencia*, Martínez Roca, Barcelona, 1972.
- [111] Luckham D. C., Vera, J., Meldal S., “Three Concepts of System Architecture”, *Technical Report N° CSL-TR-95-674*, p. 1-22, July 1995.
- [112] MALONEY, T. J., *Electrónica industrial moderna*, 5a ed., Pearson Educación, México, 2006.
- [113] MANENAT-THALMANN, N., YUAN, J., THALMANN, D., YOU, B., (eds.), *Context Aware Human-Robot and Human-Agent, Human-Computer Interaction*, Springer, Basel (Suiza) 2016.
- [114] MARTINELLI, A., TOMATIS, N., TAPUS, A., SIEGWART, R., “Simultaneous Localization and Odometry Calibration for Mobile Robots”, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, October 2003.
- [115] MARTÍNEZ, P., *La didactique des langues étrangères*, PUF, Paris, 1995.
- [116] MASURELLE, A., ESSID, S., RICHARD, G., “Multimodal Classification of Dance Movements Using Body Joint Trajectories and Step Sounds”, *Proc. International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Paris, 2013, pp. 1-4.
- [117] MAYER, R., E., *Mecanismos del pensamiento*, Concepto, México, 1987.
- [118] MC GOVERN, J., *et al.*, *A Practical Guide to Enterprise Architecture*, Prentice Hall, Upper Saddle River, NJ, 2003.

- [119] MICROSOFT. “AdaBoostTrigger”, *AdaBoostTrigger*. [En línea] [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785522\(v=ieeb.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785522(v=ieeb.10)?redirectedfrom=MSDN)
- [120] MICROSOFT. “Juegos. Kinect”, *Microsoft Store*. [En línea] <https://www.microsoft.com/es-mx/search/result.aspx?q=kinect>
- [121] MICROSOFT. “Kinect Studio”, *Kinect Studio Application Windows*. [En línea] [https://docs.microsoft.com/en-us/previous-versions/windows/kinect-1.8/hh855397\(v=ieeb.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect-1.8/hh855397(v=ieeb.10))
- [122] MICROSOFT. “RFR Progress”, *RFRProgres*. [En línea] [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785524\(v=ieeb.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785524(v=ieeb.10))
- [123] MICROSOFT. “Visual Gesture Builder”, *Visual Gesture Builder: Overview*. [En línea] [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785529%28v%3dieeb.10%](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785529%28v%3dieeb.10%28)
- [124] MICROSOFT CORPORATION, *Kinect for Windows Software Development Kit (SDK) 2.0*, Redmond, WA, 2013.
- [125] MICROSOFT CORPORATION, *Visual Studio / versión 12*. Redmond, WA, 2013.
- [126] MICROSOFT CORPORATION, *Windows 10*, Redmond, WA, 2015.
- [127] MORAVEC, H. P., “Visual Mapping by a Robot Rover”, *Proc. of the 6th IJCAI*, Tokyo, pp. 598-600, 1979.
- [128] MORRIS MANO M., *Arquitectura de computadoras*, Pearson Prentice Hall, México, 1993.
- [129] MOTTE, I., CAMPION, G., “A Slow Approach for the Control of Mobile Robots Not Satisfying the Kinematic Constraints”, *IEEE Transactions on Robotics and Automation*, vol. 16, núm. 6, pp. 875-880, 2000.
- [130] MUIR, P. F., “Modeling and Control of Wheeled Mobile Robots”. Tesis doctoral, The Robotics Institute Carnegie Mellon University, Pittsburgh, August 1988.
- [131] MUIR, P. F., NEUMAN, C. P., “Kinematic Modeling of Wheeled Mobile Robots”, *Technical Report N° CMU-RI-TR-86-12*, The Robotics Institute and The Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, June 1986.
- [132] MÜLLER, J., “L’intermedialité, une nouvelle approche interdisciplinaire: perspectives théoriques et pratiques à l’exemple de la vision de la télévision”, *Cinémas. Revue d’études cinématographiques*, vol. 10, núm. 2-3, pp. 105-134, 2000.

- [133] NEGRETE, M., SAVAGE, J., CRUZ, J., MÁRQUEZ, J., “Behavior-Based Navigation System for a Service Robot with a Mechatronic Head”, *Memorias del Congreso Latinoamericano de Control Automático*, 14-17 de oct., Cancún (Q. Roo), 2014.
- [134] NEMSER, W., “Approximative Systems of Foreign Language Learners”, *IRAL*, núm. 9, pp. 115-123, 1971.
- [135] OGATA, K., *Ingeniería de control moderna*, 5a edición, Pearson, Madrid, 2010.
- [136] OHASHI, O., OCHIAI, E., KATO, Y., “A Remote Control Method for Mobile Robots Using Game Engines”, *28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 79-84, 2014.
- [137] PAYEUR, P., NASCIMENTO, G., BEACON, J., COMEAU, G., CRETU, A., D’Aoust, V., CHARPENTIER, M., “Human Gesture Quantification: An Evaluation Tool for Somatic Training and Piano Performance”, *Proc. International Symposium on Haptic, Audio and Visual Environments and Games*, pp. 100-105, 2014.
- [138] PIAGET, J., *Psicología y epistemología*, Ariel, México, 1979.
- [139] PINEDA, L. A., RODRÍGUEZ, A., FUENTES, G., RASCON, C., MEZA, I.V., “Concept and Functional Structure of a Service Robot”, *International Journal of Advanced Robotic System*, 2015.
- [140] POLOLU CORPORATION. “Roboclaw Motor Controllers”, *Pololu. Robotics & Electronics*, [En línea] <https://www.pololu.com/category/124/roboclaw-motor-controllers>.
- [141] PUTMAN, J., *Architecting with RM-ODP*, Prentice Hall, Upper Saddle River, NJ, 2001.
- [142] QINGTANG, L., YANG, W., LINJING, W., WU PENG, H., “Design and Implementation of a Serious Game Based on Kinect”, *International Conference of Educational Innovation through Technology (EITT)*, Wuhan (China), pp. 13-18, 2015.
- [143] QUANSER CONSULTING, QuaRC / versión 2.3.603, 2011.
- [144] RAESSENS, J., “Playful Identities, or the Ludification of Culture”, *Games and Culture*, vol. 1, núm. 1, 2006, pp. 52-57.
- [145] RAHMAN, M. F., PATTERSON, D., CHEOK, A., BETTS, R., “Motor Drives”, *Power Electronics Handbook*, pp. 663-733, Academic Press, Orlando, 2001.
- [146] RAIBERT, M. H., BROWN, H. B., CHEPPONIS, M., HASTINGS, E., KOECHLING, J., MURPHY, K. N., MURTHY, S. S., STENTZ, A. J., “Dynamically Stable Legged Locomotion”, *Technical Report N° CMU-RI-TR-83-20*, The Robotics Institute and Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, December 1983.

- [147] RAINER, J., GALÁN, J., “Learning to Be a Good Tour-Guide Robot”, *International Conference on Machine Learning and Applications*, 12-14 de dic., Washington D.C., pp. 595-600, 2010.
- [148] RAINNE, V. D. B., JOSJE V., ORA O. P., SANNE V. D. V., PAUL L., “Social Robots for Language Learning: A Review”, *Review of Educational Research*, vol. 80, núm. 2, pp. 259-295, 2019.
- [149] RAJAGOPALAN, R., “A Generic Kinematic Formulation for Wheeled Mobile Robots”, *Journal of Robotic Systems*, vol. 14, pp. 77-91, 1997.
- [150] REAL ACADEMIA ESPAÑOLA, *Diccionario de la lengua española*, 23a ed., Real Academia Española, Madrid, 2014.
- [151] REAL ACADEMIA ESPAÑOLA, *Diccionario Inverso de la Real Academia Española*, 23a ed., Real Academia Española, Madrid.
- [152] RICHARDS, J., “Error Analysis and Second Language Strategies”, *Language Sciences*, núm. 17, pp. 12-22, 1971.
- [153] ROBOTZONE. “313 RPM HD Premium Planetary Gear Motor”, *Servocity*. [En línea] <https://www.servocity.com/313-rpm-hd-premium-planetary-gear-motor>
- [154] ROSCHELLE, J. *et al.*, “Changing How and What Children Learn in School with Computer-Based Technologies”, *The Future of Children*, vol. 10, núm. 2, pp. 76-10, 2000.
- [155] ROSSOMANDO, F. G., SORIA, C., CARELLI, R., “Adaptive Neural Dynamic Compensator for Mobile Robots in Trajectory Tracking Control”, *IEEE Latin America Transactions*, vol. 9, núm. 5, pp. 593-602, 2011.
- [156] RUBIO, J. J., TORRES, C., RIVERA, R., HERNÁNDEZ, C. A., “Comparison of Four Mathematical Models for Braking of a Motorcycle”, *IEEE Latin America Transactions*, vol. 9, núm. 5, pp. 630-637, 2011.
- [157] RUSSELL, S. J., NORVIG, P., *Artificial intelligence: A Modern Approach*, 3rd ed., Prentice Hall, Upper Saddle River, 2009.
- [158] RUTVEN, K., HENNESSY, S., DEANEY, R., “Incorporating Internet Resources into Classroom Practice: Pedagogical Perspectives and Strategies of Secondary School Subject Teachers”, *Computers & Education*, núm. 44, pp. 1-34, 2005.
- [159] SCHEDING, S., DISSANAYAKE, G., NEBOT, E. M., DURRANT-WHYTE, H., “Experiment in Autonomous Navigation of an Underground Mining Vehicle”, *IEEE Trans. Robot. Automat.*, vol. 15, pp. 85-95, 1999.
- [160] SCOTT, B., *An Introduction to Enterprise Architecture*, Authorhouse, Bloomington, IN, 2005.

- [161] SEGWAY, “Loomo Personal Robot”, *Loomo*. [En línea] <https://www.segway.com/loomo/>
- [162] SEGWAY, “Manual de usuario”, *Loomo*. [En línea] [https://static4.segway.com/wp-content/uploads/2020/02/UM\\_Loomo-Robot\\_EN-CH-FR-GE-IT-KR-RU-SP.pdf](https://static4.segway.com/wp-content/uploads/2020/02/UM_Loomo-Robot_EN-CH-FR-GE-IT-KR-RU-SP.pdf)
- [163] SEGWAY, “SDK robot Loomo”, *Loomo*. [En línea] <https://developer.segwayrobotics.com/developer/documents/segway-robots-sdk.html>
- [164] SELINKER, L., “Interlanguage”, *IRAL*, núm. 10, pp. 209-231, 1972.
- [165] SHEKHAR, S., “Wheel Rolling Constraints and Slip in Mobile Robots”, *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2601-2607, April 1997.
- [166] SIEGWART, R., NOURBAKSH, I. R., *Introduction to Autonomous Mobile Robots*, MIT Press, Massachusetts, 2004.
- [167] SILVA, H., “La gamification de la vie: sous couleur de jouer ?”, *Sciences du jeu*, núm. 1, pp. 1-11, 2013.
- [168] SILVA, H., *Le jeu en classe de langue*, CLE International, París, 2008.
- [169] SILVA, H., “Paradigmas y niveles del juego”, RAMOS, J. L., coord *Juego, educación y cultura*, ENAH/Conaculta, México, pp. 35-52, 1999.
- [170] SILVA, H., BROUGÈRE, G., “Jouer pour apprendre une langue étrangère: concert à 16 voix”, *Synergies Mexique*, núm. 7, pp. 51-62, 2017.
- [171] SILVA, H., BROUGÈRE, G., “Le jeu entre situations formelles et informelles d’apprentissage des langues étrangères”, *Synergies Mexique*, núm. 6, pp. 57-68, 2016.
- [172] SILVA, H., LOISEAU, M., (eds.) *Recherches et applications*, núm. 59: *Jeu(x) et langue(s): avatars du ludique dans l’enseignement/apprentissage des langues*, CLE International, Paris, 2016.
- [173] SILVA-ORTIGOZA, R., SILVA-ORTIGOZA, G., HERNÁNDEZ-GUZMÁN, V. M., BARRIENTOS-SOTELO, V. R., ALBARRÁN-JIMÉNEZ, J. M., SILVA-GARCÍA, V. M., “Trajectory Tracking in a Mobile Robot without Using Velocity Measurements for Control of Wheels”, *IEEE Latin America Transactions*, núm. 6, pp. 598-607, 2008.
- [174] SIRA-RAMÍREZ, H., AGRAWAL, S. K., *Differentially Flat Systems*, Marcel Dekker, New York, 2004.
- [175] SIRVENT, J. L., AZORÍN, J. M., IÁNEZ, E., ÚBEDA, A., FERNÁNDEZ, E., “Interfaz cerebral no invasiva basada en potenciales evocados para el control de un brazo robot”, *Revista Iberoamericana de Automática e Informática Industrial*, vol. 8, núm. 2, pp. 103-111, 2011.
- [176] SKINNER, B., F., *The Behavior of Organisms*, Appleton-Century-Crofts, New York, 1938.

- [177] SOCKETT, G., *The Online Informal Learning of English*, Palgrave Macmillan, London, 2014.
- [178] SQUIRE, K., *Video Games and Learning*, TCP, New York, 2011.
- [179] SPONG, M. W., HUTCHINSON, S., VIDYASAGAR, M., *Robot Modeling and Control*, John Wiley & Sons, Inc., Hoboken, NJ, 1989.
- [180] STEELS, L., “Grounding Language through Evolutionary Language Games”, STEELS, L., HILD, M., *Language Grounding in Robots*, Springer, New York, 2012.
- [181] STEELS, L., “Language Games for Autonomous Robots”, *Intelligent Systems*, vol. 16, núm. 5, pp. 16-22, 2001.
- [182] STEINKUEHLER, C., SQUIRE, K., BARAB, S. (eds.), *Games, Learning and Society: Learning and Meaning in the Digital Age*, Cambridge University Press, New York, 2012.
- [183] STEVENS, A., SHIELD, L., *Étude sur l’impact des technologiques de l’information et de la communication et des nouveaux médias sur l’apprentissage des langues. Rapport final*, Commission Européenne, Strasbourg, 2009.
- [184] SU, K. L., SHIAU, S. V., GUO, J. H., SHIAU, C. W., “Mobile Robot Based Online Chinese Chess Game”, *IEEE Fourth International Conference on Innovative Computing, Information and Control*, vol. 2, pp. 528-531, 2009.
- [185] SYKES, J. M., “Just, Playing Games? A Look at the Use of Digital Games for Language Learning”, *The Language Educator*, 2013.
- [186] SYKES, J., M., REINHARDT, J., *Language at Play. Digital Games in Second and Foreign Language Teaching and Learning*, Pearson, New Jersey, 2009.
- [187] TANAKA, K., YOSHIOKA, K., “Fuzzy Trajectory Control and GA-Based Obstacle Avoidance of a Truck with Five Trailers”, *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, October 1995.
- [188] TEAMVIEWER, “El acceso remoto y la asistencia remota”, *TeamViewer*. [En línea] <https://www.teamviewer.com/es-mx/#>
- [189] THE MATH WORKS, *Matlab-Simulink / versión 8.0 (R2012b)*, Natick, MA, USA, 2012.
- [190] THOMIÈRES, D., “Introduction”, *Les langues modernes*, vol. 5, pp. 7-18, 1985.
- [191] THRUN, S., BENNWITZ, M., BURGARD, W., CREMERS, A. B., DELLAERT, F., FOX, D., HANEL D., ROSENBERG, C., ROY, N., SCHULTE, J., SHULZ, D., “MINERVA: A Second-Generation Museum Tour-Guide Robot”, *IEEE International Conference on Robotics & Automation*, Detroit, Michigan, 1999.
- [192] THRUN, S., BÜCKEN, A., *Learning Maps for Indoor Mobile Robot Navigation*, Carnegie Mellon University, Pittsburgh, April 1996.

- [193] TITONE, R., *Psicolingüística aplicada*, Kapelusz, Buenos Aires, 1976.
- [194] TODD, D. J., *Walking Machines: an Introduction to Legged Robots*, Kogan Page, London, 1985.
- [195] TÜNNERMANN, B., “El constructivismo y el aprendizaje de los estudiantes”, *Universidades*, núm. 48, pp. 21-32, 2011.
- [196] US DIGITAL, “E2 Optical Kit Encoder”, US Digital, Vancouver (WA) [En línea] [https://cdn.usdigital.com/assets/datasheets/E2\\_datasheet.pdf](https://cdn.usdigital.com/assets/datasheets/E2_datasheet.pdf)
- [197] USD of Defense, *Technical Architecture Framework for Information Management (TAFIM)*, USD of Defense, 1994.
- [198] VEENMAN, C.J., HENDRIKS, E.A., REINDERS, M.J.T., “A Fast and Robust Point Tracking Algorithm”, *Proc. 15th IEEE International Conference on Image Processing*, pp. 653-657, 1998.
- [199] VEENMAN, C.J., REINDERS, M.J.T., BACKER, E., “Resolving Motion Correspondence for Densely Moving Points”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, núm. 1, pp. 54-72, 2001.
- [200] VELOSO, M., STONE, P., “Video: RoboCup Robot Soccer History1997-2011”, *Proc. International Conference on Intelligent Robots and Systems*, Vilamoura (Portugal), pp. 5452-5453, 2012.
- [201] VILLANI, V., SABATTINI, L., RIGGIO, G., LEVRATTI, A., SECCHI, C., FANTUZZI, C., “Interacting with a Mobile Robot with a Natural Infrastructure-Less Interface”, *IFAC World Congress*, vol. 5, núm. 1, pp. 12753-12758, 2017.
- [202] VOS, D. W., VON FLOTOW, A. H., “Dynamics and Nonlinear Adaptive Control of an Autonomous Unicycle: Theory and Experiment”, *29th IEEE Conference on Decision and Control*, vol. 1, pp. 182-187, 1990.
- [203] VYGOTSKY, L., *Pensamiento y lenguaje*, Lautaro, Buenos Aires, 1962.
- [204] WANG, Y., CHEN, D., “Autonomous Navigation Based on a Novel Topological Map”, *Proc. Asia-Pacific Conference on Information Processing*, Shenzhen (China), pp. 254-257, 2009.
- [205] WIENER, N. “Cybernetics or Control and Communication in the Animal and the Machine”, 2nd ed., The MIT Press, Massachusetts, 1965.
- [206] WILLIAMS, R. L., CARTER, II, B. E., GALLINA, P., ROSATI, G., “Dynamic Model with Slip for Wheeled Omnidirectional Robots”, *IEEE Trans. Robot. Automat.*, vol. 18, pp. 285-293, 2002.

- [207] WORLD ECONOMIC FORUM, *New Vision for Education: Fostering Social and Emotional Learning through Technology*, World Economic Forum, Boston, 2016.
- [208] WORLD ECONOMIC FORUM, *New Vision for Education Unlocking the Potential of Technology*, World Economic Forum, Boston, 2015.
- [209] WU, J., LV, C., ZHAO, L., LI, R., WANG, G., “Design and Implementation of an Omnidirectional Mobile Robot Platform with Unified I/O Interfaces”, *International Conference on Mechatronics and Automation*, vol. 3, pp. 410-415, 2017.
- [210] XIAOLEI, H., PENGFEI, F., YAOHONG, G., “Dynamic Kinesthetic Boundary for Haptic Teleoperation of Unicycle Type Ground Mobile Robots”, *Proc. 36th Chinese Control Conference*, vol. 2, pp. 6246-6251, 2017.
- [211] YANES N., BOUOUD, I., “Using Gamification and Serious Games for English Language Learning”, *International Conference on Computer and Information Sciences (ICCIS)*, vol. 3, pp. 7301-7312, 2019.
- [212] YANG, J. M., CHOI, I. H., KIM, J. H., “Sliding Mode Control of a Nonholonomic Wheeled Mobile Robot for Trajectory Tracking”, *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 2983-2988, Leuven, May 1998.
- [213] YANGSHENG, X., YONGSHENG, O., *Control of Single Wheel Robots*, Springer-Verlag, New York, 2005.
- [214] YU, W., ZAMORA, E., SORIA, A., “Ellipsoid SLAM: A Novel Set Membership Method for Simultaneous Localization and Mapping”, *Autonomous Robots*, vol. 40, núm. 1, pp. 125-137, 2016.
- [215] ZAFAR, T., USMAN, K. M., NAWAZ, A., AHMAD, K. F., “Smart Phone Interface for Robust Control of Mobile Robots”, *Proc. International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Portugal, vol. 1, pp. 42-46, 2016.
- [216] ZIPPY, “Batería Zippy”, US Hobbyking. [En línea] [https://hobbyking.com/en\\_us/zippy-compact-4500mah-4s-35c-lipo-pack.pdf](https://hobbyking.com/en_us/zippy-compact-4500mah-4s-35c-lipo-pack.pdf)



## Apéndice A. Modelado y control de Ludibot v1

En este apéndice se describe el modelo cinemático de Ludibot v1, cuyas condiciones de operación es importante recordar:

- El robot, considerado como un cuerpo rígido, se mueve en una superficie plana; el deslizamiento en las ruedas es mínimo y, por ello, en esta tesis se considera irrelevante.
- El robot es impulsado mediante ruedas.
- Se trata de un robot móvil del tipo unicyclo, que consta de dos ruedas de tracción colocadas en la parte inferior, además de una rueda que puede moverse libremente para mantener el robot en una posición horizontal estable. Cada una de las ruedas de tracción puede girar de manera independiente a diferentes velocidades, permitiendo que el robot gire.

### Modelado cinemático

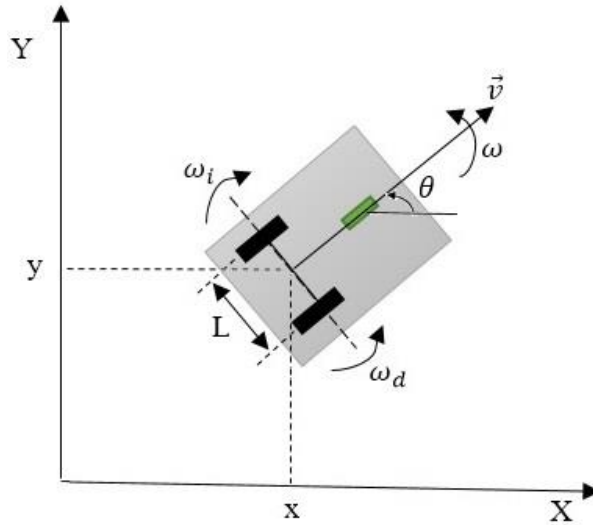
La cinemática de movimiento del robot móvil es representada por:

$$\begin{aligned}\dot{x} &= v \cos(\theta) \\ \dot{y} &= v \sin(\theta) \\ \dot{\theta} &= \omega\end{aligned}\tag{A.1}$$

donde  $(x, y, \theta)$  es el vector de estados del robot,  $v$  es la velocidad lineal,  $\omega$  es la velocidad angular del robot. Las velocidades lineal y angular para el robot mostrado se definen como:

$$\begin{aligned}v &= \frac{(\omega_d + \omega_i)r}{2} \\ \omega &= \frac{(\omega_d - \omega_i)r}{L}\end{aligned}\tag{A.2}$$

donde  $\omega_d$  y  $\omega_i$  son las velocidades de las ruedas derecha e izquierda, respectivamente, y  $L$  determina la distancia entre ambas ruedas.



**Figura A.1. Diagrama del robot móvil**

Reescribiendo las ecuaciones (A.1) de manera matricial se obtiene:

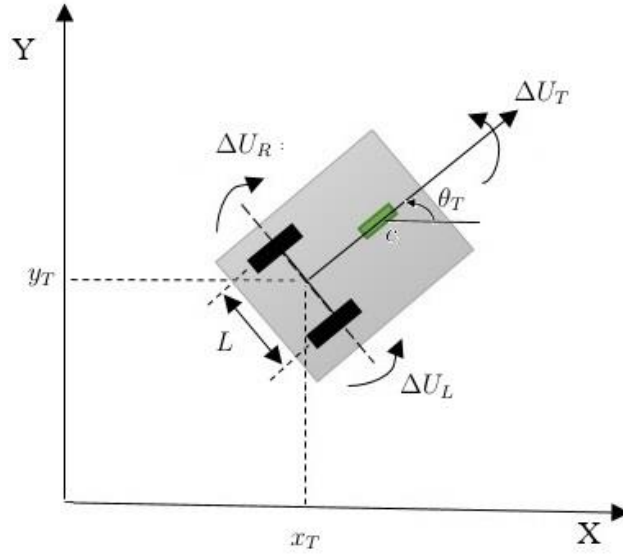
$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (A.3)$$

Debido a que la odometría es extensamente usada en la navegación de un robot móvil, es necesario llevar a cabo la localización del robot sin acumular grandes errores. Puesto que la acumulación del error de orientación y posición aumenta cuando el robot viaja a grandes distancias, describimos a continuación una metodología para estimar y corregir el modelo odómetro del robot.

### **Estimación y corrección de odometría**

El movimiento del robot ludoeducativo se detecta mediante sensores dotados de decodificadores ópticos incrementales que permiten medir la posición angular del eje de las ruedas. Es importante que en la estimación de la posición del robot no se acumulen errores.

Existe una serie de factores que pueden causar errores, tales como la resolución limitada de los sensores de posición angular, la desalineación en las ruedas, la incertidumbre en el diámetro de las ruedas, la variación en el punto de contacto de la rueda y las variaciones de contacto con el suelo.



**Figura A.2. Representación de Borenstein del robot móvil**

Para reducir los errores, se emplea el modelo de Borenstein de un robot diferencial [21][22][23] (Figura A.2). La posición puede ser estimada a partir de los desplazamientos incrementales de cada rueda en un tiempo de muestreo  $T$ . El número de pulsos incrementales de cada rueda medidos por los sensores son  $N_{L,T}$  pulsos para la rueda izquierda y  $N_{R,T}$  pulsos para la rueda derecha. El modelo de error está dado por:

$$\begin{aligned}
 \Delta U_L &= c_L N_{L,T} \\
 \Delta U_R &= c_R N_{R,T} \\
 \Delta U_T &= \frac{\Delta U_R + \Delta U_L}{2} \\
 \Delta \theta_T &= \frac{\Delta U_R - \Delta U_L}{L} \\
 \theta_T &= \theta_{T-1} + \Delta \theta_T \\
 x_T &= x_{T-1} + \Delta U_T \cos(\theta_T) \\
 y_T &= y_{T-1} + \Delta U_T \sin(\theta_T)
 \end{aligned} \tag{A.4}$$

donde  $U_T$  es el desplazamiento lineal incremental del punto  $c$ ,  $\Delta \theta_T$  es el desplazamiento incremental angular y el vector de datos  $(x_T, y_T, \theta_T)$  indica los estados de posición del robot móvil.  $c_L$  y  $c_R$  determinan los factores de conversión que traducen los pulsos del decodificador en el desplazamiento lineal de cada rueda.

Para calibrar los parámetros  $L, c_L, c_R$  se emplea un método de optimización siguiendo los siguientes pasos:

- Se selecciona una posición inicial  $(0, 0, 0)$  y una posición final  $(x_f^*, y_f^*, \theta_f^*)$ .
- Se miden las coordenadas cartesianas del punto final con respecto al punto inicial.
- Se mueve el robot de la posición inicial a la posición final  $N$  veces, siguiendo una trayectoria arbitraria y recolectando los datos de los decodificadores ópticos incrementales.
- Se aplica el método de optimización.
- Se estiman los valores de  $(L^*, c_L^*, c_R^*) = \arg \min F(b, c_L, c_R)$ .

Para reducir el error del desplazamiento lineal  $E_L$  y el desplazamiento angular  $E_A$  con respecto al desplazamiento total dado por la trayectoria, se define la función  $F(b, c_L, c_R)$  como:

$$F(b, c_L, c_R) = \sum_{i=1}^N E_{L,i} + \sum_{i=1}^N E_{A,i} \quad (\text{A.5})$$

Escribiendo  $F(b, c_L, c_R)$  en términos del desplazamiento lineal y angular se tiene que:

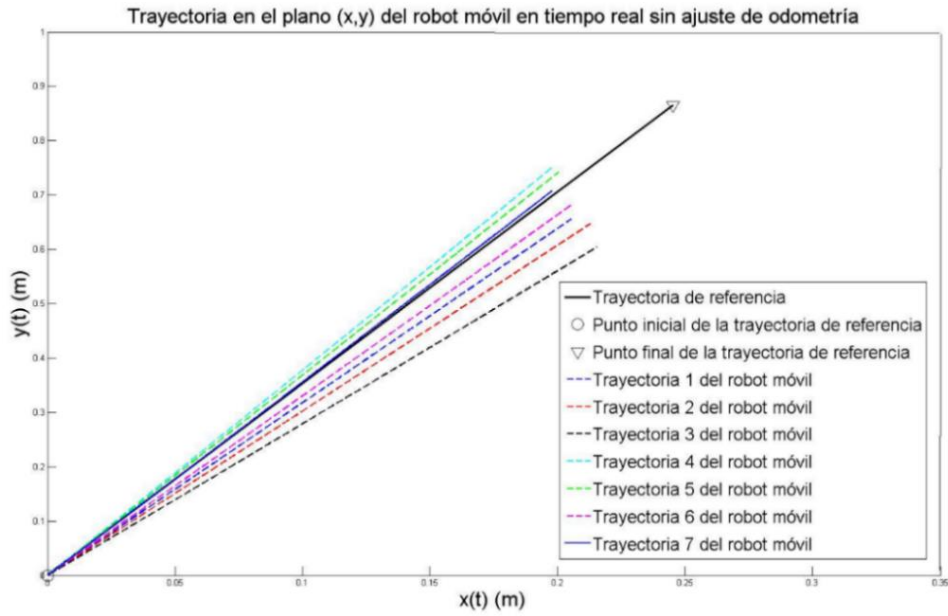
$$F(b, c_L, c_R) = \sum_{i=1}^N \frac{1}{d_i} \sqrt{(x_f^* - x_{f,i}^*)^2 + (y_f^* - y_{f,i}^*)^2} + \sum_{i=1}^N \frac{1}{d_{\theta,i}} |\theta_f^* - \theta_{f,i}^*| \quad (\text{A.6})$$

donde  $i$  define el camino,  $(x_{f,i}, y_{f,i}, \theta_{f,i})$  establece la posición calculada por la odometría,

$d_i = \sum_{j=1}^{M_i-1} \sqrt{(x_{j+1} - x_{j,i})^2 + (y_{j+1} - y_{j,i})^2}$  determina el desplazamiento lineal y  $d_{\theta,i} = \sum_{j=1}^{M_i-1} |\theta_{j+1} - \theta_{j,i}|$  fija

el desplazamiento angular.

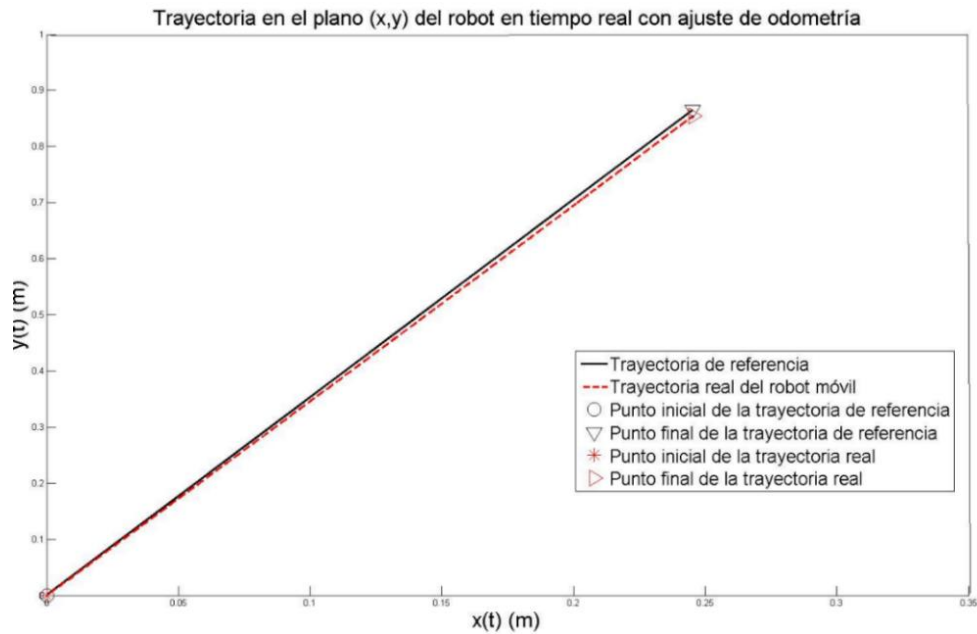
Es importante notar que  $\theta_f^*, \theta_{f,i}^* \in [-\pi, \pi]$  y  $M_i$  es el número de posiciones finales. Para la calibración de los parámetros se fijó  $N = 7$  como el número de veces que se siguió la misma trayectoria. Las posiciones finales  $(x_f^*, y_f^*, \theta_f^*)$  se generaron de manera aleatoria. En la Figura A.3 se muestran las trayectorias sin llevar a cabo la calibración de la odometría, en donde se puede notar que no se sigue la trayectoria de referencia.



**Figura A.3. Trayectorias del robot móvil sin ajuste de odometría**

En la Figura A.4 se muestra la trayectoria obtenida con los parámetros calibrados ( $L^*$ ,  $c_L^*$ ,  $c_R^*$ ). Se empleó como método de optimización el método Simplex de Nelder-Mead [104] en Matlab, mediante la función *fminsearch* que optimiza la ecuación A.6. Puede observarse que la trayectoria real del robot es muy similar a la de referencia.

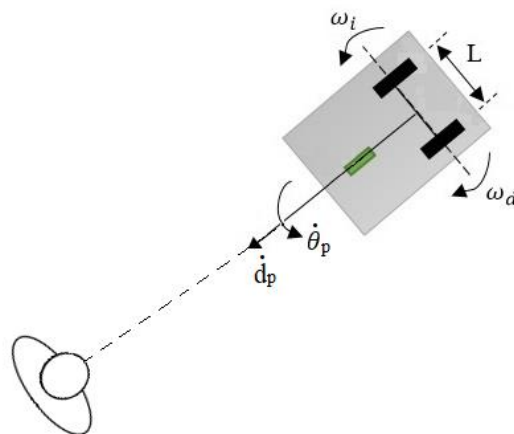
Este experimento fue realizado con los parámetros calibrados:  
 $L^* = 0.401$ ,  $c_L^* = 0.09645$ ,  $c_R^* = 0.9650$ .



**Figura A.4. Trayectoria del robot móvil con ajuste de odometría**

### Control de seguimiento de usuario

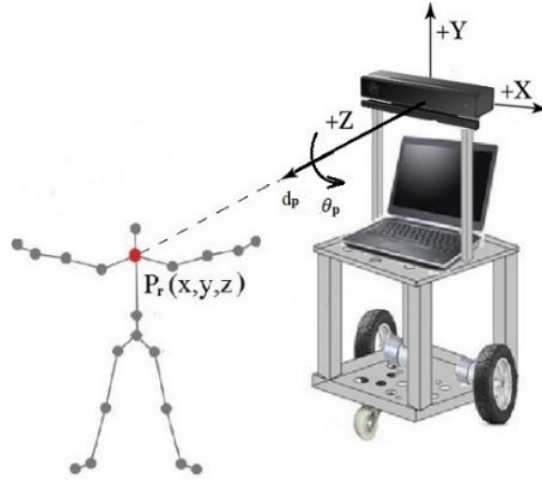
El control de posición y el seguimiento de trayectorias son las tareas más comunes de los robots móviles. Se propone aquí un modelo cinemático alternativo, que mide la distancia y el ángulo entre el usuario y el robot. Explícitamente, el modelo considera la velocidad lineal y la velocidad angular medida por el sistema de visión infrarrojo para medir la profundidad, usando el tiempo de vuelo de Kinect v2 como es ilustrado en la Figura A.5.



**Figura A.5. Distancia entre usuario y robot móvil considerando distancia y ángulo**

De esta manera, el modelo cinemático usa el marco de referencia de Kinect v2, que está situado en la parte superior del robot, cuyos grados de libertad son definidos como la distancia y el

ángulo formado entre ambos elementos. La Figura A.6 representa la relación entre las variables de distancia y ángulo para el seguimiento de usuario mediante el robot móvil.



**Figura A.6. Variables de relación distancia  $d_p$  y ángulo  $\theta_p$  entre el usuario y el robot móvil**

Las variables  $d_p$  y  $\theta_p$  son definidas con respecto al punto  $P_r$  de la persona, detectado con Kinect.

Se definen como:

$$d_p = \sqrt{Z_k^2 + X_k^2} \quad (\text{A.7})$$

$$\theta_p = \sin^{-1} \left( \frac{X_k}{\sqrt{Z_k^2 + X_k^2}} \right) \quad (\text{A.8})$$

donde  $X_k$  y  $Z_k$  son los parámetros medidos sobre los ejes del marco de referencia  $(X,Z)$  de Kinect v2.

De esta manera, se puede obtener un modelo cinemático del robot en términos de las variables del sensor Kinect v2. Este modelo permite controlar el ángulo y la distancia usando el sistema de visión infrarrojo para medir profundidad mediante el tiempo de vuelo de Kinect v2 como sensor de realimentación. Por lo tanto, el modelo cinemático para el problema de seguimiento usuario-robot móvil es determinado como:

$$\begin{aligned}\dot{d}_p &= \frac{r(\omega_d + \omega_i)}{2} \\ \dot{\theta}_p &= \frac{r(\omega_d - \omega_i)}{L}\end{aligned}\tag{A.9}$$

Definiendo  $\omega_d$  y  $\omega_i$  como las entradas del control  $u_d, u_i$  y reescribiendo el modelo de manera matricial, se obtiene:

$$\begin{pmatrix} \dot{d}_p \\ \dot{\theta}_p \end{pmatrix} = \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{pmatrix} \begin{pmatrix} u_d \\ u_i \end{pmatrix}\tag{A.10}$$

Es importante mencionar que la distancia es medida en metros y el ángulo en grados.

En general, este modelo no requiere medir el desplazamiento mediante los decodificadores ópticos incrementales. En caso de que exista deslizamiento de las llantas, los errores no serán perjudiciales, ya que el movimiento del robot depende del sistema de visión infrarrojo para medir profundidad usando el tiempo de vuelo de Kinect v2. Dicho sistema presenta la ventaja de que en el modelo no se acumulan errores en la odometría midiendo constantemente la distancia y el ángulo entre el usuario y el robot. La distancia y el ángulo entre el usuario y el robot son definidos como los grados de libertad y variables por controlar. Consecuentemente, esto permite la implementación de una estrategia de control para cada grado de libertad para el modelo (A.10) y así resolver el problema de control de seguimiento usuario-robot móvil.

Se propone una estrategia de control proporcional integral para la parte lineal y para la parte angular para el sistema (A.10) dada por:

$$\begin{aligned}u_d &= k_{P_d} e_d + k_{I_d} \int e_d dt + k_{P_\theta} e_\theta + k_{I_\theta} \int e_\theta dt \\ u_i &= k_{P_d} e_d + k_{I_d} \int e_d dt - k_{P_\theta} e_\theta - k_{I_\theta} \int e_\theta dt\end{aligned}\tag{A.11}$$

sustituyendo (B11) en (A.10) se tiene que:



$$\begin{pmatrix} \dot{d}_p \\ \dot{\theta}_p \end{pmatrix} = \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{pmatrix} \begin{pmatrix} k_{p_d} e_d + k_{I_d} \int e_d dt + k_{p_\theta} e_\theta + k_{I_\theta} \int e_\theta dt \\ k_{p_d} e_d + k_{I_d} \int e_d dt - k_{p_\theta} e_\theta - k_{I_\theta} \int e_\theta dt \end{pmatrix} \quad (\text{A.12})$$

Desarrollando (A.12) y separando las variables de estado  $\dot{d}_p$  y  $\dot{\theta}_p$  se obtienen las expresiones

$$\dot{d}_p(t) = 2 \frac{r}{2} \left( k_{p_d} e_d + k_{I_d} \int e_d dt \right) \quad (\text{A.13})$$

$$\dot{\theta}_p(t) = 2 \frac{r}{L} \left( k_{p_\theta} e_\theta + k_{I_\theta} \int e_\theta dt \right) \quad (\text{A.14})$$

donde  $e_d = d_r(t) - d_p(t)$ ,  $e_\theta = \theta_r(t) - \theta_p(t)$  son los errores de distancia y ángulo,  $d_r(t)$  y  $\theta_r(t)$  son la distancia y el ángulo de referencia, respectivamente.

Para estudiar la estabilidad del sistema y determinar las ganancias, se reescriben las ecuaciones (A.13) y (A.14) en el dominio de Laplace, obteniendo:

$$s d_p(s) = r k_{p_d} d_r(s) - r k_{p_d} d_p(s) + r k_{I_d} \frac{d_r(s)}{s} - r k_{I_d} \frac{d_p(s)}{s} \quad (\text{A.15})$$

$$s \theta_p(s) = \frac{2r}{L} k_{p_\theta} \theta_r(s) - \frac{2r}{L} k_{p_\theta} \theta_p(s) + \frac{2r}{L} k_{I_\theta} \frac{\theta_r(s)}{s} - \frac{2r}{L} k_{I_\theta} \frac{\theta_p(s)}{s} \quad (\text{A.16})$$

definiendo la función de transferencia en términos de la distancia y la distancia de referencia de la ecuación (A.15)

$$\frac{d_p(s)}{d_r(s)} = \frac{srk_{p_d} + rk_{I_d}}{s^2 + srk_{p_d} + rk_{I_d}} \quad (\text{A.17})$$

Por lo tanto, la función de transferencia (A.17) tiene el siguiente polinomio característico:

$$s^2 + srk_{p_d} + rk_{I_d} = 0 \quad (\text{A.18})$$

de manera similar, se obtiene la relación entrada-salida para la ecuación (A.16). Definiendo

$k = \frac{2r}{L}$  se tiene:

$$\frac{\theta_p(s)}{\theta_r(s)} = \frac{skk_{p_\theta} + kk_{I_\theta}}{s^2 + skk_{p_\theta} + kk_{I_\theta}} \quad (\text{A.19})$$

donde el polinomio característico de (A.19) es dado como:

$$s^2 + skk_{p_\theta} + kk_{I_\theta} = 0 \quad (\text{A.20})$$

Por lo tanto, para que todos los polos de los polinomios característicos (A.18) y (A.20) estén del lado izquierdo del plano complejo y que, además, el sistema sea estable, se debe cumplir que  $k_{p_d} > 0, k_{p_\theta} > 0, k_{I_d} > 0, k_{I_\theta} > 0$ , respectivamente.

### Control de seguimiento de trayectorias

Para que el robot realice movimientos planeados, es necesario resolver el problema de seguimiento de trayectorias de acuerdo con el control de vehículos autónomos estudiado por Canudas [33].

Este problema puede formularse y solucionarse mediante estrategias de control que permitan estabilizar el robot móvil en un punto de operación, o simplemente hacer que el robot siga de forma adecuada una trayectoria de referencia. Para cualquier trayectoria deseada se emplea el siguiente modelo:

$$\begin{aligned} \dot{x}_{ref} &= v_{ref} \cos(\theta_{ref}) \\ \dot{y}_{ref} &= v_{ref} \sin(\theta_{ref}) \\ \dot{\theta}_{ref} &= \omega_{ref} \end{aligned} \quad (\text{A.21})$$

donde se considera que  $v_{ref}, \omega_{ref}$  y sus respectivas derivadas son acotadas, y se supone que:

$$\begin{aligned}\lim_{t \rightarrow \infty} v_{ref}(t) &\neq 0 \\ \lim_{t \rightarrow \infty} \omega_{ref}(t) &\neq 0\end{aligned}\quad (A.22)$$

Esto implica que se desea seguir una trayectoria, ya que en el caso en que las velocidades fueran nulas, el problema consistiría en alcanzar una postura de referencia fija. El problema para el seguimiento de trayectorias se basa en diseñar una estrategia de control  $u = (u_d \quad u_i)^T$  tal que el error cumpla lo siguiente:

$$e = \lim_{t \rightarrow \infty} [p(t) - p_{ref}(t)] = 0. \quad (A.23)$$

Usando la ecuación de la dinámica del error (A.23) y empleando un cambio de coordenadas se tiene que:

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{ref} - x \\ y_{ref} - y \\ \theta_{ref} - \theta \end{pmatrix} \quad (A.24)$$

donde  $e_1, e_2$  son los errores de posición en el marco de referencia del robot móvil,  $e_3$  es el error en orientación y  $q_r = [x_{ref}, y_{ref}, \theta_{ref}]^T$  es la trayectoria de referencia.

Las ecuaciones de error de seguimiento son obtenidas derivando (A.24) con respecto al tiempo e introduciendo el cambio en las entradas:

$$\begin{aligned}u_d &= -v - v_{ref} \cos(e_3) \\ u_i &= \omega_{ref} - \omega\end{aligned}\quad (A.25)$$

por lo que se obtiene:

$$\dot{e} = \begin{pmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} e + \begin{pmatrix} 0 \\ \sin(e_3) \\ 0 \end{pmatrix} v_r + \begin{pmatrix} 10 \\ 00 \\ 01 \end{pmatrix} \begin{pmatrix} u_d \\ u_i \end{pmatrix} \quad (A.26)$$

## Control lineal

El control lineal resulta de linealizar el sistema (A.26) alrededor del punto de equilibrio  $e = 0, u = 0$ . Considerando  $e_3$  suficientemente pequeño, podemos emplear el siguiente sistema lineal:

$$\dot{e} = \begin{pmatrix} 0 & \omega_{ref} & 0 \\ -\omega_{ref} & 0 & 0 \\ 0 & 0 & v_{ref} \end{pmatrix} e + \begin{pmatrix} 10 \\ 00 \\ 01 \end{pmatrix} \begin{pmatrix} u_d \\ u_i \end{pmatrix} \quad (\text{A.27})$$

Si  $v_{ref}, \omega_{ref}$  son constantes, se obtiene un sistema lineal cuya matriz de controlabilidad es

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & -\omega_{ref}^2 & \omega_{ref} v_{ref} \\ 0 & 0 & -\omega_{ref} & v_{ref} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (\text{A.28})$$

En este caso, el modelo linealizado es controlable si en (A.28)  $\omega_{ref}$  o  $v_{ref}$  son diferentes de cero.

Se tiene así la estrategia de control:

$$\begin{aligned} u_d &= -k_1 e_1 \\ u_i &= -k_2 \operatorname{sgn}(v_{ref}) e_2 - k_3 e_3 \end{aligned} \quad (\text{A.29})$$

Asignando los polos del sistema en lazo cerrado iguales a las raíces del polinomio característico  $(s + 2\zeta a)(s^2 + 2\zeta a s + a^2) = 0$  con  $\zeta, a$  reales y positivas, las ganancias correspondientes del control son:

$$\begin{aligned} k_1 &= 2\zeta a \\ k_2 &= \frac{a^2 - \omega_{ref}^2}{|v_{ref}|} \\ k_3 &= 2\zeta a \end{aligned} \quad (\text{A.30})$$

Cuando  $\zeta, a$  son constantes, la ganancia se  $k_2$  incrementa y  $v_{ref}$  tiende a cero. Si se consideran ganancias en términos de  $v_{ref}, \omega_{ref}$  con  $a = (v_{ref}^2 + \omega_{ref}^2)^{1/2}$ , las ganancias quedan definidas como sigue:

$$\begin{aligned} k_1 &= 2\zeta(v_{ref}^2 + \omega_{ref}^2)^{1/2} \\ k_2 &= |v_{ref}| \\ k_3 &= 2\zeta(v_{ref}^2 + \omega_{ref}^2)^{1/2}. \end{aligned} \quad (A.31)$$

Así, las ganancias del control (A.31) son definidas en términos de  $v_{ref}, \omega_{ref}$ . En el caso particular de que  $v_{ref} = \omega_{ref} = 0$ , el sistema no es controlable y las acciones de control son cero.

### Control no lineal

Debido a que en el caso particular arriba mencionado el sistema no es controlable, se puede utilizar el control no lineal propuesto por Canudas [33]. Se busca que el sistema tenga estabilidad asintótica y se pueda solucionar el caso cuando  $v_{ref}, \omega_{ref}$  son variantes en el tiempo. El controlador está dado por:

$$\begin{aligned} u_d &= -k_1(v_{ref}, \omega_{ref})e_1 \\ u_i &= -k_2 v_{ref} \frac{\sin(e_3)}{e_3} e_2 - k_3(v_{ref}, \omega_{ref})e_3 \end{aligned} \quad (A.32)$$

donde  $k_2$  es una constante positiva y  $k_1(\bullet)$  y  $k_3(\bullet)$  son funciones continuas estrictamente positivas en  $\mathbb{R} \times \mathbb{R} - (0,0)$ . Considerando la dinámica del error y un punto de equilibrio estable para el sistema dado por  $e = 0$ , se propone la siguiente función candidata de Lyapunov:

$$V(e) = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2} \quad (A.33)$$

Derivando (A.33) con respecto al tiempo  $t$  a lo largo de las trayectorias del sistema se tiene:

$$\dot{V}(e) = k_2 e_1 \dot{e}_1 + k_2 e_2 \dot{e}_2 + e_3 \dot{e}_3$$

$$\dot{V}(e) = k_2 e_1 (u_1 + \omega_{ref} e_2) + k_2 e_2 (v_{ref} \sin(e_3) - \omega_{ref} e_1) + e_3 u_2$$

obteniéndose:

$$\dot{V}(e) = k_2 e_1 u_1 + k_2 e_2 v_{ref} \sin(e_3) + e_3 u_2 \quad (\text{A.34})$$

Sustituyendo (A.32) en (A.34) finalmente se obtiene:

$$\dot{V} = -k_1 k_2 e_1^2 - k_3 e_3^2$$

donde  $\dot{V}$  es negativa definida y garantiza que el sistema sea asintóticamente estable.

## Apéndice B. Lenguaje de programación para implementar la máquina de estados de Ludibot v2 mediante un lenguaje de marcado tipo HTML

Un autómata finito –o máquina de estado finito– es un modelo computacional que realiza cálculos en forma automática sobre una entrada para producir una salida. El comportamiento de Ludibot se basa en un modelo inspirado de una máquina de estado finito, que tiene un conjunto de variables, estados, entradas, transiciones y acciones, así como un estado inicial.

### Sintaxis HTML

La sintaxis que se utiliza para configurar y programar a Ludibot utiliza una etiqueta que encierra más etiquetas y que puede tener uno varios atributos con un valor definido. Cuando no se tiene atributo, se tiene la etiqueta de inicio y de cierre:

```
<Etiqueta></Etiqueta>
```

Cuando se tiene un atributo, la sintaxis es:

```
<Etiqueta Atributo="Valor del atributo"></Etiqueta>
```

Un programa en HTML es una colección de etiquetas, cada una con un nombre y un atributo distinto. Se sigue una estructura arborescente en la cual una etiqueta es el tronco y se introducen dentro de la etiqueta más etiquetas, que son las ramas. Para facilitar su lectura, cada nivel se indenta. Un ejemplo de un tronco con dos ramas donde cada rama tiene dos hojas sería:

```
<!-- Este es un ejemplo de un tronco -->
<Tronco>
  <RamaIzquierda>
    <HojaDerecha>
    </HojaDerecha>
    <HojaIzquierda>
    </HojaIzquierda>
  </RamaIzquierda>
  <RamaDerecha>
    <HojaDerecha>
    </HojaDerecha>
    <HojaIzquierda>
    </HojaIzquierda>
  </RamaDerecha>
</Tronco>
```

Este tipo de programación se usa en las páginas web en internet. Los comentarios se encierran entre `<!-- Este es un comentario-->` y sirven para entender que hace un programa; generalmente contienen la documentación del código fuente. En el lenguaje de Ludibot, las etiquetas son fijas y se usan para definir los elementos de la máquina de estados.

Es útil utilizar un editor de texto que reconozca la sintaxis de marcado, como por ejemplo el editor de Visual Studio. Este tipo de editores permite insertar etiquetas más rápidamente y establece colores distintos de las etiquetas, atributos y los valores de los atributos lo que permite una lectura más sencilla y evita errores.

## Subdirectorios o carpetas

Existen varias carpetas o subdirectorios en el disco duro de la computadora que son necesarios para la ejecución de Ludibot v2. Estas carpetas tienen un nombre específico y se localizan dentro de la carpeta en donde se copie el programa LudibotV2.exe. Esta carpeta de programa puede llevar cualquier nombre que se escoja.

Los archivos necesarios para la ejecución del programa LudibotV2.exe se deben de encontrar dentro de la carpeta de programa son:

- LudibotV2.exe
- LudibotV2.exe.config
- LudibotV2.pdb
- Microsoft.Kinect.dll
- Microsof.Kinect.VisualGestureBuilder.dll
- Microsoft.Kinect.VisualGestureBuilder.xml
- Microsoft.Kinect.xml

Las carpetas que debe de contener la carpeta donde se copió el programa LudibotV2.exe son cinco, a saber:

- 1) **Audios.** Es la carpeta que contiene los archivos de audio *.wav* útiles para incluir acciones sobre audios en la máquina de estados.
- 2) **Database.** Es la carpeta que contiene los archivos *.gbd* de las bases de datos para reconocer gestos.
- 3) **Imágenes.** Es la carpeta que contiene los archivos tipo *.PNG* de imágenes que el usuario copie. Pueden ser archivos que se quieren desplegar; por ejemplo, cuando un gesto se



detecte aparecerá la imagen de del gesto. Es importante notar que los nombres que lleven las imágenes de gestos deben coincidir con los gestos definidos dentro de las bases de datos .gbd.

- 4) **Vgbtechs**. Esta carpeta contiene archivos de librerías para la detección de gestos por Kinect.
- 5) **XAML**. Esta carpeta que contiene los archivos *.xaml* que definen la máquina de estados, así como el archivo de configuración del programa LudibotV2.exe.

## Configuración

El programa en donde se implementa la máquina de estados LudibotV2.exe requiere de un archivo de configuración para poder funcionar. El archivo de configuración se llama ConfiguracionLudibot.xaml y se encuentra dentro de la carpeta XAML.

ConfiguracionLudibot.xaml se usa para seleccionar dónde se encuentra el programa LudibotV2.exe. Selecciona un archivo fuente con la máquina de estados que ejecutará LudibotV2.exe. La posibilidad de seleccionar un archivo fuente permite cambiar rápidamente la aplicación de Ludibot.

Para configurar el programa, es necesario especificar la ruta para acceder al programa LudibotV2.exe y seleccionar el archivo *.xaml*, mismo que contiene la definición de una máquina de estados. Este archivo se debe encontrar en la carpeta XAML.

En el ejemplo que aparece abajo se muestra la configuración de Ludibot, cuya ruta está definida por la etiqueta Path y tres archivos fuentes *.xaml*. Se escoge el archivo que tiene el número 1. Es importante notar que siempre existe una etiqueta de inicio `<Etiqueta>` y una de fin `</Etiqueta>`. Los atributos definen tanto los valores de la ruta como el nombre de los archivos y su número.

```
<ConfigLudibot>
  <!-- Localización del programa y directorios para Ludibot-->
  <! -Esta es la carpeta en donde se copió el programa LudibotV2.exe -->
  <Path Directorio="E:\LudibotV2\"></Path>
  <!-- Se selecciona el número del archivo en Seleccion -->
  <Archivos Seleccion="1">
    <Archivo No ="1" Nombre="Menu_con_Posicionar_y_Sigueme.xaml"></Archivo>
    <Archivo No ="2" Nombre="Manos_Arriba_Abajo.xaml"></Archivo>
    <Archivo No ="3" Nombre="Aprende_A_Controlarme.xaml"></Archivo>
  </Archivos>
```

```
</ConfigLudibot>
```

Aquí se especifica el directorio donde se localiza el programa entre los marcadores `<Path></Path>` y el valor del atributo `Directorio` define la ruta:

```
Directorio="E:\LudibotV2\"
```

Para seleccionar un archivo, se especifica su número en el marcador `<Archivos Seleccion="x">`, en donde  $x$  es el archivo seleccionado. Después aparecen los marcadores que especifican el número de archivo y el nombre del archivo con terminación `.xaml`.

## Variables en el lenguaje

Las variables pueden ser de varios tipos: de posición (x,y,z), booleana, reales, de audio, UDPVar y de gesto. Las variables de posición (x,y,z) guardan posiciones de una articulación de Kinect. Las variables de audio guardan audios que luego se reproducen. Las variables UDPVar permiten establecer una comunicación para enviar una variable vía el protocolo UDP (*User Datagram Protocol*). Las variables de gesto guardan un gesto y sirven para comparar gestos. Las variables booleanas y reales sirven para poder guardar el resultado de una operación de asignación o de comparación. Se definen las operaciones más usuales de comparación (`>`, `<`, `<=`, `>=`, `==`) y de asignación (incrementar, decrementar, asignar un valor fijo o de una variable). Como Ludibot implementa una máquina de estados, a las operaciones de variables les llamamos acciones.

## Variables

Las variables se definen dentro de la etiqueta `Variables`. Es necesario definir cuántas variables se usan con el atributo `No_Variables` :

```
<Variables No_Variables="x">  
</Variables>
```

### Variable

Una variable se define con la etiqueta `Variable`. Se tiene que especificar su nombre y su tipo. También se le puede asignar, de manera opcional, un valor inicial:

```
<Variable Nombre="Nombre" Tipo="Tipo" Valor_Inicial="Valor_Inicial"></Variable>
```

Los tipos de variable válidos son booleana, real, de posición XYZ, de gesto o de audio.

### **Variables de posición XYZ**

Este tipo de variable incluye tres componentes: la posición *X*, la posición *Y* y la posición *Z* (profundidad, distancia al sensor Kinect v2). Los valores de las componentes son números reales, que representan la posición en metros, medida por el sensor Kinect v2. Si se agrega un valor inicial, se tiene que especificar los valores iniciales *X*, *Y* y *Z*, separados por comas:

```
ValorInical="25,35,50"
```

Sin un valor inicial:

```
<Variable Nombre="Pos_Mano_Derecha" Tipo="PosicionXYZ"></Variable>
```

Con un valor inicial:

```
<Variable Nombre="Contador" Tipo="Real" Valor_Inicial="0"> </Variable>
```

#### *Variable predefinida Error de posicionamiento de Loomo (ErrPosLoomo)*

Esta variable es de tipo posición XYZ. En la componente *Z* se guarda el error de posicionamiento del ángulo entre el robot y la posición deseada. Gracias a las componentes *X* y *Y* se indica la posición en el plano del robot, expresada en metros. Loomo actualiza constantemente el valor de esta variable. Se puede usar una comparación con un valor constante para poder activar una transición por medio del valor de una variable booleana.

### **Variables booleanas**

Este tipo de variable puede tomar dos valores, falso o verdadero. Se usa para comparar variables, pues utilizando operaciones de comparación se puede establecer su valor. Cuando se usa en una transición, la transición se efectúa cuando la variable tiene como valor “verdadero”.

```
<Variable Nombre="ContMayorA200" Tipo="Booleana" ValorInicial="false"></Variable>
```

## Variables reales

Este tipo de variables sirve para guardar valores reales. Se usa para almacenar valores tales como contadores o valores constantes.

```
<Variable Nombre="Constante" Tipo="Real" ValorInical="50.0"></Variable>
```

## Variables de audio

Estas variables guardan un audio, grabado en un archivo .wav.

```
<Variable Nombre="1" Tipo="Audio" Valor_Inicial="Cabeza.wav"></Variable>
```

## Variables UDPVar

Estas variables sirven para enviar una variable. Se tiene que especificar el nombre y la dirección IP y el puerto que se utiliza. Debe existir un programa de comunicación que pueda leer la información de la variable que se envía.

```
<Variable Nombre="SM_UDP_Link" Tipo="UDPVar" IPAdd="127.0.0.1" Puerto="18000"></Variable>
```

## Variables de gesto

Estas variables pueden guardar un gesto detectado por Kinect.

```
<Variable Nombre="Var_Gesto_Detectado" Tipo="Gesto" ></Variable>
```

## Máquina de estados

Los componentes por definir para la máquina de estados son los estados, las transiciones, las entradas o gestos y las acciones multimodales. Los estados sirven para designar una situación o circunstancia de Ludibot. Una transición permite cambiar de estado cuando se tiene una entrada o cuando una variable booleana es verdadera.

Las entradas para Ludibot son gestos, aunque no se descarta posteriormente incluir comandos de voz. Los gestos se asocian a un estado. Es necesario especificar la lista de los

gestos que se pueden detectar en un estado. Por cada nombre de estado que se define, debe de generarse un archivo .gdb con ese nombre. Los archivos de bases de datos .gdb se utilizan para detectar gestos de Kinect. Los nombres de los gestos que se pueden detectar en un estado deben coincidir con aquellos que se definieron durante el aprendizaje de la máquina de gestos (véase apéndice C). La interfaz despliega los gestos que se pueden detectar. Es necesario incluir una imagen .png para cada gesto, con el fin de que la interfaz pueda mostrar los gestos detectables. Además, la imagen se usa también para desplegar el gesto que se detecta.

Las acciones multimodales van desde mover al robot, desplegar una imagen o reproducir audios hasta realizar animaciones de Cozmo y Loomo. Las acciones se pueden realizar ya sea en una transición o en un estado. Cuando se efectúan en una transición, solamente se ejecutan cuando se activa la transición. Cuando se definen en un estado, se repetirán mientras la máquina de estado se encuentre en el estado donde se definieron las acciones multimodales. Las acciones se ejecutan secuencialmente según el orden con el que se definen en el archivo .xaml.

Se escogió un lenguaje de marcado para poder especificar el comportamiento de la máquina de estados de Ludibot. La aplicación tiene un analizador léxico-sintáctico del marcaje que permite la construcción de una máquina de estados. El analizador de Ludibot reconoce un conjunto cerrado de etiquetas y de atributos. Todos los estados definidos se consideran como estados finales, así, siempre Ludibot estará en uno de los estados definidos. Se tiene que escoger un estado como el estado inicial.

Todos los elementos de la máquina de estado deben de encontrarse entre la etiqueta de inicio `<Maquina_Estados>` y la etiqueta de fin `</Maquina_Estados>`:

```
<Maquina_Estados No_Repeticiones_Gesto_Transicion ="15">  
</Maquina_Estados>
```

El atributo `No_Repeticiones_Gesto_Transicion` permite configurar el número de veces que se tiene que detectar un gesto para que pueda considerarse como la entrada de una transición, permitiendo que el usuario escoja un gesto para transitar a otro estado.

Solamente el gesto *Stop* se considera sin repeticiones. Esto permite activar la transición asociada inmediatamente.

## Estados

A los estados se les asigna un nombre como puede ser “Inicio” o “Menú”. Permiten establecer temporalmente la situación o modo de Ludibot. Todos los estados deben encontrarse entre la etiqueta de inicio `<Estados>` y la etiqueta de fin `</Estados>`. Es necesario definir el número de estados. Cuando se detecta un gesto, se calcula un índice de confianza de la detección, que va de 0 a 100. El valor `Min_Confianza_de_Deteccion` establece el valor a partir del cual un gesto se considera detectado. Si se usa un valor muy bajo se puede confundir con otro gesto. Si el valor es muy elevado puede ser difícil que el robot identifique el gesto, pues sólo hasta que alcance el valor específico de confianza detectará el gesto. Generalmente, se usa una confianza de detección superior a 70%.

```
<Estados No_Estados="4" Min_Confianza_de_Deteccion="75">
</Estados>
```

### Estado

En la definición de un estado, se tiene que establecer su nombre y su número. Dentro de la etiqueta Estado se pueden definir gestos y acciones que aparecerán dentro de la etiqueta. Cuando se definen acciones, éstas se repetirán mientras la máquina de estados se encuentre en ese estado.

```
<Estado Nombre="AprendeAControlarme" Numero="1">
</Estado>
```

Al definir un estado, se debe incluir un archivo `.gdb` para detectar gestos en el estado. El nombre del archivo de la base de datos de los gestos debe coincidir con el nombre del estado seguido de `.gdb`. Si el nombre del estado es `AprendeAControlarme`, el archivo de la base de datos debe de tener el nombre `AprendeAControlarme.gdb`. El valor del atributo Nombre se especifica entre comillas.

### Estado inicial

Se escoge alguno de los estados para que sea el estado inicial. En ese estado se encontrará la máquina de estados al iniciar su ejecución. Se utiliza la etiqueta `Estado_Inicial`.

```
<Estado_Inicial Nombre="Reproduce_Parte_Aleatoriamente"></Estado_Inicial>
```

## Gestos

```
<Gestos No_Gestos="4">
```

```
</Gestos>
```

Define los gestos y el número de gestos que se puede detectar en el estado. Los gestos se definen dentro de las etiquetas `<Gestos>` y `</Gestos>`.

### Gesto

```
<Gesto Nombre="Posicionar"></Gesto>
```

Define un gesto y su nombre. El nombre del gesto debe de ser el mismo definido dentro de la base de datos `.gdb`. De lo contrario el gesto no será reconocido.

## Transiciones

Las transiciones sirven para cambiar de estado. Las transiciones deben aparecer dentro de la etiqueta de inicio `<Transiciones>` y la etiqueta de fin `</Transiciones>`. Es necesario definir el número de transiciones.

```
<Transiciones No_Transiciones="4">
```

```
</Transiciones>
```

### Transición

Una transición se define entre dos estados. Se tiene que definir el estado donde se encuentra la máquina de estado, el gesto que activa la transición y el estado al cual cambia la máquina de estados. Es posible definir acciones durante la transición. Estas acciones solamente se ejecutan cuando se realiza la transición.

```
<Transicion Estado_Inicio="Menu" Gesto="Posicionar" EstadoSiguiete="Posicionar">
```

Una transición también se puede activar cuando una variable booleana es verdadera. En este caso, se especifica el nombre de la variable correspondiente:

```
<Transicion Estado_Inicio="Reproduce_Parte_Aleatoriamente" VariableBoolTrue="Transita" EstadoSiguiete="Detectar_Gesto">
```

## Acciones

Define acciones y el número de acciones. Se puede desplegar un texto en la barra inferior de la interfaz gráfica (UI) cuando se ejecutan las acciones.

```
<Acciones No_Acciones="1" TextoBarra="Ejecutando acciones" >
</Acciones>
```

### Acción

Una acción se define con la etiqueta `<Accion>` `</Accion>`. Se tiene que definir el tipo. Según el tipo de acción, es posible determinar cierto número de atributos. Existen acciones con variables, con audios, con Kinect, con Cozmo y con Loomo.

### Acciones con variables

#### $A=B$

```
<Accion Tipo="A=B" VarA="Contador" VarB="Cero"></Accion>
```

Asignar a la variable A la variable B.

#### $R++$

```
<Accion Tipo="R++" VarResultado="Contador"></Accion>
```

Incrementa la variable R,  $R=R+1$ .

#### $R--$

```
<Accion Tipo="R--" VarResultado="Contador"></Accion>
```

Decrementa la variable R,  $R=R-1$

#### $A > B$

```
<Accion Tipo="A>B" VarResultado="ContMayorA200" VarA="Contador"
VarB="Constante"></Accion>
```

Compara si la variable A es mayor que la variable B y asigna el resultado a una variable booleana.



**$A < B$**

```
<Accion Tipo ="A<B" VarResultado="Menor" VarA="Contador" VarB="Constante"></Accion>
```

Compara si la variable A es menor que la variable B y asigna el resultado a una variable booleana.

**$A \geq B$**

```
<Accion Tipo="A>=B" VarResultado="Mayor_Igual" VarA="Contador" VarB="Constante"></Accion>
```

Compara si la variable A es mayor o igual a la variable B y asigna el resultado a la variable booleana.

**$R = A \leq B$**

```
<Accion Tipo="A<=B" VarResultado="Menor_Igual" VarA="Contador" VarB="Constante"></Accion>
```

Compara si la variable A es menor o igual a la variable B y asigna el resultado a la variable booleana.

**$R = A == B$**

```
<Accion Tipo="A==B" VarResultado="Igual" VarA="Contador" VarB="Constante"></Accion>
```

Compara si la variable A es igual a la variable B y asigna el resultado a la variable booleana.

**$R = A + B$**

```
<Accion Tipo="A+B" VarResultado ="PosDeseada" VarA="PosMano" VarB="PosTorso"></Accion>
```

Suma la variable A y la variable B y asigna el resultado.

**$R = A - B$**

```
<Accion Tipo="A-B" VarResultado ="PosDeseada" VarA="PosMano" VarB="PosTorso"></Accion>
```

Resta la variable A y la variable B y asigna el resultado.

**$R = A * B$**

```
<Accion Tipo="A*B" VarResultado ="PosDeseada" VarA="PosMano" VarB="PosTorso"></Accion>
```

Multiplica la variable A y la variable B y asigna el resultado.

### ***R=A/B***

```
<Accion Tipo="A-B" VarResultado ="PosDeseada" VarA="PosMano" VarB="PosTorso"></Accion>
```

Divide la variable A y la variable B y asigna el resultado.

### ***R=A||B (o lógico)***

```
<Accion Tipo=" A||B " VarResultado ="VarBooleanaR" VarA="VarBooleanaA"
VarB="VarBooleanaB"></Accion>
```

Calcula el valor del “o” lógico dependiendo del valor de la variable booleana A y de la variable booleana B y asigna el resultado.

### ***R=A&&B (y lógico)***

```
<Accion Tipo=" A&&B " VarResultado ="VarBooleanaR" VarA="VarBooleanaA"
VarB="VarBooleanaB"></Accion>
```

Calcula el valor del “y” lógico dependiendo del valor de la variable booleana A y de la variable booleana B y asigna el resultado.

### ***R=Rand***

```
<Accion Tipo="Rand" VarResult="Numero_Aleatorio" RandMin="1" RandMax="9"
```

Genera un número aleatorio entre *RandMin* y *RandMax* y lo guarda en *VarResult*.

## **Acciones con audio**

### ***Reproducir audio***

```
<Accion Tipo="Reproducir_Audio" VariableAudio="Variable"></Accion>
```

Se reproduce el audio contenido en la variable de audio.

### ***Reproducir audio número***

```
<Accion Tipo="Reproducir_Audio_Numero" Variable="Variable_Real"></Accion>
```

Reproduce la variable de audio cuyo nombre corresponde al valor dado a la variable. Por ejemplo, en el caso de *Variable\_Real=2*, se reproduce el audio cuyo nombre es “2”. Se usa esta acción para poder reproducir una lista de variables de audio de manera aleatoria, por ejemplo.

### ***R=AGesto==BAudio***

```
<Accion Tipo="R=AGesto==BAudio" VarResult="Parte_Tocada" VarA="Var_Gesto_Detectado"
VarB="VarAudio"></Accion>
```

Compara el nombre del gesto detectado en la variable *VarA* con el nombre de la variable de audio en la variable *VarB*. Si son iguales, asigna el valor “True” a la variable booleana *VarResult*. Se usa esta acción para el juego *De la tête aux pieds*, durante el cual se compara si el usuario tocó la parte del cuerpo indicada en la variable de audio.

### **Acciones Kinect**

#### ***Capturar la posición del esqueleto con Kinect***

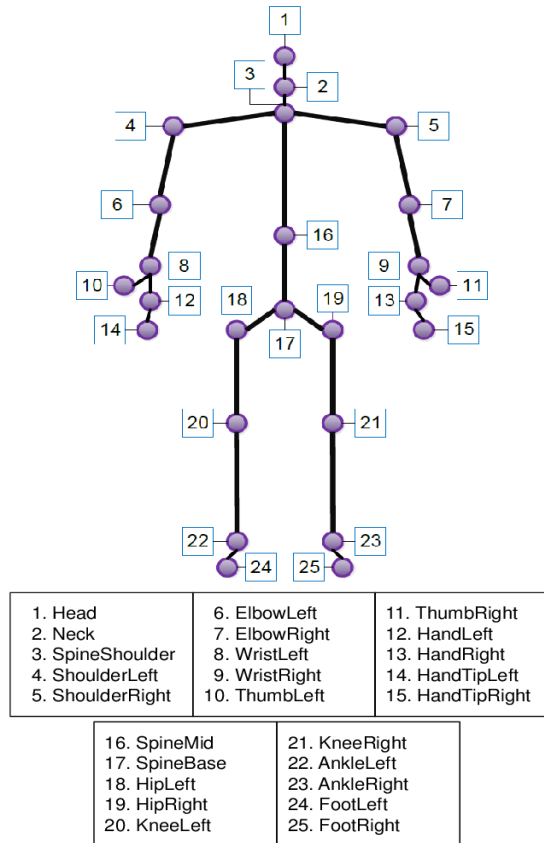
```
<Accion Tipo="Captura_Pos_Esqueleto_Kinect" Articulacion="SpineShoulder"
VariablePos="PosTorso"></Accion>
```

Captura la posición de una articulación del esqueleto de Kinect v2. El valor del atributo articulación debe ser uno de los 25 previamente definidos por Kinect v2 (véase Figura B.1). El valor capturado se asignará a las variables definidas. Las variables por capturar son reales. Los valores de las tres variables se despliegan en la barra inferior de la interfaz.

#### ***Capturar la posición de la mano derecha y multiplicar por z***

```
<Accion Tipo="Captura_Pos_Mano_Derecha_Multiplicada_Por_Z" VariablePos="Variable"
</Accion>
```

Con esta acción se puede capturar una posición en el plano *x,y* del sensor Kinect v2 y escalarla para poder lograr valores más grandes o más pequeños, como cuando se requiere posicionar el robot en una posición deseada. Captura la posición (*x,y*) de la mano derecha, multiplica por la distancia de profundidad y la asigna a la variable de posición definida. La componente *x* de la variable es igual al producto de la posición *x* de la mano derecha multiplicada por la posición *z* (profundidad) de la mano derecha. La componente *y* de la variable es igual al producto de la posición *y* de la mano derecha por la posición *z* de la mano derecha.



**Figura B.1. Atributos de las articulaciones del cuerpo dados por Kinect v2**

## Acciones con Cozmo

### *Baja a Cozmo*

```
<Accion Tipo="Baja_Cozmo" Mensaje="baja a Cozmo" ></Accion>
```

Baja la plataforma donde se encuentra Cozmo.

### *Sube a Cozmo*

```
<Accion Tipo="Sube_Cozmo" Mensaje="sube a Cozmo" ></Accion>Stop a Cozmo
```

Sube la plataforma donde se encuentra Cozmo.

### *Stop Cozmo*

```
<Accion Tipo="Stop_Cozmo" Mensaje="stop Cozmo" ></Accion>
```

Detiene el robot Cozmo.

### ***Animación de Cozmo***

```
<Accion Tipo="Animacion_Cozmo" NombreAnimacion="Nombre_de_la_Animacion" ></Accion>
```

Ejecuta la animación especificada.

### ***Decir un texto con Cozmo***

```
<Accion Tipo="Say_Text_Cozmo" Texto="Hola, soy Cozmo" ></Accion> >
```

Cozmo pronuncia el texto.

### ***Accionar los motores de Cozmo***

```
<Accion Tipo="Vels_Motores_Cozmo" VelDer="VarVelDerecha" VelIzq="VarVelIzq" ></Accion>
```

Establece las velocidades de los motores de Cozmo. Se establece la velocidad de los motores a partir del valor que tengan las variables especificadas. Las variables son de tipo real.

### ***Ejecución de un script Python***

```
<Accion Tipo="Ejecuta_Script" Script="Ejemplo.py" ></Accion>
```

Ejecuta un script Python que puede incluir llamadas a la rutina del SDK de Cozmo.

## **Acciones con Loomo**

### ***Loomo en función Seguir***

```
Accion Tipo="Seguir_Loomo" Distancia="1.5" Seguir="True" ></Accion>
```

Activa la función de seguir a la distancia especificada. Si el atributo Seguir= “true”, se activa el seguimiento. Si el atributo Seguir=“false”, se desactiva el seguimiento.

### ***Posicionar a Loomo***

```
<Accion Tipo="Posicionar_Loomo" VariablePosicion="VariableXYZ" ></Accion> >
```

Se establece como posición deseada la componente  $x$  y la componente  $y$  de la variable establecida. Se mueve el robot de la posición actual a la posición contenida en la variable de posición.

### *Animar cabeza de Loomo*

```
<Accion Tipo="Animar_Cabeza_Loomo" Animacion="Nombre_de_la_Animacion" ></Accion>
```

Ejecuta la animación especificada.

### **Otras acciones**

#### *Imprimir*

```
<Accion Tipo="Imprime" Mensaje="Imprime este mensaje"></Accion>
```

```
<Accion Tipo="Imprime" Mensaje="Pos mano derecha x,y,z"=Variable ="PosMano"></Accion>
```

Imprime un mensaje en la parte inferior de la interfaz gráfica. Si se especifica una variable, se imprime el valor de la variable.

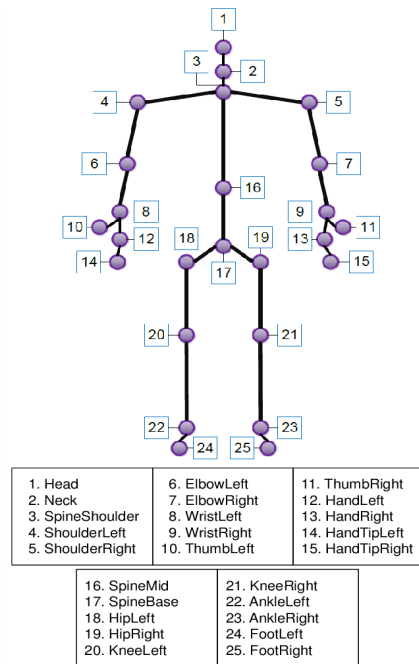
#### *Pausar*

```
<Accion Tipo="Pausar" TiempoMiliSegundos="3600"></Accion>
```

Se pausa la actividad de Ludibot durante el tiempo especificado.

## Apéndice C. Entrenamiento de gestos usando aprendizaje-máquina mediante Kinect Studio y Visual Gesture Builder

El sensor Kinect V2 recaba información que permite detectar gestos con un alto grado de precisión. Para capturar la posición  $x,y,z$  en su sistema, el sensor coordena los 25 puntos del cuerpo humano. Por ejemplo, utiliza la posición de la base del cuello (la columna vertebral a la altura del hombro o *SpineShoulder*, véase punto 3 de la Figura C.1).



**Figura C.1. Articulaciones del cuerpo dadas por Kinect v2**

El sensor identifica asimismo el estado de las manos, abiertas o cerradas. El sensor define que una mano se encuentra levantada (“mano arriba”) si la posición vertical de la componente y de la mano (punto 12 de la Figura C.1) es mayor a la componente y del punto *SpineShoulder* (punto 3 de la Figura C.1); por el contrario, determinará que la mano está abajo si la posición de la componente y de la mano es menor a la componente y del punto *SpineShoulder*. Se puede combinar el gesto “mano arriba” con el estado de la mano, abierta o cerrada, para definir gestos más complejos, por ejemplo “mano derecha arriba cerrada”.

Este enfoque, basado únicamente en la identificación del punto del esqueleto y en el estado de las manos, se utilizó en la primera versión de Ludibot. Sin embargo, el catálogo de

gestos obtenido a partir de esa información resultaba relativamente limitado. Además, era difícil determinar las características para definir más eficazmente un gesto estandarizado, dada la diversidad anatómica de los usuarios potenciales; y no estaba prevista la detección de gestos continuos –esto es, gestos en movimiento, con un inicio y un fin distintos, a diferencia de los gestos estáticos.

En Ludibot v2, se decidió explorar otra opción, a partir de la gama de herramientas que ofrece el SDK de Kinect, gracias a las cuales es posible emplear el aprendizaje-máquina para detectar un gesto con mucha mayor precisión. En efecto, dentro de las características funcionales (*features* en inglés) consideradas, Kinect puede estimar no sólo la posición respecto del esqueleto o el estado abierto o cerrado, sino también el ángulo entre articulaciones, la aceleración o la fuerza de un movimiento, entre otras.

Se parte entonces de la asociación de las posiciones que mide Kinect con un gesto que se realiza. Para lograr dicha asociación, es necesario encontrar las características funcionales que permiten identificar gestos específicos. Con ese fin, se recurre a dos algoritmos de aprendizaje-máquina para determinar el conjunto de características que representan mejor el gesto deseado. Los algoritmos de clasificación o entrenamiento prueban diferentes combinaciones de clasificadores hasta encontrar aquellos que representan mejor el gesto en cuestión. El número de combinaciones de clasificadores puede ser superior a mil; a partir de ellas, SDK selecciona los diez clasificadores más apropiados.

Las dos herramientas que permiten generar las bases de datos para la clasificación de gestos son Kinect Studio [121] y Visual Gesture Builder [123]. Kinect Studio se utiliza para grabar datos de Kinect. Visual Gesture Builder permite tanto crear y etiquetar gestos como entrenar los clasificadores. Para conocer mejor el uso combinado de ambas herramientas, es posible consultar diversos tutoriales, entre ellos “Custom Gestures End to End with Kinect and Visual Gesture Builder” (Channel 9, 2014, *Kinect for Windows Developer Blog*, 1:19’27, <https://channel9.msdn.com/Blogs/k4wdev/Custom-Gestures-End-to-End-with-Kinect-and-Visual-Gesture-Builder>) y “Custom Gestures End to End with Kinect and Visual Gesture Builder (part 2)” <http://channel9.msdn.com/Blogs/k4wdev/Custom-Gestures-End-to-End-with-Kinect-and-Visual-Gesture-Builder-part-2-> (Channel 9, 2014, *Kinect for Windows Developer Blog*, 1:00’23).



## Kinect Studio

Kinect Studio es una aplicación que permite realizar grabaciones de datos de Kinect. A un conjunto de datos que se graban se lo llamaremos aquí un dataclip, con base en la analogía con un videoclip. Se entiende por videoclip una secuencia o flujo de imágenes y sonido que forman un video. En el caso de Kinect, se añade al flujo de imágenes y sonido un flujo de datos acerca de las posiciones de las articulaciones que detecta Kinect.

Los datos que captura Kinect, además de las imágenes a color y el audio, incluyen la profundidad, un video infrarrojo, parámetros del esqueleto o cuerpo (*body*). En esta aplicación se le llama *stream* a un flujo de datos. Las siglas Nui corresponden a interfase natural de usuario o *Natural User Interface* en inglés. El reconocimiento de gestos utiliza la información de las articulaciones para funcionar. Así, se requiere grabar dataclips que como mínimo incluyan los siguientes flujos de datos:

- NUI Body Frame. Posición de las 25 articulaciones del cuerpo que detecta Kinect.
- NUI Depth. Información de profundidad.
- NUI IR. Información obtenida mediante parámetros del video infrarrojo.
- NUI Sensor Telemetry. Medición de distancias.

Kinect genera los datos relativos al esqueleto a partir de información de profundidad y parámetros infrarrojos, que son entonces flujos de datos indispensables para la operación de Ludibot.

Es posible usar otro tipo de flujos de datos, entre ellos NUI Uncompressed Color. Sin embargo, no se recomienda recurrir a flujos no indispensables, ya que recabar esa información multiplica el tamaño de por sí considerable de los archivos.

Para asignar un valor a una posición de las articulaciones, cada articulación es representada con un punto; las articulaciones están unidas por una línea. De esta manera, se hace posible recabar la información de las posiciones que toma el esqueleto.

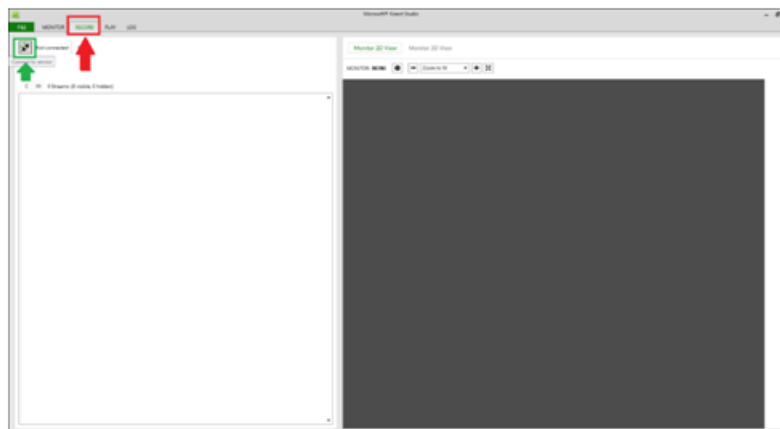
A continuación, presentamos los pasos a seguir para grabar un archivo de datos de Kinect.

A) Abrir Kinect Studio, como se muestra en la siguiente figura:



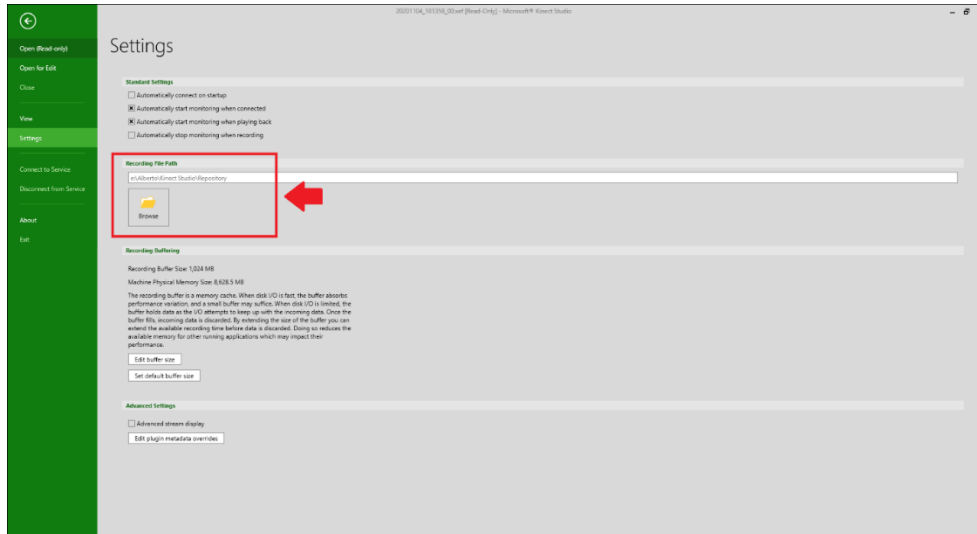
**Figura C.2. Abrir Kinect Studio en Windows**

B) En la barra de menú, seleccionar y hacer clic en Grabar (*Record*) (véase el recuadro rojo de la Figura C.3).



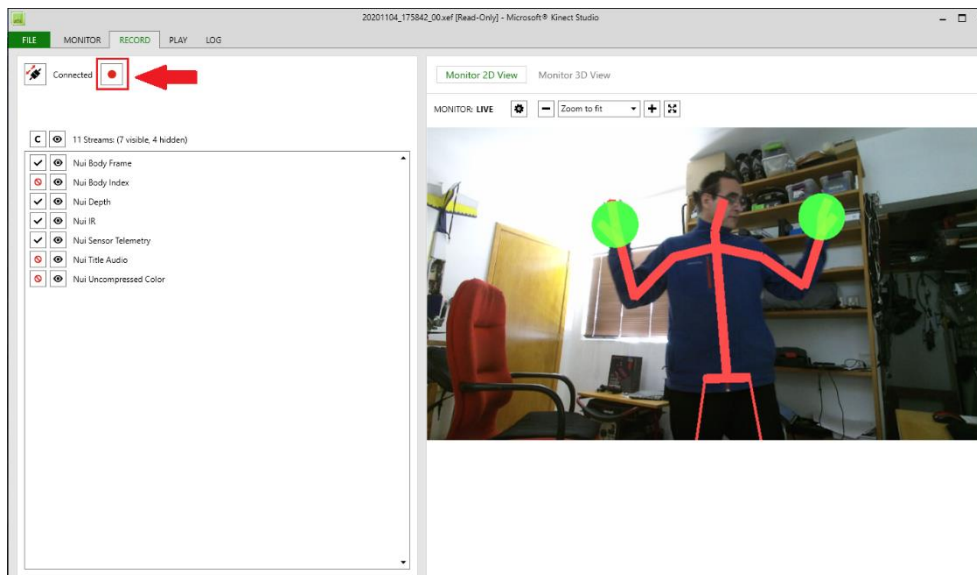
**Figura C.3. Seleccionar el modo Grabar y el botón de conexión en Kinect Studio**

C) Seleccionar *File>>Setting>>Browse* para indicar la unidad y la carpeta donde se desea guardar los videos, como se muestra en la Figura C.4. Por ejemplo, se puede seleccionar la ruta: E:\Usuarios\Alberto\Desktop\Partes\_del\_Cuerpo



**Figura C.4. Ajustes con Kinect Studio**

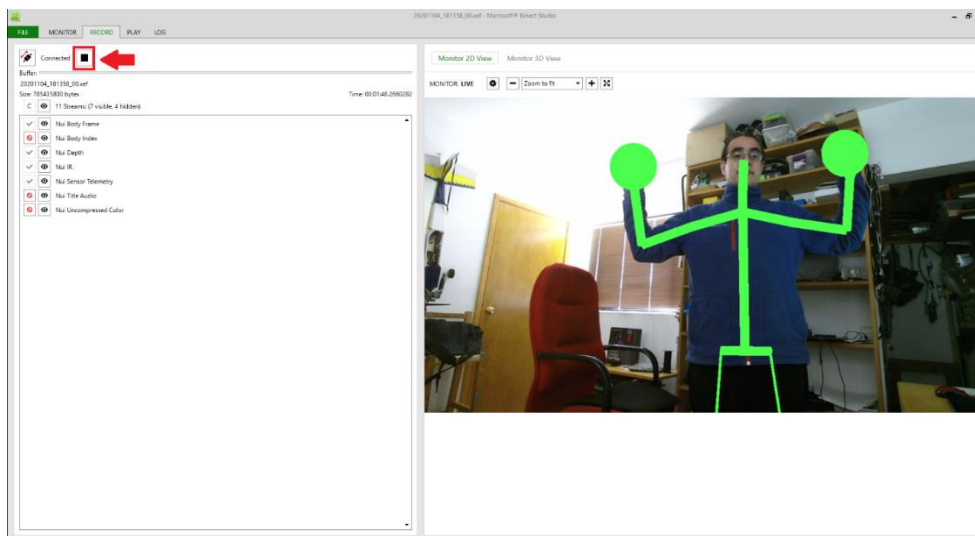
D) Realizar la conexión del Kinect v2. El estado inicial es no conectado (*Not connected*). Se da clic en el icono de conexión (recuadro verde en la Figura C.3). Cambiará al estado conectado (*Connected*), como se muestra en la Figura C.5. Automáticamente, se prenderá la cámara y aparecerán en el panel izquierdo los flujos de datos o *streams* que es posible seleccionar, ya sea para visualizarlos, ya sea para grabar. Seleccionar los flujos NUI Body Frame, NUI Depth, NUI IR y NUI Sensor Telemetry.



**Figura C.5. Estado conectado y selección de flujos de datos con Kinect Studio**

E) Al hacer un clic en el botón de grabación (recuadro rojo en las Figura C.5 y Figura C.6), se activa *Start recording file* y comienza la grabación del dataclip. Aparece en pantalla, arriba

de la lista de flujos, el nombre del archivo que se graba y su tamaño en bytes, tal como se muestra en la Figura C.6.



**Figura C.6. Grabar un Clip con Kinect Studio**

- F) Para realizar el entrenamiento de gestos, es necesario efectuar los gestos cuya detección se desea realizar durante la grabación del dataclip. Por ejemplo, en la Figura C.6, se desea llevar a cabo el entrenamiento para un gesto con las manos arriba. Se recomienda realizar varias veces el gesto por detectar; es posible también hacer otros gestos distintos para entrenar al clasificador a identificar cuándo no se realiza el gesto.
- G) Para finalizar la grabación del dataclip, una vez efectuados los gestos requeridos para el entrenamiento, basta con dar clic en el botón *Stop recording file* que se muestra en el recuadro rojo de la Figura C.6.
- H) Automáticamente, el dataclip será guardado en la carpeta seleccionada durante el paso D.
- I) Se recomienda renombrar el dataclip con una clave o con una denominación que asocie explícitamente el dataclip con el gesto trabajado. Por ejemplo, el nombre del dataclip correspondiente a la Figura C.6, inicialmente “20201104\_181358\_00.xef”, puede ser renombrado para convertirse en “manos\_arriba.xef”.

## Visual Gesture Builder

Visual Gesture Builder (VGB) permite entrenar los clasificadores de datos para realizar la detección de gestos. Construir una base de datos que permita clasificar un gesto requiere tres etapas: definir el gesto, etiquetar el gesto que se desea detectar en los dataclips y, posteriormente, efectuar el entrenamiento del clasificador.

VGB recurre a dos algoritmos de clasificación para entrenar el detector. El algoritmo de aprendizaje-máquina AdaBoost **¡Error! No se encuentra el origen de la referencia.** se utiliza para gestos estáticos, mientras que el algoritmo de regresión de bosques aleatorios o RFR **¡Error! No se encuentra el origen de la referencia.** se aplica a los gestos continuos.

Un gesto discreto tiene solamente dos estados: se realiza o no. Por ende, tiene como resultado un valor booleano verdadero o falso, por ejemplo “Sentado” o “Parado”.

Las variables que se usan en VGB para guardar el resultado de detección se llaman *tags* o etiquetas.

Un gesto continuo tiene un inicio y un fin, como puede ser “Parándose” o “Sentándose”, y tiene por resultado un valor real entre 0 y 1. Un valor de 0.65 significa que el gesto se ha completado en un 65%.

A continuación, se presenta los pasos para utilizar VGB.

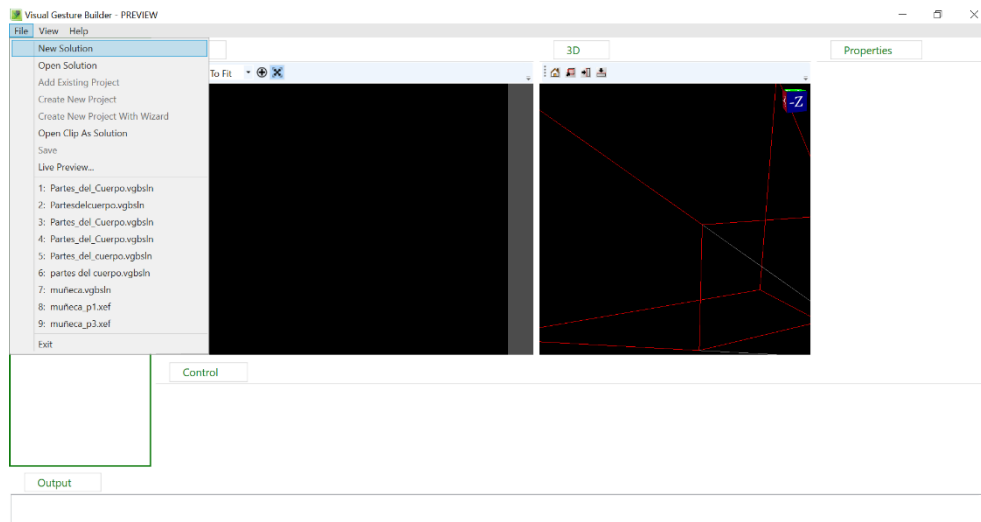
A) Instalar VGB en la computadora.

B) Abrir el programa VGB (Figura C.7).



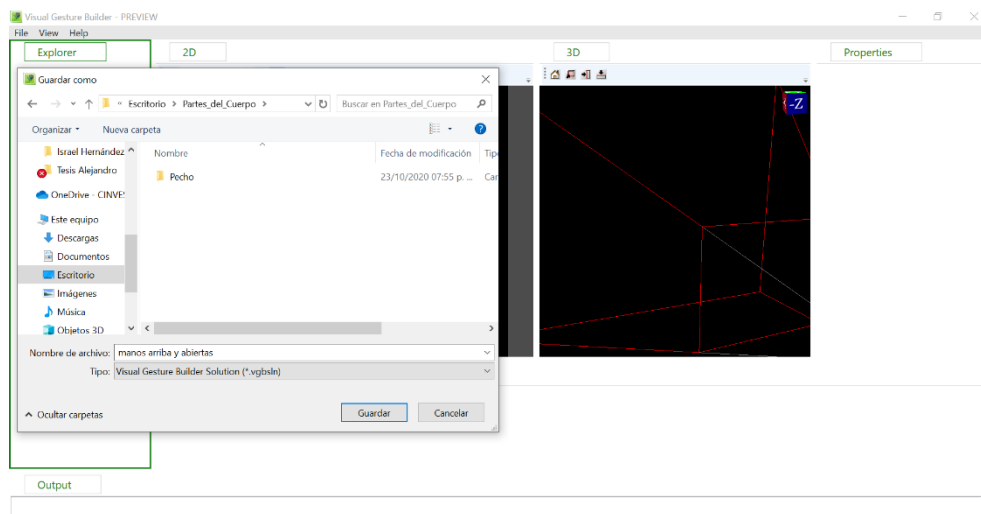
**Figura C.7. Abrir Visual Gesture Builder en Windows**

C) Dar clic en *File* >> *New Solution* para crear una nueva solución de entrenamiento.



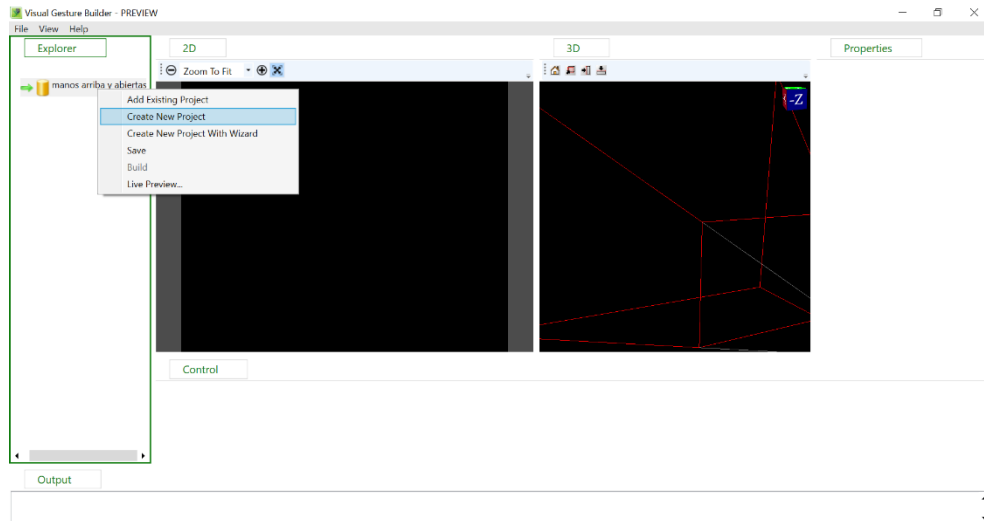
**Figura C.8. Crear una nueva solución de entrenamiento en Visual Gesture Builder**

D) Al crear la nueva solución, es necesario crear una carpeta con un nombre que permita identificar la base de datos de entrenamiento, por ejemplo: “Manos arriba y abiertas”. Se recomienda seleccionar la carpeta donde se guardan los videos de Kinect Studio.



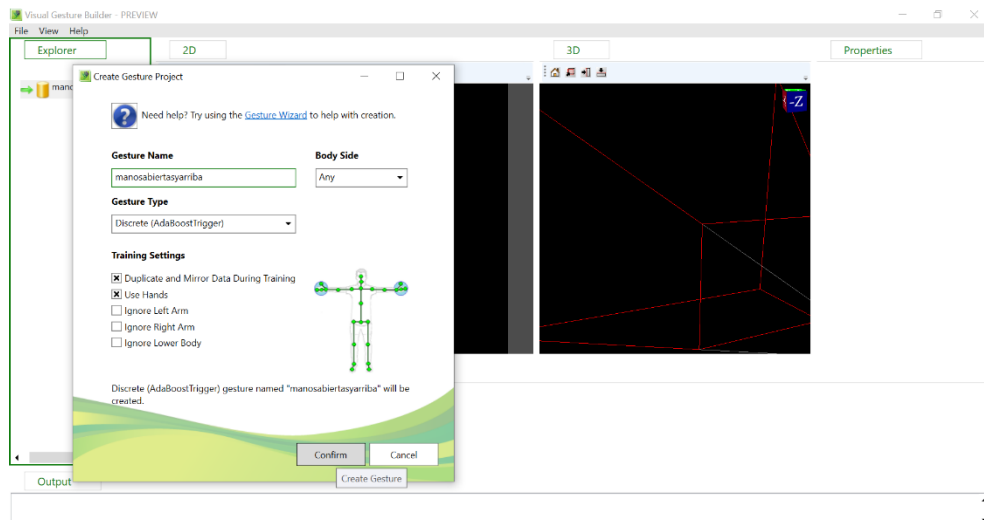
**Figura C.9. Escribir el nombre de la solución de entrenamiento en Visual Gesture Builder**

E) Cuando aparece la nueva solución, se selecciona. Se despliegan las opciones mostradas en la Figura C.10. Se da clic en *Create New Project*.



**Figura C.10. Crear un nuevo proyecto en Visual Gesture Builder**

F) Al crear el nuevo proyecto, aparecerá el menú de la Figura C.11 para ejecutar un asistente de configuración o *Wizard* para definir el gesto y sus parámetros de entrenamiento.



**Figura C.11. Seleccionar el algoritmo AdaBoost para el entrenamiento**

El nombre del gesto que se quiere detectar se define en “Gesture Name”.

A continuación, se establecen las opciones de entrenamiento. Para el lado del cuerpo o *Body Side* se puede usar *Any* para elegir ambas partes del cuerpo cuando el gesto no corresponde al lado izquierdo o derecho.

Para las configuraciones de entrenamiento o *Training Settings* se puede usar *Duplicate and Mirror Data During Training* cuando se requiere utilizar todas las articulaciones durante el entrenamiento.

El uso de manos o *Use Hands* establece si el gesto utiliza el estado de las manos, ya sean cerradas o abiertas.

Es posible excluir partes del cuerpo cuando para el gesto por detectar no es relevante, usando *Ignore Left Arm*, *Ignore Right Arm* y *Ignore Lower Body*.

Es necesario definir si el gesto es continuo o discreto, utilizando *Gesture Type* (véanse las Figura C.11 y Figura C.12).

G) El algoritmo AdaBoostTrigger [119] es un clasificador que produce un resultado binario o discreto y se utiliza para gestos discretos. Para el entrenamiento, es necesario clasificar la posición del esqueleto cuando se realiza el gesto. A esta clasificación se le denomina etiquetado.

En el caso discreto, la etiqueta tiene un valor booleano verdadero o falso, que marca la ocurrencia de un gesto como, por ejemplo, colocar la mano en el pecho. Esta marcación o etiquetado se utiliza para evaluar si se ha producido un gesto o no y para calcular el valor de confianza de la clasificación. Un valor de 50 indica que se tiene 50% de confianza de una detección adecuada del gesto.

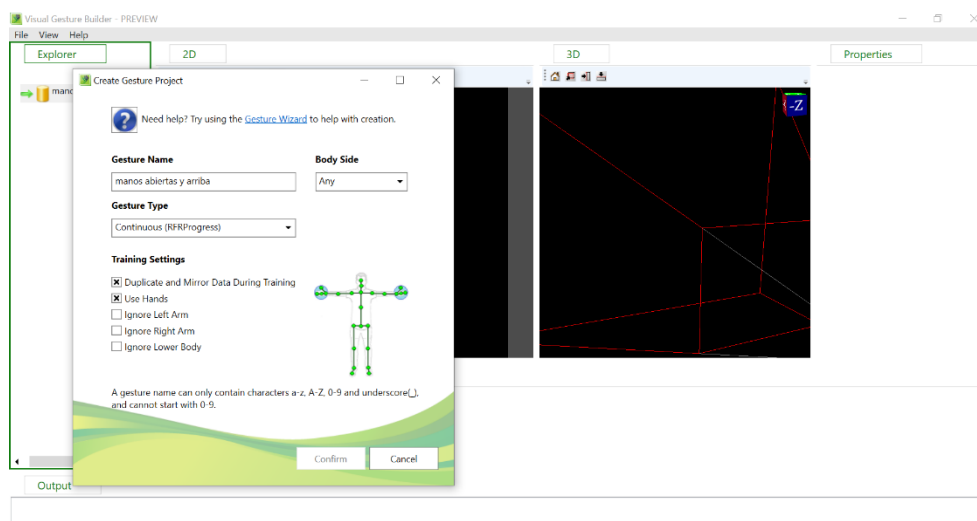
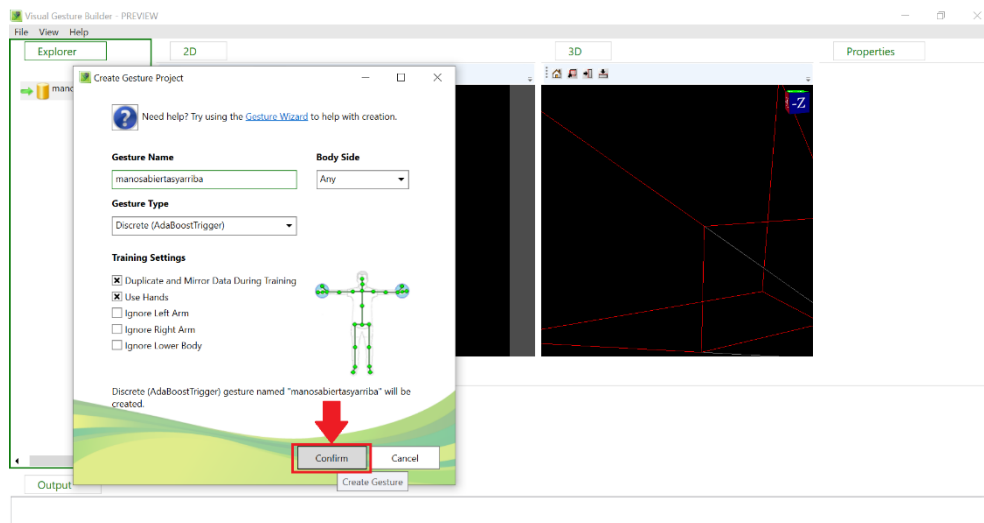


Figura C.12. Seleccionar el algoritmo RFRProgress para el entrenamiento

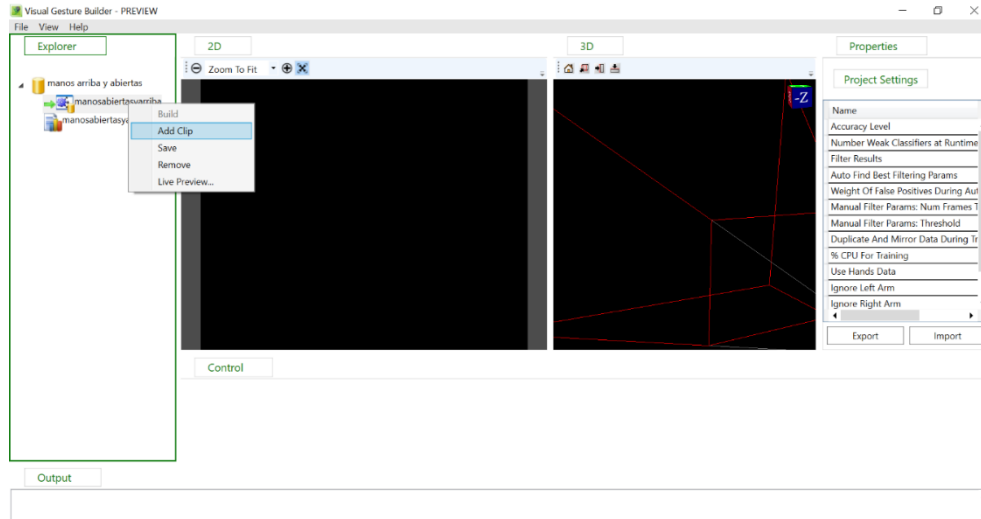


- H) El algoritmo RFR [122] es un algoritmo de clasificación que produce un resultado continuo. Utiliza el algoritmo de aprendizaje automático de regresión aleatoria RFR para determinar el progreso de un gesto realizado por un usuario. En el caso de los gestos continuos, se asocia un valor entre 0 y 1 a las posiciones del esqueleto. El etiquetado asocia al gesto una curva que representa los valores reales tomados por el resultado de la detección del gesto continuo. Así, es posible clasificar un gesto continuo como “Sentándose” o “Parándose”, distinto de los gestos discretos “Sentado” y “Parado”.
- I) Al dar clic en *Confirm* (véase Figura C.13), se crea el gesto con el nombre previamente definido. En este caso, se genera un entrenamiento para un gesto con el algoritmo de aprendizaje AdaBoost.



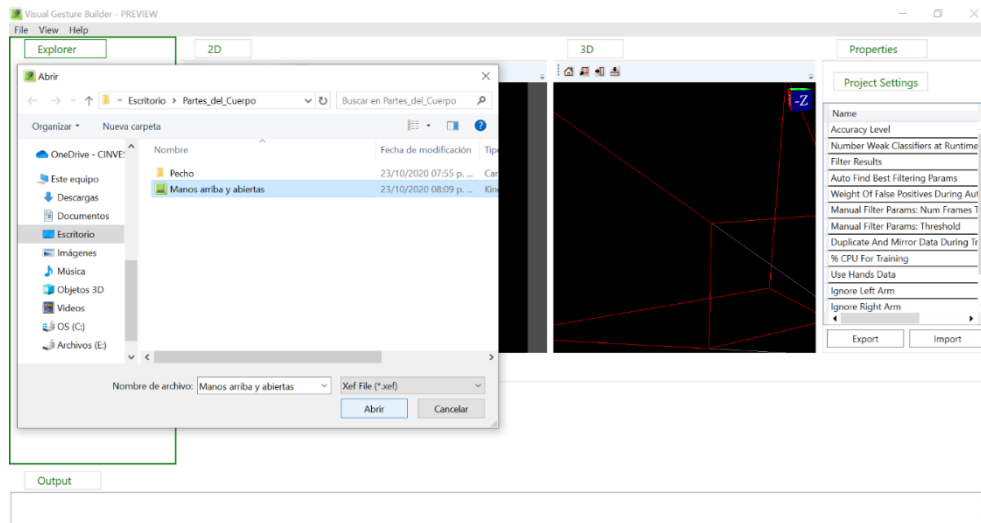
**Figura C.13. Confirmar el gesto creado para el entrenamiento**

- J) Una vez creado el gesto, se generan dos proyectos. En el primero, se agrega el dataclip capturado en Kinect Studio, dando clic en *Add clip* (véase Figura C.14).



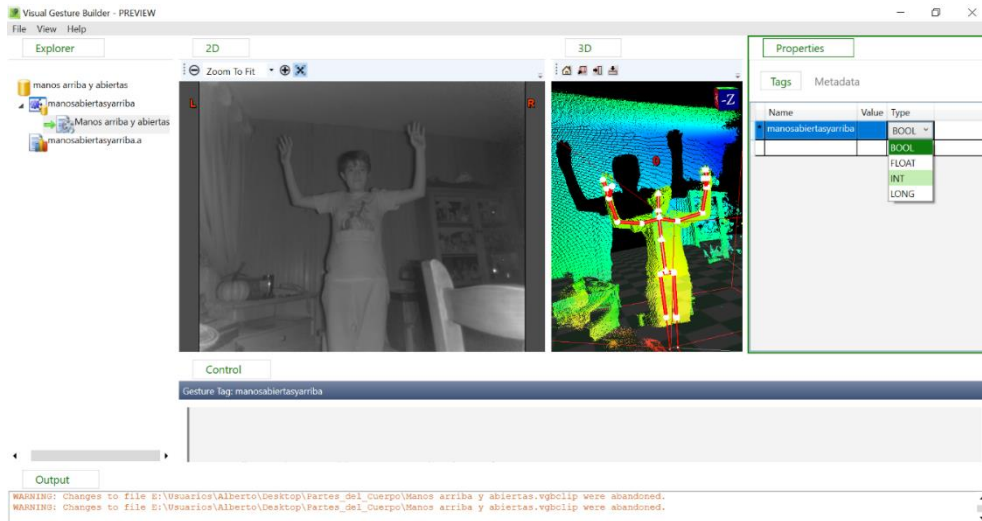
**Figura C.14. Agregar un video capturado en Kinect Studio para el entrenamiento en Visual Gesture Builder**

K) A continuación, se debe agregar el dataclip grabado con Kinect Studio y dar clic en “Abrir” (véase Figura C.15).



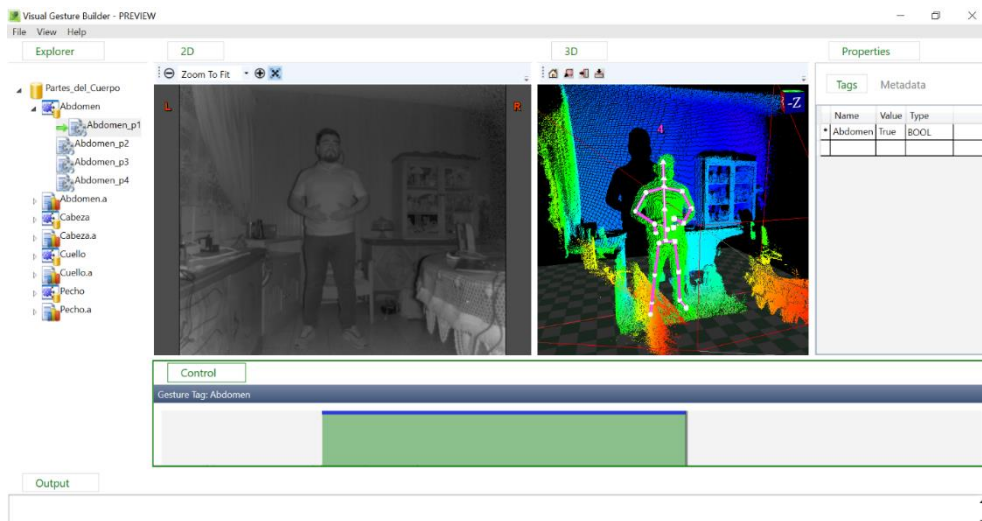
**Figura C.15. Agregar el video capturado en Kinect Studio para el entrenamiento en Visual Gesture Builder**

L) Al abrir el dataclip, se muestran en la parte superior derecha de la pantalla las propiedades (*Properties*). Es importante seleccionar en el tipo (*Type*) la opción (*Bool*). Esta opción permite llevar a cabo el entrenamiento del gesto mediante variables del tipo booleano, es decir, que sólo pueden tener dos valores (verdadero y falso), tal como se muestra en la Figura C.16.

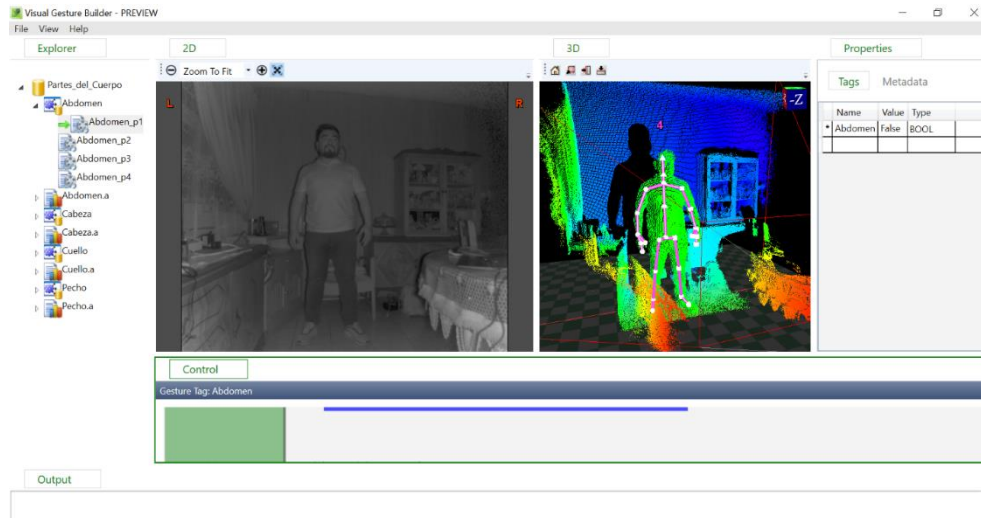


**Figura C.16. Seleccionar la opción *Bool* para el video del entrenamiento en Visual Gesture Builder**

Cabe mencionar que los entrenamientos de gestos pueden ser múltiples. Las Figura C.17 y Figura C.18 y muestra el entrenamiento para cuatro gestos que representan otras tantas partes del cuerpo: abdomen, cabeza, cuello y pecho. La opción verdadera (*true*) permite seleccionar dentro del dataclip capturado en Kinect Studio las secciones en las cuales se realiza el gesto deseado (Figura C.17). Las secciones que no contienen el gesto se designan como falsas (*false*) (figuraFigura C.18).

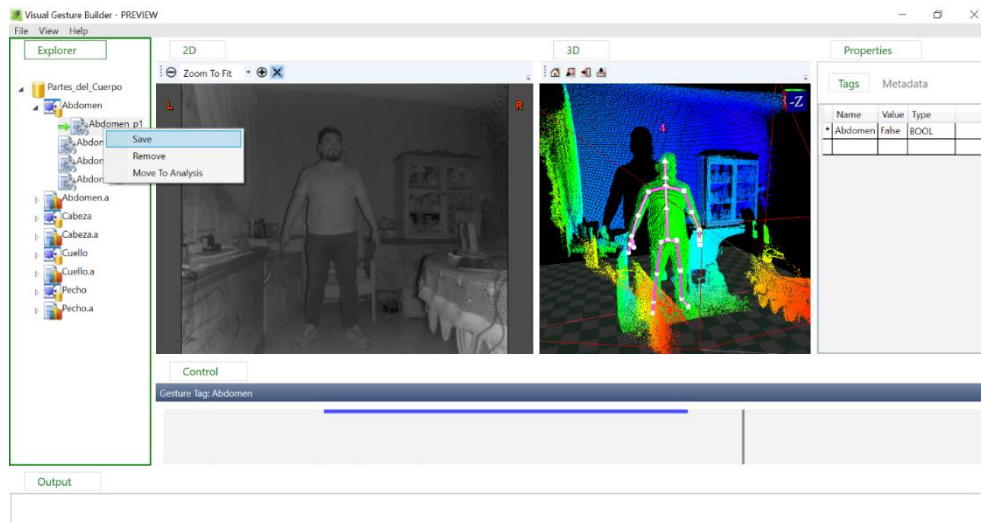


**Figura C.17. Seleccionar secciones “verdaderas” del video de entrenamiento en Visual Gesture Builder**



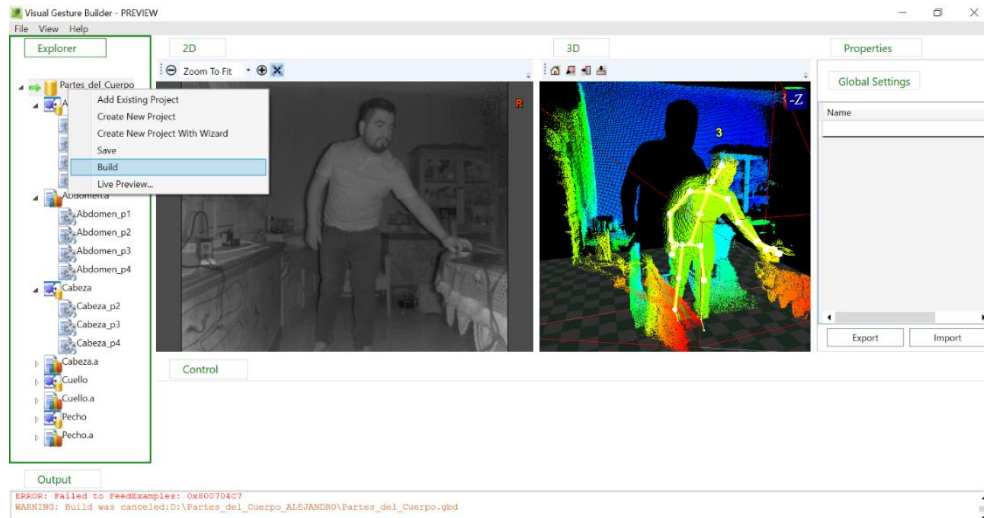
**Figura C.18. Seleccionar secciones “falsas” del video de entrenamiento en Visual Gesture Builder**

M) Tras seleccionar las secciones verdaderas o falsas del video capturado, se guarda la selección dando clic en *Save*. (Figura C.19)



**Figura C.19. Guardar el proyecto en Visual Gesture Builder**

N) A continuación, se da clic en *Build*, para construir la base de datos y realizar el entrenamiento de los gestos propiamente dicho. (Figura C.20)



**Figura C.20. Construir el proyecto en Visual Gesture Builder**

Durante el proceso de entrenamiento de los gestos definidos, VGB muestra paso a paso un texto que permite la construcción. La información brindada permite evaluar la calidad del entrenamiento y se presenta de la siguiente manera:

- Datos del gesto a entrenar y opciones de entrenamiento seleccionadas.
- Información sobre el etiquetado del dataclip grabado.
- Generación de clasificadores.
- Prueba de detección con los datos de entrenamiento.

La prueba de detección es la parte más relevante de la información de entrenamiento: indica qué tan bien funciona el clasificador del gesto usando los datos de entrenamiento. Incluye tanto el porcentaje alcanzado cuando se detecta un gesto previamente clasificado como verdadero y que al ser realizado por un usuario es detectado como verdadero como el porcentaje cuando se detecta un gesto previamente clasificado como falso y que al ser realizado por un usuario es detectado como verdadero o viceversa (gesto detectado como falso a pesar de haber sido clasificado como verdadero). Idealmente, el porcentaje de la detección de gestos positivos verdaderos debe estar cerca de 100%, mientras que los falsos positivos deben estar cerca de 0%.

Para cada gesto, se presenta la información de entrenamiento. Al final de todo el proceso aparece un mensaje según el cual la construcción del entrenamiento ha sido completada exitosamente y ha sido guardada en la dirección determinada en el paso D, con una extensión .gbd. (véanse Figura C.21, Figura C.22, Figura C.23)

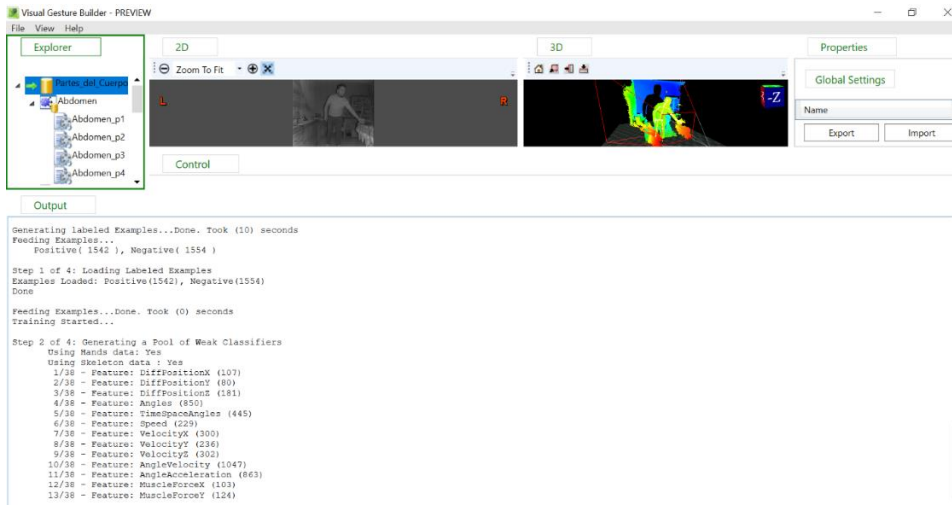


Figura C.21. Proceso de construcción del proyecto en Visual Gesture Builder

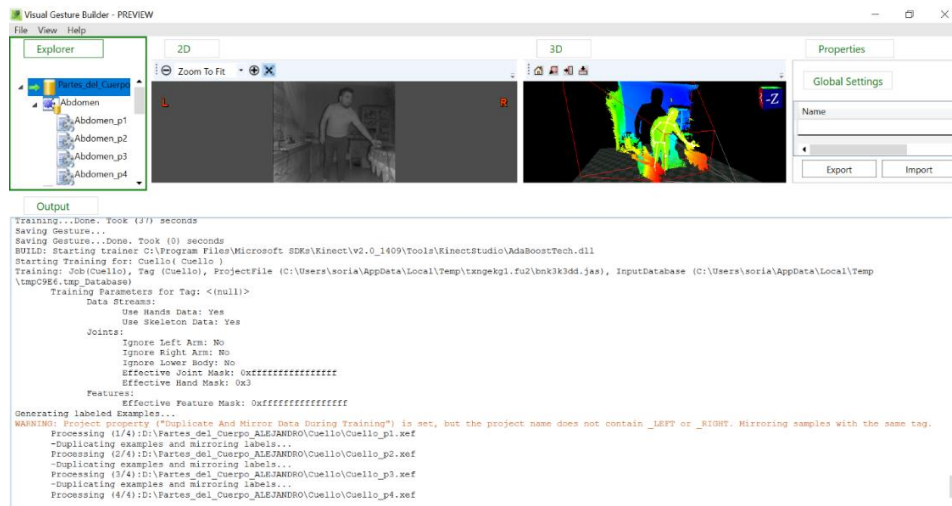


Figura C.22. Construir el proyecto en Visual Gesture Builder

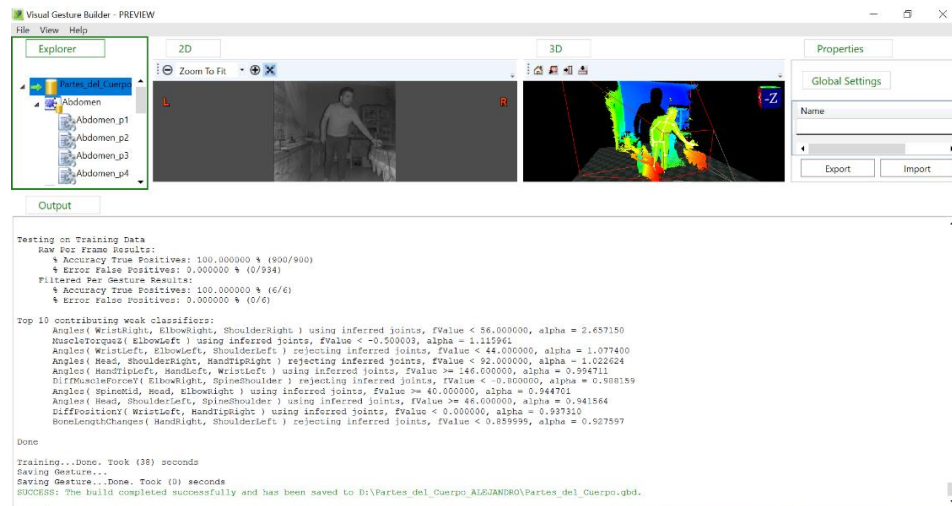
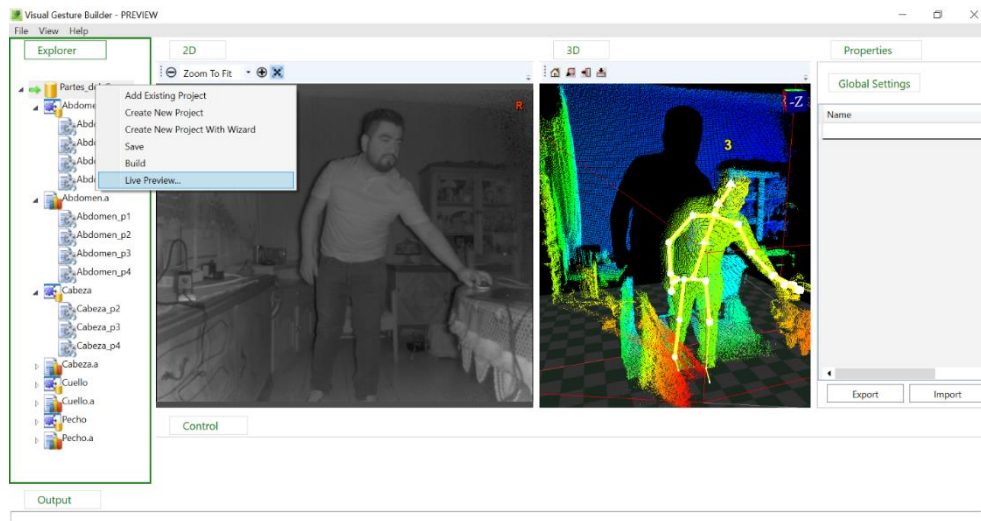


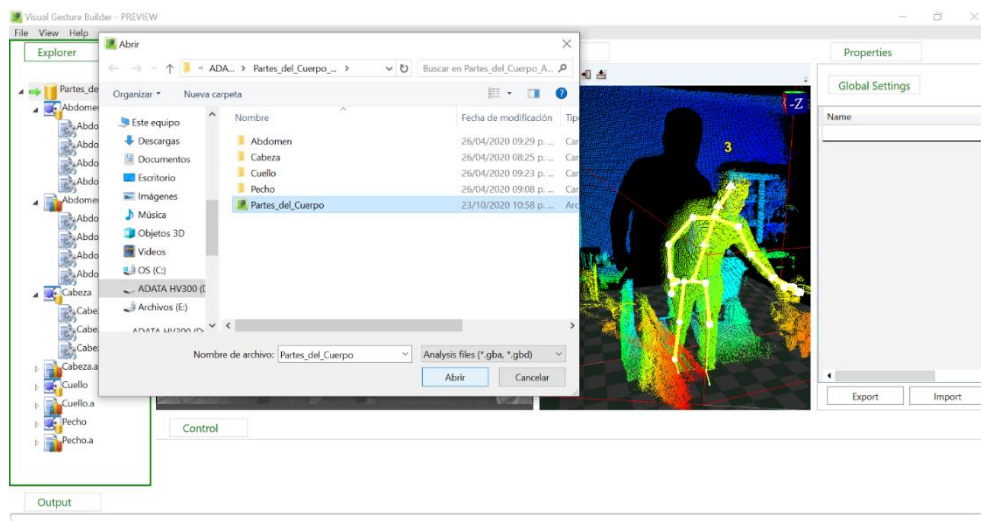
Figura C.23. Proyecto completado y guardado con éxito en Visual Gesture Builder

O) Para verificar el clasificador detecta adecuadamente los gestos, se usa un programa de prueba en vivo (véase Figura C.24). Para ello, se da clic en *Live Preview*, en donde se presenta el nivel de confianza de los gestos de la base de datos. Si un gesto se detecta, aparecerá el nivel de confianza (entre 0 y 1). Si se alcanza un valor de 1, el gesto se detecta con un 100% de confianza.



**Figura C.24. Vista previa en vivo del entrenamiento creado en Visual Gesture Builder**

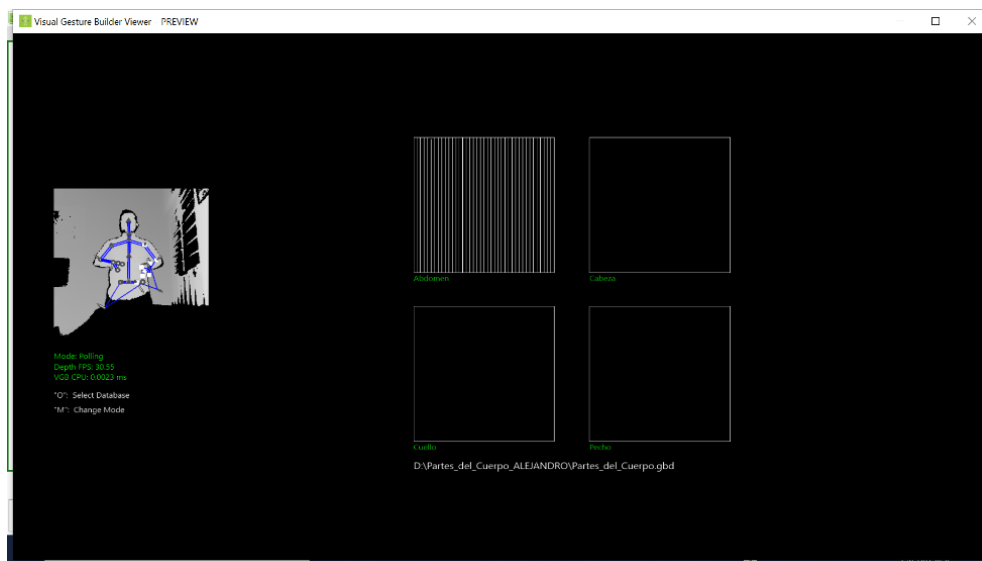
P) Al dar clic en *Live Preview*, se abrirá una ventana (Figura C.25) que permite seleccionar y abrir el archivo generado en la carpeta designada con el nombre previamente definido, con extensión *.gbd*.



**Figura C.25. Abrir el archivo de vista previa en vivo para el entrenamiento creado en Visual Gesture Builder**

Gracias al programa de prueba, es posible validar el desempeño mediante el uso de Kinect v2 en tiempo real (véase la Figura C.26). La ventana muestra la animación del cuerpo mediante Kinect v2 cuando se realizan los gestos entrenados, animación que se puede validar o no en función del nivel de confianza de detección.

En caso de que el nivel de confianza para los gestos no sea cercano a 100%, es posible puede revisar el etiquetado y/o agregar más dataclips para reforzar el entrenamiento.



**Figura C.26. Prueba y validación del entrenamiento en vivo en Visual Gesture Builder**

El archivo del entrenamiento de los gestos es generado con extensión .gbd y es guardado en la dirección mostrada en la parte inferior de la Figura C.26. Este archivo será utilizado para la implementación de la máquina de estados de Ludibot v2, mediante el lenguaje de marcado tipo HTML presentado en el apéndice B.



## Apéndice D. Uso, conexión y animaciones de Cozmo

El robot Cozmo es un robot móvil capaz de realizar un alto número de animaciones. El robot Cozmo tiene el procesador NXP Kinetis ARM Cortex M4, el procesador Expressif ESP8266, tres motores de corriente directa, decodificadores magnéticos, pantalla, dos orugas de caucho como medio de locomoción y una batería de litio. El teléfono celular se utiliza para ejecutar la aplicación Cozmo que permite utilizar el SDK escrito en Python.

En la Figura D.1 se muestra el robot conectado a un teléfono Android.

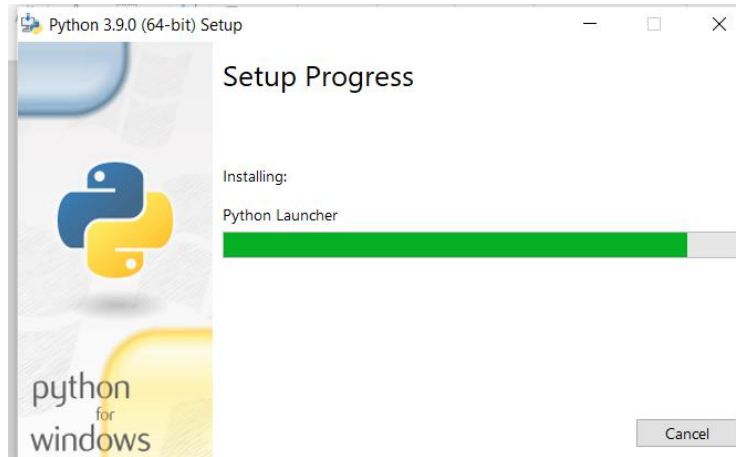


**Figura D.1. Conexión del robot Cozmo con la computadora y el teléfono celular**

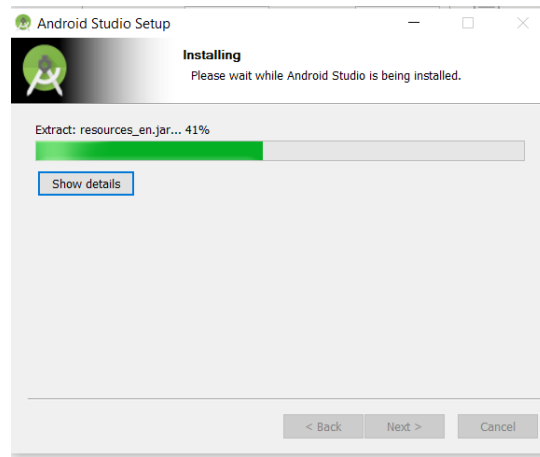
Los programas esenciales para el uso de Cozmo son: la aplicación Cozmo, el sistema operativo Android, el lenguaje de programación Python y el programa Android Debug Device (ADB).

Para utilizar Cozmo, es necesario seguir los siguientes pasos:

- A) Instalar en la computadora los programas Python (véase Figura D.2) y Android Debug Device (ADB) (véase Figura D.3).



**Figura D.2. Instalación de Python**



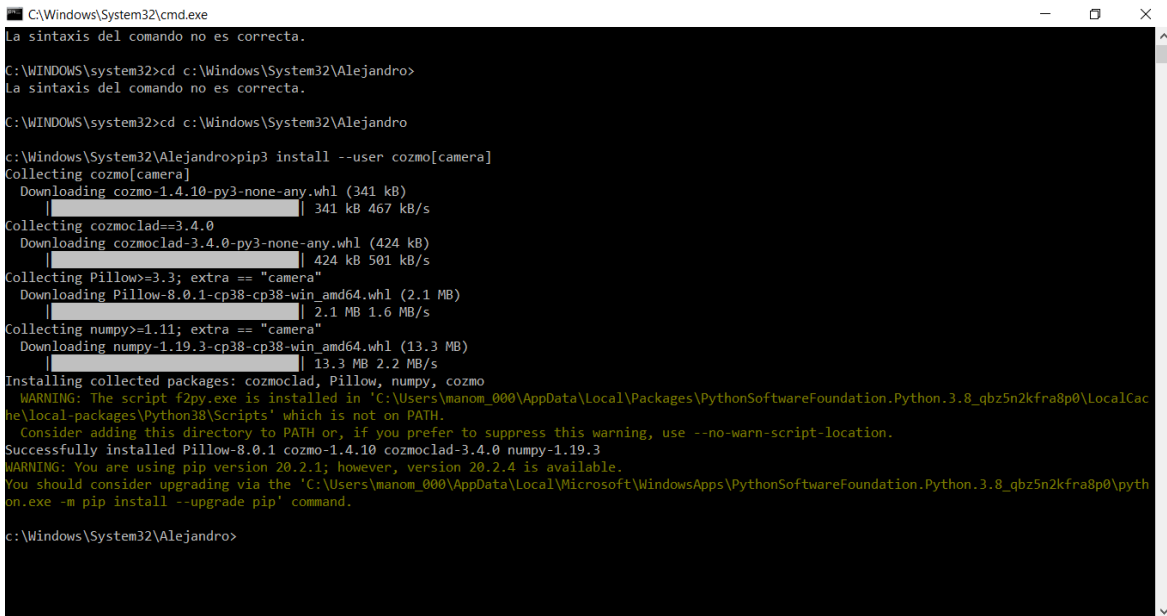
**Figura D.3. Instalación de Android Debug Device mediante Android Studio Setup**

B) Desde el Play Store, instalar la aplicación Cozmo en el teléfono Android (véase Figura D.4).



**Figura D.4. Aplicación de Cozmo para teléfono Android**

- C) Abrir en Windows la herramienta “Símbolo del sistema”, también conocida como CMD o *Command Prompt*. Al abrir CMD, es necesario añadir “cd” para indicar la dirección de la carpeta donde se trabajará. Por ejemplo: `cd c:\Windows\System32\Alejandro`
- D) Instalar el SDK para Cozmo, mediante el comando “`pip3 install --user cozmo[ camera ]`” (véase Figura D.5).



```
C:\Windows\System32\cmd.exe
La sintaxis del comando no es correcta.

C:\WINDOWS\system32>cd c:\Windows\System32\Alejandro
La sintaxis del comando no es correcta.

C:\WINDOWS\system32>cd c:\Windows\System32\Alejandro

c:\Windows\System32\Alejandro>pip3 install --user cozmo[ camera ]
Collecting cozmo[camera]
  Downloading cozmo-1.4.10-py3-none-any.whl (341 kB)
    |#####| 341 kB 467 kB/s
Collecting cozmoclad==3.4.0
  Downloading cozmoclad-3.4.0-py3-none-any.whl (424 kB)
    |#####| 424 kB 501 kB/s
Collecting Pillow>=3.3; extra == "camera"
  Downloading Pillow-8.0.1-cp38-cp38-win_amd64.whl (2.1 MB)
    |#####| 2.1 MB 1.6 MB/s
Collecting numpy>=1.11; extra == "camera"
  Downloading numpy-1.19.3-cp38-cp38-win_amd64.whl (13.3 MB)
    |#####| 13.3 MB 2.2 MB/s
Installing collected packages: cozmoclad, Pillow, numpy, cozmo
  WARNING: The script f2py.exe is installed in 'C:\Users\manom_000\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCac
he\local-packages\Python38\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Pillow-8.0.1 cozmo-1.4.10 cozmoclad-3.4.0 numpy-1.19.3
WARNING: You are using pip version 20.2.1; however, version 20.2.4 is available.
You should consider upgrading via the 'C:\Users\manom_000\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\pyth
on.exe -m pip install --upgrade pip' command.

c:\Windows\System32\Alejandro>
```

Figura D.5. Instalación del SDK de Cozmo desde el símbolo del sistema

- E) Actualizar el SDK, mediante el comando “`pip3 install --user --upgrade cozmo`”.
- F) Instalar en Windows las librerías de ADB para el uso de Cozmo, disponibles en el siguiente link: <http://cozmosdk.anki.com/docs/adb.html>. Es importante, seleccionar la carpeta donde se está trabajando para guardar dichas librerías. Como el archivo está comprimido, es necesario extraerlo en dicha carpeta.
- G) Instalar las librerías de ADB desde el CMD. Es importante seleccionar la carpeta correspondiente: `C:\Windows\System32\Alejandro\platform-tools` (véase Figura D.6).

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.1282]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Alberto Soria>adb
Android Debug Bridge version 1.0.40
Version 4797878
Installed as C:\ADB\adb.exe

global options:
-a      listen on all network interfaces, not just localhost
-d      use USB device (error if multiple devices connected)
-e      use TCP/IP device (error if multiple TCP/IP devices available)
-s SERIAL use device with given serial (overrides $ANDROID_SERIAL)
-t ID   use device with given transport id
-H      name of adb server host [default=localhost]
-P      port of adb server [default=5037]
-L SOCKET listen on given socket for adb server [default=tcp:localhost:5037]

general commands:
devices [-l]      list connected devices (-l for long output)
help             show this help message
version         show version num

networking:
connect HOST[:PORT] connect to a device via TCP/IP [default port=5555]
disconnect [HOST[:PORT]] disconnect from given TCP/IP device [default port=5555], or all
forward --list    list all forward socket connections
forward [--no-rebind] LOCAL REMOTE forward socket connection using:
```

**Figura D.6. Instalación de las librerías de Android (ADB)**

- H) Activar el teléfono Android en modo Depuración de USB, seleccionando Ajustes>>Opciones de desarrollador>>Activado>>Depuración de USB. Esto permitirá llevar a cabo el control de animaciones y movimiento desde el SDK de Cozmo mediante ADB y Python.
- I) Colocar en el CMD >>adb devices. Al dar Enter, aparecerá un código único para el teléfono Android, correspondiente al dispositivo conectado (véase Figura D.7).

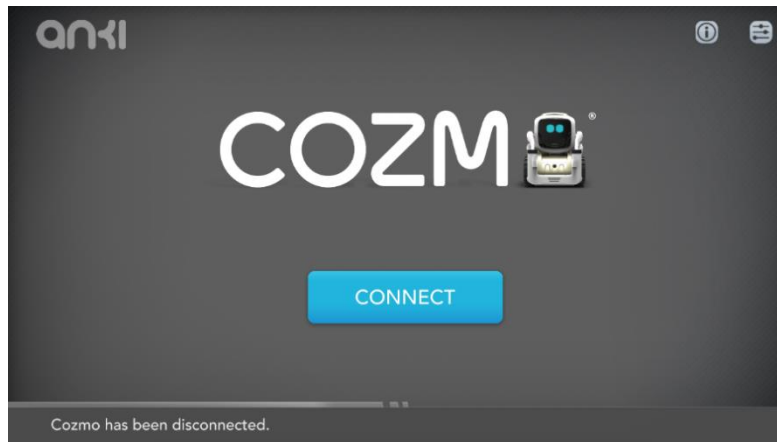
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.1282]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Alberto Soria>adb devices
List of devices attached
52036612f4df93ed device

C:\Users\Alberto Soria>
```

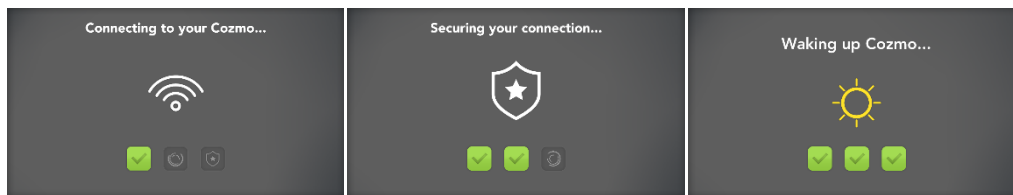
**Figura D.7. Activación del dispositivo Android desde el símbolo del sistema**

- J) Una vez realizado el punto I, es posible llevar a cabo el funcionamiento y la programación de Cozmo. Para ello, es necesario abrir la aplicación de Cozmo y conectar el robot a Cozmo mediante la base integrada, dando clic en conectar (Figura D.8).



**Figura D.8. Conexión de Cozmo desde el teléfono Android**

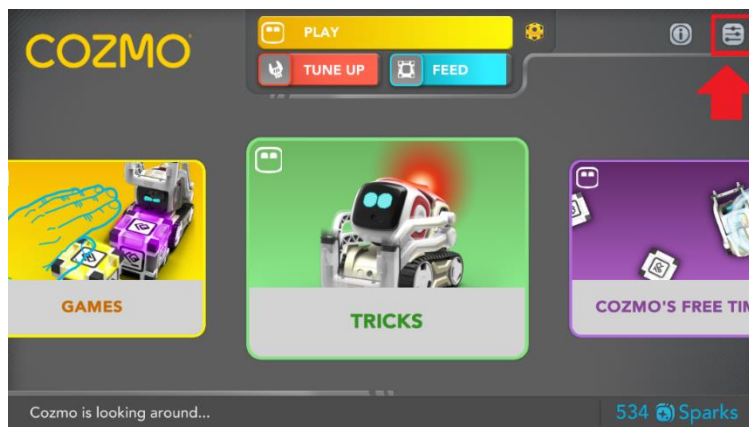
Una vez conectado Cozmo mediante el teléfono Android con su SDK, en la pantalla aparecerá el mensaje “*Connecting to your Cozmo*”, es decir, conectando a tu robot Cozmo. Cuando aparecen las tres indicaciones ✓, el robot ha sido conectado con éxito, como se muestra en la Figura D.9.



**Figura D.9. Conexión exitosa de Cozmo desde el teléfono Android**

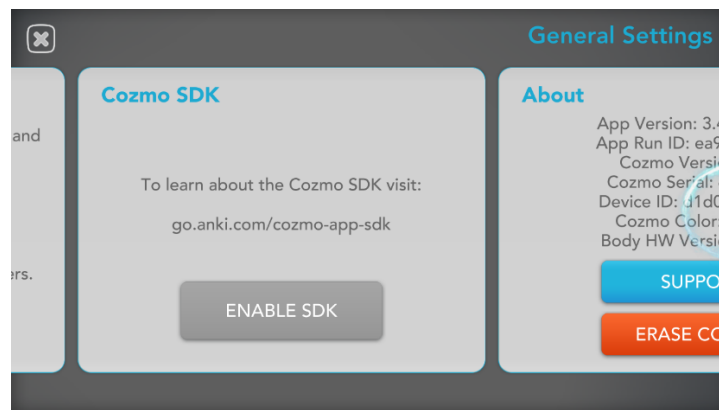
Aparecerá entonces la siguiente pantalla, mostrada en la Figura D.10. Allí se visualizan las diversas aplicaciones disponibles: jugar, aplicar trucos, programar por bloques, entre otros.

Para la programación mediante el SDK, es necesario abrir el menú de opciones en la parte superior derecha (recuadro rojo en la Figura D.10.).



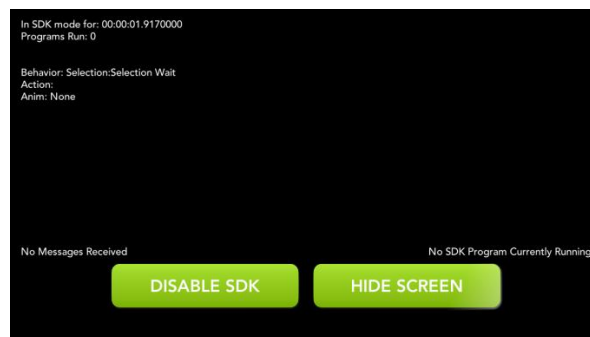
**Figura D.10. Pantalla de la aplicación de Cozmo con juegos, aplicaciones, entre otros**

Se procede entonces a habilitar el SDK, seleccionando y dando clic en la opción *Enable SDK* (véase Figura D.11), para poder llevar a cabo la programación mediante ADB y Python.



**Figura D.11. Habilitar SDK para Cozmo**

Finalmente, aparecerá la pantalla mostrada en la Figura D.12, donde se indica que Cozmo está conectado en modo SDK y que es posible llevar a cabo su programación vía Python.



**Figura D.12. Ventana donde se indica que Cozmo ha sido conectado en el modo SDK**

K) Para conocer mejor la programación de Cozmo, es posible explorar algunos ejemplos dados en el tutorial instalado en el SDK. Para ello, es necesario direccionar la carpeta donde se encuentra el archivo (Figura D.13), por ejemplo:

C:\Users\Alberto Soria\Android\cozmo\_sdk\_examples\_1.4.10\tutorials\01\_basics

```
Microsoft Windows [Versión 10.0.17763.1282]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Alberto Soria\Android\cozmo_sdk_examples_1.4.10\cozmo_sdk_examples_1.4.10\tutorials\01_basics

C:\Users\Alberto Soria\Android\cozmo_sdk_examples_1.4.10\cozmo_sdk_examples_1.4.10\tutorials\01_basics>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 6C78-693B

Directorio de C:\Users\Alberto Soria\Android\cozmo_sdk_examples_1.4.10\cozmo_sdk_examples_1.4.10\tutorials\01_basics:

28/10/2020 06:41 p. m.      <DIR>          .
28/10/2020 06:41 p. m.      <DIR>          ..
16/10/2020 09:40 a. m.      872 01_hello_world.py
16/10/2020 09:40 a. m.     1,176 02_drive_and_turn.py
16/10/2020 09:40 a. m.     1,134 03_count.py
16/10/2020 09:40 a. m.     1,292 04_drive_square.py
16/10/2020 09:40 a. m.     1,902 05_motors.py
16/10/2020 09:40 a. m.     1,765 06_sing_scales.py
16/10/2020 09:40 a. m.     1,623 07_backpack_lights.py
16/10/2020 09:40 a. m.     1,902 08_animation.py
16/10/2020 09:40 a. m.     1,808 09_cube_lights.py
16/10/2020 09:40 a. m.     2,449 10_play_sound.py
16/10/2020 09:40 a. m.     3,099 11_play_song.py
16/10/2020 09:40 a. m.     1,941 12_random_animation.py
                12 archivos            21,183 bytes
                2 dirs             6,122,086,400 bytes libres

C:\Users\Alberto Soria\Android\cozmo_sdk_examples_1.4.10\cozmo_sdk_examples_1.4.10\tutorials\01_basics>08_animation.py
2020-10-28 20:54:37,546 cozmo.general WARNING No iOS device found running Cozmo: [Error 10806] Connect call failed ('127.0.0.1', 27015)
2020-10-28 20:54:37,546 cozmo.general WARNING No Android device found running Cozmo: Failed to execute adb command ['C:\Users\Alberto Soria\Android\cozmo_sdk_examples_1.4.10\cozmo_sdk_examples_1.4.10\tutorials\01_basics\08_animation.py', 'adb.exe', '-s', '52036612f4df93ed', 'forward', '*tcp:5180', '*tcp:5180']; 0 error: unknown host service\n
A connection error occurred: No devices connected running Cozmo in SDK mode

C:\Users\Alberto Soria\Android\cozmo_sdk_examples_1.4.10\cozmo_sdk_examples_1.4.10\tutorials\01_basics>09_animation.py
2020-10-28 20:54:50,792 cozmo.general INFO App connection established. sdk_version=1.4.10 cozmo_sdk_version=1.4.0 app_build_version=00003.00004.00000
2020-10-28 20:54:50,792 cozmo.general INFO Found robot id=1
2020-10-28 20:54:50,808 cozmo.general INFO Connected to Android device serial=52036612f4df93ed
2020-10-28 20:54:51,008 cozmo.general INFO Robot id=1 serial=45a25525 initialized OK
Playing Animation Trigger 1:
Playing Animation Trigger 2: (ignoring the body track)
Playing Animation 3:
2020-10-28 20:55:06,865 cozmo.general INFO Shutting down connection
2020-10-28 20:55:06,865 cozmo.general INFO Android serial=52036612f4df93ed disconnected.

C:\Users\Alberto Soria\Android\cozmo_sdk_examples_1.4.10\cozmo_sdk_examples_1.4.10\tutorials\01_basics>
```

Figura D.13. Abriendo carpeta desde el CMD

Para visualizar los archivos que se encuentran dentro de esta dirección, basta con escribir: “dir”.

Finalmente, se escribe el nombre del archivo con su extensión .py, correspondiente a un archivo de Python. En el ejemplo de la Figura D.14, Cozmo realiza una animación.



Figura D.14. Robot Cozmo realizando una animación

L) Para conocer más animaciones de Cozmo, es posible emplear el Cozmo Animation Explorer. Para ello, es necesario instalar desde el CMD los siguientes comandos: “pip install virtualenv”, “pip install virtualenvwrapper-win” y “pip install flask”.

El siguiente link permite descargar el programa animation\_explorer.py en Python: <https://github.com/GrinningHermit/Cozmo-Animation-Explorer>.

```

Simbolo del sistema
No matching distribution found for virtual.
You are using pip version 18.0, however version 20.2.4 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Alberto Soria\Android>python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/cb/28/91f26bd088ce8e22169032100d4260614fc3da435025ff389ef14396a433/pip-20.2.4-py3-none-any.whl (1.5MB)
    100% |#####| 1.59M 1.09M/s
Installing collected packages: pip
  Found existing installation: pip 18.0
  Uninstalling pip-18.0:
    Successfully uninstalled pip-18.0
  Successfully installed pip-20.2.4

C:\Users\Alberto Soria\Android>python -m pip install --upgrade pip
Requirement already up-to-date: pip in c:\users\alberto soria\appdata\local\programs\python\python36\lib\site-packages (20.2.4)

C:\Users\Alberto Soria\Android>
C:\Users\Alberto Soria\Android>
C:\Users\Alberto Soria\Android>
C:\Users\Alberto Soria\Android>python -m pip install flask
Collecting flask
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
    100% |#####| 94 kB 315 kB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
    100% |#####| 82 kB 158 kB/s
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
    100% |#####| 125 kB 3.2 MB/s
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
    100% |#####| 298 kB 3.1 MB/s
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp36-cp36m-win_amd64.whl (16 kB)
Installing collected packages: itsdangerous, click, MarkupSafe, Jinja2, Werkzeug, flask
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 flask-1.1.2 itsdangerous-1.1.0

C:\Users\Alberto Soria\Android>

```

Figura D.15. Ejecución de comandos para el Cozmo Animation Explorer desde el CMD

Para proceder a la descarga, es necesario indicar en el CMD la dirección del archivo “animation\_explorer” (véase Figura D.15). Por ejemplo: `cd C:\Users\Alberto Soria\Android\Cozmo-Animation-Explorer-master\Cozmo-Animation-Explorer-master`

Se añade “dir” para visualizar los archivos y se escribe “animation\_explorer.py” (véase Figura D.16).

```

Simbolo del sistema - animation_explorer.py
Microsoft Windows [Versión 10.0.17763.1282]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Alberto Soria>cd C:\Users\Alberto Soria\Android\Cozmo-Animation-Explorer-master\Cozmo-Animation-Explorer-master
C:\Users\Alberto Soria\Android\Cozmo-Animation-Explorer-master\Cozmo-Animation-Explorer-master>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 6C78-693B

Directorio de C:\Users\Alberto Soria\Android\Cozmo-Animation-Explorer-master\Cozmo-Animation-Explorer-master

28/10/2020  08:32 p. m.    <DIR>          .
28/10/2020  08:32 p. m.    <DIR>          ..
28/10/2020  07:00 p. m.    <DIR>          4,048 animation_explorer.py
28/10/2020  07:00 p. m.    <DIR>          design
28/10/2020  07:00 p. m.    <DIR>          3,824 flask_helpers.py
28/10/2020  07:00 p. m.    <DIR>          686 play_animation.py
28/10/2020  07:00 p. m.    <DIR>          2,683 README.md
28/10/2020  07:00 p. m.    <DIR>          static
28/10/2020  07:00 p. m.    <DIR>          templates
28/10/2020  07:00 p. m.    <DIR>          __pycache__
                4 archivos          11,241 bytes
                6 dirs          6,138,732,544 bytes libres

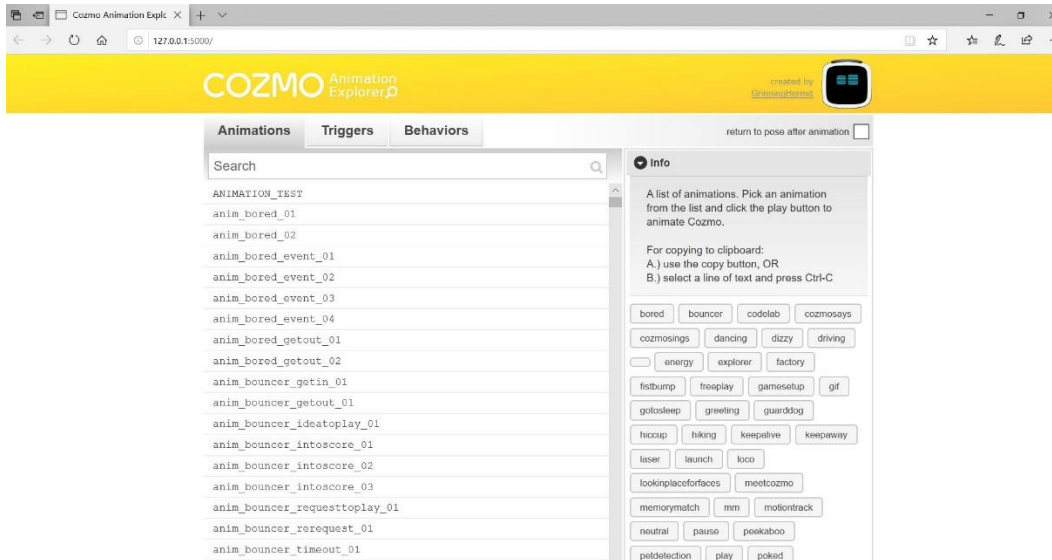
C:\Users\Alberto Soria\Android\Cozmo-Animation-Explorer-master\Cozmo-Animation-Explorer-master>animation_explorer.py
2020-10-28 21:08:33,753 cozmo.general INFO App connection established. sdk_version=1.4.10 cozmoclad_version=3.4.0 app_build_version=00003.00004.00000
2020-10-28 21:08:33,753 cozmo.general INFO Found robot id=1
2020-10-28 21:08:33,753 animation_explorer INFO Found robot id=1
2020-10-28 21:08:33,769 cozmo.general INFO Connected to Android device serial=52036612f4df93ed
2020-10-28 21:08:33,769 animation_explorer INFO Connected to Android device serial=52036612f4df93ed
2020-10-28 21:08:34,222 cozmo.general INFO Robot id=1 serial=45a25525 initialized OK
2020-10-28 21:08:34,222 animation_explorer INFO Robot id=1 serial=45a25525 initialized OK
2020-10-28 21:08:34,300 animation_explorer INFO Attempting to open browser window at 127.0.0.1:5000
* Serving Flask app "animation_explorer" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off

```

Figura D.16. Dirección del archivo Cozmo Animation Explorer desde el CMD

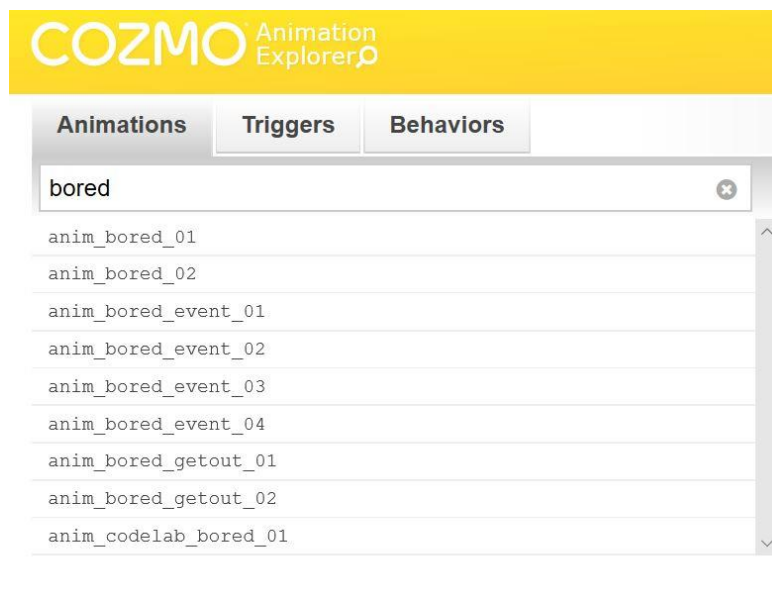
Automáticamente, se abrirá una ventana en un explorador (Figura D.17).





**Figura D.17. Ventana en línea del Cozmo Animation Explorer**

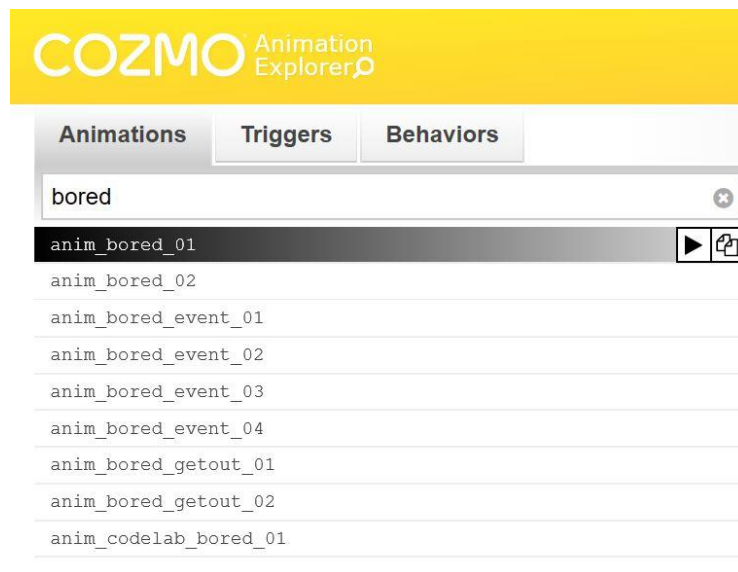
En esa ventana, es posible consultar la lista de todas las animaciones (*Animations*), acciones (*Triggers*) y comportamientos (*Behaviors*) disponibles para Cozmo (véase Figura D.18). Es importante considerar que dichos elementos se clasifican respecto a diversos estados de ánimo. Por ejemplo, cuando se selecciona *bored* (“aburrido”), se enlistan todas las animaciones de Cozmo que expresan aburrimiento.



**Figura D.18. Animaciones, acciones y comportamientos disponibles desde el Cozmo Animation Explorer**

Al seleccionar una animación, es posible elegir entre dos opciones. Eligiendo la primera opción, la animación es reproducida y Cozmo la lleva a cabo en tiempo real. La segunda opción permite

copiar el nombre de la animación para usarla en otra aplicación ejecutada mediante Python. En la Figura D.19 se muestra este proceso para la animación “anim\_bored\_01”.



**Figura D.19. Ejemplo de animación de Cozmo: “anim\_bored\_01”**

De manera similar, es posible experimentar con las acciones (*Triggers*), las cuales se activan durante un lapso de tiempo definido.

Los comportamientos (*Behaviors*) de Cozmo tienen la opción de detenerse (*Stop*). Para ello, se debe dar clic cuando se desea que termine dicho comportamiento (véase Figura D.20).



**Figura D.20. Lista de comportamientos disponibles para Cozmo**