



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

**Unidad Zacatenco**

**Departamento de Computación**

**Diseño e implementación de robots de servicio de bajo  
costo para adultos mayores**

T E S I S

que presenta

**Adrián Josué Ramírez Díaz**

para obtener el Grado de

**Maestro en Ciencias**

**en Computación**

Directores de la Tesis

**Dr. J. Guadalupe Rodríguez García**

**Dra. Sonia Guadalupe Mendoza Chapa**

Ciudad de México

Noviembre 2018



# Agradecimientos

*Por medio de las presentes líneas quiero agradecer a todas las personas e instituciones por el apoyo recibido durante el desarrollo y culminación de la presente tesis.*

*En primer lugar agradezco al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** por la beca otorgada, la cual permitió concluir mis estudios de maestría.*

*Agradezco también al Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV) por permitirme ingresar a uno de sus posgrados de maestría, al igual que por la confianza depositada al otorgarme la beca para la obtención de grado.*

*A mis directores, el Dr. José G. Rodríguez García y la Dra. Sonia G. Mendoza Chapa, que gracias a su orientación y apoyo pude concluir la presente tesis.*

*A mis sinodales, la Dra. Elsa Rubio Espino y al Dr. Amilcar Meneses Viveros por sus valiosos comentarios.*

*Al departamento de Computación, por el ameno ambiente de trabajo y colaboración ofrecida durante mi estancia en el programa de maestría.*

*A mis profesores, por compartir sus conocimientos conmigo, y en especial al Dr. Luis Gerardo de la Fraga, por las ideas acerca del procesamiento de los marcadores.*

*A Daybelis Jaramillo Olivares, quien me apoyó en cada paso hacia esta meta que ambos compartimos, gracias por todos esos momentos juntos.*

*A Jorge Salinas Lara, por ser el amigo que todo foráneo debería tener.*

*Y finalmente agradezco a mi familia por su constante apoyo, mis padres, Nitigad T. Díaz Porras y Jesús M. Ramírez García por impulsar mi crecimiento académico y personal, mis hermanos Miriam e Iván por sus constantes muestras de afecto, al igual que agradezco a mis tíos, Petra Santos y Lorenzo Ramírez, por abrirme las puertas de su hogar durante estos últimos años.*



# Resumen

El envejecimiento de la población trae consigo un incremento en la demanda de servicios de salud, a consecuencia de diversos factores que afectan a los adultos mayores, los cuales, en combinación con escenarios en los cuales el adulto mayor vive solo, en ambientes que carecen de medidas de seguridad necesarias, propician el aumento de situaciones de emergencia y un declive de la calidad de vida. Esto último ha permitido que áreas de la Computación, como el Cómputo Ubicuo, mediante la implementación de sistemas inteligentes, participen en el cuidado de adultos mayores al ofrecer soluciones enfocadas en las necesidades de este sector de la población.

Por lo anterior, en la presente tesis se describe un sistema, denominado **Sistema de Gestión de Actividades (SiGA)**, cuyo objetivo es administrar y recordar a los adultos mayores las actividades que tienen agendadas, con principal atención a las actividades relacionadas con la ingesta de medicamentos, pues una necesidad ocasionada por el deterioro cognitivo de los adultos mayores es recordar los horarios de ingesta de medicamentos, lo cual previene en gran medida situaciones de emergencia.

Para la implementación de este sistema se utilizaron dos componentes principales: un robot para entrega de mensajes y un servidor encargado de la gestión. El robot, el cual es utilizado como mecanismo de interacción entre el sistema y los adultos mayores, tiene la capacidad de desplazarse hasta la localización de los usuarios y entregarles, mediante mensajes de voz, recordatorios que describen las tareas programadas en la agenda. Por ello, en la tesis se describen las características del robot, la navegación a base de análisis e identificación de marcadores en imágenes, así como los métodos utilizados para sincronizar las actividades del robot y establecer un mecanismo de comunicación con el servidor.

El servidor, por otra parte, es el encargado de recopilar la información relacionada con la agenda del adulto mayor a través de un conjunto de interfaces. Con base en dicha información, el servidor proporciona al robot el recordatorio correspondiente en el momento adecuado, motivo por el que se tienen que realizar tareas relacionadas con la programación de actividades, la comunicación con el robot, la creación de rutas, así como la gestión de la ubicación del robot y de los usuarios del sistema, mismas que son descritas en la presente tesis.

Por último, se presenta una evaluación del sistema, mediante la verificación de todas las actividades planeadas para el mismo. De igual forma, se evalúa el funcionamiento de cada uno de los componentes, lo que permite apreciar los métodos de interacción desarrollados para los mismos y, de esta manera, identificar posibles mejoras en su funcionamiento.



# Abstract

The aging of population gives as result an increasing request of health services. Several factors can weigh the elderly people's health; there is a tendency where aged people live alone, and the environments where they live have not the required security measures. These factors and the environment of elderly people contribute to have a decline in the quality of life and an increasing number of emergency situations.

The Ubiquitous Computing has proposed solutions to these kinds of situations, through the proposal of intelligent systems to take care of elderly people giving solutions focused in the necessities of this population sector.

The lost of skills of elderly people, such as memory, gives as result that aged people frequently lose important events such as: medications ingestion, medical and familiar appointments, etc. Some of them can produce situations of risk for the old persons.

The main objective of the proposed system to reduce or even avoid these urgency situations through the scheduling of the activities of the aged people, in this system two main components where implemented: a robot and a server.

The robot gives messages to the user, it is responsible of the interaction between the users and the old persons, it has de capacity of displacing towards the user and recall him the task to be done, such as appointments or medication ingestion through voice messages.

In this document, the robot is described and the mechanisms for displacement, composed by an electro-mechanic mechanism, markers analysis through vision techniques, synchronization and communication with server, is presented.

The server is responsible of collecting and scheduling the information about users' agenda through a set of interfaces. The information must be sent to the robot at the right moment. To this end several tasks must be carried out such as: communication with the robot, path creation (if user and robot are not in the same space), robot and user location analysis, for example.

An evaluation of the system is presented, where all these activities were tested, and every single component was tested, as result possible improvements were detected.





# Índice general

<b>Índice de figuras</b>	<b>x</b>
<b>Índice de tablas</b>	<b>xii</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Planteamiento del problema . . . . .	2
1.3. Hipótesis . . . . .	4
1.4. Objetivos generales y específicos del proyecto . . . . .	4
1.5. Metodología . . . . .	5
<b>2. Conceptos Fundamentales</b>	<b>7</b>
2.1. Cómputo Ubicuo . . . . .	7
2.1.1. Definición . . . . .	7
2.1.2. Evolución . . . . .	8
2.1.3. Taxonomía de los componentes . . . . .	9
2.2. Dispositivos adjuntos . . . . .	11
2.2.1. Comunicación . . . . .	11
2.2.2. Obtención de información . . . . .	12
2.2.3. Selección de sensores . . . . .	13
2.3. Entornos Inteligentes . . . . .	15
2.3.1. Percepción . . . . .	16
2.3.2. Razonamiento . . . . .	18
2.3.3. Acción . . . . .	19
2.3.4. Interacción . . . . .	20
<b>3. Propuesta de solución</b>	<b>23</b>
3.1. Arquitectura general del sistema . . . . .	23
3.2. Componentes del sistema . . . . .	24
3.2.1. Robot . . . . .	24
3.2.2. Dispositivo del usuario . . . . .	24
3.2.3. Servidor . . . . .	25
3.2.4. Conexiones . . . . .	26
3.3. Robot de servicio . . . . .	27

3.3.1.	Diagrama de casos de uso . . . . .	27
3.3.2.	Descripción de casos de uso . . . . .	28
3.4.	Servidor . . . . .	32
3.4.1.	Casos de uso . . . . .	32
3.4.2.	Base de datos . . . . .	36
3.5.	Ubicación y navegación del robot en el entorno . . . . .	37
3.5.1.	Diseño del marcador . . . . .	39
<b>4.</b>	<b>Implementación de robot</b>	<b>41</b>
4.1.	Componentes . . . . .	41
4.1.1.	Conexión . . . . .	44
4.2.	Desplazamiento en el ambiente . . . . .	45
4.3.	Identificación de marcadores . . . . .	47
4.3.1.	Filtrado de la imagen . . . . .	48
4.3.2.	Clasificación del marcador . . . . .	53
<b>5.</b>	<b>Implementación del servidor</b>	<b>57</b>
5.1.	Gestor de interfaces . . . . .	57
5.1.1.	Parámetros de configuración . . . . .	57
5.1.2.	Gestión de medicamentos . . . . .	59
5.1.3.	Interfaz de gestión de citas . . . . .	64
5.2.	Gestor de actividades . . . . .	65
5.2.1.	Consulta de actividades . . . . .	65
5.2.2.	Generador de mensajes . . . . .	66
5.3.	Gestor de posición . . . . .	70
5.3.1.	Almacenamiento de mapas . . . . .	70
5.3.2.	Ubicación del robot . . . . .	71
5.3.3.	Ubicación de los usuarios . . . . .	74
5.3.4.	Generador de rutas . . . . .	75
<b>6.</b>	<b>Pruebas y resultados</b>	<b>77</b>
6.1.	Robot . . . . .	77
6.2.	Desplazamiento e identificación de los marcadores . . . . .	79
6.3.	Identificación del marcador . . . . .	84
6.4.	Conexión entre el servidor y el gestor de tareas . . . . .	85
6.4.1.	Generación de mensajes . . . . .	87
6.5.	Posición de los usuarios . . . . .	88
6.6.	Envío de recordatorios al robot . . . . .	90
6.7.	Prueba general del sistema . . . . .	93
<b>7.</b>	<b>Conclusiones y trabajo futuro</b>	<b>99</b>

# Índice de figuras

2.1. Taxonomía de los componentes de un sistema Ubicomp, obtenida de [32]. . .	10
2.2. Niveles de comunicación de una red BAN [30]. . . . .	12
2.3. Proceso para el reconocimiento de actividades [27]. . . . .	16
3.1. Componentes del sistema . . . . .	24
3.2. Diagrama de casos de uso del robot . . . . .	27
3.3. Diagrama de casos de uso del servidor . . . . .	32
3.4. Diagrama entidad-relación de la base de datos del sistema . . . . .	36
3.5. Ubicación de los marcadores en el entorno . . . . .	38
3.6. Diseño propuesto para el marcador. . . . .	39
4.1. Componentes del robot . . . . .	41
4.2. Conexión de los elementos del robot . . . . .	44
4.3. Filtrado de imagen por color . . . . .	48
4.4. Descomposición de imagen RGB . . . . .	49
4.5. Ecuación de la imagen . . . . .	50
4.6. Diseño propuesto para el marcador. . . . .	51
4.7. Momentos 1 y 3 de Hu de los objetos filtrados . . . . .	54
5.1. Interfaz para gestión de mapas . . . . .	58
5.2. Interfaz para gestión de adultos mayores . . . . .	60
5.3. Búsqueda de usuarios . . . . .	60
5.4. Interfaz para gestión de medicamentos . . . . .	61
5.5. Interfaz para gestión de citas . . . . .	64
5.6. Diagrama de secuencia para los eventos del generador de mensajes . . . . .	69
5.7. Representación del mapa en memoria . . . . .	70
5.8. Filtrado de identificadores . . . . .	72
5.9. Interfaz para simulación de localización de usuarios . . . . .	74
6.1. Vista frontal y lateral izquierda del robot . . . . .	77
6.2. Vista lateral derecha y trasera del robot . . . . .	78
6.3. Nivel inferior, vista lateral y frontal . . . . .	78
6.4. Crecimiento del área del marcador . . . . .	79
6.5. Posición del marcador en la prueba 1 . . . . .	80
6.6. Posición del marcador en la prueba 2 . . . . .	81

6.7. Posición del marcador en las pruebas a distancia 3 y 4 . . . . .	82
6.8. Posición del marcador en la prueba 5 . . . . .	83
6.9. Posición del marcador en la prueba 6 . . . . .	83
6.10. Errores en la clasificación de figuras . . . . .	84
6.11. Datos utilizados para la prueba de creación de mensajes . . . . .	85
6.12. Recordatorios de citas almacenados en la base de datos . . . . .	86
6.13. Recordatorios medicamentos almacenados en la base de datos . . . . .	86
6.14. Datos para verificar el envío de mensajes al servidor . . . . .	87
6.15. Ejecución del proceso gestor de tareas . . . . .	87
6.16. Recepción de recordatorio en el servidor . . . . .	88
6.17. Ejecución del gestor de posición . . . . .	89
6.18. Mapa utilizado para prueba . . . . .	89
6.19. Cambio de ubicación de los usuarios en el servidor . . . . .	89
6.20. Resultados del servidor al verificar el envío de mensajes al robot . . . . .	91
6.21. Primera etapa de la ejecución del robot en la prueba de mensajes . . . . .	92
6.22. Segunda etapa de la ejecución del robot en la prueba de mensajes . . . . .	93
6.23. Mapa del entorno para prueba integral . . . . .	94
6.24. Estado del servidor al iniciar la prueba integral . . . . .	94
6.25. Llegada del primer mensaje al servidor . . . . .	95
6.26. Resultados del robot en la prueba integral . . . . .	95
6.27. Registro del robot en el área pasillo 1 . . . . .	96
6.28. Envío de mensajes del servidor al robot . . . . .	96
6.29. Entrega de mensajes del robot al usuario . . . . .	97

# Índice de tablas

2.1. Clasificación de bioseñales por la energía asociada [23]. . . . .	13
2.2. Modos de percepción y sus aplicaciones [8]. . . . .	14
3.1. Descripción del caso de uso dirigir hacia el marcador . . . . .	28
3.2. Descripción del caso de uso mover ruedas . . . . .	29
3.3. Descripción del caso de uso registrar posición . . . . .	30
3.4. Descripción del caso de uso registrar posición . . . . .	30
3.5. Descripción del caso de uso entregar mensaje . . . . .	31
3.6. Descripción del caso de uso configurar mapa . . . . .	32
3.7. Descripción del caso de uso guardar mapa . . . . .	34
3.8. Descripción del caso de uso gestionar actividades . . . . .	34
3.9. Descripción del caso de uso gestionar medicamentos . . . . .	35
4.1. Descripción de los componentes del robot . . . . .	42
5.1. Tiempos de dosificación estandarizados . . . . .	62



# Introducción

## 1.1. Motivación

Factores, como la reducción de la fecundidad y el aumento de la esperanza de vida, traen como consecuencia una mayor proporción de adultos mayores en la población demográfica del país. En México, datos del Consejo Nacional de Población (CONAPO) indican que entre 1990 y 2017, el porcentaje de población con 60 años y más, aumentó de 6.4 % a 10.5 % <sup>1</sup> y se proyecta que el crecimiento continúe en los próximos años. El aumento del número de adultos mayores implica consecuencias para casi todos los sectores de la sociedad, tales como el mercado laboral y financiero, la demanda de bienes y servicios, al igual que las relaciones intergeneracionales, por ello, el envejecimiento de la población, según las Naciones Unidas, “esta a punto de convertirse en una de las transformaciones sociales más significativas del siglo XXI” <sup>2</sup>.

Por lo anterior, se visualiza un incremento en la demanda de servicios de salud, ya que enfermedades crónicas y degenerativas propician el deterioro de la salud, y de la calidad de vida, al hacer necesario el continuo cuidado y monitoreo de los adultos mayores. Aunado a esto, la preferencia de los adultos mayores a vivir en sus hogares, dificulta que esta supervisión se lleve a cabo, pues en muchos de los casos la independencia de los hijos y demás familiares genera escenarios de riesgo.

Estos escenarios, en los que el adulto mayor vive solo, en conjunción con la disminución de las capacidades cognitivas y motrices, así como un ambiente sin supervisión o sin las adaptaciones necesarias para la adecuada movilidad, propician el aumento de las situaciones de emergencia y declive de la calidad de vida [5, 15].

---

<sup>1</sup> Instituto Nacional de Estadística, Geografía e Informática [INEGI] (2017). *Estadísticas a propósito del día mundial de la población*. Recuperado el 10 de Octubre de 2017, de [http://www.inegi.org.mx/saladeprensa/aproposito/2017/poblacion2017\\_Nal.pdf](http://www.inegi.org.mx/saladeprensa/aproposito/2017/poblacion2017_Nal.pdf)

<sup>2</sup> Naciones Unidas (s/f). *Envejecimiento*. Recuperado el 16 de Octubre de 2017, de <http://www.un.org/es/sections/issues-depth/ageing/index.html>

El cuidado de adultos mayores implica muchas actividades, por lo que este se proporciona acorde con las necesidades y características de la persona mayor. Sin embargo, es posible identificar acciones que influyen directamente en la prevención de situaciones de emergencia, tales como la administración de la medicación, la detección de caídas, el soporte nutricional, la observación (e.g. peso y presión arterial), entre otras [7, 24].

Gracias a la identificación de acciones, estilos de vida o enfermedades que generan situaciones de emergencia, es posible la implementación de herramientas desarrolladas en el área denominada Vida Asistida por el Entorno (AAL <sup>3</sup>) [17], las cuales apoyan al personal de salud en tareas de monitoreo, detección de emergencias, comunicación e integración a la sociedad, administración de actividades y medicamentos, e inclusive tareas de ortopedia y terapia [27].

Las aplicaciones de AAL para adultos mayores son tan variadas como la cantidad de servicios requeridos, sin embargo, la mayoría de estos sistemas requieren de elaboradas instalaciones o costosos equipos, lo cual genera sistemas poco accesibles para la mayoría de personas. Por tal motivo, el presente proyecto pretende realizar un sistema funcional bajo diferentes entornos, al reducir los costos de fabricación e implementación, que proporcione tareas de monitoreo y administración.

## 1.2. Planteamiento del problema

El aumento de las capacidades de cómputo de los dispositivos electrónicos, el desarrollo de nuevas tecnologías bajo el concepto de Internet de las Cosas (IoT <sup>4</sup>), y el continuo trabajo en el campo de la Inteligencia Artificial, Domótica, y otros campos cercanos, han permitido el desarrollo de numerosos proyectos en AAL, muchos de los cuales, están enfocados al cuidado de adultos mayores.

Estos proyectos comparten objetivos, tales como: monitorear al adulto mayor, ayudar en las actividades diarias, prevenir riesgos, apoyar situaciones de emergencia, rehabilitar, detectar anomalías, entre otros. Al igual que es posible categorizar sus objetivos, es posible categorizar las herramientas que estos proyectos ocupan en tres grandes grupos: sensores móviles y usables, entornos inteligentes y robots de servicio.

El uso en conjunto de las herramientas antes mencionadas permite la creación de entornos inteligentes, que apoyan a personas mayores bajo diferentes situaciones, al adaptar sus acciones al contexto en el que se encuentren, del mismo modo que establecen prioridades en las mismas, mediante la generación de acciones proactivas, acorde con las actividades del

---

<sup>3</sup>Ambient Assisted Living

<sup>4</sup>Internet of Things



usuario.

Al contar con sistemas inteligentes, que interactúan de forma constante con las personas y toman decisiones basadas en información proveniente del entorno, es necesario que estos sistemas tengan probabilidades de fallo limitadas para garantizar la seguridad del usuario. Así, la implementación de un sistema AAL no debe interferir con las actividades cotidianas del usuario, por el contrario, debe ser transparente para el mismo, al apoyarlo a través de sugerencias en sus actividades. Otro factor importante a ser considerado es la comunicación con la sociedad, por lo que, más que aislar al usuario, el sistema AAL debe apoyar la integración de los adultos mayores a la sociedad.

El uso de robots en proyectos AAL comúnmente se relaciona con sistemas de asistencia para las Actividades de la Vida Diaria (ADL <sup>5</sup>), lo que permite categorizarlos en robots de asistencia a actividades diarias, robots de instrumentación y robots de actividades diarias mejoradas; cada una con mayor complejidad que la anterior [27]. Sin embargo, los robots con mayor complejidad no son siempre los que mayor beneficio aportan al adulto mayor, debido a la tarea que desempeñan. Al tener esto en mente, es posible la construcción de robots que realicen tareas de monitoreo y comunicación, tareas que no requieren el manejo complicados instrumentos de precisión, pero que aportan múltiples beneficios en el cuidado de adultos mayores.

Las tareas de monitoreo en adultos mayores pueden ir desde una localización geoespacial, hasta mediciones médicas como presión arterial o niveles de glucosa en la sangre. Por lo cual, la mayoría de proyectos incluyen, en la vestimenta del adulto mayor, sensores que se encargan de realizar estas mediciones. Dichos sensores, sin embargo, deben tener como característica principal, la no obstrucción de las actividades del usuario, debido a que pueden limitar la movilidad del mismo o incluso representar un riesgo. Esto hace necesario que, tanto los requerimientos de consumo de energía como la movilidad de los dispositivos, deben estar siempre pensados en evitar la no intrusión.

Al tomar en cuenta las características antes mencionadas, la creación de entornos inteligentes, que apoyen a las actividades diarias de personas con edad avanzada, resulta a menudo complicada en cuanto diseño de herramientas se refiere, lo cual genera elevados costos económicos de creación e implementación.

Por tal motivo, se propone la creación de un sistema para la asistencia de adultos mayores, el cual tenga como características principales, la fácil implementación y accesibilidad en cuanto a costos económicos se refiere. Este sistema debe abarcar tareas de monitoreo, gestión de actividades medicas, tales como citas o ingesta de medicamentos, así como la comunicación entre el adulto mayor, los miembros de la familia y el personal médico. Esto debido a que son actividades que cubren algunos de los servicios que aportan mayores

---

<sup>5</sup>Activities of Daily Living

beneficios en el cuidado de personas con edad avanzada.

Así, la forma en la cual el sistema tendrá interacción con el adulto mayor es a través de un robot de asistencia, que permita al adulto mayor obtener la información médica antes mencionada.

### **1.3. Hipótesis**

Por consiguiente, la pregunta de investigación que propone dar respuesta la presente tesis es la siguiente. ¿Es posible la construcción de un sistema de vida asistida por el ambiente, enfocado al cuidado de adultos mayores, adaptable al entorno y cuyo costo económico no sea elevado?

### **1.4. Objetivos generales y específicos del proyecto**

#### **General**

Construir un sistema, con las características de los sistemas de vida asistida por el ambiente, cuyo coste económico sea bajo, adaptable a diferentes entornos, y enfocado al cuidado de adultos mayores, de igual modo que realice tareas de gestión y recordatorios de actividades médicas.

#### **Específicos**

1. Investigar las características más importantes de algunos sistemas en el área de AAL.
2. Diseñar y construir un robot de servicio, que permita al sistema desempeñar tareas de entrega de alertas y recordatorios, desplazándose a la ubicación de los usuarios.
3. Seleccionar, diseñar e implementar de los algoritmos, que permitan al robot de servicio, la realización de las tareas antes mencionadas.
4. Definir y construir la arquitectura de comunicación entre los elementos que interactúan en el sistema.
5. Construir el sistema de gestión de recordatorios y alertas médicas.
6. Construir las interfaces de gestión del sistema, para que, tanto el personal médico, como los familiares del adulto mayor, puedan evaluar la información que recopile el sistema, y configurar parámetros de operación.
7. Verificar el correcto funcionamiento del sistema en cuanto a tiempo de respuesta de las acciones del mismo, así como a la precisión de la información obtenida.

## 1.5. Metodología

Para la construcción de la solución propuesta se presentan, a manera de lista, las actividades necesarias para cumplir con los objetivos propuestos, adaptando cada una de ellas a la solución mencionada.

- A<sub>1</sub>. Investigar proyectos relacionados con el cuidado de adultos mayores en el área de vida asistida por el ambiente.
  - 1.1. Elaboración del protocolo de tesis.
  - 1.2. Investigación de proyectos similares.
- A<sub>2</sub>. Diseño y construcción del robot de servicio.
  - 2.1. Determinación de los requerimientos.
  - 2.2. Modelado del robot.
  - 2.3. Obtención de los componentes mecánicos y tecnológicos.
  - 2.4. Construcción del robot.
  - 2.5. Pruebas.
- A<sub>3</sub>. Implementación de los algoritmos para el funcionamiento del robot.
  - 3.1. Análisis de métodos de navegación en interiores.
  - 3.2. Análisis de técnicas para detección y seguimiento de personas.
  - 3.3. Revisión de técnicas de monitoreo para adultos mayores.
  - 3.4. Implementación de la característica de navegación en interiores.
  - 3.5. Implementación del monitoreo del adulto mayor.
  - 3.6. Implementación de la interfaz de entrega de mensajes.
- A<sub>4</sub>. Construcción de la arquitectura de comunicación del sistema.
  - 4.1. Diseño de la arquitectura de comunicación.
  - 4.2. Construcción de la arquitectura de comunicación.
  - 4.3. Comunicar los subsistemas.
- A<sub>5</sub>. Construcción del sistema de gestión de alertas médicas.
  - 5.1. Revisión de sistemas de manejo de notificaciones.
  - 5.2. Revisión de algoritmos y métodos para gestión de toma de decisiones.
  - 5.3. Construcción del sistema.

**A<sub>6</sub>**. Construcción de las interfaces de gestión.

6.1. Determinación de requerimientos.

6.2. Diseño de las interfaces.

6.3. Construcción de las interfaces.

6.4. Pruebas.

**A<sub>7</sub>**. Pruebas del sistema.

# Conceptos Fundamentales

## 2.1. Cómputo Ubicuo

Al buscar en la literatura conceptos como *Pervasive Computing* [28] o *Calm Technology* [11] es inevitable referenciar el término propuesto por Weiser a principios de los noventa, *Ubiquitous Computing*. Por medio de la computación ubicua (UbiComp), Weiser describe la tendencia de los sistemas de cómputo de formar parte del ambiente en el que participan sus usuarios, al cubrir las necesidades de información de manera natural, mediante el uso de mecanismos de interacción con una alta usabilidad [34]. De esta manera, UbiComp incluye sistemas cuyos componentes participan activamente en el entorno, al fungir no solo como gestores de información, sino también como proveedores de servicios, con la capacidad de modificar dicho entorno mediante la interacción de los múltiples componentes que conforman el sistema.

### 2.1.1. Definición

Computación ubicua es un concepto que se ha modificado a lo largo del tiempo, sin embargo, se mantiene fiel a la visión de Weiser. Estos cambios han dado pie a la creación de conceptos como *Pervasive Computing* (atribuido a IMB), el cual es propuesto debido a la penetración que tienen los sistemas de cómputo ubicuo en el entorno, sin embargo, en opinión de Symonds and Khosrow-Pour [32], pueden utilizarse como sinónimos.

UbiComp es presentada como la tercer generación de la computación moderna [19]. Donde, la primer generación se encuentra representada por las computadoras *mainframe*, las cuales son generalmente propiedad de empresas y compartidas por múltiples personas; la segunda generación está representada por las computadoras personales, las cuales, como su nombre lo indica, son propiedad de una persona; la tercer generación y la cual estamos describiendo, está representada por la creación de pequeñas computadoras portátiles con la capacidad de conexión a múltiples redes, al igual que computadoras embebidas en los diferentes dispositivos que usamos, lo que origina un ambiente en el que una persona utiliza

y cuenta con múltiples computadoras con un propósito determinado [19].

Lo anterior permite establecer la cooperación entre los elementos de un sistema Ubicomp, la influencia del ambiente y una interacción humano computadora más transparente para el usuario, sin embargo, un sistema Ubicomp debe contar con las siguientes características principales [26]:

1. El sistema necesita trabajar de forma distribuida, donde cada componente cuente con los mecanismos de comunicación necesarios para una correcta interrelación entre componentes.
2. La interacción entre el sistema y el usuario se debe realizar de forma transparente, es decir, los mecanismos de acceso al sistema necesitan tener un enfoque centrado en el usuario.
3. El sistema necesita ser consciente del contexto en el que se encuentra, para optimizar su operación en el ambiente.
4. El funcionamiento del sistema puede realizarse de forma autónoma, y auto-gestionada, acorde con las acciones de los usuarios.
5. El sistema debe manejar multiplicidad de acciones e interacciones dinámicas, gestionado por una toma de decisiones y una interacción organizacional inteligentes. Esto puede implicar alguna forma de inteligencia artificial que maneje:
  - a) Interacciones incompletas y no determinísticas.
  - b) Cooperación y competencia entre miembros de una organización.
  - c) Cooperación enriquecida por el intercambio de contexto, semántica y objetos.

### 2.1.2. Evolución

La tendencia hacia la miniaturización de componentes electrónicos, junto con el crecimiento exponencial de las capacidades de cómputo y una globalización de las infraestructuras de redes, han sido factores clave en el crecimiento del cómputo ubicuo, dejando así, tres generaciones en las cuales se pueden clasificar los sistemas pertenecientes a esta área. [10].

En la primer generación se encuentran sistemas en los que el funcionamiento es autónomo y la adaptación al ambiente se desarrolla gracias a la capacidad de comunicación de todos sus componentes. La creación de componentes de red con menor costo, tamaño y consumo de energía, al igual que componentes con mayor velocidad, inalámbricos y siempre conectados, permitieron la creación de redes de componentes bajo diferentes condiciones [10].

La segunda generación incluye sistemas en los cuales, la autonomía de éstos se realiza gracias a la conciencia del contexto, en otras palabras, sistemas cuyas acciones se basan en el análisis de la información relativa al contexto en el que se encuentran, información que resulta relevante para la tarea que desempeña el mismo [2].

El contexto puede ser determinado a través de diferentes fuentes, como son: las acciones de los usuarios, la energía con la que cuenta, la información proveniente del ambiente, entre otras. Sin embargo, en cada una de estas fuentes, el contexto se modela a partir de datos de bajo nivel obtenidos mediante sensores, los cuales son accionados por características físicas del entorno en un periodo de tiempo específico [10].

En la tercera generación, la autonomía de los sistemas es obtenida, mediante la semántica del contexto, es decir, el significado de las acciones y situaciones en las que interactúa el sistema. Lo anterior da pie a sistemas altamente complejos, en los cuales sus componentes coordinan acciones para desempeñar determinadas tareas, realizan funciones en paralelo, con constante retroalimentación entre componentes y cuyo control tiende a ser disperso y descentralizado [10].

### 2.1.3. Taxonomía de los componentes

Un sistema Ubicomp, como se menciona en secciones anteriores, se encuentra construido por un conjunto de componentes que permiten la realización de diversas tareas, coordinando acciones y recopilando información del entorno. Sin embargo, estos componentes son tan variados como funciones y requerimientos tenga el sistema, por tal motivo, es necesaria una clasificación de dichos componentes.

Si bien existen diversas formas de categorizar los componentes de un sistema Ubicomp, en el presente trabajo se utiliza la clasificación propuesta por Symonds y Khosrow-Pour [32], la cual toma a los componentes como dispositivos, clasificándolos en dos categorías principales, dispositivos adjuntos y dispositivos encontrados en el ambiente, tal como se aprecia en la figura 2.1, lo que permite establecer la siguiente taxonomía:

#### 1. Dispositivos adjuntos a los humanos

Estos son dispositivos que los usuarios del sistema portan, tendiendo contacto con ellos la mayor parte del tiempo.

- a. Dispositivos portados. Estos dispositivos se dividen a su vez en tres subcategorías:
  - (1) *dispositivos móviles*, también llamados dispositivos portables, entre los que se encuentran dispositivos con computadoras de propósito general, como lo son computadoras portátiles, PDAs, teléfonos inteligentes o similares; los
  - (2) *dispositivos inteligentes* son dispositivos que permiten la identificación, autenticación y autorización de personas y posiblemente de otros propósitos; los
  - (3) *sensores en el cuerpo* son el método más utilizado para el monitoreo del estado físico y la salud de los usuarios.
- b. Prendas inteligentes. Estos dispositivos van desde prendas y accesorios aumentados por computadora hasta prototipos que se fabrican a partir de componentes estándar (teléfonos inteligentes en una funda con auriculares, etc.).

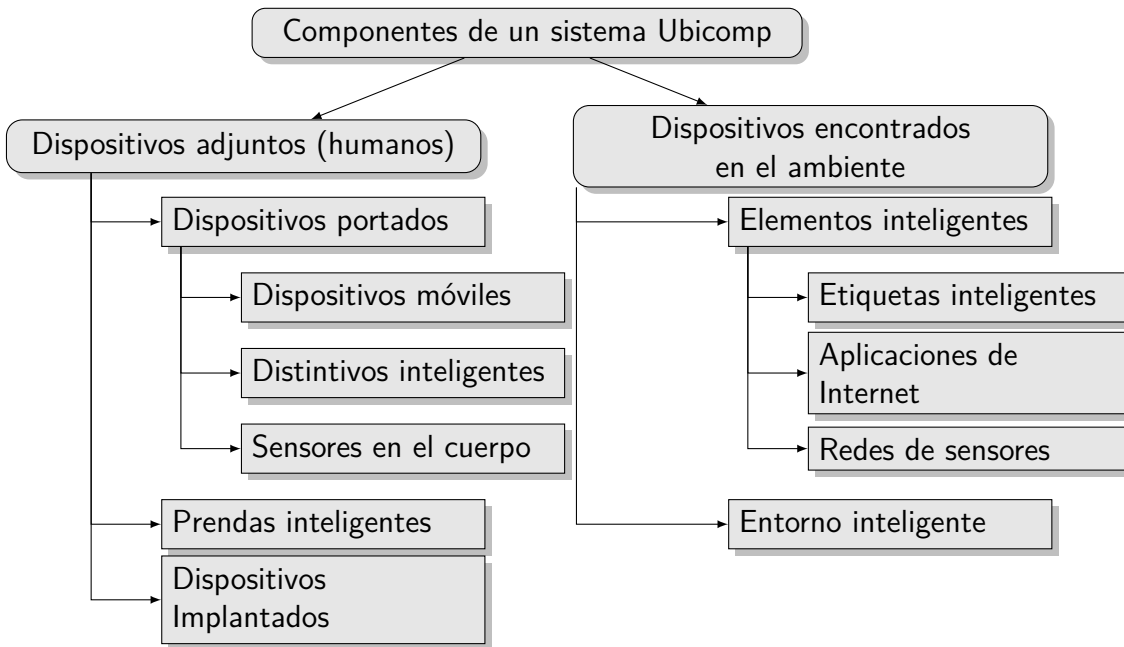


Figura 2.1: Taxonomía de los componentes de un sistema Ubicomp, obtenida de [32].

- c. Dispositivos implantados. Debido a las características (seguridad, salud, privacidad) relacionadas con dispositivos implantados en el cuerpo esta categoría no es ampliamente utilizada.

## 2. Dispositivos encontrados en el ambiente

Estos dispositivos se encuentran desempeñando tareas en el ambiente y si bien tienen interacción con el usuario, no lo hacen de forma constante como es el caso de los dispositivos adjuntos.

- a. Elementos Inteligentes, también llamados objetos físicos aumentados por computadora, son utilizados para una comunicación proactiva con los usuarios. En categoría se divide en tres subcategorías: (1) *etiquetas inteligentes*, las cuales son la forma menos sofisticada de crear elementos inteligentes, estas etiquetas permiten a otros dispositivos conocer información o funcionalidades de un objeto en determinado momento; las (2) *redes de sensores* permiten la recopilación de información del ambiente en el que se encuentra el sistema; las (3) *aplicaciones de Internet*, es una categoría en la cual se incluye el diseño de arquitecturas que permiten coordinar una serie de dispositivos a través de redes e Internet.
- b. Entornos Inteligentes. Se denota entorno inteligente a toda la atmósfera que engloba a los elementos inteligentes presentes en el sistema, es decir, a la comunicación adicional, los elementos de software requeridos y la infraestructura necesaria para convertir el conjunto de elementos inteligentes en una unidad significativa.



## 2.2. Dispositivos adjuntos

Prendas inteligentes, dispositivos móviles, componentes inteligentes, dispositivos vestibles, entre otros, han sido términos utilizados para referirse a dispositivos que pertenecen a un sistema Ubicomp, con la característica particular que los usuarios los portan en su cuerpo, o cerca de él. Al usuario mantener contacto con ellos la mayor parte del tiempo, estos dispositivos, equipados con capacidades de cómputo, almacenamiento, comunicación y monitoreo, permiten al sistema Ubicomp conocer acciones del usuario, así como su estado. Este estado se entiende como las características y mediciones obtenidas del usuario, las cuales son relevantes para el objetivo del sistema. Así pues, las mediciones pueden ir desde características de ubicación, tales como orientación, posición en el entorno, hasta mediciones biométricas, como presión sanguínea, respiración, actividad cardíaca, temperatura, entre otras.

Seneviratne *et al.* [30] clasifican estos dispositivos en tres categorías principales; accesorios, textiles inteligentes y parches inteligentes. Los accesorios abarcan dispositivos que viste en usuario y que no entran en la categoría de accesorios de ropa. Algunos ejemplos de estos dispositivos son los relojes, pulseras, anteojos inteligentes, collares, cinturones, así como algunas pinzas para la ropa. Los textiles inteligentes son aquellos dispositivos que pueden clasificarse en artículos de ropa de uso común. Estos artículos son usados en su mayoría para el monitoreo de señales fisiológicas y biomecánicas, obtener características del ambiente así como aplicaciones sensoriales y hápticas. Los parches inteligentes son dispositivos adheridos o tatuados a la piel, los cuales, al igual que los textiles inteligentes, son comúnmente usados para el monitoreo de señales fisiológicas y biomecánicas.

Si bien la clasificación de Seneviratne *et al.* [30] utiliza términos diferentes a los utilizados por Symonds y Khosrow-Pour [32], es factible decir que al hablar de accesorios inteligentes se refiere a los dispositivos portados, igual es el caso de los textiles inteligentes, encontrados en la taxonomía de Symonds y Khosrow-Pour como prendas inteligentes. En el caso de los parches inteligentes, éstos se encuentran en la categoría de sensores en el cuerpo, al ser un tipo de sensores que realizan un análisis más específico.

### 2.2.1. Comunicación

Si bien la clasificación de Seneviare *et al.* [30] es más reducida, estos autores presentan el concepto de redes de área corporal (BAN), con la cual se busca describir la extensión de comunicación ocupada por los sensores presentes en el cuerpo. A partir de la definición de la red BAN, se puede hablar de tres tipos de comunicaciones presentes en los dispositivos adjuntos: comunicación Intra-BAN, comunicación Inter-BAN y comunicación Beyond-BAN, tal como se aprecia en la figura 2.2.1.

**Intra-BAN.** La comunicación intra-BAN hace referencia a una comunicación entre sensores colocados en el cuerpo. Para ello se necesita un manejo eficiente de la energía, por tal

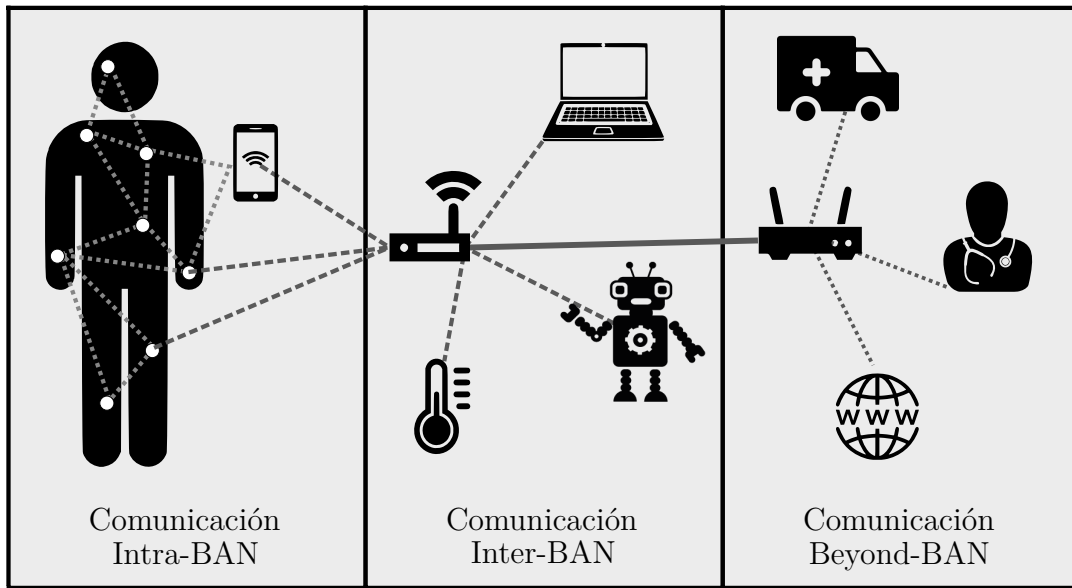


Figura 2.2: Niveles de comunicación de una red BAN [30].

motivo se han propuesto esquemas de comunicación con el aprovisionamiento de Calidad de Servicio. Otras propuestas existentes incluyen el uso del cuerpo humano como canal de comunicación entre sensores, mediante el uso del tejido corporal. Entre las opciones más utilizadas para la transición de información se encuentran las incorporadas por el estándar IEEE 802.15, como lo son, Bluetooth (IEEE 802.15.1), ZigBee (IEEE 802.15.4) y WiMedia (IEEE 802.15.3).

**Iter-BAN.** La comunicación inter-BAN incluye el envío de información desde dispositivos personales como teléfonos inteligentes o diversos sensores a puntos de acceso, ya sea mediante una infraestructura de red o mediante un enlace *ad-hoc*. Entre las opciones más utilizadas para este tipo de comunicación se encuentran tecnologías como WLAN, Bluetooth, ZigBee, 3G/4G, entre otras.

**Beyond-BAN.** La comunicación beyond-BAN conecta la red del sistema Ubicomp con Internet y otras redes. Usualmente esta comunicación incluye la persistencia de información mediante una base de datos y una interfaz de acceso web, para facilitar la comunicación entre usuarios del sistema y proveedores de servicios, como los servicios de salud.

### 2.2.2. Obtención de información

Anteriormente se mencionó que los dispositivos adjuntos al cuerpo permiten el análisis y el monitoreo de señales biométricas, fisiológicas o biomecánicas. Estas bioseñales tienen como característica principal un cambio durante el tiempo o el espacio. Estas bioseñales se asocian con varios tipos de energía, permitiendo así clasificarlas en seis grupos principales; eléctricas, mecánicas, térmicas, radiantes, magnéticas y químicas. [23]

Tabla 2.1: Clasificación de bioseñales por la energía asociada [23].

Tipo de energía	Parámetros	Ejemplos de bioseñales
Eléctrica	Voltaje, corriente, resistencia, capacitancia, inductancia, etc.	Electrocardiograma, electroencefalografía, electromiografía, electrooculograma.
Mecánica	Desplazamiento, velocidad, aceleración, fuerza, presión, flujo, etc.	Presión sanguínea, velocidad de la onda de pulso.
Térmica	Temperatura, flujo del calor, conducción, etc.	Temperatura central del cuerpo, temperatura de la piel.
Radiante	Luz visible, ondas de radio infrarojo, etc.	SpO <sub>2</sub> * , fotopleletismograma.
Magnética	Flujo magnético, intensidad de campo, etc.	Magnetoencefalografía, medidor de flujo.
Química	Composición Química, pH (derivado de diversas fuentes), etc.	Glucosa, colesterol, creatina quinasa.

\* SpO<sub>2</sub> presenta el porcentaje de moléculas de hemoglobina unidas al oxígeno.

Con el uso de sensores biomédicos se puede cuantificar los niveles de energía presente en el cuerpo (tabla 2.1), niveles que corresponden a fenómenos fisiológicos de interés para el sistema Ubicomp. Al traducir esta energía detectada a otro tipo de energía, generalmente una señal eléctrica, es posible la obtención de información mediante el procesamiento, almacenamiento y envío de esta señal. La utilidad de estas señales radica en el análisis histórico de las mismas, ya que al detectar cambios, se identifican posibles anomalías en el funcionamiento del proceso fisiológico correspondiente.

### 2.2.3. Selección de sensores

El tipo adecuado de sensor, como en el caso de los sensores para prendas inteligentes, se relaciona con el tipo de medición realizada, por lo que se debe identificar la característica que se desea conocer del usuario, o incluso del entorno, para así determinar el sensor correcto. Algunos de los sensores diseñados permiten la medición de posición, detección de químicos, humedad, niveles de luz, radiación, temperatura, sonido, tensión, presión, velocidad, dirección (véase la tabla 2.2), llegando incluso a la detección fisiológica para apoyar el monitoreo de la salud.

Se ha mencionado que sensores implantados en dispositivos adjuntos al cuerpo de los usuarios permiten analizar determinados procesos fisiológicos y biomecánicos de los mismos, por lo que es de gran importancia la selección correcta del tipo correcto de sensor, dicho lo anterior, en esta sección se presentan algunas sugerencias realizadas por Scilingo, Lanatà y Tognetti en [29] para el monitoreo realizado a través de prendas inteligentes.

Tabla 2.2: Modos de percepción y sus aplicaciones [8].

Tipo de percepción	Usos comunes
Tensión y presión.	Pisos, puertas, camas, sofás, balanzas.
Posición, dirección y movimiento.	Seguridad, localización, seguimiento, detección de caídas.
Luz, radiación y temperatura.	Seguridad, localización, seguimiento, cuidado de la salud, eficiencia energética.
Sólidos, líquidos y gases.	Seguridad y salud, monitoreo, eficiencia de riego.
Etiquetas inteligentes.	Usado en la identificación de personas y objetos.
Sonido.	Seguridad, control de volumen, reconocimiento de voz.
Imagen.	Seguridad, identificación, entendimiento del contexto.

### Sensores para el monitoreo biomecánico

Los sensores biomecánicos son usados para el registro de parámetros cinemáticos de los segmentos del cuerpo. El análisis de los gestos y el movimiento corporal pueden ser medios para detectar anomalías en el movimiento, relacionados con una patología específica, de igual forma son utilizados para contextualizar actividades físicas específicas.

*Técnicas estándar.* Las técnicas estándar que permiten el análisis de movimiento se basan en sistemas estereofotogramétricos, magnéticos y electromecánicos, teniendo como principal ventaja la precisión obtenida, sin embargo, este tipo de sistemas operan bajo un área restringida, o en algunos requieren partes molestas para los usuarios.

*Sensores de movimiento inercial.* En esta categoría se incluyen sensores como acelerómetros y giroscopios, los cuales tienen como principal ventaja el consumo de energía y el área ocupada, sin embargo, el análisis de la información proporcionada se vuelve más complejo.

### Sensores para la captura de señales fisiológicas

Los dispositivos adjuntos a los usuarios son utilizados comúnmente en aplicaciones del área de la salud y el cuidado de personas, por ello se requiere de un monitoreo constante de las señales fisiológicas de los usuarios.

*Actividad respiratoria.* La actividad respiratoria requiere el análisis de la cantidad y velocidad del aire que entra y sale de los pulmones, dicho análisis se puede realizar de forma directa o indirecta. Para el caso de un análisis directo se realiza por medio de espirómetros, sin embargo este método no es muy recomendado para las prendas inteligentes, por el echo de llevarse un dispositivo a la boca. Los métodos indirectos usan desplazamientos del pulmón transmitidos a la pared del tórax y viceversa, algunas técnicas indirectas recomendadas son

pletismografía inductiva respiratoria, pletismografía de impedancia, sensores piezorresistivos y/o neumografía piezoeléctrica.

*Respuesta galvánica de la piel.* La respuesta galvánica de la piel es una de las técnicas más explotadas en el diseño de prendas inteligentes debido a la facilidad en su implementación y su relación con los parámetros más significativos en el campo de la neurociencia. Para el análisis de esta característica se realizan dos tipos de mediciones; la primera se denomina exosomática, ya que la corriente en la que se basa la medición se introduce desde el exterior. El segundo tipo de medición, la cual es menos usada, se llama endosomática, pues la fuente de voltaje es interna.

*Oximetría de pulso.* La oximetría de pulso es un método no invasivo para monitorear la saturación de oxígeno en la sangre, el frecuencia cardíaca y amplitud de pulso. Un oxímetro de pulso ilumina en dos longitudes de onda un lecho de tejido y mide la señal de luz transmitida.

*Sensores térmicos radiantes.* La temperatura del cuerpo es usualmente obtenida por sensores en contacto con la piel, sin embargo esta forma de realizar la medición es subjetiva, ya que depende de la parte del cuerpo donde se realice la medición y es sensitiva al incremento de la circulación sanguínea. Por lo anterior se sugiere el uso de radiómetros, los cuales permiten determinar la temperatura interna del cuerpo a través de microondas.

*Sistemas de actividad cardiopulmonar.* Al diseñar un sistema de esta índole se espera que permita el monitoreo tanto de la actividad pulmonar como la cardíaca. El método más común es el uso del electrocardiógrafo, el cual proporciona información sobre la actividad eléctrica del corazón, sin embargo el agregarlo a una prenda inteligente se vuelve una tarea complicada. Así pues en la literatura se proponen técnicas como el uso de radares Doppler de microondas, lo que permite determinar el movimiento del corazón en lugar de las características eléctricas. Al ser esta técnica posible el uso de tecnologías como Ultra WideBand permite analizar este movimiento con un bajo consumo de energía.

## 2.3. Entornos Inteligentes

Al presentar la taxonomía de los elementos que componen un sistema Ubicomp [sección 2.1.3], se hace mención del entorno inteligente, al cual se define como la conjunción de los diferentes elementos de software, la infraestructura de comunicación y los distintos dispositivos inteligentes ubicados en el ambiente gestionado por un sistema Ubicomp [32]. Esta definición permite identificar que la gestión de un entorno inteligente representa una parte importante en la construcción de un sistema Ubicomp, por ello, el área denominada como inteligencia ambiental (AmI), se enfoca al desarrollo de este tipo de sistemas.

Por lo anterior, es posible definir a la Inteligencia Ambiental como los mecanismos que de forma proactiva, pero sensible, gestionan el entorno de sus usuarios, dando soporte a

las actividades cotidianas de los mismos. Al mencionar la característica “sensible” de estos sistemas AmI, se hace referencia al uso de sistemas inteligentes, los cuales aprenden y conocen las preferencias del usuario, lo que les proporciona la capacidad de mostrar empatía o reaccionar acorde con el estado de ánimo del usuario, o ante una situación imperante [3].

AmI se encuentra estrechamente relacionada con diversas áreas de las ciencias de la computación, tales como redes, sensores, interfaces humano-computadora, robótica y/o sistemas multiagente, sin embargo, es necesario mencionar que a pesar de la relación con estas áreas, no debe confundirse con ninguna de ellas, pues ninguna cubre completamente a la inteligencia ambiental, debido a que en AmI se reúnen múltiples recursos proporcionados por las áreas mencionadas, he incluso más, con el objetivo de brindar servicios flexibles e inteligentes a usuarios que actúan en los entornos gestionados. Lo anterior lleva a Cook *et al.* a clasificar la contribución de las áreas en cinco categorías: percepción, razonamiento, acción, interacción y seguridad [8].

### 2.3.1. Percepción

La Inteligencia Ambiental es un área diseñada para trabajar con entornos del mundo real, por lo que es indispensable el uso de mecanismos que permitan a los algoritmos inteligentes obtener las características del entorno en el cual trabajan, para así tomar decisiones pertinentes sobre el mismo.

El enlace entre el entorno y el poder computacional de los sistemas AmI se realiza a través de actuadores y sensores, los actuadores permiten a los algoritmos realizar cambios en el ambiente. Por otra parte, los sensores permiten la identificación del estado de ambiente, los cambios que sufre y las acciones que realizan los usuarios. Sin embargo esta identificación de estados, cambios y acciones no se realiza de forma directa, pues el sensor entrega al sistema un conjunto de valores a través del tiempo, los cuales representan alguna característica determinada, por ejemplo, el nivel de luz, humedad, presencia de químicos, temperatura, sonido, presión, posición, velocidad, dirección, etc.

Existen diversas formas y algoritmos que permiten identificar acciones y estados a partir de los datos recolectados de sensores, sin embargo, en todos ellos se realiza un análisis de los datos proporcionados por el sensor en periodos de tiempo, teniendo como principal objetivo el filtrado de datos que son relevantes en un momento determinado.

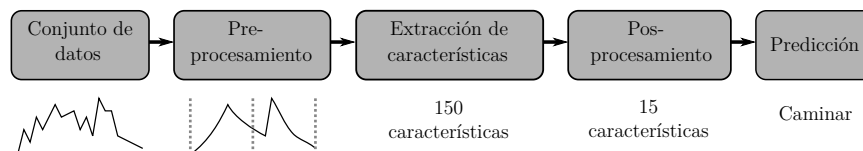


Figura 2.3: Proceso para el reconocimiento de actividades [27].

Rashidi y Mihailidis [27] describen el proceso de obtención de información de sensores como un proceso diversas etapas, tal y como se muestra en la figura 2.3. Primero, se realiza una captura de datos provenientes de un sensor a una frecuencia determina, después es necesaria una etapa de filtrado de datos, con la cual se realiza una remoción de ruido. El siguiente paso es extraer las características estadísticas y morfológicas de los segmentos de señales, como lo es la media, desviación estándar, amplitud, transformada de Fourier, entre otras. El paso de posprocesamiento se encarga de reducir el número de características extraídas, aplicando selección de características y técnicas de reducción de dimensionalidad. Obtenido ese conjunto de características, el siguiente paso es seleccionar, acorde a las características obtenidas, el correspondiente estado, acción o cambio en el entorno [27] .

## Visión por computadora

Uno de los métodos más utilizados para el análisis del entorno y sus integrantes es el uso de cámaras, mediante las cuales se obtienen imágenes que permiten determinar elementos y cambios en el ambiente. Por esta razón, el reconocimiento de actividades a partir de cámaras es uno de los métodos mas utilizados para la percepción del contexto, pues estas técnicas proporcionan información de contexto muy detallada. Sin embargo, debido a la cantidad de información que puede contener una imagen, se enfrentan diversas dificultades como múltiples variaciones en entornos naturales, complejidad algorítmica y problemas de privacidad [27].

Por lo anterior, es necesario establecer niveles de abstracción de la información para su análisis. El modelo JDL (*Joint Directors of Laboratories*) presenta cinco niveles de abstracción, llamados: evaluación de características en señales (L0); evaluación de entidades (L1); evaluación de situaciones (L2); evaluación de impacto (L3) y evaluación del proceso (L4) [13].

Los niveles de abstracción de información más bajos, L0 y L1 en el modelo JDL, se enfocan en el pre-procesamiento de la imagen y el análisis de los objetos contenidos en ella, por lo que se incluyen procesos como la localización y calibración de la cámara, al igual que la detección y seguimiento de objetos. Los niveles de abstracción altos L2 y L3, tienen como objetivo la describir las relaciones entre los objetos en el escenario percibido, relaciones expresadas a través de términos simbólicos como acciones o intensiones en lugar de medidas numéricas. En este nivel de abstracción se realizan acciones como la identificación y reconocimiento de actividades; la construcción de modelos; y la detección y evaluación de situaciones que tienen relevancia en determinados escenarios (ej., situaciones de emergencia). El nivel L4 contiene tareas enfocadas en planificar y realizar procedimientos para mejorar la fusión entre los niveles altos y bajos [13].

### 2.3.2. Razonamiento

Como se ha mencionado, el enlace entre los algoritmos inteligentes y el mundo real se realiza a través de sensores y actuadores, sin embargo, para hacer que los servicios otorgados por estos actuadores sean adaptables, receptivos y beneficiosos para los usuarios, es necesario para el sistema AmI realizar varios tipos de razonamiento, como modelado, la predicción y reconocimiento, la toma de decisiones o el razonamiento espacio temporal [8].

**Modelado.** Los sistemas AmI tienen, entre otras características, la capacidad de responder de forma diferente ante distintos usuarios, ya que a través de la constante interacción con ellos, es posible reconocer y hacer distinciones entre los mismos, identificando sus preferencias y habilidades, lo que conlleva una mejor interacción y respuesta del sistema. Por tal motivo, el modelado de usuarios (UM) se relaciona comúnmente con el análisis de la interacción entre el usuario y el sistema, al igual que con el desarrollo de modelos cognitivos para la creación de métodos de interacción e interfaces. [8, 16]

Los elementos en un modelo de usuario puede incluir representaciones de objetivos, planes, preferencias o tareas de uno o mas tipos de usuarios; la clasificación de usuarios en subgrupos o estereotipos; la creación de suposiciones sobre el usuario en función de un historial y la generalización y clasificación de historias de interacción con usuarios. Así mismo, al tener sistemas que gestionan un entorno, la interacción de los usuarios es determinado contexto es indispensable conocerla, por lo que, al diseño de modelos es importante incluir características como la localización de los usuarios, el estado del entorno e incluso el entorno social en el que se encuentra [16].

**Reconocimiento y predicción de actividades.** En la sección 2.3.1 se presentó el proceso mediante el cual se realiza la extracción de características, tomando como base los datos obtenidos de sensores, así mismo se menciona que, a partir de la clasificación de esas características, es posible la identificación de determinadas actividades. Sin embargo, el uso de datos que se originan en sensores físicos permite la identificación de acciones simples del usuario, como movimientos, gestos, desplazamientos, etc. Por lo que, en muchos de los casos, es necesario repetir el proceso mencionado pero tomando como entrada las acciones más básicas.

Lo anterior permite clasificar las actividades obtenidas de los usuarios acorde con su nivel de resolución, como movimientos simples, acciones, actividades, grupos de actividades y multitud de actividades. Este etiquetado de actividades es realizado de diversas formas, pero un enfoque ampliamente utilizado es el uso de algoritmos supervisados, los cuales se basan en el etiquetado por medio de entrenamiento, como lo son los árboles de decisión, redes neuronales, razonamiento causa-efecto, modelos de mezcla y los modelos gráficos. Dentro de estos últimos se encuentran técnicas como las cadenas de Markov, Redes bayesianas dinámicas, el modelo oculto de Markov y los campos aleatorios condicionales [27].



**Toma de decisiones.** El proceso de toma de decisiones consiste en extraer y determinar la información necesaria para automatizar la selección de las actividades que realiza un sistema. Esta información proviene de la etapa de percepción, en la cual se identifica un conjunto de características que proporcionan el contexto en el que se encuentra el sistema. Para ello se emplean diversos algoritmos que permiten establecer filtros adaptativos que analizan la información del contexto estableciendo un conjunto de reglas que determinen las acciones a seguir [14].

### 2.3.3. Acción

Los sistemas Ubicomp se vinculan con el entorno a través de sensores y actuadores, permitiendo reconocer el entorno e interactuar con el mismo. Para lograr esta interacción y modificación del entorno, los sistemas Ubicomp usan dispositivos inteligentes y de asistencia que permiten la ejecución de acciones y afectar el entorno de los usuarios. Otro mecanismo de interacción de los sistemas Ubicomp son los robots de servicio. Por ejemplo, un robot de asistencia permite al adulto mayor, superar sus limitaciones físicas mediante el apoyo en la realización de tareas cotidianas, lo cual ocasiona que se obtenga un incremento en el grado de independencia por parte del adulto mayor [27].

### Robots de asistencia

Un robot de asistencia permite al adulto mayor, superar sus limitaciones físicas mediante el apoyo en la realización de tareas cotidianas, lo cual ocasiona que se obtenga un incremento en el grado de independencia por parte del adulto mayor [27].

El uso de robots de asistencia son más evidentes en áreas de aplicación como vida asistida, guías autónomas, asistencia por robots y robots sociales. Sin embargo, esto no quiere decir que se limiten a estas áreas, pues, en el enfoque seguido por la investigación actual, los robots de asistencia son creados con una limitación o barrera específica es mente, para así generar los mecanismos necesarios para superarla mediante una tecnología determinada [22].

Al tener robots que realizan una tarea específica, muchos de ellos trabajan en conjunto, es decir, forman parte de un ambiente inteligente dedicado al cuidado de un adulto mayor. Al tener robots con capacidad de comunicación en una red, se opta por destinar parte del procesamiento del mismo a computadoras con mayor capacidad de procesamiento, lo cual da pie a los servicios robóticos en la nube (CSR). Los servicios robóticos en la nube pueden definirse como la integración de diferentes agentes que permiten, de manera eficiente, efectiva y robusta, la cooperación entre robots, entornos inteligentes y humanos [4].

Así pues, se han desarrollado proyectos como en Bonaccorsi *et al.*[4], en donde se utilizan los conceptos robot como servicio y software como servicio. En el concepto robot como

servicio, se accede a todas las características del robot a través de una interfaz encontrada en una red. El mismo enfoque tiene software como servicio, solo que los servicios que se tienen no pertenecen a un robot, sino a un componente de software. Esto permite la integración de ambientes inteligentes donde la gestión, acceso y comunicación de todos los componentes se realiza a través de una red, lo cual permite el uso de elementos de Internet de las Cosas.

### 2.3.4. Interacción

Para mayor aceptación de los sistemas Ubicomp se requiere que la interacción con sus usuarios se realice de forma natural. Por ello, además de los dispositivos de interacción con el ambiente, se debe establecer un método de comunicación adecuado entre el sistema Ubicomp y los usuarios, enfocándose en las características fisiológicas del usuario así como el contexto en el que se encuentre.

### Conciencia de contexto

En la presentación de la evolución de los sistemas Ubicomp (sección 2.1.2), se menciona que a partir de la segunda generación, dichos sistemas adquirieron autonomía a través de la conciencia del contexto, es decir, el uso de información proveniente del entorno en el que se encuentra el sistema; la acciones de los usuarios; o incluso el mismo estado del sistema; con el objetivo de determinar las acciones que realizan [10].

Por lo tanto, la conciencia del contexto puede ser descrita como la habilidad de una aplicación para conocer su entorno o situación física, con el objetivo de responder de forma proactiva e inteligente basando en dicha conciencia [36]. Acorde con el nivel de contexto necesario para una aplicación, el contexto puede representarse en tres niveles: contexto de bajo nivel, contexto de alto nivel; y relaciones situacionales. En el contexto de bajo nivel se encuentra la información obtenida de sensores, sin embargo, esto no quiere decir que solo contenga información de dispositivos físicos, por lo cual es posible clasificar los sensores de la siguiente manera [36]:

- Los *sensores físicos* hacen referencia a dispositivos que capturan magnitudes físicas, como posición, temperatura, etc.
- Los *sensores virtuales* son aplicaciones de software o servicios así como datos semánticos, por ejemplo datos de redes sociales o información sobre dispositivos.
- Los *sensores lógicos* son una combinación de sensores físicos y virtuales agregando información adicional obtenida de bases de datos o archivos.

El contexto de nivel alto o contexto inferido se obtiene a partir de información recopilada en niveles de contexto más bajo, lo que genera la siguiente clasificación:

- *Contexto de dispositivos.* Incluye conectividad, costo de comunicación, recursos.
- *Contexto de usuarios.* Incluye perfil, posición geográfica o situación social.
- *Contexto temporal.* Incluye fechas, días, meses, estaciones, años, etc.

## Interfaces de usuario

El diseño de interfaces de interacción con los usuarios es una tarea compleja basada en investigación, técnicas, métodos, guías de diseño, así como una amplia gama de campos relacionados, como lo son: psicología cognitiva, al aplicar metáforas y modelos mentales; experiencia de usuario, mediante el uso de maquetas por ejemplo; arquitectura de información, con el uso de modelos de datos compartidos o almacenes de datos; diseño de comunicación, aplicando comunicación visual o auditiva, entre otros; ingeniería de interfaz de usuario, por medio de técnicas como la creación de prototipos; estética a través de diseño industrial, ingeniería de usabilidad; interacción centrada en el usuario; o factores humanos, como lo es la capacidad humana o la ergonomía [19].

A pesar de la diversidad de técnicas y conceptos implementados en el desarrollo de interfaces, es necesario que se mantengan las siguientes reglas de diseño de interfaces [19]

- *Facilidad.* Es necesario que al aprender a interactuar con una nueva interfaz, el usuario no requiera aprender nuevas habilidades o lenguaje de comandos complejo.
- *Distracción.* La interfaz no debe demandar constante atención.
- *Flujo cognitivo.* Al ser sistemas presentes en múltiples partes de un entorno, se debe permitir al usuario mantener un enfoque total en la tarea.
- *Manuales.* No se requiere que el usuario necesite leer un manual para operar el sistema.
- *Transparencia.* El usuario no debe tener la necesidad de tener en mente el estado del sistema al operar las interfaces.
- *Modos de operación.* Evite que el sistema responda de forma diferente a la normal, dado el mismo estímulo de entrada debido a que existe información oculta.
- *Miedo a la operación.* Proporcione medios para deshacer acciones, ya que de lo contrario el usuario tendrá miedo al realizar acciones sobre la interfaz.
- *Notificaciones.* La retroalimentación proporcionada al usuario puede incluir interacciones con su entorno físico.
- *Sentidos.* Es posible que las interfaces se basen en una amplia gama de entradas humanas, como los sentidos.
- *Valores predeterminados.* Las buenas interfaces deben de explotar el contexto conocido o lo que se pueda deducir del mismo.

Las reglas anteriores corresponden al enfoque del mínimo esfuerzo en las interfaces, ya que el usuario no debería de requerir habilidades especiales para poder usarlas. El mas claro ejemplo de esta necesidad ocurre en interfaces de sistemas enfocados al cuidado de adultos mayores. Estos sistemas deben tomar en cuenta que los usuarios tienen limitadas capacidades fisiológicas, motrices y en ocasiones cognitivas, lo que genera la necesidad de interfaces que soporten problemas como la debilidad visual, necesidad para recordar.

## Propuesta de solución

### 3.1. Arquitectura general del sistema

El sistema propuesto, denominado Sistema de Gestión de Actividades (SiGA), consta de tres componentes principales, **Servidor**, **Robot** y **Supervisor de usuarios**. El *servidor* tiene como objetivo procesar y administrar los datos utilizados por el sistema, proporcionar las interfaces necesarias para la configuración y funcionamiento, y apoyar al robot en tareas que implican un alto coste computacional. El *robot* por su parte, se encarga de la interacción con el ambiente y el usuario, mediante el desplazamiento del mismo a través de las áreas que conforman el entorno inteligente, o que permite la interacción con el usuario (mensajes, supervisión, comunicación). Por último, el *dispositivo del usuario* se encarga de recopilar datos relevantes para el sistema, correspondientes al estado en el que se encuentran los usuarios, al igual que información que permite el funcionamiento del sistema.

Al igual que el sistema, el *servidor* consta de tres subcomponentes: gestor de posición, gestor de interfaces y gestor de actividades. El gestor de posición es el encargado de almacenar, obtener y procesar la información correspondiente a la ubicación de los elementos y personas en el entorno. El *gestor de interfaces* se encarga de recopilar la información relacionada con la agenda del adulto mayor, al igual que permitir la configuración del sistema. El gestor de actividades es el encargado de establecer la relación entre los subcomponentes anteriores, ya que, con base en la información obtenida por ellos, determina el lugar y momento adecuado para entregar las notificaciones.

En la figura 3.1, se observan los componentes principales del sistema, la interacción entre estos componentes, al igual que los mecanismos que los conforman, descritos a continuación.

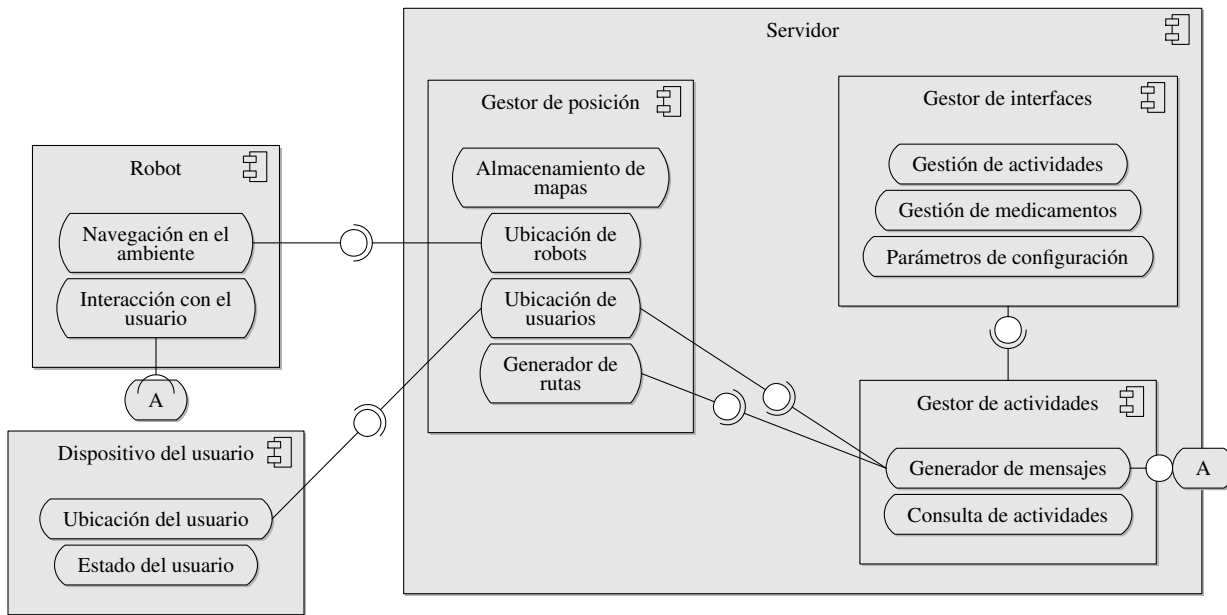


Figura 3.1: Componentes del sistema

## 3.2. Componentes del sistema

### 3.2.1. Robot

*Navegación en el ambiente.* El método de navegación utilizado por el robot hace uso de puntos de referencia visuales, por consiguiente, el mecanismo de navegación del robot se enfoca en la calibración de los movimientos del robot a partir de datos extraídos de imágenes capturadas por la cámara. Por esta razón, este mecanismo incluye el filtrado, extracción, clasificación e interpretación de datos presentes en la imagen, a fin de obtener el marcador utilizado como punto de referencia.

*Interacción con el usuario.* El robot es el método a través del cual el sistema interactúa con los adultos mayores, mediante el monitoreo de los mismos y entrega de recordatorios, por esta razón, el mecanismo de interacción con los usuarios incluye la síntesis de texto a voz, al igual que captura de imágenes para su posterior envío al servidor.

### 3.2.2. Dispositivo del usuario

*Ubicación del usuario.* Este mecanismo forma parte del dispositivo portado por el usuario, y tienen como objetivo proporcionar al *gestor de posición*, ubicado en el servidor, la localización de los diferentes usuarios, lo que permite desplazar al robot hacia las áreas donde se requiera interactuar.

*Estado del usuario.* Otro método de monitoreo a través del cual el sistema analiza el estado

del adulto mayor es el proporcionado por el mecanismo estado del usuario, el cual permite, a través de sensores, identificar problemas o cambios en algún usuario.

### 3.2.3. Servidor

#### Gestor de posición

*Almacenamiento de mapas.* Este mecanismo permite la persistencia del mapa que describe el entorno de acción, el cual se ingresa al sistema por medio de la interfaz de creación de mapas, ubicada en el gestor de interfaces.

*Ubicación de robots.* Los marcadores que permiten al robot navegar en el entorno contienen un identificador, el cual facilita determinar el área en la que se encuentra el robot, por lo tanto, en este mecanismo se procesan las imágenes provenientes del robot a fin de identificar el área en la que se encuentra.

*Ubicación de usuarios.* Establece un método de comunicación entre el servidor y el dispositivo del usuario, con el objetivo de informar al sistema la localización de los diferentes usuarios participantes en el entorno gestionado.

*Generador de rutas.* El generador de rutas proporciona al gestor de actividades los movimientos necesarios para que el robot se desplace entre las diferentes áreas, a modo de localizar a un usuario e interactuar con él.

#### Gestor de interfaces

*Gestión de actividades.* Las interfaces para la gestión de actividades permiten al personal responsable (médico o familiares) del paciente agendar citas y programar los recordatorios que deberá entregar el robot.

*Gestión de medicamentos.* El mecanismo de gestión de medicamentos proporciona las interfaces necesarias para programar los medicamentos que el adulto mayor debe tomar, el periodo de tiempo que debe hacerlo, la frecuencia a la que debe tomarlo, y las indicaciones al momento de la ingesta.

*Parámetros de configuración.* Los parámetros de configuración permiten ajustar el comportamiento del sistema, por lo que en este conjunto de interfaces se anexa información bajo la cual se describe el entorno y el funcionamiento del sistema, por ejemplo el mapa del entorno y descripción de marcadores.

#### Gestor de actividades

*Generador de mensajes.* El generador de mensajes es el proceso encargado de determinar el momento adecuado para entregar los mensajes, decisión basada en la ubicación de los usuarios y del robot.

*Consulta de actividades.* Así como es necesario almacenar el mapa del entorno, también se requiere que la agenda de los adultos mayores sea persistente, pues en caso se fallo, no se debe ser necesario ingresar nuevamente las tareas programadas con anterioridad. Almacenadas las actividades, es necesario un proceso de monitoreo que evalúe la información y determine el momento en el cual debe ser procesada la información almacenada.

### 3.2.4. Conexiones

La interacción entre los diferentes componentes del sistema varía acorde con los requerimientos establecidos por cada tipo de comunicación y la información requerida por el componente, por lo cual, se presentan los diferentes tipos de conexiones utilizadas por el sistema.

Envío imágenes. Este tipo de conexión se realiza entre los mecanismos de navegación en el ambiente y ubicación de robots. Para proporcionar al servidor las imágenes necesarias que permitan determinar el identificador presente en los marcadores. Esto con el objetivo de evitar que el robot realice un mayor procesamiento, por consiguiente, en esta comunicación el robot envía un conjunto de imágenes y el servidor regresa como respuesta las acciones que el robot debe realizar.

Envío de señales. Como el dispositivo portado por el usuario, encargado de la ubicación y monitoreo del mismo no cuenta con alto poder de procesamiento, se limita a detectar magnitudes en el entorno (por ejemplo la intensidad de una señal), y a enviarlas al gestor de posición para su posterior análisis, lo que permite establecer el estado del usuario o la posición que este tiene en el entorno.

Envío de ubicaciones. Una vez identificadas por el gestor de posición las ubicaciones de los usuarios y del robot dentro del entorno inteligente, la información permanece disponible para que el gestor de actividades acceda tanto a la ubicación en la que se encuentran los usuarios, como los posibles caminos que el robot debe seguir para interactuar con cada uno de ellos. Por esta razón, el gestor de posición y el gestor de actividades mantienen una comunicación bidireccional en la cual se transfieren datos relacionados con la navegación y ubicación.

Mensajes al robot. Otro método de comunicación entre el robot y el servidor, además del envío de imágenes, es el envío y recepción de textos. Este tipo de mensajes permite la sincronía entre los procesos del servidor y del robot ya que con ellos se coordina la entrega de los recordatorios para los usuarios, la dirección a la que debe dirigirse el robot, cambios en las acciones del robot así como la verificación del estado del mismo.



### 3.3. Robot de servicio

Anteriormente se presentan las acciones desempeñadas por el robot al igual que la interacción que tiene con los otros elementos presentes en SiGA, sin embargo, en esta sección se describe a mayor detalle las tareas desempeñadas así como los elementos que lo componen, por ello, se presenta al robot como un sistema independiente que se encuentra en constante interacción con el resto de subsistemas presentes en SiGA.

#### 3.3.1. Diagrama de casos de uso

Un caso de uso (CU) permite la descripción de una tarea en particular desempeñada por un sistema, la cual busca cumplir un objetivo específico, sin embargo, para comprender el funcionamiento de un sistema es necesario analizar la relación entre las múltiples tareas existentes y la interacción con elementos externos a dicho sistema [25]. Por otra parte, un actor es un rol desempeñado por una entidad que interactúa con el sistema en cuestión, por ejemplo una persona, organización u otro sistema [25].

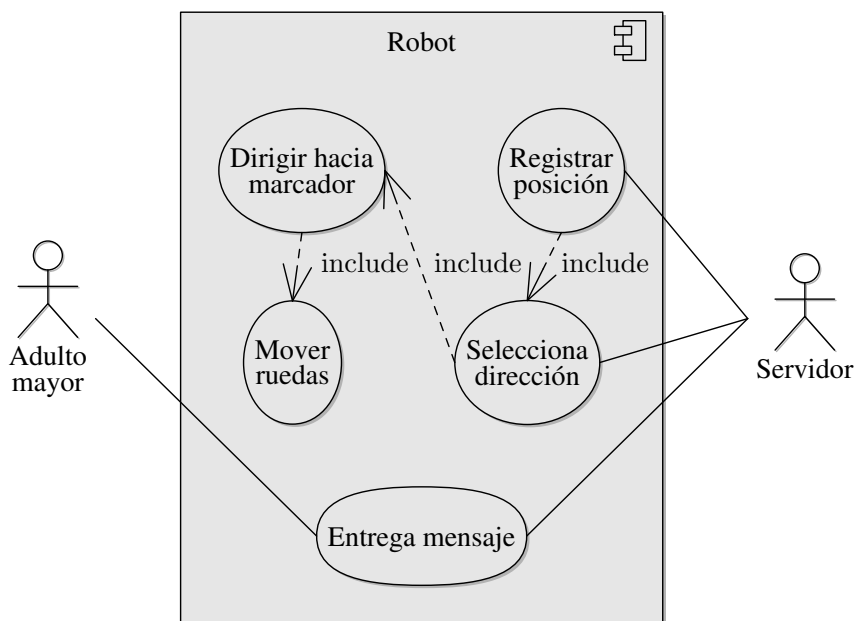


Figura 3.2: Diagrama de casos de uso del robot

Por ello, el diagrama de casos de uso permite establecer una vista general del comportamiento de un sistema, a través de la definición de los casos de uso presentes en él, las relaciones entre dichos casos y los actores que intervienen en ellos [1, 12]. Por lo anterior, se presenta en la figura 3.2 el diagrama de casos de uso correspondiente al funcionamiento del correspondiente al funcionamiento del robot.

### 3.3.2. Descripción de casos de uso

Tabla 3.1: Descripción del caso de uso dirigir hacia el marcador

<b>ID</b>	<b>CU-R1</b>																				
<b>Título</b>	<b>Dirigir hacia el marcador</b>																				
Breve descripción	Presenta el proceso a través del cual el robot, ubicado en determinada posición, ejecuta los movimientos necesarios para llegar a un marcador.																				
Actores	Ninguno																				
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>▪ Debe existir un marcador en el entorno que puede ser captado por la cámara.</li> <li>▪ Es conveniente que la cámara del robot esté dirigida hacia el marcador.</li> <li>▪ El entorno permite el desplazamiento del robot hacia el marcador.</li> </ul>																				
<b>Postcondiciones</b>	<ul style="list-style-type: none"> <li>▪ El robot se ubica frente al marcador destino, disponible para la capturar de imágenes que permitan la identificación del marcador.</li> </ul>																				
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><i>Paso</i></th> <th><i>Acción</i></th> </tr> </thead> <tbody> <tr><td>1.</td><td>El robot captura una imagen.</td></tr> <tr><td>2.</td><td>El robot selecciona los objetos de color magenta en el marcador.</td></tr> <tr><td>3.</td><td>El robot clasifica los objetos por forma, seleccionado los marcadores.</td></tr> <tr><td>4.</td><td>Se seleccionan el objeto con mayor área.</td></tr> <tr><td>5.</td><td>Se verifica con base al área si el robot se encuentra cerca del marcador.</td></tr> <tr><td>6.</td><td>Se verifica la posición del marcador dentro de la imagen.</td></tr> <tr><td>7.</td><td>Se determina la dirección del robot.</td></tr> <tr><td>8.</td><td>Se mueve el robot a la dirección seleccionada.</td></tr> <tr><td>9.</td><td>Regresa al paso inicial.</td></tr> </tbody> </table>	<i>Paso</i>	<i>Acción</i>	1.	El robot captura una imagen.	2.	El robot selecciona los objetos de color magenta en el marcador.	3.	El robot clasifica los objetos por forma, seleccionado los marcadores.	4.	Se seleccionan el objeto con mayor área.	5.	Se verifica con base al área si el robot se encuentra cerca del marcador.	6.	Se verifica la posición del marcador dentro de la imagen.	7.	Se determina la dirección del robot.	8.	Se mueve el robot a la dirección seleccionada.	9.	Regresa al paso inicial.
<i>Paso</i>	<i>Acción</i>																				
1.	El robot captura una imagen.																				
2.	El robot selecciona los objetos de color magenta en el marcador.																				
3.	El robot clasifica los objetos por forma, seleccionado los marcadores.																				
4.	Se seleccionan el objeto con mayor área.																				
5.	Se verifica con base al área si el robot se encuentra cerca del marcador.																				
6.	Se verifica la posición del marcador dentro de la imagen.																				
7.	Se determina la dirección del robot.																				
8.	Se mueve el robot a la dirección seleccionada.																				
9.	Regresa al paso inicial.																				
<b>Secuencias alternativas</b>	<table border="1"> <thead> <tr> <th colspan="2">Robot perdido</th> </tr> <tr> <th><i>Paso</i></th> <th><i>Acción</i></th> </tr> </thead> <tbody> <tr><td>1.</td><td>El robot captura una imagen.</td></tr> <tr><td>2.</td><td>El robot no encuentra objetos de color magenta.</td></tr> <tr><td>3.</td><td>El robot robot gira hasta encontrar un marcador que le permita orientarse.</td></tr> <tr><td>4.</td><td>El robot continua con la ejecución normal.</td></tr> </tbody> </table>	Robot perdido		<i>Paso</i>	<i>Acción</i>	1.	El robot captura una imagen.	2.	El robot no encuentra objetos de color magenta.	3.	El robot robot gira hasta encontrar un marcador que le permita orientarse.	4.	El robot continua con la ejecución normal.								
Robot perdido																					
<i>Paso</i>	<i>Acción</i>																				
1.	El robot captura una imagen.																				
2.	El robot no encuentra objetos de color magenta.																				
3.	El robot robot gira hasta encontrar un marcador que le permita orientarse.																				
4.	El robot continua con la ejecución normal.																				
<b>Excepciones:</b>	<table border="1"> <tr> <td>En caso de no encontrar ningún marcador después de una búsqueda, el robot detendrá el proceso e informará el error.</td> </tr> </table>	En caso de no encontrar ningún marcador después de una búsqueda, el robot detendrá el proceso e informará el error.																			
En caso de no encontrar ningún marcador después de una búsqueda, el robot detendrá el proceso e informará el error.																					

La tabla 3.1 describe el CU *dirigir hacia el marcador* (CU-R1), el cual se relaciona con el CU *mover ruedas* (CU-R2), descrito en la tabla 3.2, a través de una relación de inclusión. La relación indica que CU-R1 hace uso de CU-R2 para realizar sus tareas, es decir, cuando se menciona el movimiento del robot en CU-R1, se hace referencia a las tareas realizadas por CU-R2, .

Tabla 3.2: Descripción del caso de uso mover ruedas

<b>ID</b>	<b>CU-R2</b>										
<b>Título</b>	<b>Mover ruedas</b>										
Breve descripción	Recibida la configuración de las ruedas se realiza el proceso de accionamiento de las ruedas. Los movimientos posibles para cada rueda son: avanzar, detener y retroceder.										
Actores	Ninguno										
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>▪ El dispositivo que acciona los motores que mueven las ruedas del robot se conecta al controlador principal del robot a través del puerto USB, por lo que dicha conexión debe estar activa.</li> </ul>										
<b>Postcondiciones</b>	<ul style="list-style-type: none"> <li>▪ Las ruedas se encuentran en el estado indicado por el controlador.</li> </ul>										
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><i>Paso</i></th> <th><i>Acción</i></th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>El controlador realiza el enlace entre el gestor de movimiento y el gestor principal del robot a través del puerto USB.</td> </tr> <tr> <td>2.</td> <td>Se coloca a la espera de una orden.</td> </tr> <tr> <td>3.</td> <td>Se recibe la orden y codifican en movimientos.</td> </tr> <tr> <td>4.</td> <td>Se envía la señal de activación.</td> </tr> </tbody> </table>	<i>Paso</i>	<i>Acción</i>	1.	El controlador realiza el enlace entre el gestor de movimiento y el gestor principal del robot a través del puerto USB.	2.	Se coloca a la espera de una orden.	3.	Se recibe la orden y codifican en movimientos.	4.	Se envía la señal de activación.
<i>Paso</i>	<i>Acción</i>										
1.	El controlador realiza el enlace entre el gestor de movimiento y el gestor principal del robot a través del puerto USB.										
2.	Se coloca a la espera de una orden.										
3.	Se recibe la orden y codifican en movimientos.										
4.	Se envía la señal de activación.										
<b>Excepciones:</b>											
	En caso de no tener una conexión con el puerto UBS, se informa del error.										

La tabla 3.3 describe el CU *registrar posición* (CU-R3), donde se indica al servidor el área en la que se encuentra el robot. Este proceso es requerido por el CU *selecciona dirección* (CU-R4), presente en la tabla 3.4, pues una vez se informa al servidor una nueva dirección para el robot, este decide la ruta que deberá seguir.

Tabla 3.3: Descripción del caso de uso registrar posición

<b>ID</b>	<b>CU-R3</b>										
<b>Título</b>	<b>Registrar posición</b>										
Breve descripción	Describe el proceso de registro en una nueva ubicación, es decir, el proceso ocurrido una vez que el robot alcanza alguno de los marcadores en el entorno										
Actores	Servidor										
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>El servidor está a la espera de recepción de imágenes.</li> </ul>										
<b>Postcondiciones</b>	<ul style="list-style-type: none"> <li>El servidor determina el marcador más cercano al robot.</li> <li>El robot queda a la espera de ordenes.</li> </ul>										
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><i>Paso</i></th> <th><i>Acción</i></th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>El robot captura un conjunto de imágenes.</td> </tr> <tr> <td>2.</td> <td>El servidor activa el puerto de recepción.</td> </tr> <tr> <td>3.</td> <td>El robot envía las imágenes al servidor.</td> </tr> <tr> <td>4.</td> <td>El robot queda a la espera de nuevas ordenes.</td> </tr> </tbody> </table>	<i>Paso</i>	<i>Acción</i>	1.	El robot captura un conjunto de imágenes.	2.	El servidor activa el puerto de recepción.	3.	El robot envía las imágenes al servidor.	4.	El robot queda a la espera de nuevas ordenes.
<i>Paso</i>	<i>Acción</i>										
1.	El robot captura un conjunto de imágenes.										
2.	El servidor activa el puerto de recepción.										
3.	El robot envía las imágenes al servidor.										
4.	El robot queda a la espera de nuevas ordenes.										
<b>Excepciones:</b>	En caso de no tener una respuesta del servidor el robot permanecerá a la espera.										

Tabla 3.4: Descripción del caso de uso registrar posición

<b>ID</b>	<b>CU-R4</b>
<b>Título</b>	<b>Selecciona dirección</b>
Breve descripción	Ocurre cuando el servidor determina la dirección que el robot debe seguir.
Actores	Servidor
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>El robot envió un conjunto de imágenes al servidor.</li> </ul>
<b>Postcondiciones</b>	<ul style="list-style-type: none"> <li>El robot inicia el proceso “Dirigir hacia el marcador con un nuevo objetivo”</li> </ul>

*Continúa en la siguiente página*

Tabla 3.4 – Continuación de la página previa

<b>Secuencia normal</b>	
<i>Paso</i>	<i>Acción</i>
1.	El robot recibe una solicitud.
2.	El robot inicial la configuración de giro.
3.	El robot captura imágenes mientras gira.
4.	Se busca un marcador en las imágenes capturadas.
5.	Se fija el marcador objetivo para el CU-R1.
<b>Excepciones:</b>	
En caso de no recibir instrucciones el robot debe permanecer en posición.	

El caso de uso *entregar mensaje* (CU-R5), descrito en la 3.5, es el encargado de la entrega de mensajes. Los mensajes son entregados por el servidor y sintetizados a voz por el robot, por ello es necesario que exista relación entre el sistema y el adulto mayor.

Tabla 3.5: Descripción del caso de uso entregar mensaje

<b>ID CU-R5</b>	
<b>Título</b>	<b>Entregar mensaje</b>
Breve descripción	Es el proceso encargado codificar a voz los mensajes para el usuario.
Actores	Servidor, Adulto mayor.
<b>Precondiciones</b>	
<ul style="list-style-type: none"> <li>▪ El robot se encuentra en el área requerida.</li> </ul>	
<b>Secuencia normal</b>	
<i>Paso</i>	<i>Acción</i>
1.	El robot se ubica en un área determinada y permanece a la espera de ordenes.
2.	El servidor envía el mensaje en forma de texto.
3.	El robot recibe el mensaje y codifica dicho mensaje a voz.
<b>Excepciones:</b>	
En caso de no recibir ninguna orden, el robot detendrá el proceso e informará el error.	

## 3.4. Servidor

### 3.4.1. Casos de uso

El diagrama de casos de uso del servidor, presente en la figura 3.3 busca describir el funcionamiento dinámico del sistema encargado de administrar las acciones del servidor. Al igual que el diagrama de casos de uso del robot, el diagrama del servidor muestra las interacciones del servidor con los actores con los que se comunica, como lo son: los usuarios externos al entorno ubicuo (médico y familiares del adulto mayor), el robot y los adultos mayores.

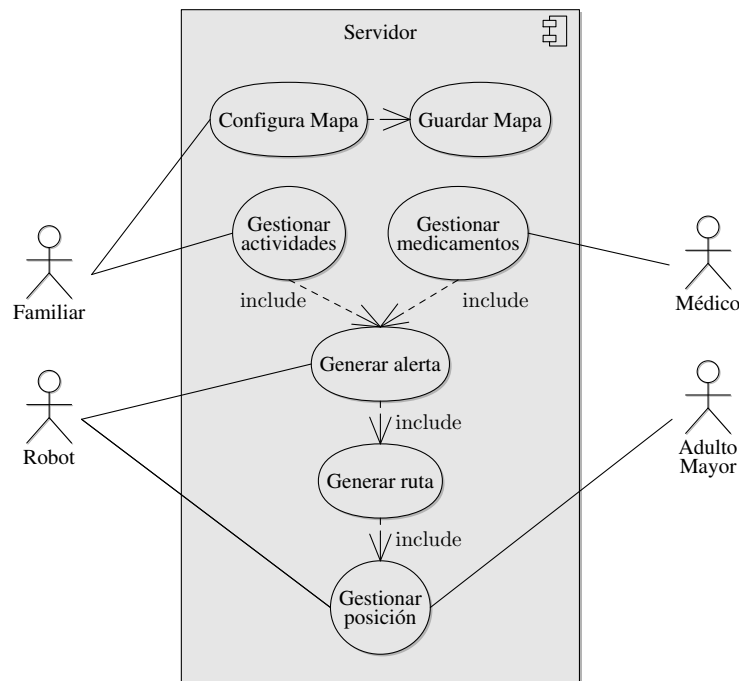


Figura 3.3: Diagrama de casos de uso del servidor

El primer caso de uso, *configurar mapa* (CU-S1), describe el proceso a través del cual se configuran los mapas que representan el entorno de navegación. Este proceso incluye la creación, modificación y actualización del mapa.

Tabla 3.6: Descripción del caso de uso configurar mapa

<b>ID</b>	<b>CU-S1</b>
<b>Título</b>	<b>Configurar mapa</b>
	Breve descripción
	Este caso de uso describe el proceso a través del cual se crea, configura y modifica el mapa que describe el entorno.

*Continúa en la siguiente página*

Tabla 3.6 – Continuación de la página previa

Actores	
Familiar.	
<b>Precondiciones</b>	
<ul style="list-style-type: none"> <li>■ El familiar, o la persona responsable de la gestión del servidor tiene acceso a la interfaz de configuración de mapas.</li> </ul>	
<b>Postcondiciones</b>	
<ul style="list-style-type: none"> <li>■ El mapa debe contener las áreas que describan el entorno de navegación del robot.</li> </ul>	
<b>Secuencia normal</b>	
<i>Paso</i>	<i>Acción</i>
1.	El familiar ingresa el nombre del área.
2.	El familiar selecciona las figuras que se encuentran en el marcador.
3.	El familiar presiona el botón Agregar área.
<b>Secuencias alternativas</b>	
Agregar camino	
<i>Paso</i>	<i>Acción</i>
1.	Clic derecho sobre el área origen.
2.	Clic sobre el área destino.
3.	Seleccionar el tipo de camino correspondiente (Izquierdo o Derecho).
Eliminar camino	
<i>Paso</i>	<i>Acción</i>
1.	Clic derecho sobre el camino a eliminar.
2.	El sistema elimina el camino.
Eliminar área	
<i>Paso</i>	<i>Acción</i>
1.	Doble clic sobre el área a eliminar.
2.	Confirmar la acción presionando el botón aceptar.
3.	El sistema elimina los caminos relacionados al área, después elimina el área.
<b>Excepciones:</b>	
En caso de existir un mapa previamente configurado, la interfaz de gestión lee la información almacenada en la base de datos, y presenta dicha información.	

El mapa utilizado por el sistema es el que se encuentra almacenado en la base de datos, por lo que el caso de uso *guardar mapa*, presente en la tabla 3.7 tiene como objetivo actualizar la base de datos con la información obtenida desde la interfaz de configuración.

Tabla 3.7: Descripción del caso de uso guardar mapa

<b>ID</b>	<b>CU-S2</b>										
<b>Título</b>	<b>Guardar mapa</b>										
Breve descripción	Guarda la información obtenida del proceso de configuración de mapa.										
Actores	Ninguno.										
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>▪ El mapa previamente configurado en la interfaz.</li> </ul>										
<b>Postcondiciones</b>	<ul style="list-style-type: none"> <li>▪ El mapa almacenado en la base de datos.</li> </ul>										
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><i>Paso</i></th> <th><i>Acción</i></th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>El familiar presiona el botón de guardar mapa en la interfaz.</td> </tr> <tr> <td>2.</td> <td>La interfaz verifica que todos los campos se ingresaron.</td> </tr> <tr> <td>3.</td> <td>La interfaz empaqueta la información para su envío.</td> </tr> <tr> <td>4.</td> <td>El servidor actualiza la información en la base de datos.</td> </tr> </tbody> </table>	<i>Paso</i>	<i>Acción</i>	1.	El familiar presiona el botón de guardar mapa en la interfaz.	2.	La interfaz verifica que todos los campos se ingresaron.	3.	La interfaz empaqueta la información para su envío.	4.	El servidor actualiza la información en la base de datos.
<i>Paso</i>	<i>Acción</i>										
1.	El familiar presiona el botón de guardar mapa en la interfaz.										
2.	La interfaz verifica que todos los campos se ingresaron.										
3.	La interfaz empaqueta la información para su envío.										
4.	El servidor actualiza la información en la base de datos.										
<b>Excepciones:</b>	En caso de existir un error en la información correspondiente al mapa, se informará por medio de la interfaz, el motivo por el cual se genera el error.										

Otra de las tareas necesarias para el funcionamiento del servidor es el almacenamiento y gestión de las actividades, así como la gestión de medicamentos. Por lo anterior, el caso de uso *gestionar actividades* (CU-S3) descrito por la tabla 3.8, describe la programación de citas y tareas que requiere recordar al adulto mayor. Para los recordatorios relacionados con medicamentos, el médico puede agendar las fechas en las que el usuario debe realizar la toma de medicamentos a través del proceso descrito por el caso de uso *gestionar medicamentos*, descrito por la tabla 3.9 (CU-S4).

Tabla 3.8: Descripción del caso de uso gestionar actividades

<b>ID</b>	<b>CU-S3</b>
<b>Título</b>	<b>Gestionar actividades</b>
Breve descripción	Permite al médico o familiar del paciente generar un recordatorio para el adulto mayor, a través de una interfaz.
Actores	Familiar, médico

*Continúa en la siguiente página*



Tabla 3.8 – Continuación de la página previa

<b>Precondiciones</b>	
<ul style="list-style-type: none"> <li>▪ El gestor de base de datos debe encontrarse activo para almacenar los datos correspondientes.</li> </ul>	
<b>Secuencia normal</b>	
<i>Paso</i>	<i>Acción</i>
1.	El usuario de la interfaz escribe una breve descripción de la cita, esta descripción será usada para generar el recordatorio.
2.	El usuario selecciona la fecha y hora a la que ocurrirá la cita.
3.	El usuario presiona el botón de guardado.
4.	El sistema almacena la información proporcionada por el usuario.
<b>Excepciones:</b>	
En caso de fallo en el almacenamiento de la información, o de información incompleta, la interfaz debe mencionar al usuario sobre el error ocurrido.	

Tabla 3.9: Descripción del caso de uso gestionar medicamentos

<b>ID</b>	<b>CU-S4</b>
<b>Título</b>	<b>Gestionar medicamentos</b>
Breve descripción	Gestionar medicamentos permite al médico del adulto mayor agregar la información correspondiente a los medicamentos que debe tomar.
Actores	Médico.
<b>Precondiciones</b>	
<ul style="list-style-type: none"> <li>▪ El gestor de base de datos debe encontrarse activo para almacenar los datos correspondientes.</li> </ul>	
<b>Secuencia normal</b>	
<i>Paso</i>	<i>Acción</i>
1.	El médico debe de ingresar el nombre del medicamento.
2.	El médico ingresa la fecha de inicio del del tratamiento.
3.	El médico ingresa la duración del tratamiento en días.
4.	Es necesario seleccionar la frecuencia con la que se debe ingerir el medicamento.
5.	También se solicita una hora opcional en la que es recomendable que se realice la ingesta del medicamento.
6.	Se presiona el botón <i>Guardar</i> para guardar la información ingresada.
<b>Excepciones:</b>	
En caso de fallo en el almacenamiento de la información, o de información incompleta, la interfaz debe mencionar al usuario sobre el error ocurrido.	

### 3.4.2. Base de datos

Un diagrama entidad-relación (ER) es llamado así porque permite la organización de los datos a través de entidades, mientras expresa la conexión entre dichos datos por medio de relaciones [21]. Este diagrama es el punto de partida en la creación de bases de datos relacionales pues las entidades se migran hacia tablas y las relaciones se representan a través de las claves foráneas de estas tablas. Por lo anterior, en la figura 3.4 se muestra el diagrama ER correspondiente al diseño de la base de datos utilizada en el componente servidor del sistema SiGA.

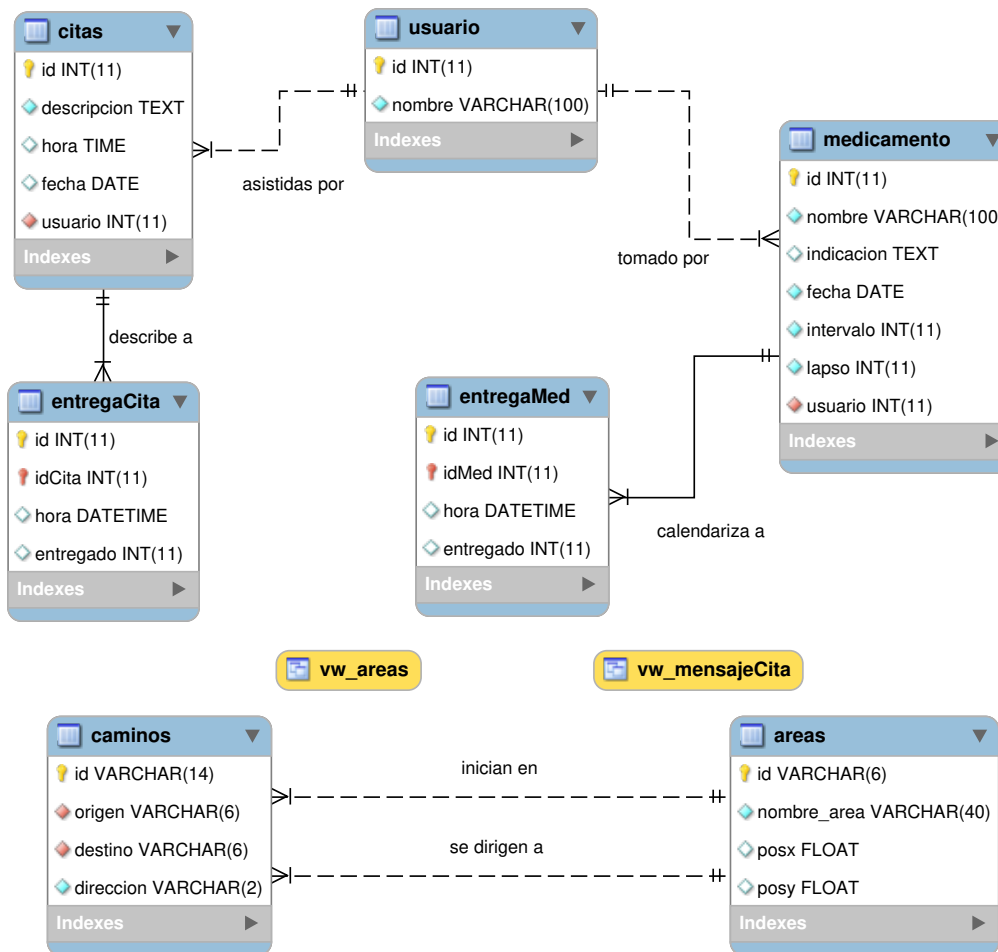


Figura 3.4: Diagrama entidad-relación de la base de datos del sistema

Las entidades utilizadas en el sistema son siete: *usuario*, *citas*, *medicamento*, *entregaCita*, *entregaMed*, *caminos* y *áreas*. *Usuario* representa a los adultos mayores encontrados en el entorno gestionado por SiGA. *Citas* almacena la información correspondiente a los eventos próximos en los que participará el usuario, tales como citas con el médico o reuniones

familiares. La entidad *medicamento* guarda información correspondiente a la medicación de los usuarios, almacenando datos como el nombre del medicamento, las indicaciones del médico al igual que las fechas necesarias para construir un horario de medicación. Al crear una nueva cita o recetar un medicamento, se genera un conjunto de mensajes o recordatorios para cada evento, estos mensajes son los que serán entregados por el robot, por ello, las entidades *entregaCita* y *entregaMed* permiten almacenar esta información. Las entidades llamadas *áreas* y *caminos* permiten almacenar el mapa de navegación, ya que, *áreas* almacena, como su nombre lo indica, las áreas que en las cuales se encuentran los diferentes marcadores, por otra parte, y un camino establece una conexión entre dos áreas, por ello, la entidad *caminos* contiene los datos necesarios para almacenar estas conexiones.

Las relaciones utilizadas por el diagrama ER son seis: *asistidas por*, *tomado por*, *describe a*, *calendariza a*, *inicia en*, y *se dirige a*. La relación *asistida por* se establece entre citas y usuarios, indicando que un usuario asiste a muchas citas, es decir, es una relación uno a varios. En la relación *tomado por* indica que un medicamento es tomado por un usuario, sin embargo, al ser expresado como una relación del tipo uno a muchos, indica que un usuario puede tomar uno o más medicamentos. Las relaciones *inicia en* y *se dirige a* permiten la construcción del mapa correspondiente al entorno, ya que almacenadas las áreas, el nodo inicial de un camino se representa con la relación *inicia en*, de igual forma, el final del camino se especifica con la relación *se dirige a*.

### 3.5. Ubicación y navegación del robot en el entorno

Los sistemas de posicionamiento en interiores (IPS) permiten ubicar objetos o personas dentro de un entorno determinado. Este tipo de sistemas de posicionamiento son utilizados en escenarios donde las características de precisión requeridas no permiten el uso de mecanismos como el sistema de posicionamiento global (GPS). Los IPS determinan la ubicación a través de magnitudes de referencia [6] obtenidas a partir de elementos presentes en el entorno (punto de referencia) y puede ser expresada a través de sistemas basados en coordenadas, o por medio de referencias simbólicas (“sala”, “comedor”) [35].

Los puntos de referencia utilizados por los IPS varían acorde con las características del sistema, ofreciendo diferentes magnitudes físicas a ser evaluadas, como lo son, señales de radio, la cantidad de luz o mediante el análisis de objetos en imágenes [31].

Es posible obtener diferentes niveles de exactitud al medir la posición de un objeto utilizando un IPS, pues depende de las características y fiabilidad del análisis de los puntos de referencia, sin embargo, el uso de sistemas IPS con puntos de referencia complejos suelen incrementar el costo económico de implementación, por consiguiente, para cumplir con el objetivo de crear un sistema de bajo costo, los nodos utilizados en el sistema SiGA son puntos de referencia visuales, es decir, elementos en el entorno que pueden ser analizados a través de imágenes capturadas por el robot y de los cuales es posible la extracción de información que permita ubicar al robot en una zona determinada.

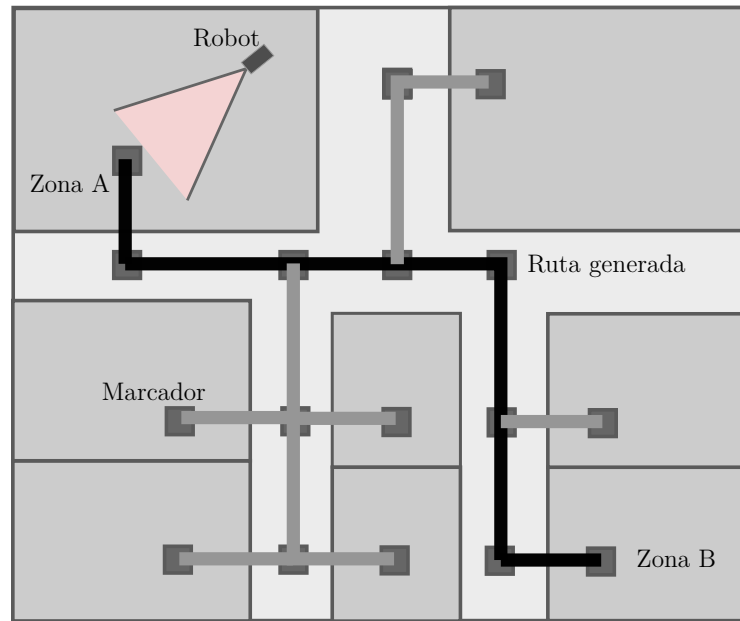


Figura 3.5: Ubicación de los marcadores en el entorno

La figura 3.5 presenta, a manera de ejemplo, la forma en la que los puntos de referencia identifican las diferentes áreas que componen al entorno gestionado por el sistema SiGA. De igual forma, es posible observar los caminos que conectan a dichas áreas; cada uno de estos caminos se pueden establecer si, ubicado el robot dentro de un área  $A$  puede observar al marcador que identifica al área  $B$ , entonces se dice que existe un camino  $AB$ . Es importante mencionar que, aunque exista el camino  $AB$  esto no quiere decir que el camino  $BA$  también exista, por ello, al momento de crear el grafo que representa al mapa de navegación del entorno, se debe considerar un grafo dirigido.

El procedimiento de navegación y posicionamiento del robot se compone por dos tareas principales: la selección de rutas y el desplazamiento del robot hacia marcadores. La selección de rutas es una tarea de gestión desempeñada por el servidor, la cual consiste en identificar el área en la que se encuentra el robot a partir de un grupo de imágenes en las cuales se encuentran los puntos de referencia que representan a dicha área. De ser necesario, esta tarea también le indica al robot la dirección en la que debe moverse para encontrar un segundo marcador, es decir, selecciona el camino que debe tomar.

Por otra parte, en desplazamiento del robot hacia el marcador es una tarea desempeñada por el robot, la cual consiste en, capturar imágenes del entorno, identificar posición en la que se encuentra el punto de referencia y generar las ordenes necesarias para desplazar al robot hacia dicho punto, calibrando los movimientos del robot para evitar el desvío del mismo.

### 3.5.1. Diseño del marcador

Debido a la gran cantidad de información obtenida a través de una cámara, el marcador, también llamado punto de referencia, debe resaltar en el entorno para su fácil filtrado. Esto conlleva un diseño que permita la fácil identificación y clasificación del marcador dentro del conjunto de elementos encontrados en las imágenes capturadas por el robot, de igual forma, el diseño debe facilitar la codificación de la información que dicho marcador contiene.

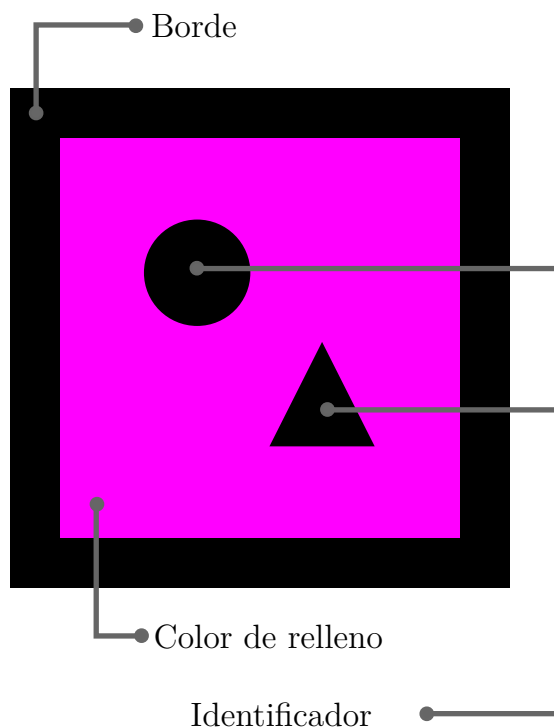


Figura 3.6: Diseño propuesto para el marcador.

En la figura 3.6, se presenta el diseño propuesto para el marcador de posición utilizado por el sistema SiGA, el cual cuenta con tres características principales.

- **Color de resalte.** Esta característica permite seleccionar un conjunto de objetos en el cual se encuentran los marcadores del sistema, pues, en la primera etapa del proceso de búsqueda de marcadores en las imágenes se seleccionan los objetos que coincidan con el color de resalte del marcador. Por esta razón es importante que el color de resalte no sea común en el entorno donde se capturan las imágenes, ya que esto permite reducir el número de objetos evaluados en etapas posteriores del proceso mencionado.

- **Borde.** Debido a que el color de resalte no asegura que todos los objetos seleccionados sean marcadores es necesaria una segunda clasificación, en la cual se evalúen los objetos por forma, esta clasificación es posible gracias al *borde*, pues además de delimitar el marcador, proporciona una forma básica que permite distinguir los marcadores de otros objetos.
- **Identificador.** Es el elemento a partir del cual se determina la posición del robot en el entorno, ya que cada identificador es único y se encuentra ligado a una zona específica del ambiente. Por este motivo, el identificador se conforma por una o más figuras específicas que permitan su fácil codificación.

# Implementación de robot

## 4.1. Componentes

La selección de los elementos que se usaron para construir el robot está guiada por dos objetivos principales: la reducción de costos y la funcionalidad requerida. Por ello se optó por construir un robot con componentes de bajo costo, que permitan el desplazamiento del robot en áreas donde solo se requiera de movimientos básicos, es decir, moverse hacia adelante, retroceder, giros hacia la izquierda y giros hacia la derecha.

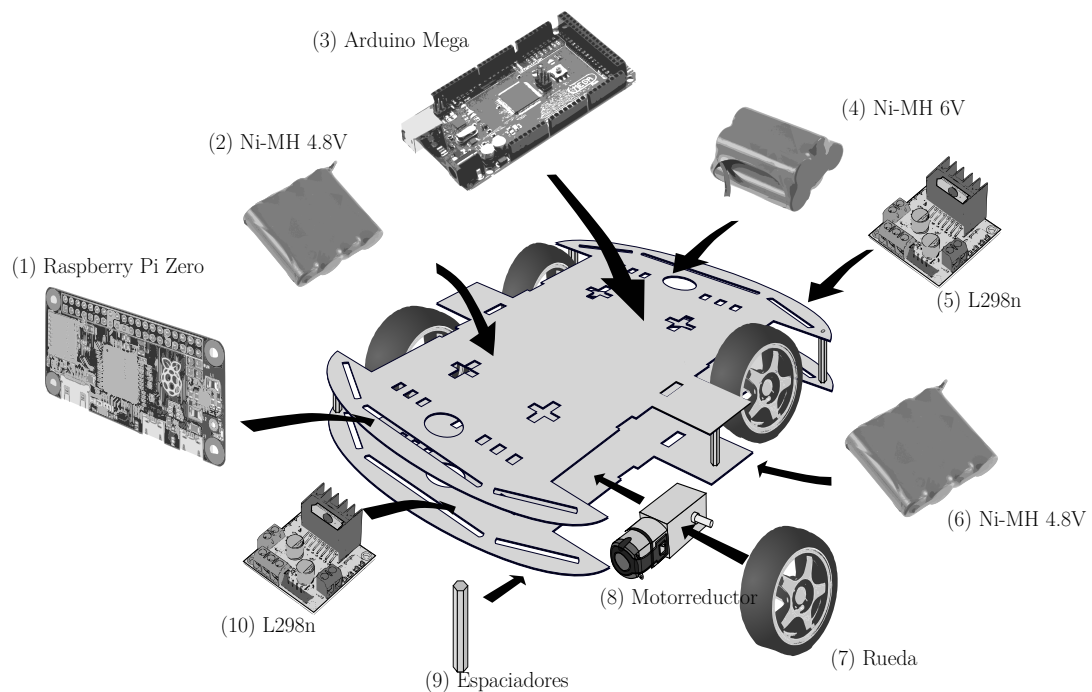


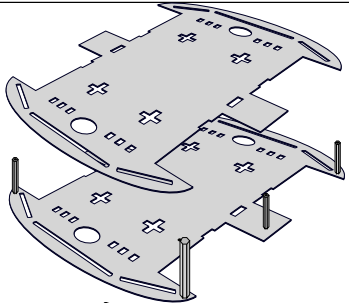
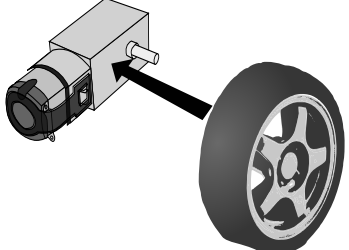
Figura 4.1: Componentes del robot

El primer componente seleccionado es un chasis con tracción en las cuatro ruedas, impulsado por cuatro motorreductores B01 1:48. Los motorreductores son alimentados con un paquete de baterías NiMH de 6.0V, y accionados a través de dos controladores duales para motores L298N. La señal de activación de los motores se genera mediante una tarjeta *Arduino Mega 2560*, la cual se encarga de codificar las ordenes producidas por el controlador principal del robot, una computadora *Raspberry Pi Zero W*. Ambas tarjetas son alimentadas por dos paquetes de baterías de cuatro celdas NiMH con 4.8 v.

Para ubicar los componentes antes mencionados, las dos placas de acrílico que componen el chasis son separadas por un conjunto de espaciadores de metal, esto crea dos niveles en los cuales se colocan todos los componentes. En el nivel inferior se ubican los componentes que permiten el desplazamiento del robot, como lo son: los cuatro motorreductores que soportan a las llantas, los dos controladores de los motorreductores, y un paquete de baterías de 4.8 v. En el nivel superior se colocan los componentes de procesamiento: la tarjeta raspberry con su correspondiente cámara, la arduino mega, que permite generar las ordenes para los controladores, un paquete de baterías de 4.8 v. y un paquete de baterías de 6.0 v., tal y como se muestra en la figura 4.1.

La tabla 4.1 contiene una breve descripción de los componentes antes mencionados a través de la presentación de las características principales de cada uno de ellos.

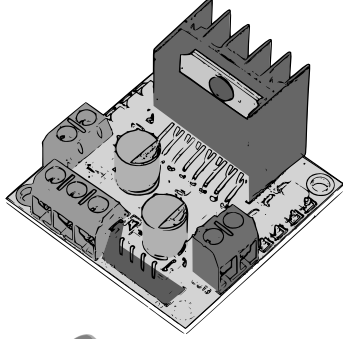
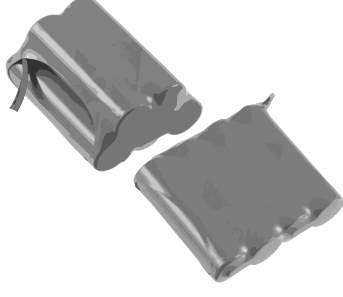
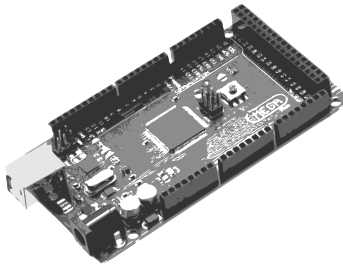
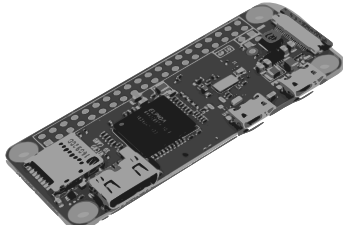
Tabla 4.1: Descripción de los componentes del robot

	<p><i>Chasis 4WD</i> El chasis del robot consta de dos placas de acrílico de 26cm x 17.5cm con 3mm de espesor, seis separadores de 3.5 cm.</p>
	<p><i>Motorreductor con caja reductora 6v 1:48</i> Los motorreductores tienen un voltaje de operación de 6v a 8v, velocidad de 175 rpm, con un tamaño de 7cm x 2.2cm x 1.8cm. Las llantas utilizadas son de plástico y tienen un diámetro de 6cm con 2.6cm de ancho.</p>

*Continúa en la siguiente página*



Tabla 4.1 – Continuación de la página previa

	<p><i>Controlador dual para motores L298N</i></p> <p>El módulo L298N permite controlar dos motores de corriente continua, por lo cual se requiere de dos componentes para tener un control independiente para cada motor. L298N es alimentado con con 6 a 12v, y proporciona cuatro salidas, dos para cada motor, al igual que cuatro puertos de control.</p>
	<p><i>Paquetes de baterías NiMH</i></p> <p>Los paquetes de baterías utilizados están compuestos por celdas de 1.2 v. Por ello, los utilizados en el robot proporcionan 4.8 v. y 6.0 v. Estos paquetes de baterías proporcionan 2200 mAh a través de un conector JR hembra con 3 pines.</p>
	<p><i>Tarjeta Arduino Mega 2560</i></p> <p>Contiene 54 pines digitales que pueden ser usados como entradas o salidas, de los cuales 15 proporcionan una salida PWM. Esto permite agregar diversos sensores al robot, lo que brinda la capacidad de agragar nuevas funciones al robot. Utiliza 5 v. como voltaje de operación, y tiene como medidas 10.1 cm de largo y 5.3 cm de ancho.</p>
	<p><i>Raspberry Pi Zero W</i></p> <p>El controlador principal del robot es la computadora Raspberry Pi Zero, la cual cuenta con un CPU single-core, 512 MB de RAM, Bluetooth 4.1, conexión inalámbrica 802.11 b/g/n, conector CSI para cámara, puerto micro USB, puerto mini HDMI, así como 40 pines GPIO.</p>

### 4.1.1. Conexión

Se ha mencionado que la tarjeta *Raspberry pi zero w* es la encargada de procesar la información obtenida por el robot y controlar sus acciones a fin de cumplir las tareas proporcionadas por el servidor. Para ello se requiere que el resto de componentes del robot interactúen a fin de transformar las ordenes en acciones.

Esta interacción de los componentes se realiza con conexiones alámbricas, a través de las cuales se transmiten diversas órdenes. Entre las principales tareas que desempeña el robot se encuentran las correspondientes al desplazamiento del mismo en el ambiente. Por ello, muchas de las conexiones presentes en el diseño del robot se orientan a la activación de los motores a partir de ordenes creadas por el controlador principal, es decir la tarjeta Raspberry.

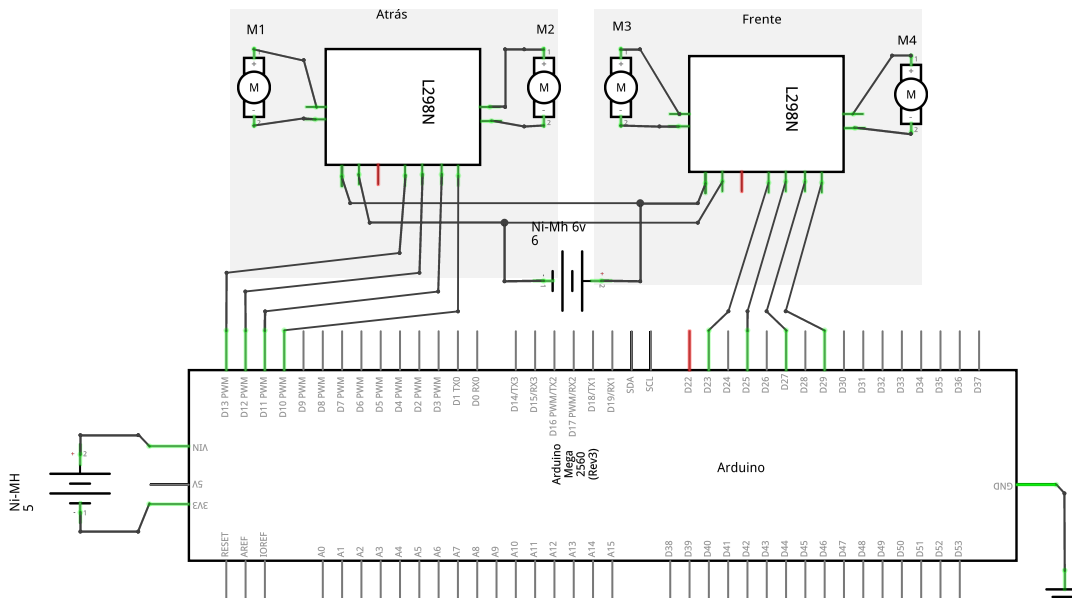


Figura 4.2: Conexión de los elementos del robot

El proceso de activación de los motores inicia cuando el controlador principal indica un cambio en los motores. Para ello, la Raspberry transmite a la tarjeta Arduino Mega 2560 una orden codificada en una cadena de texto que indica la configuración deseada para uno o más de los motores, la cual es enviada a través de una conexión serial USB, mediante el uso de los correspondientes puertos en ambas tarjetas. Cuando se determina que la orden fue transmitida por completo a la tarjeta Arduino, se establece en los puertos digitales de salida la configuración correspondiente a la orden recibida.

Los controladores L298N se conectan a la tarjeta Arduino por medio de cuatro salidas digitales cada uno. Es decir, se requiere de dos señales digitales para especificar el estado

(girar, girar en sentido contrario, detener) de cada motorreductor, por esta razón, al establecer una configuración en los pines de salida digitales de la tarjeta Arduino, los controladores obtienen esta configuración y activan el movimiento del motorreductor correspondiente.

El esquema de conexión utilizado para la tarjeta Arduino, los controladores L298N y los motorreductores se presenta en la figura 4.2, donde también se observa la conexión con dos de las tres fuentes de energía utilizadas, ya que tercera se encarga de alimentar a la tarjeta Raspberry.

## 4.2. Desplazamiento en el ambiente

En la sección anterior se describe el proceso a través del cual se activan las ordenes de movimiento del robot, este proceso forma parte del mecanismo de navegación en el ambiente, ya que otorga la capacidad de movimiento al robot. Sin embargo, el proceso a través del cual el sistema es consciente de la posición del robot y selecciona los movimientos necesarios para desplazarse de un área a otra, es denominado como mecanismo de navegación en el ambiente.

Como se menciona en la sección 3.5, en el capítulo anterior, para desplazar y ubicar al robot entre las diferentes áreas que conforman al entorno, el sistema de navegación utiliza como puntos de referencia un conjunto de marcadores visuales, los cuales forman un mapa a través del cual se posiciona y desplaza al robot. Para utilizar estos marcadores es necesario que el sistema SiGA capture y procese diferentes imágenes, por medio de las cuales obtiene la información necesaria para determinar la posición en la que se encuentra el robot, o incluso, calibrar los movimientos para desplazarlo a una zona específica.

Una parte importante en el proceso de navegación es la capacidad del robot para detectar los marcadores en las imágenes obtenidas del entorno y desplazarse hacia ellos, lo que permite al robot navegar en el ambiente, establecer caminos entre áreas o incluso reconocer lugares si llega a perderse.

Para que el robot inicie el desplazamiento hacia un marcador se requiere de una orden de inicio proporcionada por el servidor. Recibida la orden, el robot captura un imagen, determina la posición de los marcadores que se encuentran en ella y selecciona el marcador objetivo. Una vez elegido el marcador, el robot determina el tipo de movimiento que se debe realizar, es decir: girar a la derecha, girar hacia la izquierda o seguir recto. Este proceso se repite múltiples veces hasta que el robot considera que se encuentra a una distancia favorable para que el servidor pueda analizar el identificador presente en el marcador alcanzado.

**Algoritmo 1:** Dirigir el robot hacia un marcador

---

**Datos:** rango, lapsoT, areaMax, erroresMax

```

1  detener = false
2  errores = 0
3  giro = Centrado
4  mientras  $\neg$  detener hacer
5      detener el robot
6      Obtener imagen
7      Identificación de marcadores en imagen
8      si imagen contiene al menos un marcador entonces
9          marcador = Marcador con mayor área en imagen
                                     // Área es el número de pixeles
10         si marcador.area < areaMax entonces
11             si marcador está dentro del rango entonces
12                 giro = Centrado
13                 El robot avanza recto
14             en otro caso
15                 si marcador está por abajo del rango entonces giro = Izquierda
16                 si marcador está por arriba del rango entonces giro = Derecha
17                 El robot gira hacia la dirección giro por lapsoT milisegundos
18                 El robot avanza recto
19             fin
20         en otro caso
21             El robot gira hasta centrar el marcador en la imagen
22             El robot envía imágenes al servidor
23             detener = true
24         fin
25     en otro caso
26         si errores < erroresMax entonces
27             errores = errores + 1
28             El robot gira hacia la dirección opuesta de giro por lapsoT milisegundos
29             El robot avanza recto
30         en otro caso
31             mientras imagen no contenga marcadores hacer
32                 Obtener imagen
33                 Identificación de marcadores en imagen
34                 El robot gira en dirección opuesta de giro
35             fin
36             errores = 0
37         fin
38     fin
39 fin

```

---

El algoritmo 1 describe a mayor detalle, el proceso a partir del cual se desplaza el robot hacia el marcador. Para ello requiere de cuatro parámetros de configuración llamados *rango*, *lapsoT*, *areaMax* y *erroresMax*. Estos parámetros permiten ajustar el algoritmo a los movimientos del robot, ya que por el tipo robot utilizado no es posible establecer ordenes con movimientos exactos. *Rango* es un parámetro que establece la posición que puede tener un marcador en la imagen para considerarse centrado, y que el robot pueda desplazarse en línea recta hacia él. El *lapsoT* es un lapso de tiempo en milisegundos, y controla el tiempo que el robot gira dada una orden de giro, si el tiempo es muy alto los giros del robot serán más bruscos, por otra parte, si el tiempo es muy pequeño tardará más tiempo en centrar la imagen.

*AreaMax* representa el área que debe tener un marcador en la imagen para considerarlo cercano, por lo tanto es la variable que detiene el desplazamiento hacia el marcador. Si este parámetro es muy pequeño el robot se detendrá más lejos del marcador, lo que puede ocasionar problemas al analizar el identificador dentro de dicho marcador. Por el contrario, si *AreaMax* es muy grande, el robot no se detendrá a tiempo y perderá contacto con el marcador, lo que puede provocar una colisión entre el robot y el marcador.

Puede ocurrir que en uno de los giros realizado por el robot, el marcador se ubique fuera del campo de visión de la cámara, por ello es necesario un giro en la dirección contraria, cada que esto ocurre un contador de errores se incrementa, si el error se repite muchas veces de manera continua el robot entra en un estado donde se considera perdido, por ello requiere de buscar el marcador más cercano, este proceso de búsqueda puede desviarlo de la ruta original por lo que es necesario un parámetro que controle el número de errores permitidos antes de considerar al robot como perdido, este parámetro es *erroresMax*.

### 4.3. Identificación de marcadores

El procedimiento para desplazar al robot en el entorno hace evidente la importancia de una correcta identificación de marcadores dentro de las imágenes, ya que este mecanismo permite coordinar y dirigir los movimientos de desplazamiento del robot.

Esta identificación de marcadores es un proceso compuesto por dos etapas principales: el filtrado de la imagen y la clasificación de los objetos. El filtrado de la imagen es el proceso a partir del cual se seleccionan las áreas de la imagen donde se localizan posibles marcadores, esta selección se realiza mediante un análisis del color de los objetos dentro de la imagen, gracias a que el marcador cuenta con un color que resalta en el entorno.

La imagen 4.3 presenta el resultado obtenido por el proceso de filtrado de la imagen. En este resultado se aprecia la selección de marcadores, pero también se observan otros objetos que comparten el color de resalte con el marcador. Por esta razón, es necesaria la segunda etapa del proceso de identificación de marcadores, es decir, la clasificación de objetos.

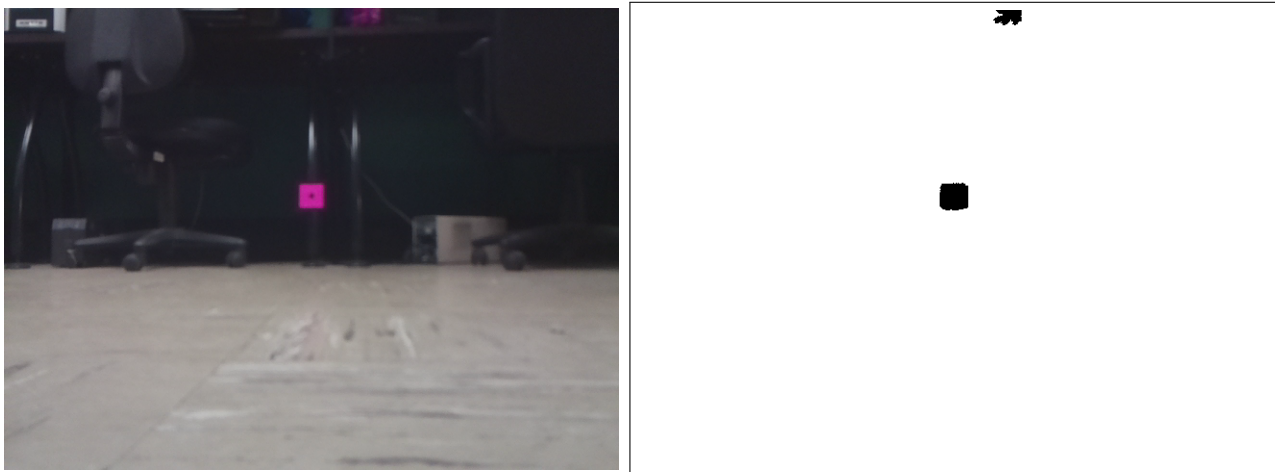


Figura 4.3: Filtrado de imagen por color

La clasificación de objetos permite extraer diferentes características a los objetos seleccionados por el filtro anterior, estas características propician la creación de clases, y por consiguiente, la selección de los marcadores agrupando los objetos presentes en la imagen por la forma que éstos poseen.

### 4.3.1. Filtrado de la imagen

Debido a la gran cantidad de información que posee una imagen, la tarea de extracción e identificación de objetos se complica si los objetos de interés no presentan características que les permitan resaltar en el entorno, lo que conlleva un mayor número de errores o un aumento en el tiempo de procesamiento. Por esta razón la elección del color de resalte para los marcadores resulta indispensable para un rápido filtrado y selección de los mismos. El diseño del marcador contempla el uso del color magenta como la elección para el color de resalte, debido a su poco uso en los entornos, por consiguiente, para realizar la extracción de marcadores en una imagen es necesario seleccionar aquellos objetos cuyo color sea magenta.

Para analizar los diferentes colores dentro de la imagen, es necesario utilizar un modelo de representación de color que permita describir colores a través de valores numéricos. El modelo más extendido es el modelo RGB, el cual permite la descripción de un color a través de la intensidad de tres colores primarios (canales): rojo, verde y azul [20].

Gracias a que las imágenes obtenidas por el robot se encuentran en el esquema de color RGB, no es necesario un proceso de adaptación del modelo de color. Por esta razón el proceso de selección de objetos se basa en el análisis de los canales que describen a cada píxel, lo que conlleva una selección de grupos de píxeles que cumplen con las características que definen al color magenta.

En el modelo RGB, el color magenta se representa como  $(255, 0, 255)$ , con un rango que

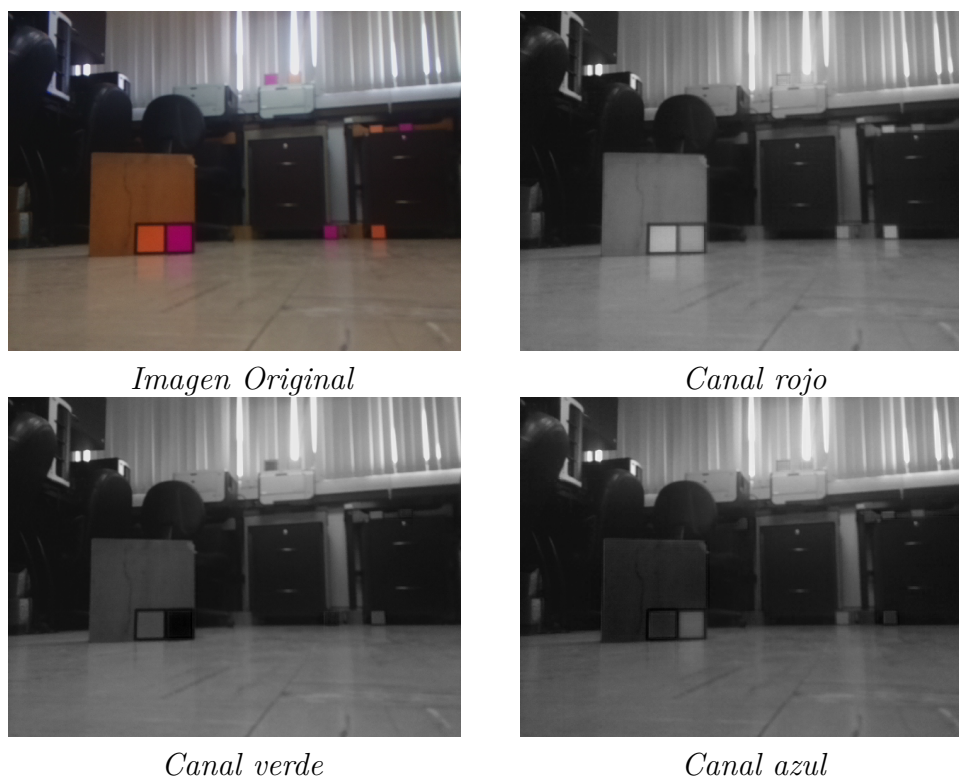


Figura 4.4: Descomposición de imagen RGB

va desde 0 a 255 para representar los niveles de saturación de cada canal. Esto indica que el color magenta presenta una saturación total en los canales correspondientes al color rojo y azul, y una nula saturación en el canal verde.

Con esta información es posible descomponer una imagen a color en tres imágenes en escala de grises, donde cada imagen resultante de esta descomposición representa a uno de los tres canales que componen al modelo RGB, tal y como se aprecia en la figura 4.4.

Las imágenes resultantes del proceso anterior dieron pie a la creación de filtros que seleccionan niveles altos y bajos de saturación en imágenes en escala de grises. Esto permite filtrar los píxeles con alta saturación para las imágenes que corresponden al canal rojo y azul y filtrar los píxeles donde la saturación es cercana a cero en el canal verde. Después se realiza un proceso de intersección entre los píxeles seleccionados de las tres imágenes y se obtienen aquellos que representan al color magenta.

La aplicación de estos filtros permite la selección de marcadores cuando la imagen obtenida resulta con una nitidez buena, y el color del marcador se logra apreciar claramente, sin embargo, en imágenes con poca o mucha iluminación el marcador no se selecciona por completo, o en el peor de los casos, no se encuentra. Por esta razón, se buscó estandarizar la iluminación para todas las imágenes por medio de un proceso de ecualización de la imagen.

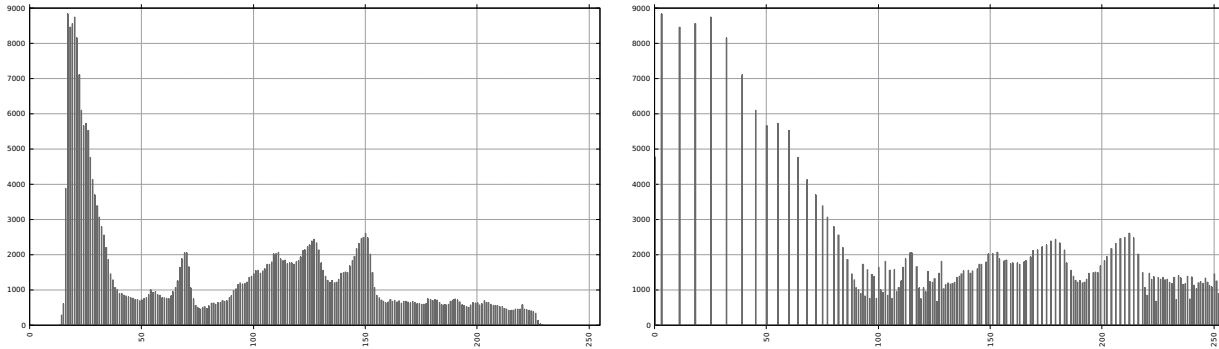


Figura 4.5: Ecuación de la imagen

El histograma de una imagen consiste en contar cuántos píxeles se encuentran en cada nivel de saturación posible, en nuestro caso 256 valores. Mediante este histograma se puede observar si una imagen contiene una composición muy clara u oscura, ya que la mayor cantidad de píxeles se encontrará en alguno de los extremos del histograma. La ecualización de la imagen busca obtener una distribución lo más uniforme posible, al distribuir la concentración de píxeles en los diferentes niveles de saturación. El resultado de este proceso se puede observar en la figura 4.5.

Para ecualizar una imagen a través de su histograma se utiliza una función de transformación del tipo  $y = f(x)$ , donde  $x$  representa el valor actual de un píxel y  $y$  representa el nuevo valor de intensidad [33], función descrita en la ecuación 4.1.

$$T(k) = (L - 1) \sum_{i=0}^k p(i), \forall k = 0, 1, \dots, L - 1 \quad (4.1)$$

Donde  $p(i)$  es la probabilidad de ocurrencia del  $i$ -ésimo nivel de intensidad (dada por el cálculo previo del histograma), y donde el rango de valores válidos para las intensidades se encuentra entre 0 y  $L - 1$ .

Como ya se mencionó, el proceso de ecualización fue realizado en las imágenes obtenidas de la descomposición de la imagen en color. Esto redujo la afectación de la luminosidad en las imágenes, lo que a su vez conlleva a una mejor selección del color magenta, sin embargo, al realizar las pruebas con múltiples imágenes obtenidas por el robot, los marcadores seleccionados por el filtro no tenían una forma bien definida, por ello se optó por cambiar a un modelo de color en el cual la iluminación fuera independiente del color seleccionado.

### Modelo de color HSV

El modelo HSV, al igual que el modelo RGB, se forma de tres valores: matiz, saturación y valor, sin embargo, este modelo no se basa en intensidades de colores primarios, sino en la



definición de una tonalidad del color, el nivel de saturación de dicho color y la cantidad de brillo presente.

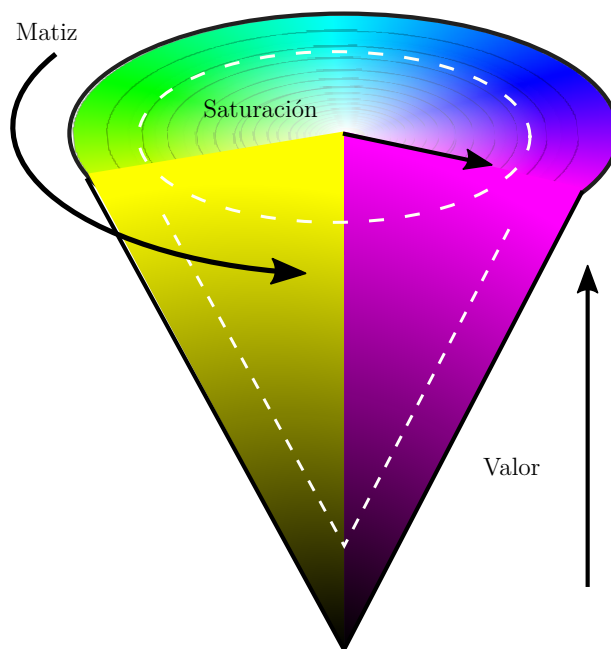


Figura 4.6: Diseño propuesto para el marcador.

Para ejemplificar el uso del modelo HSV, se hace referencia al *cono del modelo HSV*, presente en la figura 4.6. En esta representación, el campo correspondiente al *matiz* se presenta en un rango de, entre  $0^\circ$  y  $360^\circ$  y forma la base del cono, este campo permite la selección de un color determinado, por ejemplo, con  $300^\circ$  el color seleccionado es el magenta.

Seleccionado el matiz, se indica el nivel de *saturación* del color, es decir, qué tanto brillo o tendencia al blanco tiene este color, este campo se presenta en porcentajes y, mientras menor sea dicho porcentaje, más blanco será el color. Por último se selecciona el *valor*, el cual, al igual que la saturación, se presenta en porcentajes, sin embargo éste indica qué tan oscurecido se encuentra el color, es decir, mientras menor sea el porcentaje correspondiente al valor, el color se encuentra más negro.

Para hacer uso del modelo de color HSV es necesario un proceso de conversión de modelos de color, pues como se menciona anteriormente, las imágenes que son obtenidas del robot se encuentran en un formato RGB. Para realizar esta conversión de modelos se hace uso de las ecuaciones 4.2, 4.3 y 4.4.

$$\begin{aligned} \max &= \max(R, G, B) \\ \min &= \min(R, G, B) \\ \Delta &= \max - \min \end{aligned}$$

**Matiz.**

$$H = \begin{cases} 0^\circ, & \Delta = 0 \\ 60^\circ \left( \frac{G-B}{\Delta} \right), & \max = R \\ 60^\circ \left( \frac{B-R}{\Delta} + 2 \right), & \max = G \\ 60^\circ \left( \frac{R-G}{\Delta} + 4 \right), & \max = B \end{cases} \quad (4.2)$$

**Saturación**

$$S = \begin{cases} 0, & \max = 0 \\ \frac{\Delta}{\max}, & \max \neq 0 \end{cases} \quad (4.3)$$

**Valor**

$$V = \max \quad (4.4)$$

Donde  $R, G$  y  $B$  son valores entre 0 y 1.

Gracias a la forma en la que son expresados los colores en el modelo HSV, el manejo de la luminosidad y la gama de colores se realiza de forma independiente, lo que permite una selección del color sin interferencia de los niveles de luz en la imagen.

Por lo anterior, el proceso de filtrado de marcadores consiste en convertir cada píxel presente en la imagen a su representación en el modelo HSV. Hecho esto, se seleccionan los píxeles en los cuales el matiz se encuentra dentro de un rango cercano a los 300 grados, ya que es el ángulo en el que se encuentra el color magenta. Es posible que el píxel posea un matiz dentro del rango, pero con un nivel de saturación o brillo muy bajo, lo que significa que el color se encuentra muy cercano al negro. Por esta razón es necesario establecer un rango mínimo para los niveles de saturación y valor, con lo que se eliminan píxeles negros con tendencia a un magenta oscuro.

Este proceso de filtrado de píxeles de color magenta puede ser visto como una binarización de la imagen, ya que un píxel puede tener dos valores, negro o blanco, esto permite la identificación de objetos en la imagen, sin embargo, como ya se mencionó, existen objetos en el entorno que comparten color con los marcadores pero no forma, por esta razón, es necesaria una clasificación de los objetos por forma.

### 4.3.2. Clasificación del marcador

A pesar de un filtrado de color eficiente, esto no garantiza que el ambiente se encuentre libre de objetos con colores similares al utilizado en el marcador, por lo cual es necesaria la extracción de características que tomen en cuenta la forma de los objetos, con el objetivo de analizar si el objeto encontrado tiene la forma de un marcador.

Para establecer un proceso de clasificación de objetos es necesario establecer un conjunto de características que permitan la creación de clases, estas características pueden ir desde una medida como el perímetro o área del objeto, hasta características más complejas.

Para realizar la clasificación de los marcadores se debe tomar en cuenta que el robot puede realizar la captura de la imagen desde diferentes distancias del marcador, o incluso desde varios puntos de vista, por lo cual las características seleccionadas para realizar la clasificación no deben ser afectadas por las transformaciones que pueda sufrir la imagen. Una técnica utilizada para la extracción de características es el cálculo de los denominados *momentos de Hu*, los cuales son un grupo de 7 valores que permanecen invariantes a transformaciones de traslación, rotación y cambios de escala en objetos de dos dimensiones, por esta razón, los momentos de Hu son las características sobre las cuales el clasificador fue construido.

#### Algoritmo de clasificación

Un algoritmo de clasificación utilizado cuando se desea clasificar objetos es el algoritmo de los *k vecinos más próximos*, el cual clasifica un nuevo objeto dentro de una clase al evaluar la distancia que existe entre las características de ese nuevo objeto y los vectores más representativos de las clases posibles.

Por consiguiente, la etapa de entrenamiento de dicho algoritmo consiste en determinar un conjunto de vectores característicos de cada clase, después, en la etapa de clasificación se calcula la distancia euclidiana de el nuevo vector hacia cada uno de los vectores característicos de las clases, clasificando al nuevo objeto dentro de la clase a la que pertenecen los vectores que tengan la menor distancia. Este clasificador depende de el número de vectores característicos de cada clase por lo que, mientras mas vectores sean utilizados de cada clase, mejor será el clasificador, sin embargo, esto ocasiona que la etapa de clasificación sea más lenta.

Una solución propuesta es realizar el cálculo de las medias estadísticas de los vectores que conforman cada clase, lo que permite tomar las medias de las clases como vectores característicos. Así, en la etapa de clasificación, la distancia solo es calculada de cada nuevo vector a los vectores de las medias de cada clase, lo que permite una clasificación mas rápida.

Para implementar este algoritmo de clasificación es necesario un conjunto de objetos extraídos de imágenes capturadas por el robot, en las cuales, se seleccionaron previamente los objetos correspondientes a marcadores, así como los objetos que no entran dentro de esta clase. De este conjunto de objetos ya clasificados, el 60% conforma el conjunto de entrenamiento, es decir, los objetos que permitieron crear los vectores característicos utilizados por el algoritmo. El 40% restante se utilizó para realizar una evaluación del clasificador.

En esta evaluación se observó que el uso de las medias como valores característicos no lograban una correcta clasificación de todos los marcadores, debido a la dispersión que presentaban los objetos que no pertenecen a la clase marcadores. Por esta razón aplicar el algoritmo de *k vecinos más próximos* con medias como vectores característicos no es factible. Sin embargo, al realizar el cálculo estadístico de las medias para los momentos de Hu, es posible apreciar que los valores que corresponden a la clase marcadores se encuentran por debajo de los valores obtenidos por otro tipo de objetos, tal y como se observa en la figura 4.7.

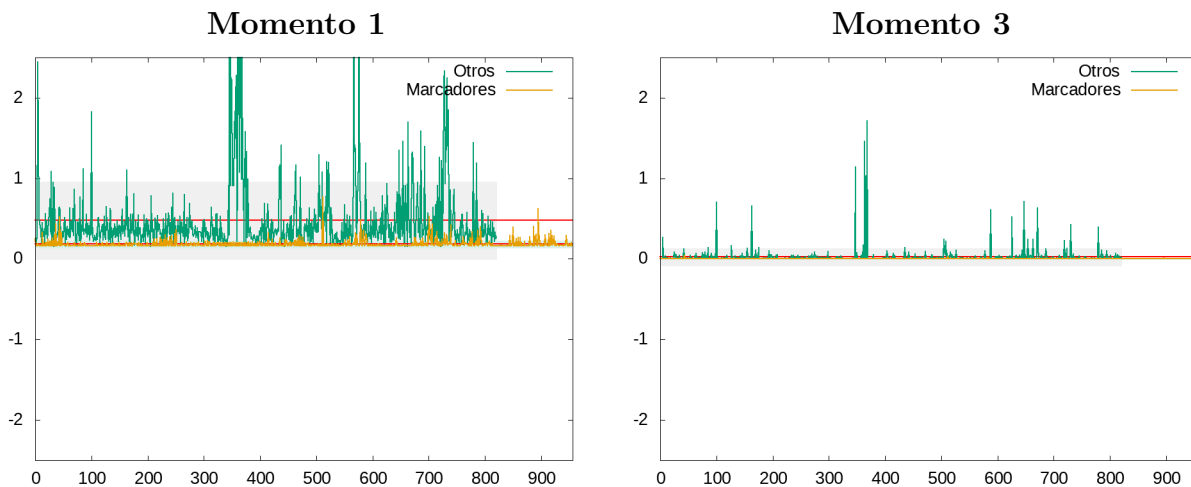


Figura 4.7: Momentos 1 y 3 de Hu de los objetos filtrados

Por lo anterior es posible seleccionar un límite a partir del cual es posible realizar el filtrado de objetos, es decir, los objetos que superan el rango establecido en ciertos momentos de Hu, serán considerados como objetos que no pertenecen a la clase marcadores. Para establecer este límite de filtrado se requiere encontrar el nivel óptimo donde el número de errores cometidos por el clasificador se reduce al mínimo. Se considera como error de clasificación cuando un marcador es descartado o cuando otro objeto es seleccionado como marcador.

Para encontrar los niveles óptimos de filtrado para cada momento de Hu se utiliza un algoritmo genético de representación real, en el cual, cada individuo representa un conjunto de siete niveles de filtrado (uno para cada momento de Hu), a través de vectores con una

representación real. Establecidos los individuos del algoritmo genético como niveles de filtrado, la función de evaluación del algoritmo genético consiste en analizar los niveles de cada individuo y contar el número de errores cometidos al realizar la clasificación de los objetos contenidos en el banco de pruebas. Con base en esto, el algoritmo genético busca encontrar el conjunto de niveles en el que se cometan menos errores.

Establecido el nivel óptimo de filtrado, basta con seleccionar los momentos de  $H_u$  mas significativos, pues el uso de todos los momentos ocasiona que se incremente el número de errores cometidos en la clasificación. Todo lo anterior dio como resultado el uso del primer y tercer momento de  $H_u$ , con niveles de clasificación en 0.207342207431793 y 0.000300077896100 respectivamente, para realizar la clasificación marcadores.

Con la clasificación de marcadores implementada en el robot, el mecanismo de navegación descrito anteriormente permite al servidor conocer la ubicación del robot, así como desplazarlo a las áreas en las cuales se requiera entregar un mensaje a un usuario.



## Implementación del servidor

El robot es el elemento central del sistema SiGA a través del cual se realiza la interacción con los usuarios y el entorno. Sin embargo, a pesar de ser el principal método de interacción, los componentes que realizan la gestión de toda la información se encuentran en el servidor del sistema. Esta gestión abarca desde las interfaces de captura de información, hasta los controles de las actividades del robot, por lo que para facilitar la administración de todas las funcionalidades proporcionadas, el servidor se compone de tres elementos principales: el gestor de posición, el gestor de interfaces y el gestor de actividades.

### 5.1. Gestor de interfaces

Para que el sistema SiGA pueda realizar la entrega de mensajes a los usuarios, requiere de un conjunto de datos clasificados en tres categorías principales: los parámetros de configuración, la información correspondiente a las actividades del usuario y la información relacionada a la medicación del mismo. Para ingresar esta información al sistema SiGA, el servidor proporciona las interfaces de captura necesarias, así como los métodos de validación y almacenamiento correspondientes.

#### 5.1.1. Parámetros de configuración

Los parámetros de configuración permiten especificarle al sistema la información relativa al entorno en el que dicho sistema trabaja. Por esta razón, el servidor incluye interfaces que permiten la descripción del entorno administrado por el sistema, como es el mapa de navegación del robot o los usuarios que participan en el entorno inteligente.

## Interfaz de gestión de mapas

La interfaz de configuración de mapas tiene como objetivo que el usuario indique al sistema las áreas pertenecientes al entorno, y las rutas que conectan a dichas áreas; a fin de proporcionar la información suficiente para que el robot pueda desplazarse por el entorno al igual que establecer puntos de referencia para la ubicación de los usuarios.

### Mapas Captura de mapas en el sistema

The interface is divided into three main sections:

- Gestión de mapa:** A map visualization showing four nodes: 'sala', 'cocina', 'p1', and 'p2'. Arrows indicate a clockwise path: sala → cocina → p1 → p2 → sala.
- Agregar área:** A form with an input field for 'Area' (placeholder: 'Ingrese el nombre del área') and four marker options: a circle, a square, a star, and a triangle. A green 'Agregar área' button is below the options.
- Caminos:** A table listing paths between nodes.

Origen	Destino	Izq	Der
sala	cocina	<input type="radio"/>	<input checked="" type="radio"/>
cocina	p1	<input type="radio"/>	<input checked="" type="radio"/>
p1	p2	<input type="radio"/>	<input checked="" type="radio"/>
p2	sala	<input type="radio"/>	<input checked="" type="radio"/>

Below the map is a blue button labeled 'Guardar Mapa'.

Figura 5.1: Interfaz para gestión de mapas

La interfaz se compone por tres áreas principales: el formulario para alta de áreas, el gestor del mapa, y la selección de rutas, presentes en la figura 5.1, las cuales son descritas a continuación.

**Alta de áreas.** Permite establecer el enlace entre el marcador físico utilizado para la identificación de áreas y la información contenida por el sistema, ya que es en este formulario permite ingresar el identificador presente en el marcador a través de una selección de las figuras.

Al presionar el botón Agregar área, se verifica que, tanto el nombre del área como el identificador hayan sido capturados. Realizada esta verificación, se revisa que el



identificador proporcionado no se utilizó en otra área, pues crearía problemas en el mecanismo de navegación.

**Gestor del mapa.** El gestor del mapa es el área donde se definen las rutas a través de las cuales el robot se desplaza. Para crear una ruta, el usuario debe dar clic derecho en el área de origen y clic izquierdo en el área destino. Creada la ruta, el sistema valida que esta ruta no haya sido creada anteriormente. Al indicar el origen la ruta, el sistema valida que dicha área no sirva de origen a más de dos rutas, (uno hacia la izquierda y otro a la derecha), esto para funcionamiento del mecanismo de navegación del robot. Otra funcionalidad de esta área es eliminar las rutas y áreas del mapa. Para eliminar un camino basta con dar clic derecho sobre él. Para eliminar un área del mapa es necesario dar doble clic sobre ella y confirmar la eliminación. Realizada la confirmación el sistema elimina, como primer paso, las rutas relacionados a dicha área, después, realiza la eliminación del área.

**Selección de rutas.** Como se ha mencionado, una vez que el robot alcanza un punto de control, el servidor le indica hacia qué dirección debe girar (izquierda / derecha) para alcanzar el siguiente marcador. Por esta razón el área de selección de rutas permite seleccionar la posición del marcador destino con respecto al origen, en otras palabras, al estar el robot en el marcador “origen”, esta interfaz permite seleccionar hacia qué dirección debe girar para alcanzar el marcador “destino”.

### Interfaz para gestión de adultos mayores

La interfaz para la gestión de adultos mayores permite ingresar al sistema SiGA el nombre de los usuarios que participan en el entorno gestionado, y a los cuales serán entregados los mensajes generados por el servidor. Para realizar el alta de un usuario basta con ingresar el nombre en el campo correspondiente y presionar guardar, al hacer esto el servidor valida el nombre y lo guarda en la base de datos.

Conforme los usuarios se almacenan en el sistema, la interfaz presenta una lista de los usuarios dados de alta. En dicha lista, para cada usuario se incluyen dos opciones: editar y eliminar (5.2). Al presionar eliminar, toda la información correspondiente al usuario es borrada de la base de datos, lo que incluye, información relacionada a citas y medicación. Al presionar editar, se carga el nombre del usuario en el campo de edición, lo que permite la actualización del dato.

#### 5.1.2. Gestión de medicamentos

La interfaz para gestión de medicamentos permite al médico del adulto mayor especificar la medicación de los usuarios del sistema. Para agregar un medicamento, el médico tiene que buscar al usuario en la interfaz de búsqueda de usuarios, la cual se muestra en la figura 5.3. La interfaz de búsqueda permite filtrar los usuarios cuyo nombre coincida con el texto

**Usuarios** Gestión de adultos mayores

Nombre del usuario [Editar]

Usuario 1

✕ Cancelar Guardar

Usuario 1	Editar Eliminar
Usuario 2	Editar Eliminar
Usuario 3	Editar Eliminar

Figura 5.2: Interfaz para gestión de adultos mayores

introducido en la caja de búsqueda.

**Medicamentos** Gestión de medicamentos

Selecciona el Usuario

Teclee nombre del usuario Buscar

Usuario 1
Usuario 2
Usuario 3

Figura 5.3: Búsqueda de usuarios

### Carga de medicamentos

Una vez el médico realizó la búsqueda del usuario, la interfaz presenta el formulario de alta de medicamentos, mostrado en la figura 5.4 (izquierda). Este formulario permite el registro del medicamento a través de la captura de cinco campos: nombre, indicaciones, duración, tomar cada, y fecha inicio. *Nombre* corresponde al nombre con el que se identifica el medicamento, *indicaciones* representa las instrucciones al momento de realizar la ingesta

del medicamento (en caso de requerirse), *duración* determina los días que el usuario debe tomar el medicamento. *Tomar cada* representa las horas que deben pasar entre cada ingesta. Por último, *fecha inicio* permite que el médico indique la fecha en la que se debe iniciar el tratamiento, lo cual ofrece la posibilidad de programar la medicación del usuario.

## Medicamentos Gestión de medicamentos

### Agregar medicamento

Nombre del medicamento

Indicaciones

Duración (días) Tomar cada (horas)

Fecha inicio

octubre 2018						
lu	ma	mi	ju	vi	sá	do
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

### Medicamentos activos y próximos

- 2018-10-12 **Medicamento 1** tomar durante 6 días cada 3 horas
- 2018-10-13 **Medicamento 2** tomar durante 8 días cada 3 horas

Figura 5.4: Interfaz para gestión de medicamentos

Cuando el médico presiona el botón guardar, además de almacenar los campos ingresados en la base de datos, el servidor realiza la programación de los horarios en los cuales se deben entregar los recordatorios de toma de medicamento. Para ello, en el servidor fueron configurados los tiempos de dosificación estandarizados propuestos por la Universidad de Florida [18] presentes en la tabla 5.1.

Tabla 5.1: Tiempos de dosificación estandarizados

Intervalo	Tiempos estandarizados
Cada 3 horas	00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, 21:00
Cada 4 horas	01:00, 05:00, 09:00, 13:00, 17:00, 21:00
Cada 6 horas	06:00, 12:00, 18:00, 24:00
Cada 8 horas	08:00, 16:00, 24:00
Cada 12 horas	09:00, 21:00
Cada 24 horas	09:00

Por lo anterior, cuando el médico asigna una fecha de inicio, si ésta corresponde al día en el que se está haciendo el registro del medicamento, la programación de actividades iniciará con la hora más próxima a la indicada en la tabla 5.1, si es un día diferente, la programación inicia con la primera hora correspondiente al intervalo.

De requerirse una configuración distinta para los tiempos de dosificación, el servidor almacena la hora inicial de cada periodo en el archivo `libphp/horarios.php`, tal y como se aprecia en el código 5.1.

```

1      <?php
2          //Frecuencia de ingesta, hora de primera toma
3          $horarios = [
4              "3"  => "0",
5              "4"  => "1",
6              "6"  => "6",
7              "8"  => "8",
8              "12" => "9",
9              "24" => "9"
10         ];
11     ?>

```

Código 5.1: Configuración de los tiempos de dosificación

El primer campo representa la frecuencia de ingesta y el segundo campo la hora en la cual se debe realizar la primera toma. Si son modificados estos datos es posible crear una nueva configuración para el horario de entrega de recordatorios.

## Listado y eliminación

En la figura 5.4 (derecha) se observa una lista de los medicamentos programados para el usuario. Esta lista se actualiza al momento de ingresar a la interfaz para la captura de medicamentos, y en ella se muestran los medicamentos que se encuentran activos para un usuario, es decir, aquellos que se encuentra consumiendo o cuyo tratamiento está próximo a empezar. Esta lista se actualiza cada que un nuevo medicamento es agregado al usuario. Entre la información que presenta se encuentra la fecha de inicio del tratamiento, el

nombre del medicamento así como la periodicidad de las tomas y la duración del tratamiento.

En esta lista también se encuentra un botón de eliminación en cada uno de los medicamentos. Como su nombre lo indica, el botón permite eliminar el medicamento de la programación del usuario, al igual que elimina las alertas programadas para este tratamiento. Una vez eliminados estos datos, la interfaz presenta el mensaje de eliminación exitosa, o en su defecto, muestra la alerta de un error ocurrido.

### 5.1.3. Interfaz de gestión de citas

La interfaz para gestión de citas permite, tanto a los médicos como a los familiares del adulto mayor, programar reuniones o mensajes que deben ser entregados a una hora determinada. Para programar un nuevo evento es necesario, como en el caso de la interfaz para la gestión de medicamentos, seleccionar el usuario al cual debe ser entregado el recordatorio.

Una vez que se selecciona el usuario, la interfaz muestra el formulario para la creación de citas, para ello solo se requiere ingresar el mensaje que debe ser entregado al usuario, la fecha y la hora. Este formulario puede observarse en la imagen 5.5.

Figura 5.5: Interfaz para gestión de citas

Cuando el usuario de la interfaz presiona el botón guardar, los datos se almacenan en la base de datos. Al mismo tiempo, se crea un conjunto de copias del mensaje, las cuales serán entregadas a diferentes horas antes de que ocurra el evento programado. Para configurar el número de copias del mensaje y las horas a las cuales deben ser entregadas es necesario modificar en el servidor el archivo `libphp/horarios.php`.

```

1      <?php
2          $horaIniEntrega = "6:00:00";
3          $numeroMensajes = 5;
4          $tiemposEntrega = [5, 3, 2, 1, 0];
5      ?>

```

Código 5.2: Configuración de las horas de entrega

En el código 5.2, el campo `$horaIniEntrega` permite configurar la primera hora a la que el sistema puede iniciar con la entrega de recordatorios de citas. Esta hora previene que los usuarios sean molestados durante las horas de descanso, pues, si algunas de las copias ocurre antes de la hora de inicio especificada, estas copias son eliminadas.

El siguiente campo de configuración es `$numeroMensajes`, el cual define el número de copias que se crean para cada evento. Para configurar las horas de entrega de estas copias el vector `$tiemposEntrega` especifica cuantas horas antes del evento se debe programar cada uno de los mensajes.

En la figura 5.5 (derecha) también se aprecia la lista donde se muestran los medicamentos que el usuario está consumiendo. Esta lista también es el método para retirar medicación al usuario, pues en cada medicamento se encuentra un botón que permite eliminarlo.

## 5.2. Gestor de actividades

El gestor de actividades es el componente de software encargado de generar los eventos dentro del servidor correspondientes a la entrega de mensajes. Estos eventos son producidos por la información ingresada a través de las interfaces gestión de medicamentos y gestión de citas.

Para determinar el momento adecuado en el que debe ser entregado un mensaje es necesario un proceso que verifique en lapsos de tiempo la existencia de mensajes dentro de dicho lapso. De existir mensajes, se debe determinar el estado del robot y verificar si es el momento adecuado para realizar la entrega de mensajes. Si el robot no se encuentra en un estado disponible para la entrega de mensajes, se deben generar las acciones correspondientes para colocar al robot en un estado adecuado para realizar la entrega. Por esta razón, las tareas de consulta de actividades y generador de mensajes describen el proceso a partir del cual, el servidor configura las actividades del robot con el objetivo de entregar los mensajes a los usuarios.

### 5.2.1. Consulta de actividades

Las interfaces de gestión de medicamentos y citas almacenan en la base de datos un conjunto de mensajes que deben ser entregados por el robot en diferentes fechas y horas. Debido a que, tanto la llegada de esta información, así como los momentos de entrega de la misma son eventos completamente aleatorios, es necesario que un proceso que se encuentre de forma continua revisando la base de datos a fin de determinar si es el momento de entregar un mensaje.

La “*consulta de actividades*” es un proceso del servidor que se ejecuta en lapsos de tiempo definido, en estas ejecuciones se realizan diferentes consultas a la base de datos, con el objetivo de localizar los mensajes que deben ser entregados dentro del lapso de tiempo

definido por el momento en que se realiza la consulta, y hasta el momento en el que se realiza la siguiente ejecución del proceso.

```
1      SELECT * FROM vw_mensajeCita
2      WHERE horaMsj >= "%1" AND horaMsj < "%2"
3      ORDER BY horaMsj
```

Código 5.3: Consulta para la obtención de mensajes

En el fragmento de código 5.3 se ejemplifica una de las consultas que realiza el proceso *consulta de actividades*, la cual recibe como parámetro el lapso de tiempo en el que debe realizarse la búsqueda de mensajes.

Esta consulta regresa los datos obtenidos proporcionados por la vista `vw_mensajeCita`, en la cual se muestra la hora y día en la que debe ser entregado el mensaje, el usuario al que va dirigido, en mensaje correspondiente, y en el caso de las citas, el día y la hora en la que la cita tendrá efecto.

Una vez que son obtenidos los mensajes de la base de datos, éstos son enviados al proceso encargado del envío de mensajes al robot, sin embargo, para realizar el envío, es necesaria la sincronización de actividades con el robot.

### 5.2.2. Generador de mensajes

El proceso *generador de mensajes* es el encargado de sincronizar las actividades realizadas por el robot a fin de entregar los mensajes obtenidos a partir de las interfaces para la gestión de medicamentos y citas. Este proceso puede ser visto como el módulo de control principal del servidor, ya que es el encargado de coordinar todos los eventos producidos por el robot, el proceso de consulta de actividades e inclusive, el gestor de posición de los usuarios. Para que se logre la sincronía con el robot y con otros procesos dentro del servidor, se hace uso de una comunicación a través de mensajes, los cuales son enviados y recibidos mediante *WebSockets*.

El generador de mensajes cuenta con un mecanismo de recepción de mensajes bidireccionales, permaneciendo siempre activo para la recepción de estos mensajes ya que es un proceso asíncrono. Los mensajes que recibe este proceso pueden ser de dos tipos: de texto y binarios. Los mensajes de texto permiten la coordinación con los demás componentes del sistema, pues a través de ellos se envían cadenas en las cuales se encuentran contenidas órdenes o estados de los componentes. Por otra parte, los mensajes binarios permiten la recepción de imágenes, lo que a su vez permite determinar al robot la ubicación del robot, mecanismo que será descrito en la sección 5.3.2



## Tipos de mensajes de texto

Como se mencionó, el proceso *generador de mensajes* se comunica con el resto de componentes del servidor y con el robot a través de mensajes de texto. Estos mensajes cambian dependiendo del componente al que son enviados o desde el componente que son recibidos, por lo que a continuación se describen los tipos de mensajes recibidos o producidos por el generador de mensajes.

### ■ *Mensajes recibidos.*

*Registro de componentes.* La recepción de un mensaje del tipo registro de componentes, es utilizada para informar al generador de mensajes que se ha establecido la conexión con algún componente del sistema, ya sea el robot, el proceso de consulta de actividades, o el gestor de posición del usuario. Este mensaje se recibe solo una vez durante la ejecución del servidor y permite almacenar el mecanismo de comunicación con dichos componentes.

*Recepción del actividad.* Este tipo de mensaje es el que produce el proceso de consulta de actividades. En otras palabras, es el mensaje que indica que es el momento en el que el robot debe entregar un recordatorio. Los datos contenidos en este mensaje son: el usuario, la hora del evento y el mensaje que debe ser entregado.

*Actualización de posición.* Un mensaje de este tipo informa que el usuario ha cambiado de localización dentro del entorno, ya sea que se localiza en un área nueva o que ha salido o entrado del entorno. Este evento permite que el generador de mensajes conozca la posición del usuario, y en caso de ser necesario, se cambie la ubicación del robot o se entregue algún mensaje.

*Llegada del robot.* Mientras el robot se encuentra en camino hacia un marcador, el servidor no puede indicarle nuevas instrucciones puesto que la tarea principal desempeñada por el robot es la navegación en el entorno. Una vez el robot alcanza una nueva zona, envía las correspondientes imágenes que permitirán identificar su nueva localización. Una vez que el robot terminó de enviar todas las imágenes, envía un mensaje de *llegada del robot*. Este mensaje de llegada informa al servidor que puede iniciar con la identificación de la posición, al igual que indicar que el robot se encuentra disponible para el envío de nuevas ordenes.

### ■ *Mensajes enviados.*

*Aviso de registro.* Este mensaje se envía cuando un mensaje de registro de componentes es recibido, y funciona como acuse de recibo para que el componente sepa que el registro se realizó de manera exitosa.

*Mensajes de dirección.* Este tipo de mensajes son enviados al robot para indicarle la dirección en la que debe buscar un nuevo marcador. Al ser enviado un mensaje

de este tipo, el servidor no interactúa con el robot hasta recibir un nuevo mensaje que informe que el robot ha alcanzado el objetivo.

*Mensajes de comunicación.* Si tanto el robot como el usuario se encuentran en la misma zona, y existen recordatorios para el usuario se produce un mensaje de comunicación. Este tipo de mensaje es enviado al robot para indicarle el mensaje que debe reproducir en dicha zona.

*Mensaje de terminación de recordatorios.* Cada que se produce un mensaje de comunicación, el robot tarda un determinado tiempo en reproducirlo, por esta razón, si se requiere reproducir más de un recordatorio, el robot encola el resto de recordatorios recibidos, hasta que termina de reproducirlos. El mensaje de terminación de recordatorios indica al robot que no existen más mensajes por entregar, por lo que debe esperar la llegada de este mensaje para iniciar con los mensajes de dirección en caso de haber recibido alguno.

## Eventos

Los eventos que ocurren en el generador de mensajes son iniciados por la llegada de algún mensaje, por ejemplo el cambio de posición de un usuario o la necesidad de entregar un recordatorio. Debido a que los mensajes pueden llegar en cualquier momento, es necesario que las acciones del robot sean sincronizadas a fin de evitar la pérdida de información y errores de entrega de mensajes.

La figura 5.6 ejemplifica forma en la que los mensajes crean la interacción entre los elementos del sistema. En esta figura se observan dos de los principales procesos de interacción de los elementos. El primero de ellos es el registro de los componentes, el cual se realiza una vez que establecen la conexión con el generador de mensajes. El segundo proceso de interacción ejemplifica la llegada del robot a un punto de control.

Este segundo proceso inicia con el envío de imágenes por parte del robot, seguido del mensaje que indica que todas las imágenes fueron enviadas. Cuando este mensaje es recibido por el *generador de mensajes* comienza a realizar el análisis de las imágenes a fin de determinar la posición del robot. Mientras esto ocurre, el proceso encargado de la localización de los usuarios puede informar el cambio de posición de alguno de ellos, por lo que, el generador de mensajes actualiza la nueva posición del usuario.

Por otra parte, el proceso encargado de la consulta de actividades en la base de datos envía los recordatorios que necesita entregar el robot. Esto puede ocurrir en cualquier momento, por lo cual, el generador de mensajes debe almacenarlos hasta que el robot se encuentre disponible para la recepción de ordenes.

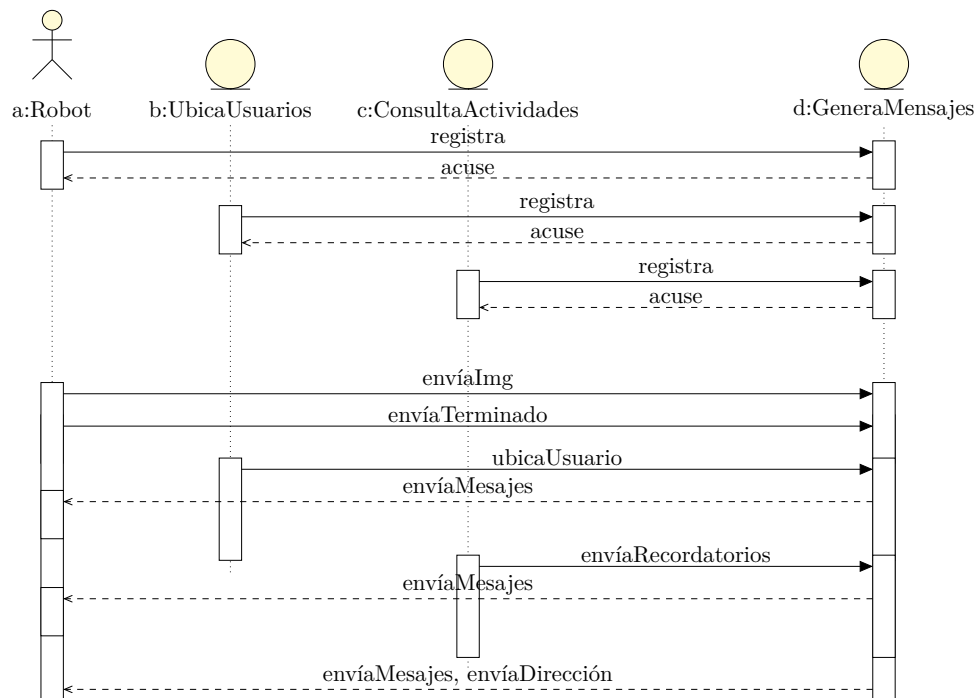


Figura 5.6: Diagrama de secuencia para los eventos del generador de mensajes

Una vez que el generador de mensajes determinó la posición del robot a través de las imágenes recibidas, verifica si existen recordatorios almacenados que deban ser entregados, si esto ocurre, busca entre los recordatorios recibidos si alguno debe ser entregado en la posición actual del robot, lo que crea un conjunto de mensajes de reproducción enviados al robot. Terminado este proceso, el servidor verifica si existen mensajes pendientes, lo que indica la necesidad de desplazar al robot. Con ello inicia la búsqueda de un camino, lo que origina un mensaje de dirección.

Si bien la figura 5.6 muestra un proceso donde el robot ya se encuentra en un punto de control, el proceso seguido cuando el robot se encuentra en desplazamiento es muy similar. Si se recibe un mensaje de cambio de posición del usuario o la entrega de recordatorios, éstos se almacenan hasta que el robot alcanza el nuevo punto de control y se realiza el proceso de entrega correspondiente.

Por último, si el robot alcanza un punto de control antes que el proceso de consulta de actividades envíe algún recordatorio, el robot permanece a la espera hasta que sea recibido un recordatorio nuevo que requiera modificar el estado del robot.

## 5.3. Gestor de posición

### 5.3.1. Almacenamiento de mapas

En la sección 5.1.1 se describe la interfaz de que permite la creación del mapa, sin embargo, al ser guardada en la base de datos el mapa solo requiere de dos tablas: *áreas* y *caminos*. En la primer tabla se almacenan las áreas que conforman el entorno, para ello se requiere guardar el marcador codificado para cada área, el nombre del área, así como otros datos que permiten dar persistencia al diagrama utilizado por la interfaz. La tabla *caminos* almacena referencias al área que representa el origen del camino, el área que representa el destino y la dirección que debe seguir el robot para moverse del origen al destino.

Esta representación es útil para lograr la persistencia del mapa sin embargo, no resulta la representación óptima al momento de realizar operaciones, como lo son el trazado de rutas, o incluso la ubicación del robot o los usuarios. Por esta razón, cuando el proceso *generador de mensajes* es iniciado en el servidor, realiza la carga del mapa en memoria con una representación mas adecuada para la búsqueda de rutas y el manejo de ubicaciones.

$$\begin{array}{l}
 \text{nombres} = \begin{pmatrix} \text{área 1} \\ \text{área 2} \\ \text{área 3} \\ \vdots \\ \text{área } n \end{pmatrix} \quad \text{id} = \begin{pmatrix} 0001 \\ 0010 \\ 0011 \\ \vdots \\ 0111 \end{pmatrix} \quad \text{caminos} = \begin{array}{ccccc}
 & 1 & 2 & 3 & \dots & n \\
 \begin{pmatrix} 0 & d & i & \vdots & 0 \\ 0 & 0 & 0 & \vdots & 0 \\ 0 & 0 & i & \vdots & d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & i & d & \vdots & 0 \end{pmatrix} & 1 \\ & & & & & 2 \\ & & & & & 3 \\ & & & & & \vdots \\ & & & & & n
 \end{array}
 \end{array}$$

Figura 5.7: Representación del mapa en memoria

La figura 5.7 muestra las estructuras de datos utilizadas para la representación del mapa en tiempo de ejecución. Los dos vectores iniciales contienen en nombre de las áreas, y la codificación del identificador de cada área. Por otra parte, las rutas que relacionan a las diferentes áreas se representan a partir de la matriz de adyacencia, en la cual, el índice de las filas representa al área origen del camino y las columnas representan el área destinos. El tipo de elementos contenidos en esta matriz son caracteres, los cuales tienen tres posibles valores: 0, *d*, *i*. El carácter '0' es utilizado para indicar que no existe un camino, los caracteres '*d*' e '*i*' son utilizados para indicar que existe un camino, y el robot debe girar hacia la derecha o la izquierda respectivamente, para alcanzar el marcador destino.

### 5.3.2. Ubicación del robot

Como se menciona en secciones anteriores, una vez que el algoritmo de desplazamiento lleva al robot hasta un punto cercano a un marcador, éste envía un conjunto de imágenes al servidor para determinar el identificador del marcador alcanzado, pues, a través del identificador se conoce el área en la que se localiza el robot.

Cada que el servidor recibe una imagen por parte del robot, el primero debe realizar un proceso de análisis de imágenes que permita reconocer las figuras presentes en el marcador. Dependiendo de las figuras encontradas en el identificador del marcador, se crea una cadena que representa un área dentro del mapa. Esta cadena es almacenada de forma provisional, pues el robot envía múltiples imágenes para evitar errores que pueden surgir al momento al capturar alguna imagen.

Cuando el robot termina de enviar todas las imágenes, se realiza un análisis de los identificadores extraídos de ellas, este análisis consiste en determinar el identificador que más veces fue encontrado en las imágenes. El identificador con más apariciones es el identificador del área en la que se encuentra el robot, lo que conlleva una actualización de la localización del robot y una limpieza de los datos almacenados temporalmente.

#### Identificación de figuras en el marcador

El proceso para determinar el identificador de cada marcador se basa en el reconocimiento de las figuras que se encuentran dentro del mismo, para ello es necesario el procesamiento de las imágenes recibidas.

El primer paso en el proceso de identificación de figuras es determinar el área que abarca el marcador dentro de la imagen, para ello se realiza una selección del color magenta como la descrita en el filtrado de marcadores del robot, presente en la sección 4.3.1. Esto permite seleccionar el marcador en la imagen, sin embargo, este proceso afecta parte de los píxeles que pertenecen a las figuras del identificador, pues algunos de los píxeles adquieren una tonalidad magenta, por esta razón, es necesario filtrar, dentro del área que abarca el marcador, los píxeles más oscuros, ya que ellos representan los objetos.

Para filtrar los píxeles más oscuros dentro del área magenta es necesario un filtro adaptativo, que permita la identificación de colores oscuros sin verse afectado por los problemas ocasionados por los cambios de iluminación en el ambiente. Para ello es necesario examinar la luminosidad de los píxeles magenta, en busca de los valores mas altos y bajos. Con estos dos valores se puede realizar una normalización de los niveles de luminosidad en los píxeles del marcador, y establecer un nivel estático a partir del cual un marcador es considerado como píxel oscuro.

En la figura 5.8 se observa el resultado obtenido del proceso de filtrado del color magenta (2) a partir de una imagen capturada por el robot (1), este proceso coloca en color negro los

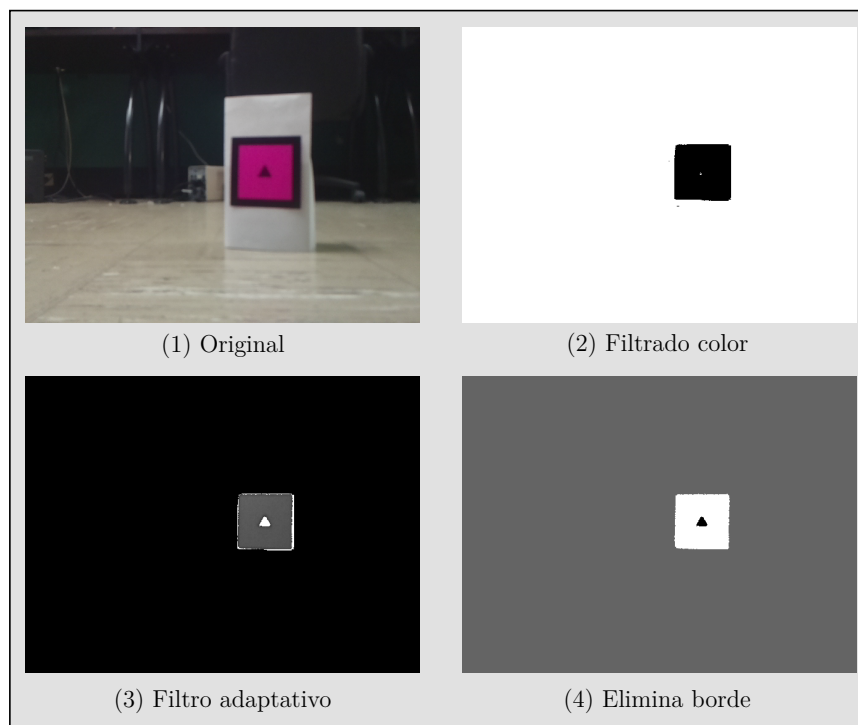


Figura 5.8: Filtrado de identificadores

píxeles considerados como seleccionados. En esta selección de píxeles es necesario eliminar los más oscuros a través del filtro adaptativo, esto da como resultado la imagen (3), en la cual, los píxeles pertenecientes al identificador y borde del marcador están marcados en color blanco, y en gris (valor correspondiente a su luminosidad) los píxeles magenta.

Al analizar la imagen obtenida del filtrado adaptativo se observa que, el identificador se encuentra seleccionado en color blanco, por lo que basta con eliminar el contorno del marcador, el cual también se encuentra seleccionado en color blanco. Para eliminar este contorno es necesario borrar los píxeles blancos que se encuentran entre el fondo de la imagen y el color magenta, lo que da como resultado la imagen (4), en la que se observa el identificador en negro.

### Clasificación de figuras

Una vez que las figuras que conforman el identificador fueron seleccionadas de la imagen, es necesario un proceso de clasificación para poder determinar el tipo de figura. Para ello se utilizan los momentos de Hu, los cuales, como en el caso de la clasificación del marcador, sirven como características representativas de las diferentes clases. Para la construcción del clasificador se utilizó un conjunto de imágenes de prueba obtenidas por el robot, de las cuales fueron extraídas las figuras que conformaban los identificadores,

clasificándolos de forma manual en cuatro clases: *estrella*, *círculo*, *cuadrado* y *triángulo*, pues son los objetos que el usuario puede seleccionar en la interfaz para la creación de mapas.

Clasificadas las figuras que conforman el banco de pruebas, se calcularon los momentos de Hu para todos los objetos, lo que permite observar agrupaciones de las diferentes clases en algunos de los momentos. Con la información observada a partir del análisis de los momentos calculados, se pudo establecer el clasificador descrito en el algoritmo 2.

---

**Algoritmo 2:** Clasificador de objetos
 

---

**Datos:** Momentos de Hu del objeto

```

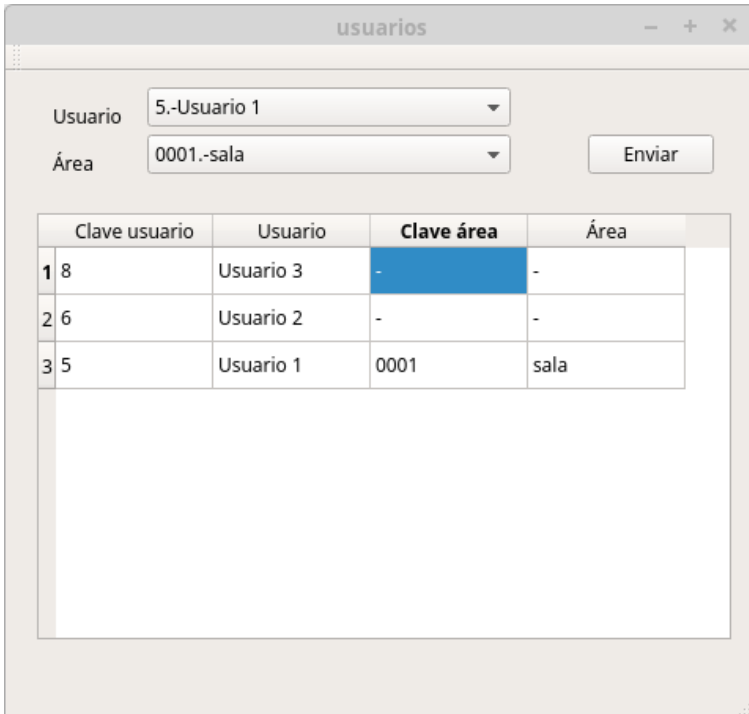
1 si  $0.158 \leq momento0 \wedge momento0 < 0.170$  entonces
2   si  $0.1641 \leq momento0 \wedge momento0 \leq 0.169$  entonces
3     Es un cuadrado
4   en otro caso
5     si  $0.1585 \leq momento0 \wedge momento0 \leq 0.1641$  entonces
6       Es un círculo
7     en otro caso
8       No se reconoce
9     fin
10  fin
11 en otro caso
12  si  $0.170 \leq momento0 \wedge momento0 < 0.375$  entonces
13    si  $0.0005 \leq momento2 \wedge momento2 \leq 0.004$  entonces
14      Es un triángulo
15    en otro caso
16      Es una estrella
17    fin
18  en otro caso
19    Es un cuadrado
20  fin
21 fin

```

---

### 5.3.3. Ubicación de los usuarios

Para que el sistema entregue los recordatorios a los usuarios debe conocer la ubicación de cada uno de ellos, esta ubicación es proporcionada por el gestor de ubicaciones. Para la implementación del sistema se realizó una simulación de ubicaciones de los usuarios, por lo cual fue construida la interfaz a partir de la cual es posible cambiar la ubicación del usuario, e informar al servidor de dicho cambio.



	Clave usuario	Usuario	Clave área	Área
1	8	Usuario 3	-	-
2	6	Usuario 2	-	-
3	5	Usuario 1	0001	sala

Figura 5.9: Interfaz para simulación de localización de usuarios

La figura 5.9 muestra la interfaz a partir de la cual se simula la posición de cada uno de los usuarios en el sistema. Esta interfaz realiza una conexión con la base de datos y extrae una lista de los usuarios, de igual forma, obtiene la lista de las áreas que conforman el mapa, con lo cual configura los elementos de selección.

Obtenidos estos datos, realiza un conexión con el servidor para registrarse como gestor de posición de los usuarios. Con el registro realizado, permite la selección de un usuario y un área determinada, lo que conlleva la creación del mensaje que informa al servidor que un usuario se localiza en un área determinada.

La interfaz también cuenta con una tabla en la cual se puede observar la localización de cada usuario, lo que permite un mejor control de la distribución del usuario para las pruebas del sistema.



### 5.3.4. Generador de rutas

En la sección 5.3.1 se muestra la estructura de datos utilizada para representar el mapa. En esta representación las rutas cuentan con una dirección, pero no con un peso. Bajo estas consideraciones, la creación de rutas que permitan el desplazamiento del robot entre las diferentes áreas se realiza mediante el uso del algoritmo de Dijkstra, el cual permite la exploración de un grafo al identificar la ruta óptima a partir de un nodo origen [9].

---

#### Algoritmo 3: Algoritmo de Dijkstra

---

**Datos:** Grafo  $G$ , nodo\_origen  $s$

```

1 para  $u \in V[G]$  hacer
2    $d[v] = \infty$ 
3    $antecesor[v] = NULL$ 
4 fin
5  $d[s] = 0$ 
6 agrega (cola, ( $s, d[s]$ ))
7 mientras cola no es vacía hacer
8    $u = \text{extraer\_Mínimo}(\text{cola})$ 
9   para todo  $v \in \text{listaAdyacentes}[u]$  hacer
10    si  $d[v] > d[u] + 1$  entonces
11       $d[v] = d[u] + 1$ 
12       $antecesor[v] = u$ 
13      agrega(cola, ( $v, d[v]$ ))
14    fin
15  fin
16 fin

```

---

El algoritmo de Dijkstra obtenido de [37], presente en el pseudocódigo 3, muestra la generación de los antecesores óptimos de cada uno de los nodos, al tomar como nodo inicial el nodo *origen*. Con esta información se puede crear la ruta mas mas corta a cualquier nodo, pues al estar ubicado en el nodo origen, basta con recorrer, desde el nodo destino los nodos identificados como antecesores, hasta alcanzar el nodo origen. Una vez se determina el camino hacia un área, tomando como origen ubicación del robot, se identifica la siguiente área que debe visitar y por consiguiente, la dirección hacia la cual debe de girar el robot y comenzar con el algoritmo de desplazamiento.

Este proceso de identificación de rutas se genera cada que el robot necesita desplazarse hacia una nueva área, lo que permite la actualización del área de origen cada que llega a un nodo intermedio, con el objetivo de cambiar la ruta si el usuario, es decir, el objetivo al que se debe desplazar el robot, modifica su posición.



## Pruebas y resultados

### 6.1. Robot

Uno de los principales resultados de la tesis es la creación del robot para la entrega de recordatorios, este robot, por sus características de diseño, puede desplazarse en entornos donde el piso no contenga variaciones significativas, como lo pueden ser escalones o barreras altas. Si bien el desplazamiento del robot se restringe a entornos con un piso liso, al estar pensado para funcionar en viviendas de adultos mayores, esta limitante no representa un inconveniente mayor. Por otra parte, al contar con un diseño sencillo, el costo en la creación del robot se redujo.

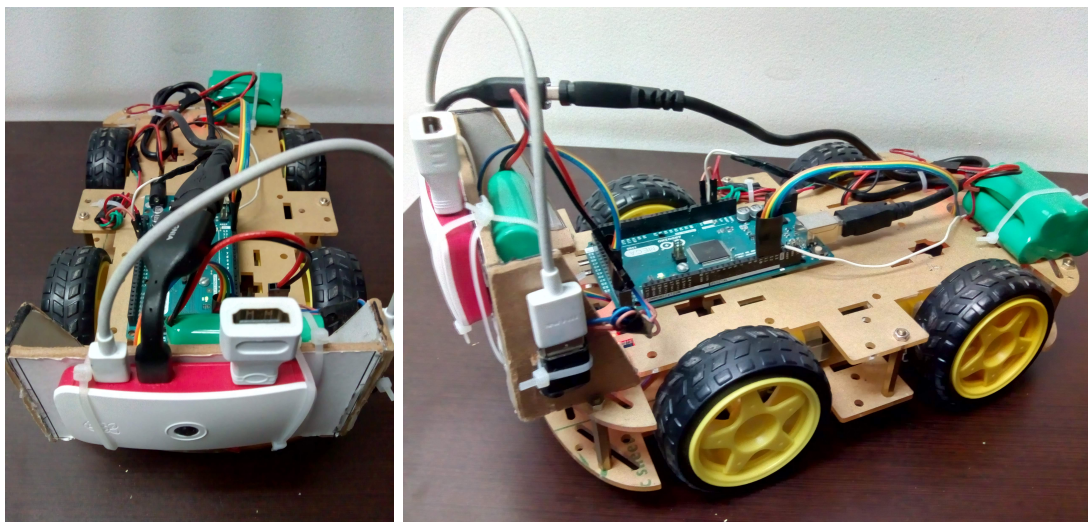


Figura 6.1: Vista frontal y lateral izquierda del robot

En las figuras 6.1 y 6.2, se presentan cuatro vistas del robot construido. En la vista frontal (6.1, izquierda) se aprecia, dentro de la carcasa roja, la *Raspberry pi zero w* con su

módulo de cámara, la cual cuenta con tres puertos de conexión: el puerto de alimentación (blanco), el puerto de datos (negro) y el puerto mini HDMI (desconectado). El puerto de alimentación micro USB se conecta al conector hembra USB-B que se muestra en la vista lateral izquierda (6.1). El puerto de datos es un conector micro USB, por esta razón se utiliza un cable adaptador OTG para conectarlo con la Arduino mega.

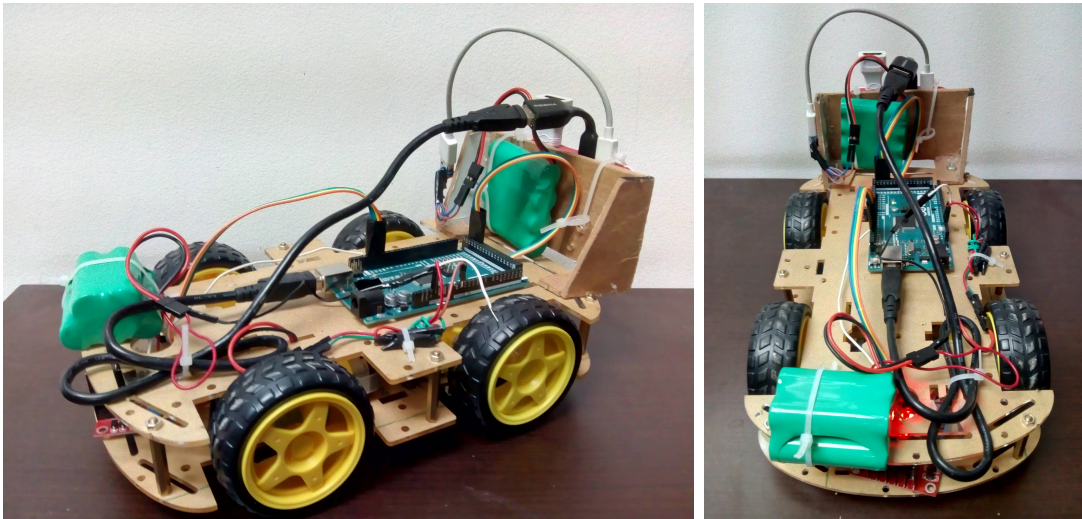


Figura 6.2: Vista lateral derecha y trasera del robot

En la figura 6.2 se aprecia la ubicación de las baterías utilizadas para alimentar a la Raspberry y al conjunto de motores. También se aprecia la posición en la que es colocada la Arduino, tarjeta encargada de codificar la señal de activación de los motores ubicados en el nivel inferior del robot. En dicho nivel, visto en la figura 6.3, se localizan también los dos controladores de motor y la batería encargada de la alimentación de la Arduino.

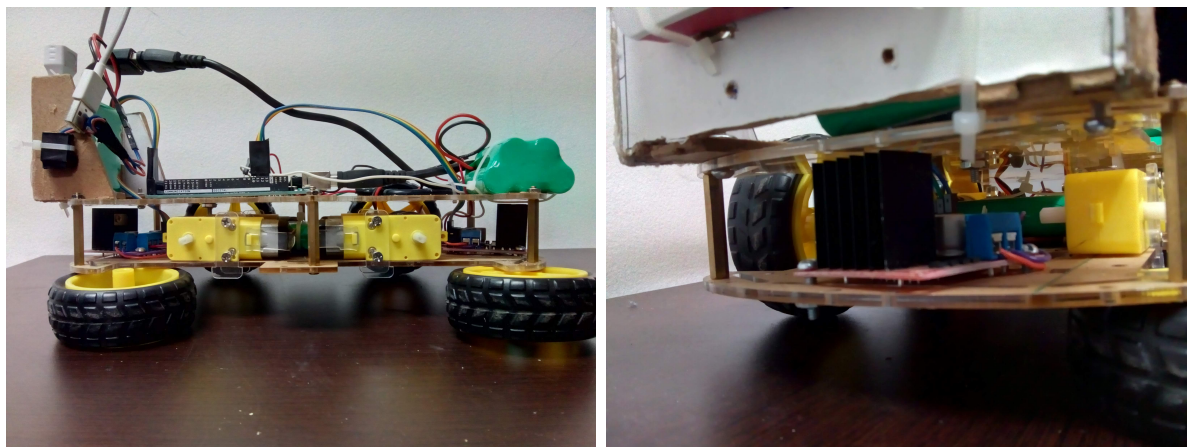


Figura 6.3: Nivel inferior, vista lateral y frontal

## 6.2. Desplazamiento e identificación de los marcadores

Para determinar si el método de navegación utilizado funciona correctamente, es necesaria la verificación de los diferentes procesos relacionados con la navegación. Entre estos procesos se encuentra el desplazamiento del robot hacia los marcadores. Dónde se evalúa, además del desplazamiento, el funcionamiento del robot, pues las ordenes generadas por el análisis de las imágenes, se deben transferir correctamente hacia las ruedas del robot.

Para realizar la evaluación del desplazamiento hacia los marcadores se debe verificar que el robot coordina los movimientos de desplazamiento a fin de mantener al marcador centrado en las imágenes. Por lo anterior, las pruebas realizadas para este proceso consisten en ubicar al robot en diferentes posiciones con respecto al marcador, he iniciar la orden correspondiente para que el robot inicie el desplazamiento. Alcanzado el marcador, el robot se detiene y pueden ser extraídos los datos generados por la prueba.

Los datos generados por la prueba son producidos cada vez que el robot realiza la captura de una imagen, pues en ese momento se determina el siguiente movimiento de desplazamiento. Si dentro de la imagen capturada se ha podido localizar un marcador, los datos que se generan son: el número de imágenes que han sido capturadas; el área del marcador encontrado; y la posición del dentro del marcador con respecto al eje horizontal. En caso que la imagen no contenga un marcador, solo es necesario capturar el número de imagen, pues con ello se entiende que el marcador no fue encontrado.

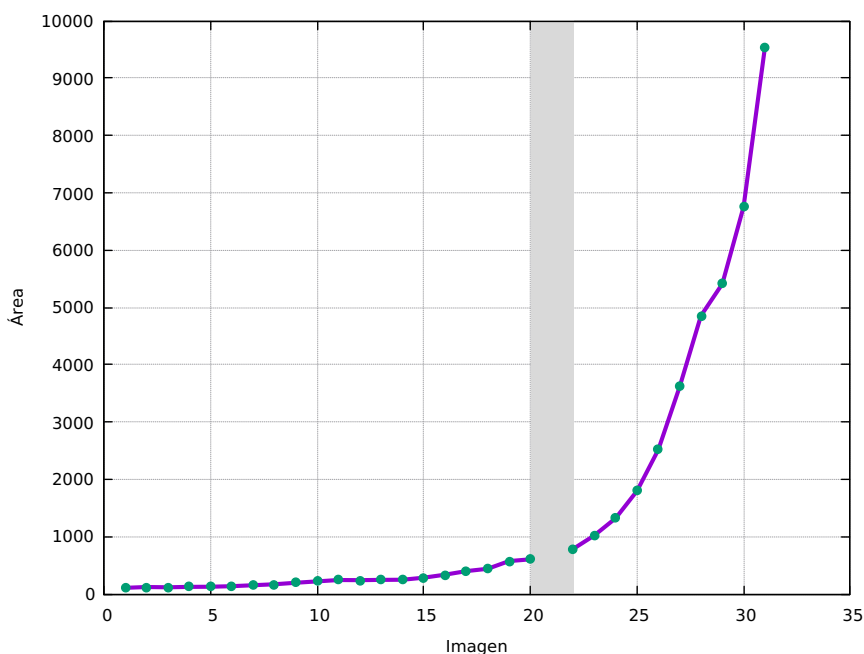


Figura 6.4: Crecimiento del área del marcador

Con base en los datos generados por las pruebas, es posible observar diferentes características como el incremento del área del marcador en las imágenes conforme el robot se acerca al punto de control, o incluso los mecanismos de control de desplazamiento que permiten la corrección de la dirección con base en la posición del marcador.

La figura 6.4 muestra el crecimiento del área del marcador dentro de la imagen, conforme el robot se acerca. Esto permite analizar que dicho crecimiento no se realiza de forma lineal, lo que indica que a mayor cercanía el crecimiento esperado debe ser mayor, lo cual puede afectar la calibración necesaria para determinar la cercanía del robot con el marcador. También es posible observar que en la imagen número 20, el robot no encontró ningún marcador, sin embargo, el mecanismo de corrección de la ruta le permitió localizarlo nuevamente en la imagen 21.

Con el análisis del área del marcador se establece un límite a partir del cual se considera que el robot está lo suficientemente cerca de dicho marcador, esto permite determinar la distancia óptima para enviar al servidor un conjunto de imágenes que permitan codificar el identificador del marcador. De modo similar a la evaluación del área, es necesario verificar el control de dirección del robot, proceso encargado del desplazamiento del mismo.

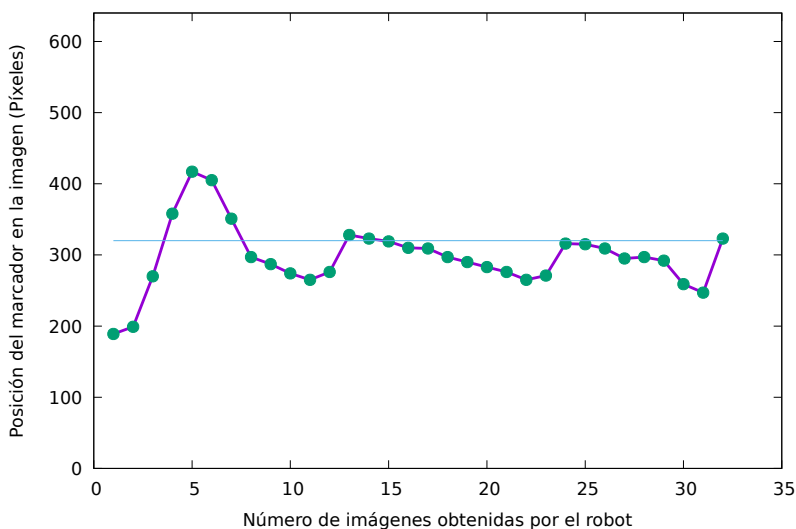


Figura 6.5: Posición del marcador en la prueba 1

Para describir algunas de las características observadas en el desplazamiento del robot se seleccionaron seis pruebas, las cuales se presentan a través de gráficas encontradas en las figuras 6.5, 6.6; que corresponden a pruebas en donde el robot se colocó a seis metros de distancia del marcador, las pruebas de larga distancia se presentan en la figura 6.7, y las pruebas donde se realizaron pérdidas del marcador se encuentran en las figuras 6.8 y 6.9.

Las gráficas antes mencionadas muestran la posición del marcador dentro de las imágenes que el robot capturó durante la prueba, por lo cual, se resaltó el centro de la imagen con una línea horizontal. Esto permite que observar que, si un marcador se encuentra por encima de los 320 px, el marcador se encuentra a la derecha de la imagen y por consiguiente a la derecha del robot. Por el contrario, si el centro del marcador se encuentra por debajo de los 320 px, el marcador se encuentra a la izquierda del robot.

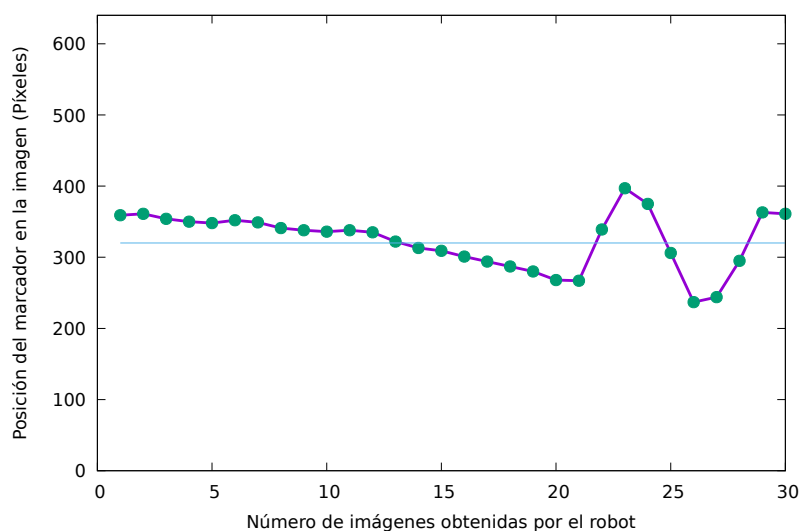


Figura 6.6: Posición del marcador en la prueba 2

En las pruebas 1 y 2, el robot fue colocado a una distancia de 6 m de un marcador con 10 cm de lado. En estas pruebas la orden indicada para el robot fue la correspondiente al desplazamiento hacia al marcador y se almacenó la posición del marcador en el eje horizontal para cada imagen que el robot obtuvo.

En la gráfica 6.5 se observa que en las primeras imágenes obtenidas, el marcador se encontraba en una posición inferior a los 200 px, es decir, el marcador se encontraba a la izquierda de la posición el robot, motivo por el cual se observa el ajuste realizado por el robot en las siguientes imágenes capturadas. A partir de la imagen 14 el robot logra centrar el marcador, pero debido a que el robot presenta una tendencia de girar hacia la derecha mientras avanza recto es necesario realizar ajustes continuos.

El caso que se muestra en la figura 6.6, se observa que el robot inicia con un marcador centrado, por ello, la orden realizada por el robot es avanzar recto, esta orden se mantiene durante las primeras 20 imágenes, sin embargo, se observa, al igual que en la prueba anterior, la tendencia del robot de avanzar recto con un pequeño desplazamiento hacia la derecha, lo que ocasiona que el robot realice ajustes para mantener el marcador centrado en la imagen 21. En esta prueba también se observa que, al encontrarse cercano al marcador, los giros que realiza el robot tienen un mayor efecto en la posición del marcador en la imagen, lo que a

su vez ocasiona un incremento en las correcciones.

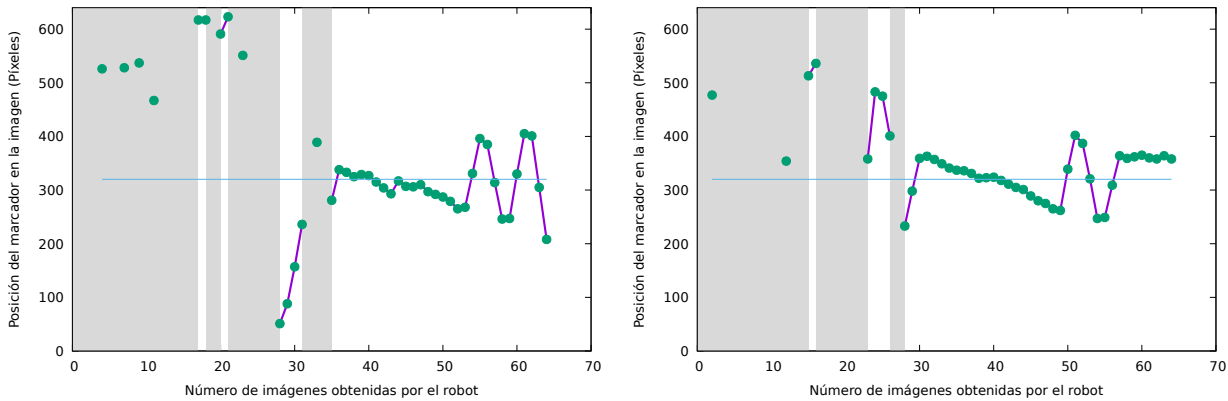


Figura 6.7: Posición del marcador en las pruebas a distancia 3 y 4

Para las pruebas 3 y 4, el robot se colocó a una distancia mayor que las pruebas anteriores, con el objetivo de analizar la distancia conveniente a la que el robot realiza la identificación de los mismos. En la primer gráfica de la figura 6.7 (izquierda), se muestran los resultados generados por una prueba en donde el robot fue ubicado a 8 metros de distancia de un marcador con 10 cm de largo. El resultado observado muestra que el robot no pudo ubicar al marcador al inicio de la prueba, por esta razón inició con el procedimiento de búsqueda de un marcador, esto permitió que se encontraran algunos marcadores, sin embargo, se perdió múltiples veces. Al llegar a la imagen 28, el robot se encontraba a mayor cercanía con el marcador, lo que permitió la ubicación de un mayor número de marcadores, y si bien, lo perdió de vista en alguna ocasión, se logró una ubicación más rápida.

En la prueba número 4 presente a la derecha de la figura 6.7, el robot tuvo un comportamiento similar, pues perdió de vista al marcador en múltiples ocasiones. Los resultados de estas pruebas indican que a distancias mayores a los 6 metros el robot presenta mayores dificultades para localizar marcadores de 10 cm, por lo consiguiente una opción útil en ambientes donde las distancias que debe recorrer el robot son mayores, es utilizar marcadores con mayor tamaño.

El procedimiento descrito para los casos en los que el robot se encuentra perdido es analizar el último movimiento realizado, si éste fue un giro, el robot realiza un giro al lado opuesto. Este procedimiento está descrito de esta forma, debido a que el robot suele perderse cuando realiza un ajuste para mantener centrado al marcador. Sin embargo, en la prueba 5, presente en la figura 6.8, la pérdida del marcador se realizó cuando el robot se dirigía en línea recta. La acción tomada por el robot en estos casos es retroceder levemente, esto debido a que la pérdida del marcador pudo ser ocasionada por el ajuste de la cámara. La ejecución de esta prueba permite evaluar que el sistema realizó el procedimiento indicado, lo que dio como resultado la relocalización del marcador.



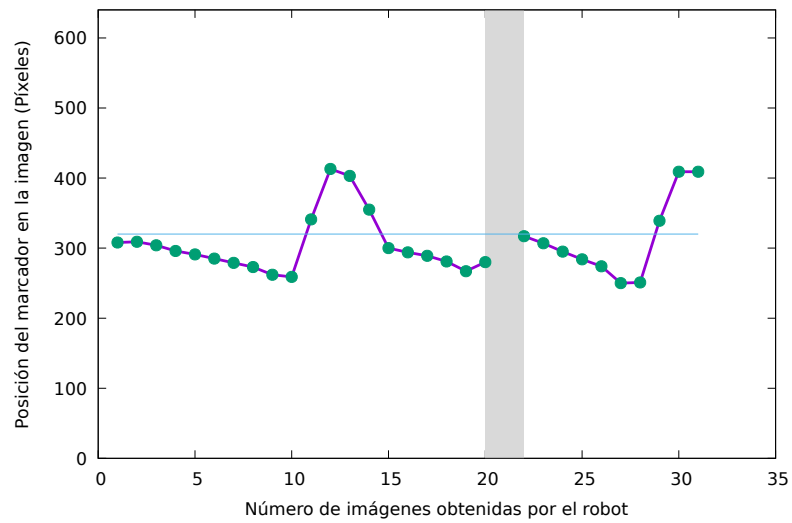


Figura 6.8: Posición del marcador en la prueba 5

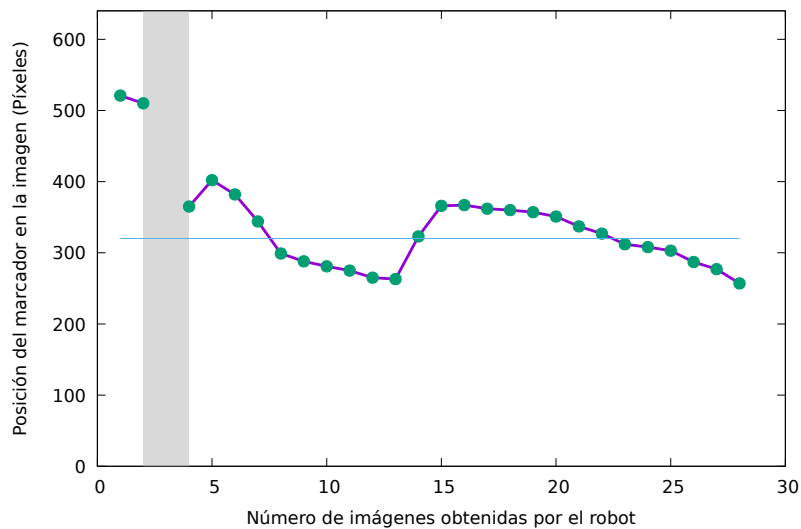


Figura 6.9: Posición del marcador en la prueba 6

En la sexta prueba seleccionada, presente en la figura 6.9 la pérdida del marcador se presenta en las primeras imágenes capturadas por el robot, con el marcador ubicado en un extremo de la imagen. Esta prueba ejemplifica los casos en los que, el marcador se sale del campo de visión del robot debido al giro de ajuste realizado, sin embargo, como se describe en el párrafo anterior, el robot realiza un giro al lado opuesto para ubicar al marcador nuevamente en la imagen.

### 6.3. Identificación del marcador

La correcta extracción de la información contenida en el marcador es una etapa de suma importancia en la navegación del robot, pues a través de este proceso se determina la localización del robot.

Como se describe en el capítulo de implementación, el proceso comienza con la llegada de un conjunto de imágenes al servidor, las cuales indican que el robot alcanzó un punto de control. Este conjunto de imágenes es procesado a fin de extraer las figuras que conforman el identificador del marcador. Una vez las figuras son extraídas, el servidor clasifica las dichas figuras acorde con su forma, lo que permite la creación de un identificador.

Para probar el correcto funcionamiento de este proceso se utilizaron tres tipos de marcadores: marcadores con una figura; marcadores con dos figuras; y marcadores con tres figuras. Para cada una de las clases de marcadores se capturaron 300 imágenes por medio del robot. La ubicación del robot para realizar la captura de las imágenes se determinó a través del algoritmo de desplazamiento del robot, es decir, al iniciar el algoritmo de seguimiento del marcador desde diferentes puntos de partida, la posición final del robot permitió la obtención del marcador con diferentes poses.

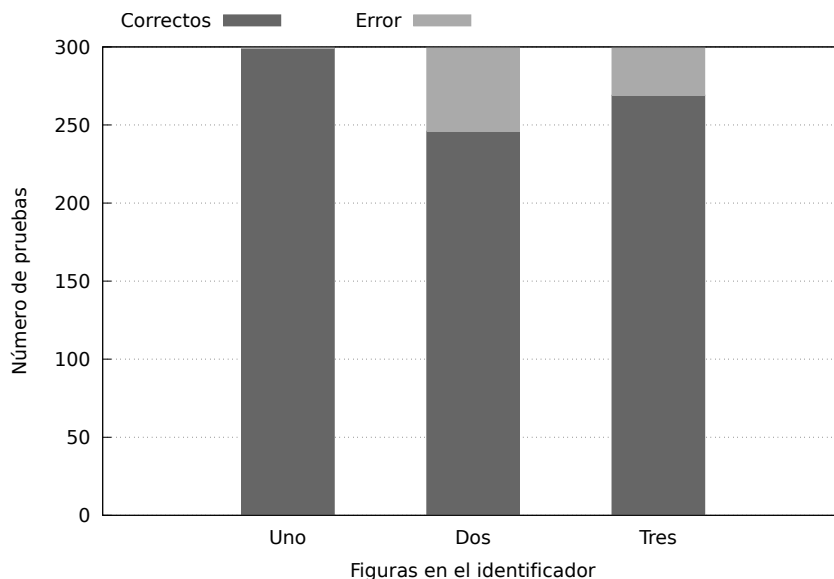


Figura 6.10: Errores en la clasificación de figuras

Para cada una de las imágenes, el resultado obtenido del proceso de identificación del marcador fue comparado con el valor real, esto dio como resultados los datos expresados en la gráfica presente en la figura 6.10. En esta gráfica se observa que, para el caso de marcadores con una sola figura como identificador la precisión alcanzada fue mayor que con las clases con mayor número de figuras.

Si bien, a primera instancia el error ocurrido en las clases con mayor número de figuras en el identificador puede ser atribuido a un fallo en el proceso de clasificación de figuras, un análisis más detallado de las imágenes capturadas por el robot muestra que el error se ocasiona debido que, en algunos de los casos, las imágenes enviadas por el robot no contienen en su totalidad al marcador, por esta razón, al realizar el procesamiento del mismo se omite parte de la información.

Debido a lo anterior, se fue incrementado el número de figuras enviadas al marcador, las cuales permiten determinar la posición del robot mediante un conteo del identificador con mayor número de apariciones en las imágenes. También se anexó al proceso de navegación del robot, un subproceso de centrado, el cual es activado una vez que el robot alcanza al marcador. En este subproceso el robot realiza pequeños giros a modo de centrar el marcador, para así enviar al servidor imágenes con el marcador en mejor posición. Con las medidas descritas anteriormente, el error mostrado en las pruebas se redujo a fin de no interferir con la determinación de la posición del robot.

## 6.4. Conexión entre el servidor y el gestor de tareas

Las interfaces para la gestión de citas y medicamentos permiten, tanto a los familiares como médicos del adulto mayor programar los mensajes que deberán ser entregados al adulto mayor. Una vez que la información es ingresada a las interfaces, el gestor de tareas verifica el momento adecuado para enviar los mensajes hacia el servidor, el cual se encarga de coordinar las acciones del robot con el fin de entregar dichos mensajes.

Para probar el correcto funcionamiento del gestor de tareas es indispensable verificar, en primera instancia, que la programación de las tareas se realiza de forma correcta. Por esta razón, en el sistema se realizó la programación de dos citas así como la ingesta de un medicamento, información que se presenta en la figura 6.11.

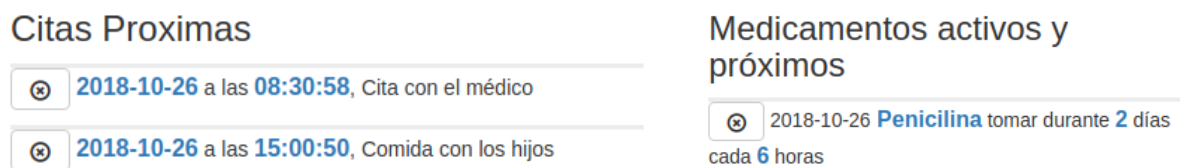


Figura 6.11: Datos utilizados para la prueba de creación de mensajes

La fecha seleccionada para el inicio de los tres eventos fue el 26 de Octubre del 2018. En esta fecha se agendó una cita con el médico a las 8:30 de la mañana, una comida con los hijos a las 15:00 horas y el inicio de la ingesta de Penicilina durante dos días.

La configuración del servidor para las horas de entrega define la creación de 5 recordatorios, el primero de ellos 5 horas antes del evento, los siguientes mensajes se entregan 3, 2 y 1 hora antes del evento, dejando el último mensaje para la hora programada. Por otra parte, la hora programada para el iniciar la entrega de recordatorios es a las 6:00 am, por lo que el sistema no puede programar recordatorios antes de esta hora.

```
mysql> select * from vw_mensajeCita;
+-----+-----+-----+-----+-----+-----+
| id | horaEvento | fechaEvento | usuario | mensaje | horaMsj |
+-----+-----+-----+-----+-----+-----+
| 18 | 08:30:58 | 2018-10-26 | 5 | Cita con el médico | 2018-10-26 06:30:58 |
| 18 | 08:30:58 | 2018-10-26 | 5 | Cita con el médico | 2018-10-26 07:30:58 |
| 18 | 08:30:58 | 2018-10-26 | 5 | Cita con el médico | 2018-10-26 08:30:58 |
| 20 | 15:00:50 | 2018-10-26 | 5 | Comida con los hijos | 2018-10-26 10:00:50 |
| 20 | 15:00:50 | 2018-10-26 | 5 | Comida con los hijos | 2018-10-26 12:00:50 |
| 20 | 15:00:50 | 2018-10-26 | 5 | Comida con los hijos | 2018-10-26 13:00:50 |
| 20 | 15:00:50 | 2018-10-26 | 5 | Comida con los hijos | 2018-10-26 14:00:50 |
| 20 | 15:00:50 | 2018-10-26 | 5 | Comida con los hijos | 2018-10-26 15:00:50 |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> █
```

Figura 6.12: Recordatorios de citas almacenados en la base de datos

Con base en la información anterior, los recordatorios de citas almacenados en la base de datos deben ser tres para la cita con el médico y 5 para la comida con los hijos, esto debido a que la cita con el médico es tan temprano que, la configuración del servidor impide la creación los mensajes correspondientes a las 5 y 3 horas antes del evento. La figura 6.12 muestra la programación de recordatorios de citas que será leída por el gestor de tareas, esta información se encuentra almacenada en la base de datos y permite verificar que la programación se realiza conforme a lo estipulado en la configuración del servidor.

```
mysql> select * from vw_mensajeMed;
+-----+-----+-----+-----+-----+-----+
| id | horaEvento | fechaEvento | usuario | mensaje | horaMsj |
+-----+-----+-----+-----+-----+-----+
| 14 | 2018-10-26 06:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-26 06:00:00 |
| 14 | 2018-10-26 12:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-26 12:00:00 |
| 14 | 2018-10-26 18:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-26 18:00:00 |
| 14 | 2018-10-27 00:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-27 00:00:00 |
| 14 | 2018-10-27 06:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-27 06:00:00 |
| 14 | 2018-10-27 12:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-27 12:00:00 |
| 14 | 2018-10-27 18:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-27 18:00:00 |
| 14 | 2018-10-28 00:00:00 | 2018-10-26 | 5 | Penicilina | 2018-10-28 00:00:00 |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Figura 6.13: Recordatorios medicamentos almacenados en la base de datos

La configuración del servidor para los horarios de ingesta de medicamentos indica que, en medicamentos donde el lapso de ingesta es cada seis horas, el horario de inicio de la

medicación se programa a las seis de la mañana. Con base en esta información, se deben crear 8 mensajes para el tratamiento configurado, tal y como se presenta en la figura 6.13. En dicha figura se presentan los mensajes programados para la información capturada.

### 6.4.1. Generación de mensajes

Verificada la correcta programación de los recordatorios con base en la información proporcionada a través de las interfaces, es necesario comprobar que el gestor de tareas informa al servidor acerca de los recordatorios que fueron programados en el tiempo exacto que son requeridos.

```
mysql> select * from vw_mensajeCita where id = 21;
+-----+-----+-----+-----+-----+-----+
| id | horaEvento | fechaEvento | usuario | mensaje | horaMsj |
+-----+-----+-----+-----+-----+-----+
| 21 | 15:57:50 | 2018-10-25 | 5 | Mensaje de prueba | 2018-10-25 14:57:50 |
| 21 | 15:57:50 | 2018-10-25 | 5 | Mensaje de prueba | 2018-10-25 15:57:50 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Figura 6.14: Datos para verificar el envío de mensajes al servidor

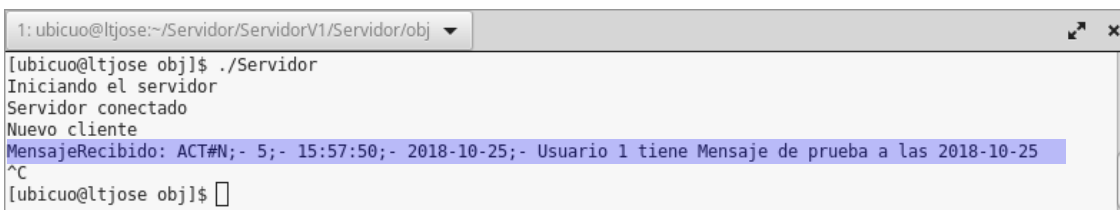
Para probar el mecanismo de entrega de mensajes se programó un recordatorio de prueba a través de la interfaz para programación de citas. Este recordatorio generó un mensaje que debía ser entregado a las 15:57:50, tal y como lo muestra la figura 6.14.

```
2: ubicuo@ltjose:~/Dropbox/GestionTareas/build-gestionTareas-Desktop-Debug
[ubicuo@ltjose build-gestionTareas-Desktop-Debug]$ ./gestionTareas
Conectando con el servidor
Tiempo: 2018-10-25 14:57:12
[2018-10-25 14:57:12] Enviando datos
QSqlDatabasePrivate::removeDatabase: connection 'conecta' is still in use, all queries will cease to work.
-----
Mensaje rec: Registrado
Tiempo: 2018-10-25 14:57:22
[2018-10-25 14:57:22] Enviando datos
QSqlDatabasePrivate::removeDatabase: connection 'conecta' is still in use, all queries will cease to work.
-----
Tiempo: 2018-10-25 14:57:32
[2018-10-25 14:57:32] Enviando datos
QSqlDatabasePrivate::removeDatabase: connection 'conecta' is still in use, all queries will cease to work.
-----
Tiempo: 2018-10-25 14:57:42
[2018-10-25 14:57:42] Enviando datos
Enviando: ACT#N;- 5;- 15:57:50;- 2018-10-25;- Usuario 1 tiene Mensaje de prueba a las 2018-10-25
QSqlDatabasePrivate::removeDatabase: connection 'conecta' is still in use, all queries will cease to work.
-----
Tiempo: 2018-10-25 14:57:52
[2018-10-25 14:57:52] Enviando datos
QSqlDatabasePrivate::removeDatabase: connection 'conecta' is still in use, all queries will cease to work.
-----
```

Figura 6.15: Ejecución del proceso gestor de tareas

Con el mensaje almacenado en la base de datos, la prueba comenzó al arrancar el proceso principal del servidor. El segundo proceso en iniciarse fue el gestor de tareas, el cual envió el mensaje de registro al servidor una vez fue iniciado. Realizado el registro del gestor de tareas en el servidor, el gestor de tareas comenzó con una lectura periódica de la base de datos, a fin de identificar si existía algún mensaje programado para el lapso de tiempo estipulado (10 segundos).

En la figura 6.15 se muestra la ejecución realizada por el gestor de tareas, en donde se aprecian los accesos a la base de datos cada 10 segundos, iniciando a las 14:57:12 hrs. En la figura también se observa que al realizar el cuarto acceso a la base de datos, esto a las 14:57:42, el gestor de tareas detectó que debía informar al servidor la presencia de un mensaje dentro de ese lapso de tiempo, por lo cual, realizó el envío mediante la conexión previamente realizada, continuando con las lecturas periódicas sin encontrar nuevos mensajes. Esto comprueba que el gestor de tareas realiza el filtrado de los mensajes de forma correcta.



```
1: ubicuo@ltjose:~/Servidor/ServidorV1/Servidor/obj
[ubicuo@ltjose obj]$ ./Servidor
Iniciando el servidor
Servidor conectado
Nuevo cliente
MensajeRecibido: ACT#N;- 5;- 15:57:50;- 2018-10-25;- Usuario 1 tiene Mensaje de prueba a las 2018-10-25
^C
[ubicuo@ltjose obj]$
```

Figura 6.16: Recepción de recordatorio en el servidor

En la figura 6.16 se muestra la ejecución del servidor, en la cual se aprecian las impresiones generadas por el proceso de arranque del mismo. Una vez listo el servidor, se informa que fue recibida una nueva conexión a través del mensaje **Nuevo Cliente**, esta conexión corresponde al proceso para la gestión de tareas estableciendo el mecanismo de comunicación entre ambos procesos. Una vez la comunicación fue establecida, la impresión en pantalla generada por el servidor corresponde a la recepción de un mensaje generado por el gestor de tareas, el cual corresponde al mensaje almacenado en la base de datos.

## 6.5. Posición de los usuarios

Para que el robot entregue los mensajes producidos por el gestor de tareas, el sistema SiGA debe conocer la ubicación de los usuarios que participan en el entorno. Por esta razón se implementó una interfaz para el gestor de posición, la cual permita registrar la localización de los diferentes usuarios dentro del entorno. Esta interfaz obtiene los datos correspondientes a las áreas y usuarios del sistema desde la base de datos, lo que permite ubicar al usuario dentro de un área.

Por lo anterior, es necesario comprobar que el servidor del sistema SiGA es informado de los cambios correspondientes a la ubicación del usuario. Esta comprobación se realiza

ejecutando el servidor y el gestor de posición, para posteriormente generar un conjunto de cambios de posición de los usuarios en el gestor de ubicaciones, mientras se comprueba que la ubicación es actualizada de forma correcta en el servidor.

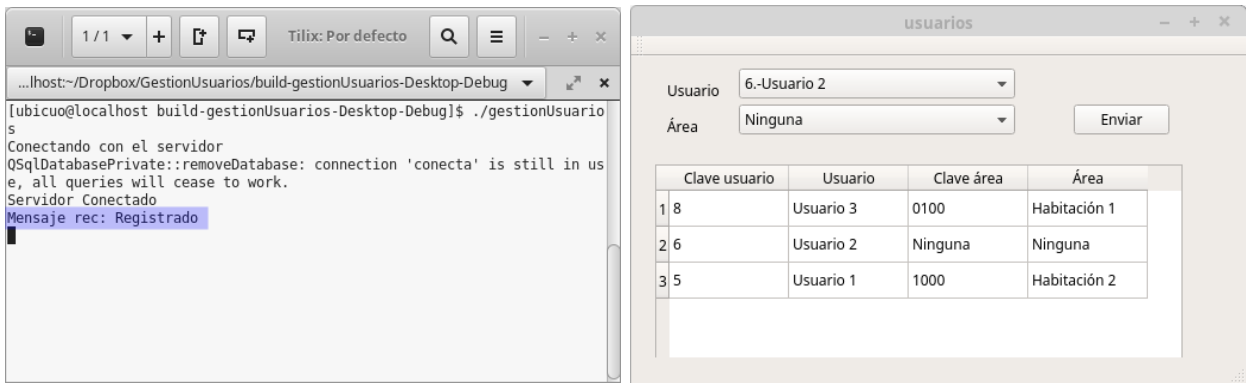


Figura 6.17: Ejecución del gestor de posición

En la figura 6.17, se muestra el resultado obtenido al ejecutar el gestor de posición. En el resultado en consola se puede apreciar el mensaje a través del cual el servidor informa que el gestor de posición se conectó con el servidor de forma correcta. Una vez se realiza la conexión, se utilizó la interfaz realizar los siguientes cambios:

- Usuario 1 con id 5 registra su llegada al área *Sala*
- Usuario 2 con id 6 registra su llegada al área *Comedor*
- Usuario 3 con id 8 registra su llegada al área *Habitación 1*
- Usuario 1 con id 5 registra su llegada al área *Habitación 2*
- Usuario 2 con id 6 se retira del entorno.

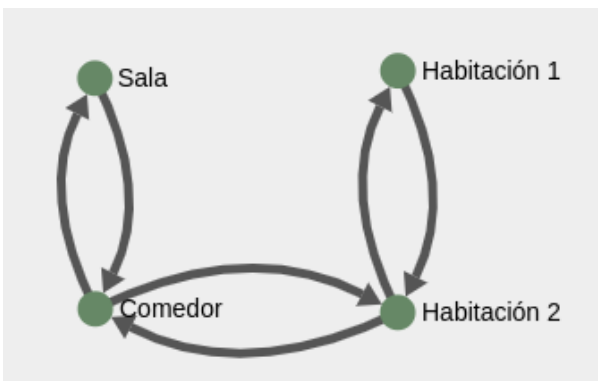


Figura 6.18: Mapa utilizado para prueba

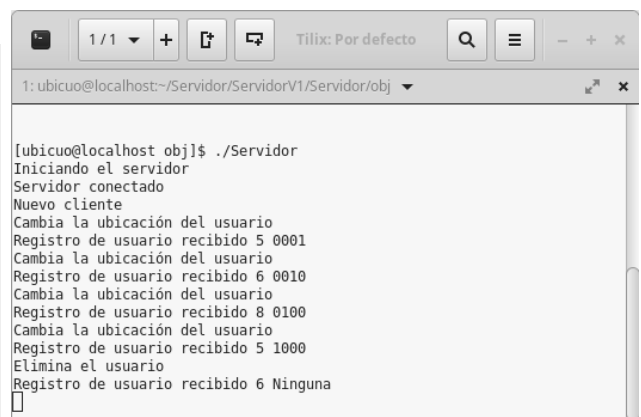


Figura 6.19: Cambio de ubicación de los usuarios en el servidor

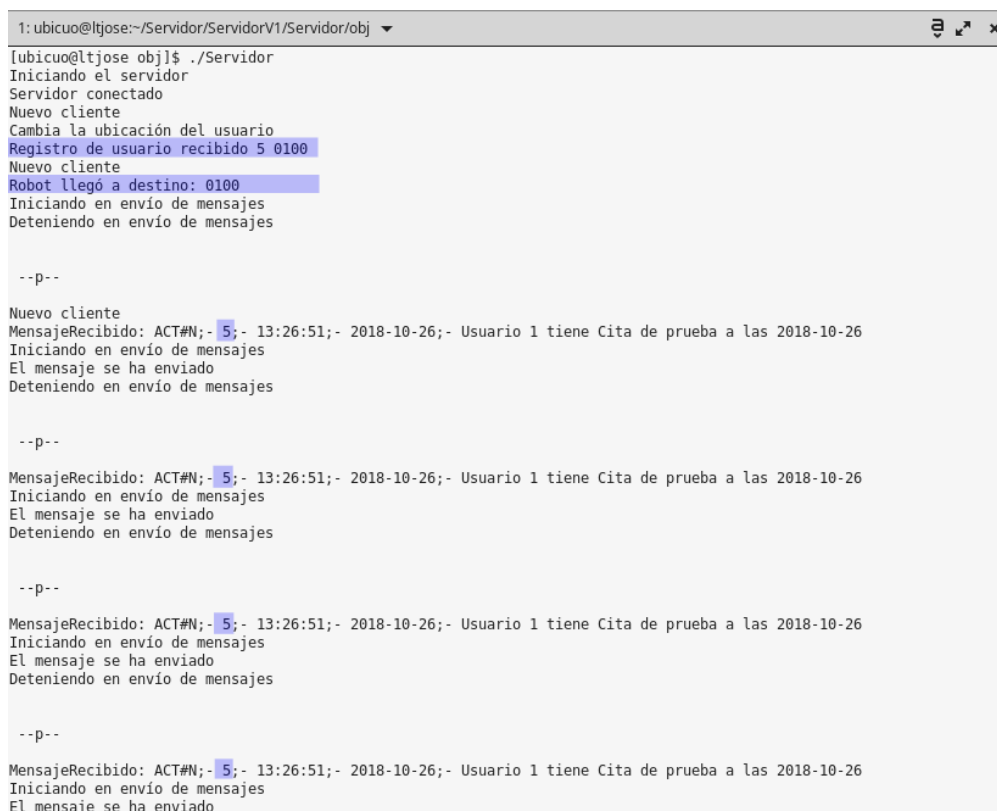
Las áreas registradas para esta prueba son cuatro: sala [0001], comedor [0010], habitación 1 [0100] y habitación 2 [1000], tal y como se aprecia en la figura 6.18; por esta razón, en la figura 6.19 se muestra el resultado obtenido en el servidor, en los cuales se aprecian los diferentes cambios en las ubicaciones, lo que corrobora la comunicación entre el gestor de posición y la actualización en el servidor.

## 6.6. Envío de recordatorios al robot

En las secciones anteriores se evaluó el desplazamiento del robot, así como la comunicación del servidor con los gestores de posición y de tareas. Esta comunicación permite que el servidor conozca la información que debe ser entregada, así como la ubicación del objetivo. Por consiguiente, la siguiente característica que debe ser evaluada es la capacidad del robot para establecer comunicación con el servidor a modo de obtener la información correspondiente a los mensajes y entregarlos al usuario.

El objetivo de esta verificación no es evaluar el desplazamiento del robot, por esta razón en esta prueba el robot se encuentra frente a un marcador que representa un área determinada, misma que será utilizada para indicar la localización del usuario. En otras palabras, el robot se encuentra en el área destino, por lo que no será necesario que se realice un desplazamiento.





```
1:ubicuo@ltjose:~/Servidor/ServidorV1/Servidor/obj
[ubicuo@ltjose obj]$ ./Servidor
Iniciando el servidor
Servidor conectado
Nuevo cliente
Cambia la ubicación del usuario
Registro de usuario recibido 5 0100
Nuevo cliente
Robot llegó a destino: 0100
Iniciando en envío de mensajes
Deteniendo en envío de mensajes

--p--

Nuevo cliente
MensajeRecibido: ACT#N;- 5;- 13:26:51;- 2018-10-26;- Usuario 1 tiene Cita de prueba a las 2018-10-26
Iniciando en envío de mensajes
El mensaje se ha enviado
Deteniendo en envío de mensajes

--p--

MensajeRecibido: ACT#N;- 5;- 13:26:51;- 2018-10-26;- Usuario 1 tiene Cita de prueba a las 2018-10-26
Iniciando en envío de mensajes
El mensaje se ha enviado
Deteniendo en envío de mensajes

--p--

MensajeRecibido: ACT#N;- 5;- 13:26:51;- 2018-10-26;- Usuario 1 tiene Cita de prueba a las 2018-10-26
Iniciando en envío de mensajes
El mensaje se ha enviado
Deteniendo en envío de mensajes
```

Figura 6.20: Resultados del servidor al verificar el envío de mensajes al robot

El primer paso en la prueba es arrancar el proceso correspondiente en el servidor, ya que el resto de elementos del sistema inician con sus tareas una vez han establecido comunicación con el mismo. El siguiente paso para la prueba consiste en iniciar el gestor de posición de los usuarios, a través del cual se realiza el registro de la ubicación del usuario dentro del área en la que se encuentra el robot. Esto da como resultado en el servidor el primer mensaje resaltado en la figura 6.20, **Registro de usuario recibido 5 0100**, el cual indica que el usuario con el identificador número 5 se localiza en el área con identificador 0100.

Con la ubicación del usuario registrada dentro del servidor, el siguiente paso es iniciar el proceso controlador del robot, al cual se accede por medio de una conexión *ssh* previamente establecida con la *Raspberry* en el robot. El proceso controlador del robot verifica los componentes que conforman al robot, para posteriormente establecer una comunicación con el servidor. Una vez que se establece esta comunicación, el robot inicia con una búsqueda de marcadores, sin embargo, al haber sido colocado frente al marcador, el robot solo realiza los ajustes necesarios para enviar las imágenes que permiten al servidor determinar su ubicación, la ejecución de este proceso se aprecia en la figura 6.21.

Una vez que el robot termina de enviar las imágenes al servidor, éste determina la ubicación del robot y actualiza los registros correspondientes, esto se aprecia en el segundo mensaje resaltado en la figura 6.20. Registrada la ubicación del robot y el usuario en el

```

1: pi@raspberrypi: ~/Documents/Codigos/robot/compila
network -lqt5core -lptnread
pi@raspberrypi:~/Documents/Codigos/robot/compila $ ./robot
Iniciando síntesis de voz...
[OK]
Conectando con la arduino...
[OK]
Iniciando el robot
Mensaje Reproducido [OK]
Iniciando el robot
Mensaje Reproducido [OK]
Iniciando conexión con el servidor
Conexión con el servidor realizada [OK]
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
Buscando marcador
marcador !
marcador !
Guiando al marcador . . . [32039]

```

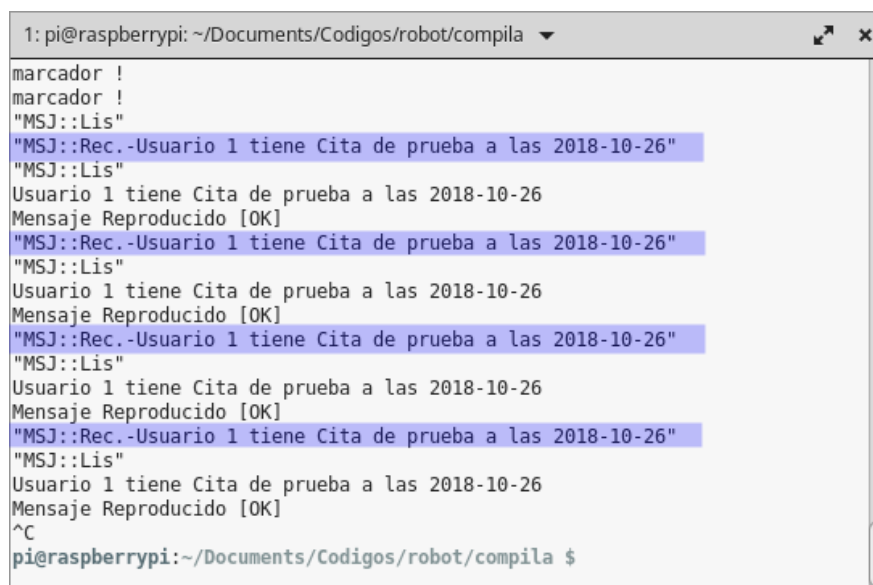
Figura 6.21: Primera etapa de la ejecución del robot en la prueba de mensajes

servidor, se inicia la ejecución del gestor de tareas, el cual, para efectos de la prueba, se programa para enviar una serie de mensajes al usuario previamente registrado. Por esta razón, en la figura 6.20 se aprecia la recepción de una serie de mensajes, en los cuales se resalta que el destinatario corresponde con el usuario registrado dentro del entorno.

Cada vez que el servidor recibe un mensaje del gestor de tareas, verifica la ubicación del destinatario y la posición del robot, y como para esta prueba ambas ubicaciones son iguales, el servidor realiza el envío del mensaje correspondiente al robot, el cual se encarga de entregarlo al usuario.

La figura 6.22 resalta la llegada de mensajes al robot provenientes del servidor, los cuales son impresos en pantalla por el proceso controlador, pero también se entregan al usuario a través de un proceso de síntesis de voz, con lo cual se concluye la prueba de envío de mensajes al robot.

Es importante señalar que el orden de ejecución de los procesos propuesto para esta prueba limita el número de mensajes mostrados en pantalla, lo que permite apreciar de forma ordenada, a través de las figuras presentes en esta sección, la conexión del gestor de posición, el registro de la ubicación del usuario, la activación del robot y su registro dentro de un área, al igual que la generación y envío de mensajes. Sin embargo, la única restricción de ejecución dentro del sistema SiGA sería ejecutar el servidor al inicio, por lo tanto, los demás componentes del sistema pueden ser ejecutados en cualquier orden. De igual forma,



```
1: pi@raspberrypi: ~/Documents/Codigos/robot/compila
marcador !
marcador !
"MSJ::Lis"
"MSJ::Rec.-Usuario 1 tiene Cita de prueba a las 2018-10-26"
"MSJ::Lis"
Usuario 1 tiene Cita de prueba a las 2018-10-26
Mensaje Reproducido [OK]
"MSJ::Rec.-Usuario 1 tiene Cita de prueba a las 2018-10-26"
"MSJ::Lis"
Usuario 1 tiene Cita de prueba a las 2018-10-26
Mensaje Reproducido [OK]
"MSJ::Rec.-Usuario 1 tiene Cita de prueba a las 2018-10-26"
"MSJ::Lis"
Usuario 1 tiene Cita de prueba a las 2018-10-26
Mensaje Reproducido [OK]
"MSJ::Rec.-Usuario 1 tiene Cita de prueba a las 2018-10-26"
"MSJ::Lis"
Usuario 1 tiene Cita de prueba a las 2018-10-26
Mensaje Reproducido [OK]
^C
pi@raspberrypi:~/Documents/Codigos/robot/compila $
```

Figura 6.22: Segunda etapa de la ejecución del robot en la prueba de mensajes

las acciones de generación de mensajes, cambios de ubicación del usuario o la llegada del robot a un área son eventos asíncronos, por lo que el servidor es capaz de almacenar la información entregada por todos los componentes hasta el momento en el que se logre cumplir con todos los requerimientos para la entrega de un mensaje, es decir, que exista un mensaje, y que el destinatario del mensaje y el robot se encuentren en la misma área.

## 6.7. Prueba general del sistema

En la prueba anterior se verifica la capacidad del sistema para comunicar a los diferentes procesos que lo conforman, a fin de llevar un mensaje desde la información proporcionada a través de las interfaces hasta un usuario en el ambiente. Esta funcionalidad es el objetivo del sistema SiGA, pues ejemplifica la capacidad de un sistema ubicuo para aprovechar la información proporcionada por el contexto (ubicación del usuario, fecha, hora) para coordinar las acciones del mismo a fin de apoyar en una tarea.

No obstante, para decir que el sistema realiza todas las tareas planeadas es necesario comprobar que el robot es capaz de desplazarse en el entorno para llevar un mensaje a un usuario, al hacer uso del mecanismo de navegación descrito, al mismo tiempo que aprovecha información agregada en el sistema (mapa). Por esta razón, en la última prueba del funcionamiento del sistema se ejemplifica la forma en la que el sistema SiGA interactúa con el entorno mediante un mecanismo inteligente (robot) a fin de cumplir un objetivo.

Para esta prueba, el mapa utilizado para representar al entorno se compone de seis áreas: cocina [0010], comedor [0011], pasillo 1 [0100], sala [1000], pasillo 2 [1100] y

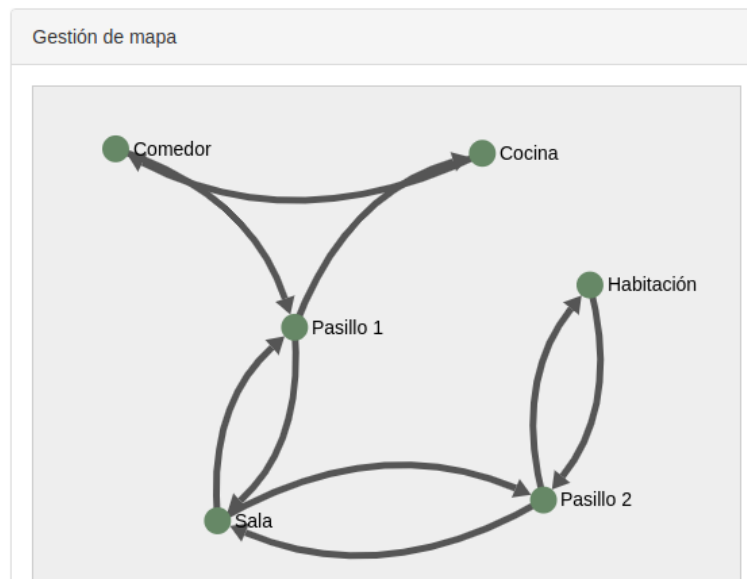


Figura 6.23: Mapa del entorno para prueba integral

habitación [1110]. Estas áreas se relacionan a través de nueve rutas preestablecidas, las cuales corresponden con las observadas en el mapa de la figura 6.23.

Con base en este mapa, el robot fue colocado en el área sala [1000], y el usuario final fue localizado en el área cocina [0010], por consiguiente, el sistema debe identificar las direcciones de giro necesarias para desplazar al robot y entregar el mensaje.

```
[ubicuo@ltjose obj]$ ./Servidor
Iniciando el servidor
Servidor conectado
Nuevo cliente
Cambia la ubicación del usuario
Registro de usuario recibido 5 0010
Nuevo cliente
Robot llegó a destino: 1000
Iniciando en envío de mensajes
Deteniendo en envío de mensajes

--p--
```

Figura 6.24: Estado del servidor al iniciar la prueba integral

La figura 6.24 muestra el estado en el que se encuentra el servidor al iniciar la prueba, en donde se aprecia la llegada del usuario usuario 5 al área 0100. Después de la llegada del usuario se aprecia la conexión del robot al servidor, en dónde se observa que el servidor identifica el área en la que se localiza el robot como 1000, sin embargo, como todavía no ha

recibido ningún mensaje para ser entregado envía al robot la orden `--p--`, la cual indica al robot permanecer en su sitio hasta nuevo aviso.

```
Nuevo cliente
MensajeRecibido: ACT#N;- 5;- 13:26:51;- 2018-10-26;- Usuario 1 tiene Cita de prueba a las 2018-10-26
Iniciando en envío de mensajes
Deteniendo en envío de mensajes
0 0010 "Cocina"
1 0011 "Comedor"
2 0100 "Pasillo 1"
3 1000 "Sala"
4 1100 "Pasillo 2"
5 1110 "Habitación"
n      l      n      n      n      n
n      n      l      n      n      n
r      n      n      l      n      n
n      n      l      n      r      n
n      n      n      r      n      l
n      n      n      n      l      n
0      2      3
Llamar al gestor de mapas

--l--
```

Figura 6.25: Llegada del primer mensaje al servidor

El siguiente evento recibido por el servidor es la llegada de un mensaje para el usuario con id 5. Al llegar este mensaje el servidor identifica que el usuario se encuentra activo dentro del entorno y que el robot se encuentra en un área diferente, esto genera la necesidad de desplazar al robot hacia una nueva área, por lo tanto realiza el análisis del entorno con el gestor de posición el cual lee la información correspondiente almacenada en la base de datos, y mediante la información correspondiente a la posición del usuario y la posición del robot determina que la orden que debe ser enviada al robot es `--l--`, la cual indica que el robot debe buscar un nuevo marcador al cual desplazarse realizando un giro hacia la izquierda, estos cambios realizados en el servidor se aprecian en la figura 6.25.

<pre>Guiando al marcador . . .[42854] marcador ! marcador ! marcador ! marcador ! marcador ! marcador ! "MSJ::Lis" "MSJ::Lis" "Izq" Girando hacia la izq 0 Buscando marcador con la dir I</pre>	<pre>Guiando al marcador . . .[35358] marcador ! marcador ! marcador ! marcador ! marcador ! marcador ! "MSJ::Lis" "Der" Girando hacia la der 0 Buscando marcador con la dir D</pre>
---	--

Figura 6.26: Resultados del robot en la prueba integral

La figura 6.26 muestra el resultado obtenido durante la ejecución del robot, en la imagen del lado izquierdo se aprecia que el robot realiza el envío de imágenes al servidor para

registrar su ubicación. Como ya se expuso, el robot permaneció a la espera de nuevas ordenes hasta que, como se aprecia en la imagen izquierda, recibió la orden *Izq*. Con esta orden comenzó con el proceso de búsqueda y desplazamiento hacia el marcador ubicado en la dirección izquierda. Una vez el robot alcanza ese nuevo marcador, envía las imágenes al servidor para determinar su nueva posición, donde obtiene como respuesta la orden *Der*, la cual se observa en la imagen del lado derecho de la figura 6.26.

```
Robot llegó a destino: 0100
Iniciando en envío de mensajes
Deteniendo en envío de mensajes
QSqlDatabasePrivate::addDatabase: duplicate connection name 'qt_sql_default_connection', old connection removed.
0 0010 "Cocina"
1 0011 "Comedor"
2 0100 "Pasillo 1"
3 1000 "Sala"
4 1100 "Pasillo 2"
5 1110 "Habitación"
n      l      n      n      n      n
n      n      l      n      n      n
r      n      n      l      n      n
n      n      l      n      r      n
n      n      n      r      n      l
n      n      n      n      l      n
0      2
Llamar al gestor de mapas

--r--
```

Figura 6.27: Registro del robot en el área pasillo 1

Esta nueva orden fue generada gracias a que el servidor determinó que el robot había alcanzado el área pasillo 1, la cual se presenta con el identificador *0100*. Una vez se identificó la nueva posición del robot, el gestor de posición actualizó la ruta lo que dio como resultado la orden *--r--* que se observa en la figura 6.27.

```
MensajeRecibido: ACT#N;- 5;- 13:26:51;- 2018-10-26;- Usuario 1 tiene Cita de prueba a las 2018-10-26
Robot llegó a destino: 0010
Iniciando en envío de mensajes
El mensaje se ha enviado
El mensaje se ha enviado
Deteniendo en envío de mensajes

--p--
```

Figura 6.28: Envío de mensajes del servidor al robot

Antes de que el robot alcance el tercer marcador, el gestor de tareas envía los mensajes que deben ser entregados, sin embargo, como el robot se encuentra en camino, los mensajes son almacenados en una cola dentro del servidor. Al registrarse la llegada del robot a la tercer área, el servidor verifica que tanto el robot como el usuario se encuentren en la misma ubicación. Al ser esto correcto, revisa todos los mensajes almacenados para determinar cuales de ellos son dirigidos al usuario alcanzado. Esto da como resultado el envío de los

mensajes almacenados hacia al robot para que éste los reproduzca, tal y como se aprecia en la figura 6.28.

```
Guiando al marcador . . .[46348]
marcador !
marcador !
marcador !
marcador !
marcador !
marcador !
"MSJ::Rec.-Usuario 1 tiene Cita de prueba a las 2018-10-26"
"MSJ::Rec.-Usuario 1 tiene Cita de prueba a las 2018-10-26"
"MSJ::Lis"
Usuario 1 tiene Cita de prueba a las 2018-10-26
Mensaje Reproducido [OK]
Usuario 1 tiene Cita de prueba a las 2018-10-26
Mensaje Reproducido [OK]
```

Figura 6.29: Entrega de mensajes del robot al usuario

Una vez el robot recibe los mensajes, realiza el proceso para sintetizarlos a voz, acción con la que el sistema SiGA crea una interacción con el usuario. Esto se puede observar en la figura 6.29.





## Conclusiones y trabajo futuro

Un sistema de cómputo ubicuo debe tener la capacidad de conocer e interpretar el contexto del entorno en el que se encuentra, lo que le permite orientar sus acciones con el fin de proporcionar una mejor interacción con el usuario. Esto trae consigo un incremento en la información manipulada por el sistema, ya que no solo se enfoca en las tareas objetivo, sino en coordinar sus acciones acorde con el contexto, preferencias y restricciones ajustadas por el usuario.

Por consiguiente, el diseño de sistemas de cómputo ubicuo trae consigo diversos retos como: la interpretación de la información obtenida del entorno, la coordinación de los múltiples elementos que componen al sistema, la adición de las preferencias y restricciones relacionadas a los usuarios, así como la creación de componentes que contemplen los cambios que puede sufrir el entorno. Todo sin perder de vista las tareas para las cuales el sistema fue creado.

Dicho lo anterior, el diseño del sistema SiGA, además de enfocarse en la gestión y entrega de recordatorios para los adultos mayores, contempla mecanismos que le permite extraer información del entorno, como en el caso del subsistema de navegación construido para ubicar y desplazar al robot dentro del entorno, ya que dicho robot representa el método principal de interacción entre el sistema SiGA y los adultos mayores.

La construcción e implementación del robot presentó diferentes retos, pues uno de los principales objetivos del sistema SiGA es mantener el coste económico bajo. Esto trajo como consecuencia que el robot diseñado no cuente con una capacidad de procesamiento alta, ni con capacidades de movimiento muy precisas, por ejemplo la desviación del robot al indicarle movimientos en línea recta, ocasionada por la diferencia en la velocidad de giro de los motores utilizados. Aunado a esto, las modificaciones realizadas en el entorno no debían de representar un coste alto, por lo tanto, se optó por un mecanismo de posicionamiento y navegación basado en el análisis de imágenes.

Al implementar técnicas de visión para la navegación del robot en componentes con bajas capacidades de procesamiento se genera la necesidad de buscar mecanismos de filtrado que no requieran altas capacidades de cómputo, o bien, implementar un enfoque de servicios robóticos en la nube, el cual permite que parte del procesamiento requerido por un robot sea realizado por componentes con mayor capacidad de cómputo. El sistema SiGA separa la navegación del robot en dos etapas: el desplazamiento del robot a un marcador y la creación de rutas. Con esta separación de tareas, el sistema hace uso de los dos enfoques antes mencionados, pues fue necesaria una optimización del proceso de filtrado de imágenes para poder desplazar al robot hacia un marcador, pero además se implementó la estructura de comunicación necesaria entre el robot y el servidor para que este último pudiera apoyar en las tareas relacionadas con la extracción de información de las imágenes, y la identificación de rutas definidas en el mapa.

Para lograr el desplazamiento del robot hacia un marcador, se exploran técnicas de filtrado de objetos con el uso de diferentes esquemas de color, lo que da como resultado un algoritmo de detección de marcadores basado en el modelo de color HSV. También fue necesaria la calibración de movimientos con base en la información extraída de las imágenes, por esta razón, la construcción del robot implementa procesos de comunicación entre los diferentes componentes del robot, los cuales permiten definir acciones de alto nivel como movimientos o giros a partir de ordenes creadas por componentes de bajo nivel, como lo es la tarjeta arduino.

El servidor, además de gestionar la ubicación del robot, debe coordinar las tareas relacionadas con la gestión de recordatorios, por ello, a fin de crear una separación entre las diferentes tareas, se optó por construir múltiples componentes que desarrollaran actividades específicas, los cuales son controlados por medio de un componente central. Entre estos componentes se encuentran las interfaces de captura y programación de actividades, el proceso encargado de crear e informar el momento en que debe ser entregado un mensaje, el proceso encargado de gestionar la navegación del robot, y el proceso que se encarga de determinar la ubicación de los usuarios.

Para lograr una coordinación de todos estos componentes es necesario el uso de algoritmos que permitan sincronizar procesos en sistemas distribuidos, por esta razón, es necesario mantener una constante comunicación con todos los procesos a fin de coordinar las tareas, con ello se logra que el sistema ubicuo mantenga un control constante sobre los componentes y pueda conocer los cambios en el contexto para así determinar el momento adecuado para realizar una actividad.

En conclusión, la creación de un sistema de cómputo ubicuo, de bajo costo, orientado a cumplir un objetivo como lo es el apoyo a adultos mayores para recordar actividades y horarios de medicación, puede presentar limitaciones ocasionadas por el restricciones iniciales del problema, como son las capacidades de procesamiento de los componentes o las limitaciones en movilidad y costos, sin embargo, gracias a los mecanismos de comunicación que hoy en día presentan componentes de hardware para la creación de dispositivos inteligentes, es posible

construir una infraestructura de comunicación de componentes que distribuyan las tareas realizadas por el sistema y hagan factible la implementación de sistemas como el descrito en este documento.

## Trabajo futuro

Durante el diseño, creación y evaluación del sistema SiGA se descubrieron posibilidades que, al ser implementadas, pueden mejorar algunos aspectos del mismo, por lo que a continuación se presentan algunos de los posibles desarrollos que pueden mejorar el funcionamiento del sistema.

- Por medio de las pruebas realizadas al robot se pudo identificar una desviación del mismo al realizar desplazamientos en línea recta, es decir, una tendencia del robot de moverse hacia la derecha, por ello un trabajo a futuro sería explorar otros diseños de robots que presenten mejoras en el desplazamiento.
- Al evaluar el proceso de desplazamiento del robot hacia el marcador, se observó que a mayor cercanía con el marcador, el ajuste del robot para centrar el marcador debería de ser mínimo, por esta razón, uno de los trabajos a futuro puede ser la creación de un mecanismo de ajuste adaptativo, el cual cambie el tiempo de giro en relación con el área del marcador dentro de la imagen.
- El sistema fue configurado para el uso de marcadores de color magenta, sin embargo, gracias a la forma en la que se implementó el filtro de color, es posible crear una interfaz que permita al usuario definir el color de los marcadores utilizados.
- Una parte importante en los sistemas ubicuos es el uso de mecanismos inteligentes que adapten sus acciones respecto al contexto, por lo cual, otro punto de mejora del sistema SiGA es la implementación de un mecanismo que permita la creación de recordatorios inteligentes, el cual aproveche la información relacionada con el desplazamiento y ubicación de los usuarios dentro del entorno.
- Se requiere de la implementación de un sistema de localización de usuarios, ya que para efectos de prueba del sistema, la posición de los usuarios fue simulada a través de interfaces de software.
- Las pruebas realizadas al sistema se desarrollaron en un ambiente controlado con usuarios simulados, por esta razón se requiere una evaluación del mismo en entornos reales.
- A través del uso de dispositivos móviles las interfaces pueden ser mejoradas a fin de presentar múltiples formas de interactuar con el sistema, al mismo tiempo que pueden servir como fuentes de información que apoyen a las tareas realizadas.



# Bibliografía

- [1] *Use Case Modeling*, pages 49–82. Apress, Berkeley, CA, 2007.
- [2] Unai Alegre, Juan Carlos Augusto, and Tony Clark. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, 117(Supplement C):55 – 83, 2016.
- [3] Juan Carlos Augusto, Hideyuki Nakashima, and Hamid Aghajan. *Ambient Intelligence and Smart Environments: A State of the Art*, pages 3–31. Springer US, Boston, MA, 2010.
- [4] M. Bonaccorsi, L. Fiorini, F. Cavallo, R. Esposito, and P. Dario. *Design of Cloud Robotic Services for Senior Citizens to Improve Independent Living and Personal Health Management*, pages 465–475. Springer International Publishing, Cham, 2015.
- [5] Joanna Buscemi, Jeremy Steglitz, and Bonnie Spring. Factors and predictors of cognitive impairment in the elderly. *Translational Behavioral Medicine*, 2(2):126–127, Jun 2012.
- [6] Claude Caron, Daniel Chamberland-Tremblay, Cédric Lapierre, Pierre Hadaya, Stéphane Roche, and Mokhtar Saada. *Indoor Positioning*, pages 1011–1019. Springer International Publishing, Cham, 2017.
- [7] L. J. Cartagena, A. Kang, S. Munnangi, A. Jordan, I. C. Nweze, V. Sasthakar, A. Boutin, and L. D. George Angus. Risk factors associated with in-hospital mortality in elderly patients admitted to a regional trauma center after sustaining a fall. *Aging Clinical and Experimental Research*, 29(3):427–433, Jun 2017.
- [8] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277 – 298, 2009.
- [9] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271, December 1959.
- [10] Alois Ferscha. Pervasive computing in the large: Socio-technical fabric. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '10, pages 5–5, New York, NY, USA, 2010. ACM.
- [11] J. Fiaidhi, W. Chou, and J. Williams. Mobile computing in the context of calm technology. *IT Professional*, 12(3):14–17, May 2010.

- 
- [12] Joseph Frantiska. *Use Case Diagrams*, pages 1–8. Springer International Publishing, Cham, 2018.
- [13] Juan Gómez-Romero, Miguel A. Serrano, Miguel A. Patricio, Jesús García, and José M. Molina. Context-based scene recognition from visual data in smart homes: An information fusion approach. *Personal Ubiquitous Comput.*, 16(7):835–857, October 2012.
- [14] Rachel E. Goshorn, Deborah E. Goshorn, Joshua L. Goshorn, and Lawrence A. Goshorn. *Behavior Modeling for Detection, Identification, Prediction, and Reaction (DIPR) in AI Systems Solutions*, pages 669–700. Springer US, Boston, MA, 2010.
- [15] Marianne Granbom, Susanne Iwarsson, Marianne Kylberg, Cecilia Pettersson, and Björn Slaug. A public health perspective to environmental barriers and accessibility problems for senior citizens living in ordinary housing. *BMC Public Health*, 16(1):772, Aug 2016.
- [16] Alejandro Jaimes. *Data Mining for User Modeling and Personalization in Ubiquitous Spaces*, pages 1015–1038. Springer US, Boston, MA, 2010.
- [17] I. Johnson and P. Ianes. *Frail Elderly Persons and Smart Home Technologies*, pages 119–123. Springer International Publishing, Cham, 2018.
- [18] Erin Jones. Standardized dosing times. In Randy C. Hatton, editor, *Drugs & Therapy Bulletin*, volume 21, page 4. Shands HealthCare’s Publication, Nov. / Dec. 2007.
- [19] John Krumm. *Ubiquitous Computing Fundamentals*. Chapman & Hall/CRC, 1st edition, 2009.
- [20] Dongqing Li, editor. *RGB*, pages 1791–1791. Springer US, Boston, MA, 2008.
- [21] Qing Li and Yu-Liu Chen. *Entity-Relationship Diagram*, pages 125–139. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [22] Claudia Loitsch, Michael Schmidt, and Gerhard Weber. Position paper: Accessible human-robot interaction (ahri). In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA ’15, pages 16:1–16:4, New York, NY, USA, 2015. ACM.
- [23] Eric McAdams, Asta Krupaviciute, Claudine Gehin, Andre Dittmar, Georges Delhomme, Paul Rubel, Jocelyne Fayn, and Jad McLaughlin. *Wearable Electronic Systems: Applications to Medical Diagnostics/Monitoring*, pages 179–203. Springer US, Boston, MA, 2011.
- [24] Gro Næss, Marit Kirkevold, Wenche Hammer, Jørund Straand, and Torgeir Bruun Wyller. Nursing care needs and services utilised by home-dwelling elderly with complex health problems: Observational study. 17, 12 2017.

- 
- [25] A Olivé. *Use Cases*, pages 337–351. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [26] Stefan Poslad. *Ubiquitous computing : smart devices, environments and interactions*. Wiley, Chichester, U.K, 2009.
- [27] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE J. Biomedical and Health Informatics*, 17(3):579–590, 2013.
- [28] D. Saha and A. Mukherjee. Pervasive computing: a paradigm for the 21st century. *Computer*, 36(3):25–31, Mar 2003.
- [29] Enzo Pasquale Scilingo, Antonio Lanatà, and Alessandro Tognetti. *Sensors for Wearable Systems*, pages 3–25. Springer US, Boston, MA, 2011.
- [30] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne. A survey of wearable devices and challenges. *IEEE Communications Surveys Tutorials*, PP(99):1–1, 2017.
- [31] Ojaswa Sharma, Jalaj Pandey, Hammad Akhtar, and Gaurav Rathee. Navigation in ar based on digital replicas. *The Visual Computer*, May 2018.
- [32] Judith Symonds and Mehdi Khosrow-Pour. *Ubiquitous and Pervasive Computing: Concepts, Methodologies, Tools, and Applications*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2009.
- [33] Haifeng Wang and Yi Zhang. Image enhancement algorithm using brightness preserving multiple-interval histogram equalization. In Yuhang Yang and Maode Ma, editors, *Proceedings of the 2nd International Conference on Green Communications and Networks 2012 (GCN 2012): Volume 2*, pages 647–654, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [34] Mark Weiser. The computer for the 21st century. 265:94, 09 1991.
- [35] Moustafa Youssef. *Indoor Localization*, pages 1004–1010. Springer International Publishing, Cham, 2017.
- [36] Ö. Yürür, C. H. Liu, Z. Sheng, V. C. M. Leung, W. Moreno, and K. K. Leung. Context-awareness for mobile sensing: A survey and future directions. *IEEE Communications Surveys Tutorials*, 18(1):68–93, Firstquarter 2016.
- [37] Nivio Ziviani. *Diseño de algoritmos con implementaciones en Pascal y C*, page 279. Thomson-Paraninfo, Madrid, 2007.