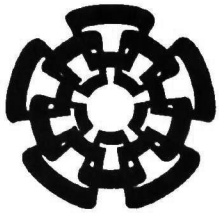
CT-682-551
DOW-7012

XX(199681.1)



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Modelo de Memoria Semántica para Criaturas Virtuales Basado en Neurociencias

Tesis que presenta:
Ana Karina Jaime Oliver

para obtener el grado de:
Maestro en Ciencias

en la especialidad de:
Ingeniería Eléctrica

Director de Tesis
Dr. Félix Francisco Ramos Corchado

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, Agosto de 2011

CT00586
CT-682-SS1
27-08-2012
Jan-2012
$

ID: 199522-2001

Modelo de Memoria Semántica para Criaturas Virtuales Basado en Neurociencias

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Ana Karina Jaime Oliver
Ingeniero en Computación
Universidad de Guadalajara 2004-2008

Becario de CONACYT, expediente no. 332407

Director de Tesis
Dr. Félix Francisco Ramos Corchado

CINVESTAV del IPN Unidad Guadalajara, Agosto de 2011

Resumen

Dotar a las criaturas virtuales de la habilidad de tener un comportamiento humano nos da la certeza obtener resultados útiles cuando las sometamos a alguna clase de estímulo.

Una parte fundamental del comportamiento humano esta dada por la memoria. La memoria nos ayuda a analizar nuestro ambiente y reaccionar en base al conocimiento que hemos adquirido. Un tipo de memoria es la memoria semántica, esta almacena las características de los objetos que se encuentran en el ambiente así como la relación que estos tienen entre si.

Existen varios grupos de investigación que han tratado de resolver este problema, la mayoría ha utilizado la psicología como fundamento para proponer arquitecturas cognitivas. La psicología se basa en observaciones, no tiene un consenso en base a sus resultados.

Por otro lado, la neurociencia analiza el funcionamiento del cerebro. Cuando se realiza alguna actividad, observa como reaccionan las diferentes zonas del cerebro. Como resultado concluyen cual es la función de las áreas del cerebro en dichas tareas. Por esa razón la elegimos como fundamento para nuestra propuesta.

Mediante la neurociencia identificamos las áreas del cerebro que intervienen en la memoria semántica, modelamos su comportamiento y lo emulamos.

Se diseño y desarrollo un sistema distribuido para simular el modelo propuesto. El resultado fue un sistema capaz de identificar los objeto que se encontraban en el ambiente, reconociéndolos en base a sus características y comparándolos con otros objetos previamente almacenados.

Abstract

Providing the virtual creatures of the ability to have human behavior certainly gives us achieving useful results when subjected to some kind of stimulus. A fundamental part of human behavior is given by the memory. Memory helps us analyze our environment and react based on the knowledge we have acquired.

A type of memory we have is semantic memory; it stores the characteristics of the objects in the environment and the relationship they have with each other. Several research groups have tried to solve this problem, they have been used the psychology as a basis to propose cognitive architectures. The psychology is based on observations, no consensus on the basis of their results.

On the other hand, the neuroscience analyzes the brain function. When the brain performing an activity, the neuroscientists see how the brain areas react to different stimulus. As a result the neuroscientists conclude the function of each brain areas. For that reason we chose it as the basis for our proposal.

By neuroscience we identified the brain areas involved in semantic memory, we model and emulate their behavior.

We designed and development a distributed system to simulate the proposed model. The result was a system capable of identifying the objects in the environment, recognizing them based on their characteristics and comparison with other previously stored objects.

Agradecimientos

A mi familia, que me apoyo incondicionalmente haciendo mi vida mas fácil para poder concentrarme en mi maestría.

A Gustavo por acompañarme, guiarme, ayudarme y cuidarme.

A Andrea, Everardo, Francisco y Alberto por haberme acompañarme durante estos años y permitirme ser su amiga y compañera de aventuras.

A mi asesor el Doctor Félix Ramos, por guiarme durante este periodo de mi vida.

A CONACYT, por apoyarme económicamente durante estos dos años y hacer posible realizar este trabajo de investigación.

Índice general

Índice de tablas

Índice de figuras

Glosario

A

Aferencia Transmisión aferente [16].

Aferente *Anat. y Biol.* Dicho de una transmisión anatómica: Que transmite sangre, linfa, otras sustancias o un impulso energético desde una parte del organismo a otra que respecto a ella es considerada central [16].

Almacenamiento Es el mantenimiento de la información a través del tiempo [6].

Amígdala Órgano formado por la reunión de numerosos nódulos linfáticos [16].

Arquitectura Cognitiva Una arquitectura cognitiva es una estructura teórica y un conjunto de mecanismos para cognición humana [23], brinda la infraestructura necesaria para un sistema inteligente. Un arquitectura incluye esos aspectos de un agente cognitivo que esta constantemente a través del tiempo y a través de diferentes dominios de aplicación [26].

C

Codificación Es el proceso por el cual la información nueva es registrada o procesada para posteriormente almacenarla [6].

Cognición f. conocimiento (acción y efecto de conocer) [16].

Cornu Ammonis El cornu ammonis es una parte simple de la corteza. Tiene una estructura heterogénea debido a diferentes aspectos de sus neuronas piramidales. El cornu ammonis esta dividido en cuatro campos llamadas CA1, CA2, CA3 y CA4 [15].

Corteza Entorrinal Es la parte inferior del lóbulo piriforme que se extiende al segmento del giro parahipocampal [15].

Corteza Perirrinal La corteza perirrinal área cortical esencial en el lóbulo medio temporal interconectando la formación hipocampal con otras partes del lóbulo limbico y con el temporal lateral y con las cortezas occipitotemporales de asociación [19].

D

Declarativa Que declara o explica de una manera perceptible algo que de suyo no es o no está claro. [16].

E

Eferencia f. *Biol.* Transmisión de sangre, linfa, otras sustancias o un impulso energético, desde una parte del organismo a otra que con respecto a ella es considerada periférica [16].

Estímulo Agente físico, químico, mecánico, etc., que desencadena una reacción funcional en un organismo [16].

F

Formación Hipocampal La formación hipocampal es un grupo de áreas del cerebro que consisten en el giro dentado, el hipocampo, el subículo, presubículo, parasubículo y la corteza entorrinal [3].

I

Identificación Acción y efecto de identificar o identificarse [16].

Identificar Reconocer si una persona o cosa es la misma que se supone o se busca [16].

Interfaz (Del ingl. *interface*, superficie de contacto). f. *Inform.* Conexión física y funcional entre dos aparatos o sistemas independientes [16].

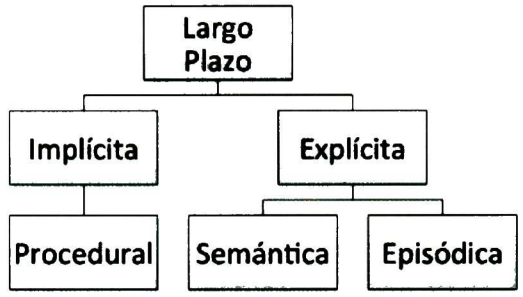
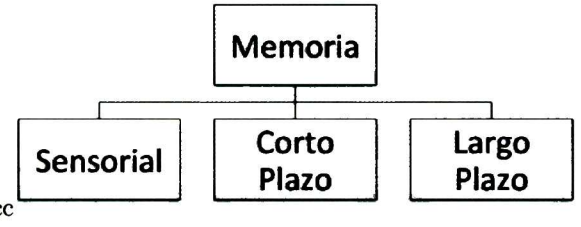


Figura 1: **Categorías de la memoria a largo plazo**



cc

Figura 2: **Jerarquía de memoria propuesta por Atkinson y Shiffrin**

M

Memoria Memoria es el proceso en el que el conocimiento se codifica, almacena y recupera [21]. Atkinson y Shiffrin [5] dividen la memoria en tres categorías: memoria sensorial, memoria a corto plazo y memoria a largo plazo como se ve en la Figura 2.

Memoria A Corto Plazo Recibe información selecta proveniente de la memoria sensorial y de la memoria a largo plazo. La información almacenada decae o se olvida en alrededor de 30 segundos, pero mediante una de las etapas de la memoria llamada *recuperación* puede mantenerse por un periodo mas largo de tiempo, mientras se desee o se necesite la información. El conocimiento almacenado en la memoria a corto plazo ayuda en la generación de la memoria a largo plazo [31].

Memoria A Largo Plazo Es el almacén permanente de las memorias consolidadas. [31] la divide en dos grandes categorías: Implícita y Explícita (Figura 1).

Memoria Asociativa Es aquella memoria que es capaz almacenar un patrón y relacionarlo

con otro previamente almacenado [22].

Memoria Episódica Almacena todos los eventos en los que hemos estado involucrados, estos son escenas (imágenes) que se almacenan de manera secuencial [33].

Memoria Explícita Es aquella memoria que puede ser recuperada (o recordada) y su contenido puede ser descrito con palabras [33]. La memoria explícita se divide en dos categorías: semántica y episódica.

Memoria Semántica Almacena hechos ó significados de los elementos que conforman el ambiente [33].

Memoria Sensorial Se refiere a la forma automática en la que se almacenan las características sensoriales los estímulos entrantes para la subsecuente integración con los estímulos presentados previamente y/o recuperar la información [1]. Almacena la información que los cinco sentidos (visión, audición, gusto, tacto y olfato) obtienen del ambiente durante un periodo corto de tiempo.

Memoria Implícita También conocida como memoria procedural, incluye información obtenida durante el aprendizaje de nuevas habilidades y nuevos hábitos que no pueden ser descritas con palabras, Almacena habilidades y hábitos tales como habilidades motoras o respuestas emocionales [33].

Memoria Visual Mantiene la información general y particular acerca de los objetos en la escena visual [5], [20].

Middleware el término *middleware* se aplica al estrato software que provee una abstracción de programación, así como un enmascaramiento de la heterogeneidad subyacente de las redes, hardware, sistemas operativos y lenguajes de programación [10].

N

Neurociencia La Neurociencia es el estudio del sistema nervioso, avanza en el entendimiento del pensamiento humano, emociones y comportamiento. Los neurocientíficos utilizan herramientas computacionales para examinar moléculas, células nerviosas, redes neuronales, áreas cerebrales y comportamiento. De estos estudios, aprenden como el sistema nervioso se desenvuelve y funciona normalmente, además analizan las causas de los desordenes neuronales [17].

P

Patrón Modelo que sirve de muestra para sacar otra cosa igual [16].

Percepción f. Acción y efecto de percibir [16].

Percibir tr Recibir por uno de los sentidos las imágenes, impresiones o sensaciones externas [16].

Predecir tr. Anunciar por revelación, ciencia o conjetura algo que ha de suceder [16].

Predicción f. Acción y efecto de predecir [16].

R

Reconocer Examinar con cuidado algo o alguien para enterarse de su identidad, naturaleza y circunstancia [16].

Reconocimiento f. Acción y efecto de reconocer o reconocerse [16].

Recuperación f. Acción y efecto de recuperar o recuperarse [16].

Recuperar tr. Volver a tomar o adquirir lo que antes se tenía [16].

S

Subículo Es la principal salida de información del Hipocampo. Esta dividido en cuatro áreas: prosubículo, el mismo subículo, presubículo y parasubículo [15].

T

Tálamo *Anat* Conjunto de núcleos voluminosos, de tejido nervioso, situados a ambos lados de la línea media en los hemisferios cerebrales, por encima del hipotálamo. Se enlaza con casi todas las regiones del encéfalo e intervienen en la regulación de la sensibilidad y de la actividad de los sentidos [16].

Acronimos

A

AVI Áreas Visuales para la Identificación

C

CA1 Cornu Ammonis sección uno

CA2 Cornu Ammonis sección dos

CA3 Cornu Ammonis sección tres

CODOP Código de operación

CPR Corteza Perirrinal

G

GD Giro Dentado

L

LCS Learning Classifier System

M

MW Middleware

N

NGL Núcleo Geniculado Lateral

S

SB Subículo

Capítulo 1

Introducción

1.1. Descripción del problema

En los últimos años ha crecido el interés por diseñar criaturas virtuales cuyo comportamiento sea más humano. Criaturas no autónomas o con comportamientos predefinidos tienen un área de aplicación reducida.

Dotar a las criaturas virtuales de la habilidad de tener un comportamiento humano nos da la certeza obtener resultados útiles cuando las sometamos a alguna clase de estímulo, lo que significa que, si sometemos a la criatura virtual a un estímulo esta debe de reaccionar como lo haría un humano.

Una criatura virtual con este tipo de comportamiento sera útil para diferentes tareas;

- podemos someterla a situaciones de contingencia (incendios, terremotos o inundaciones) para analizar sus respuestas y utilizarlas para realizar planes de para disminuir perdidas humanas
- personajes de vídeo juegos más interesantes
- desarrollar personajes virtuales capaces de representar a un sospechoso de un crimen, un paciente, un comprador potencial entre otros. Estos personajes pueden ayudar en la capacitación de policías, médicos o vendedores.

Para llegar a obtener este comportamiento las criaturas virtuales deben contar las siguientes habilidades:

- percibir su ambiente

analizarlo

- aprenderlo
- hacer planes

tomar decisiones

Las habilidades antes listadas necesitan alguna clase de almacén donde puedan buscar información o guardarla. Como los humanos, las criaturas virtuales deben de contar con un sistema de memoria donde almacenar todo lo que observan y aprenden de su ambiente. La información almacenada es utilizada para diferentes tareas como la generación de planes y la toma de decisiones.

Existes varias arquitecturas cognitivas que han tratado de desarrollar mecanismos que simulen el proceso de memoria de una forma más humana. Estas han utilizado un enfoque psicológico. La psicología les ofrece un listado de reacciones en base a estímulos. Esto es, describe la forma en que actúa el ser humano a una gama de estímulos, las arquitecturas antes mencionadas tratan de simular las acciones realizadas, el problema de este enfoque es que no existe un consenso en la psicología de ningún proceso, lo que tienen que hacer estas arquitecturas es seguir alguna línea de investigación e ignorando el resto.

Ahora existe otra ciencia que estudia el comportamiento humano con una perspectiva diferente. La neurociencia estudia el comportamiento humano ha un nivel celular, utiliza herramientas computacionales para examinar moléculas, nervios, áreas cerebrales [17]. En esta ciencia, a diferencia de la psicología, existe un consenso en los resultados. Lo que significa que el descubrimiento que tengan difícilmente sera refutado, por lo que la neurociencia es una herramienta muy útil en el estudio del comportamiento.

El problema de proponer un modelo de memoria es muy complejo, ya que la memoria consiste de varios procesos que administran diferentes tipos de información. La información almacenada pueden ser descripciones de las características de los elementos que conforman el ambiente (memoria semántica), eventos acontecidos (memoria episódica) ó habilidades motoras y respuestas emocionales adquiridas (memoria procedural) [33]. Cada una de estas clasificaciones implica una tarea de investigación.

La memoria semántica es uno de los primeros retos a superar, ya que esta le proveerá la habilidad a las criaturas virtuales de identificar y aprender los elementos que conforman el ambiente. Con esta habilidad la criatura sera capaz de realizar tareas como evadir obstáculos, utilizar los elementos de su ambiente correctamente, lo que implica que se relacione con ellos.

En este documento se propone un modelo de memoria semántica, basado en los resultados obtenidos por la neurociencia, capaz de reconocer y aprender objetos en base a sus características esenciales como lo haría un ser humano.

1.2. Propuesta

Se desarrollo un modelo de memoria semántica, cada módulo del modelo representa un área cerebral involucrada en el proceso de la memoria semántica según los resultados obtenidos por la neurociencia. La memoria se encuentra distribuida en toda la corteza cerebral [31] por lo que para simular el modelo propuesto de eligió un sistema distribuido, ya que nos permitía ejecutar todos los módulos del modelo en diferentes maquinas trabajando en conjunto como sucede en el cerebro. A continuación se detalla estos dos componentes:

Se propuso un modelo de memoria semántica y su implementación informática basado en los resultados obtenidos por la neurociencia.

- El modelo de memoria semántica consta de un grupo de nodos.
- Cada nodo representa una área cerebral.
- Las áreas cerebrales modeladas están involucradas en el proceso de la memoria.
- Las tres etapas de la memoria (codificación, almacenamiento y recuperación) fueron modeladas.
- El intercambio de información entre los nodos se hizo en base a los conocimientos propuestos por la neurociencia de como funciona la memoria.

- Se desarrollo un sistema distribuido que ayudara en la simulación del modelo.
  - Se implemento un middleware.
  - El middlewre provee la posibilidad de intercambiar información entre nodos que se encuentran distribuidos en la red.
  - La información que intercambian los nodos esta definida por el modelo neurocientifico.

Como ya se menciono el modelo fue probada en un sistema distribuido ,el cual consta de diferentes nodos, distribuidos en la red, que simulan el comportamiento de las áreas cerebrales involucradas en el proceso de la memoria semántica.

La simulación consistió en el paso de mensajes entre estos nodos, los mensaje enviados fueron generados de tal forma que representaran los estímulos que existirían en el cerebro humano cuando se trata de identificar un objeto. Estos mensajes debían ser en el mismo orden y con la misma información que los estímulos del cerebro. Al final de la ejecución se debía mostrar los objetos identificados. Analizando los resultados tenemos que por un lado, el ordenamiento en el tiempo de los mensajes en nuestra .ªrquitectura.^{es} el mismo que en el cerebro; por otro lado el contenido de estos mensajes en nuestra propuesta funcionalmente son los mismos que son transmitidos en el cerebro.

1.3. Organización del documento

La organización del documento es la siguiente:

Capítulo 2 describe los modelos de memoria desarrollados por otras arquitecturas cognitivas.

Capítulo 3 detalla el modelo de memoria propuesto.

- **Capítulo 4** detalla el sistema que se desarrollo para probar el modelo propuesto

Capítulo 5 describe los dos casos de estudio implementados, así como los resultados obtenidos de cada uno de ellos.

Capítulo 6 lista las conclusiones a las que se llego durante el desarrollo de esta investigación, además de mencionar el trabajo futuro.

Capítulo 2

Otras aproximaciones de manejo de memoria

2.1. Introducción

Porque elegirlas: Arquitecturas cognitivas Manejo de memoria Dividen la memoria Enfoque psicológico

A continuación se listan tres arquitecturas cognitivas que proponen modelos de memoria. Fueron elegidas en base a las siguientes características:

Tienen módulos de memoria.

- Dividen la memoria en base al tipo de información.

Tienen un enfoque psicológico.

Son las mas reconocidas y referenciadas.

Estas arquitecturas son SOAR, ACT-R y EPIC. Este documento propone un modelo de memoria que es parte de una arquitectura cognitiva por lo que la comparativa que aquí se realiza es solo con los sistemas de memoria de las arquitecturas antes mencionadas.

2.2. SOAR

SOAR (State Operator and Result) es una arquitectura cognitiva utilizada para construir agentes con inteligencia, utilizados para resolver desde problemas rutinarios hasta complejos

problemas abiertos. La arquitectura utiliza métodos de solución de probemas y es capaz de aprender nuevas tareas [24].

Comenzó como una cama de pruebas para la Teoría de la Cognición de Newell [29], en la cual se denota a la *inteligencia* como "la habilidad de usar todo el conocimiento de uno mismo para alcanzar objetivos" De ahí el fundamento de que toda la memoria debe ser accesible para la solución de un problema. Cabe mencionar que la arquitectura es centralizada y la memoria está prácticamente en un sólo lugar.

2.2.1. Sistema de Memoria

El manejo de la memoria se hace mediante un sistema de producción posterior a OPS-5 [9] que está basado en reglas y utiliza una máquina de inferencia de encadenamiento hacía adelante. Dichas reglas son del tipo SI-ENTONCES, lo cual según la teoría de Newell se denota como "cognición-acción" La resolución de problemas requiere de un procedimiento de inferencia en el que se utilicen el conocimiento almacenado y el actual para producir un comportamiento (acción) que lleve al agente a su meta.

En la Figura 2.1 se puede observar como el sistema de memoria se divide en subsistemas que son utilizados de distinta manera según el problema que se trata de resolver. Primeramente, la división se hace entre las memorias de largo y corto plazo. Las memorias de largo plazo se almacenan como producciones clasificadas en procedurales, semánticas y episódicas. En cambio, las memorias de corto plazo se almacenan en la memoria de trabajo, la cual representa el estado actual del agente [12].

En general, la memoria de trabajo se construye con las entradas de los sensores del agente y se hace un "match" con las condiciones en la memoria de largo plazo para generar una acción, llevada a cabo a través de actuadores. Si las condiciones en el "match" empatan completamente, las acciones se agregan a la memoria de trabajo.

El sistema trata de emular las etapas conocidas de la memoria: codificación, almacenamiento y recuperación, para las cuales se definen distintas dimensiones [11]. A continuación se explica el funcionamiento de cada etapa y sus dimensiones.

Codificación

La arquitectura determina cuando se requieren nuevas condiciones (cuando se ha llegado a un punto muerto). Hecho esto, determina que se va a aprender (conocimiento temporal que ayudará a evitar el punto muerto). Y finalmente, determina cuando se debe adquirir el conocimiento. A éste fenómeno se le denota como "chunking" Cada "chunk" es una regla (o conjunto de reglas) que se añade a la memoria de largo plazo [25].

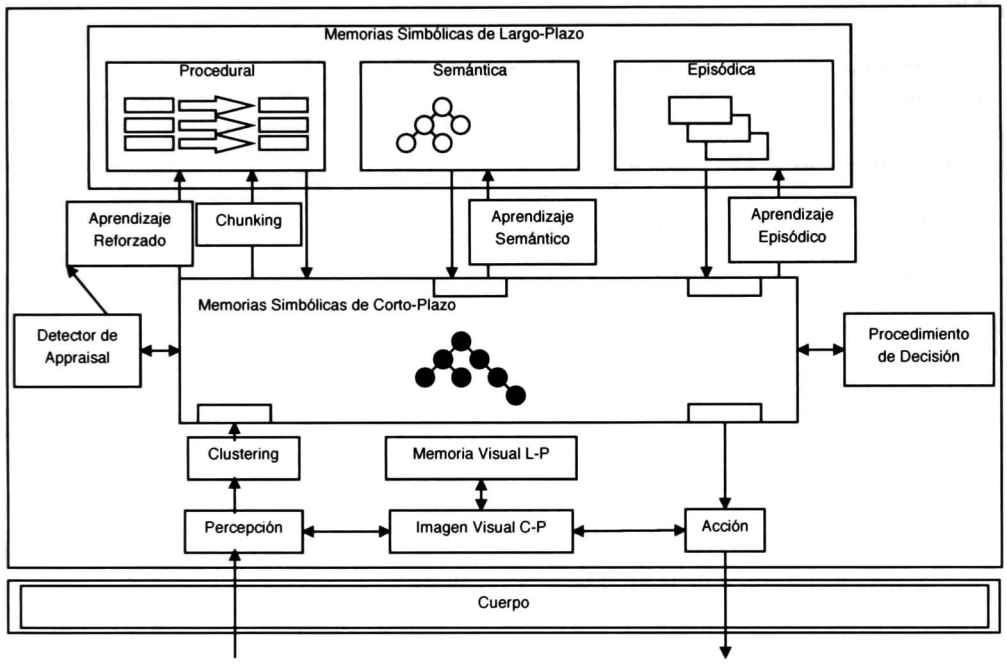


Figura 2.1: **SOAR** Sistema de memoria de la arquitectura SOAR.

Iniciación Conjunta las condiciones eventuales que disparan los procesos de codificación y almacenamiento.

Determinación Cuando se inicia, el sistema selecciona los atributos que componen el conocimiento que será almacenado.

Integración El sistema decide como se añadirá el nuevo conocimiento.

Almacenamiento

Se produce cuando se han codificado las estructuras contenidas en la memoria de trabajo, las cuales se almacenan como nuevas producciones procedurales, semánticas y/o episódicas.

Granularidad El conocimiento almacenado varía en cuanto a su accesibilidad y modificación. Tiene rangos desde diminuto (a nivel simbólico), moderado (un episodio), hasta grueso (todo el conocimiento).

Dinamismo El conocimiento puede cambiar en el tiempo, por ejemplo para sesgar la recuperación o el olvidó.

Recuperación

Se produce cuando las condiciones empatan con la memoria de trabajo. Se generan las acciones que posiblemente lleven a la resolución del problema.

Accesibilidad El conocimiento codificado puede variar en el grado al cual se expone a otros mecanismos de la arquitectura cognitiva.

Iniciación Como en la codificación, se conjuntan las condiciones que disparan el proceso.

Determinación de pistas Cuando se inicia, el sistema une el estado actual, el conocimiento, el contexto, y/o meta-data para seleccionar o crear la pista de recuperación.

- **Selección** Cuando se da una pista, el sistema aplica una política para determinar como el conocimiento se empata con la pista. Ésto puede estar restringido por el tiempo, cálculo, y/o el número de resultados requeridos.

Resultado Al seleccionar, el sistema puede representar el conocimiento arbitrariamente.

2.2.2. Ventajas

Entre las ventajas del sistema de memoria de SOAR se pueden observar las siguientes:

- El uso de diferentes tipos de memoria (semántica, episódica y procedural).

  El fenómeno de "chunking" permite construir nuevas condiciones que ayudan a la resolución del problema en cuestión.

  Se evita a toda costa el bloqueo de la solución en puntos muertos.

  La división en las conocidas etapas de la memoria brinda un marco de trabajo que puede ser utilizado para la investigación de los procesos de memoria en el ser humano.

- La resolución de una amplia gama de problemas.

  - Entre los ejemplos, cabe mencionar uno en el cual distintos agentes (que representan tanques de guerra) tienen la meta de obtener la mayor cantidad de puntos que se obtienen al destruir a los demás. Los agentes son capaces de: utilizar su radar para determinar la existencia de enemigos, armas, energia, obstaculos y teletransportadores; escuchar los sonidos del ambiente para ampliar la línea de vista del radar; moverse en el ambiente; y disparar a los enemigos. En la Figura 2.2 se puede observar uno de los pasos de la simulación.

2.2.3. Desventajas

Se encontraron distintas desventajas en el sistema de memoria, a saber:

- La memoria de los seres humanos es distribuida, en el caso de SOAR la memoria es centralizada.

  Es necesario todo el conocimiento para resolver un problema.

  SOAR tiene su fundamento , por lo que se basa en conocimientos adquiridos por observaciones los cuales no explican las bases al origen de dicho comportamiento.

- Debido a que la memoria en SOAR se base en la teoría de Newell, esta se aleja de la realidad por que la memoria no se encuentra en un solo lugar sino distribuida en la corteza cerebral [33], el humano no utiliza toda la memoria para resolver problemas solo la información mas reciente o relevante [7].

Figura 2.2: **Un ejemplo de aplicación de la arquitectura SOAR** Los agentes (tanques de colores) tienen la meta de obtener la mayor cantidad de puntos destruyendo a los demás agentes.

2.3. ACT-R

ACT-R [18] ,[4] es una arquitectura cognitiva que se utiliza de una manera similar a un lenguaje de programación. Está diseñada en base al modelado de suposiciones acerca de la cognición humana, las cuales se han obtenido mediante resultados derivados de diversos experimentos psicológicos.

Es posible la creación de nuevos modelos, lo cual permite la agregación de nuevas suposiciones acerca de tareas particulares. Los modelos son escritos en el lenguaje de ACT-R, se ejecutan y se obtienen resultados que, al contrario de distintas arquitecturas conocidas, pueden ser comparados directamente con aquellos que se obtienen en experimentos sobre seres humanos reales. Dichos resultados son las medidas cuantitativas que se usan en la psicología cognitiva: tiempo en el que se realiza la tarea, precisión, y datos neurológicos parecidos a los obtenidos de las fMRI (functional Magnetic Resonance Imaging).

La arquitectura cognitiva ha sido utilizada para crear modelos en distintos dominios de la psicología cognitiva: aprendizaje y memoria, solución de problemas y toma de decisiones, comunicación y lenguaje, atención y percepción, desarrollo cognitivo, y diferencias entre individuos. Por otro lado, algunas aplicaciones fuera de la psicología son: interacción hombre-máquina, educación (sistemas de tutoréo cognitivo), agentes para ambientes de entrenamiento, e interpretación de datos fMRI.

Como se puede ver en la Figura 2.3, la arquitectura se divide en tres componentes principales: módulos, buffers, y un "matcher" de patrones.

Los módulos a su vez se dividen en:motores-perceptuales, que son los responsables de interactuar con el ambiente; y de memoria, declarativa y procedural.

Los buffers son las interfaces entre ACT-R y los módulos, existe un buffer por cada módulo.

El "matcher" de patrones busca producciones que empaten con el estado actual de los buffers. Las producciones ejecutadas pueden modificar los buffers y por ende cambiar el estado del sistema.

2.3.1. Sistema de Memoria

Como se dijo arriba, el sistema de memoria de ACT-R se divide en dos: la memoria declarativa, que consiste de hechos; y la memoria procedural, constituída por producciones. Estos módulos constituyen el núcleo cognitivo de la arquitectura.

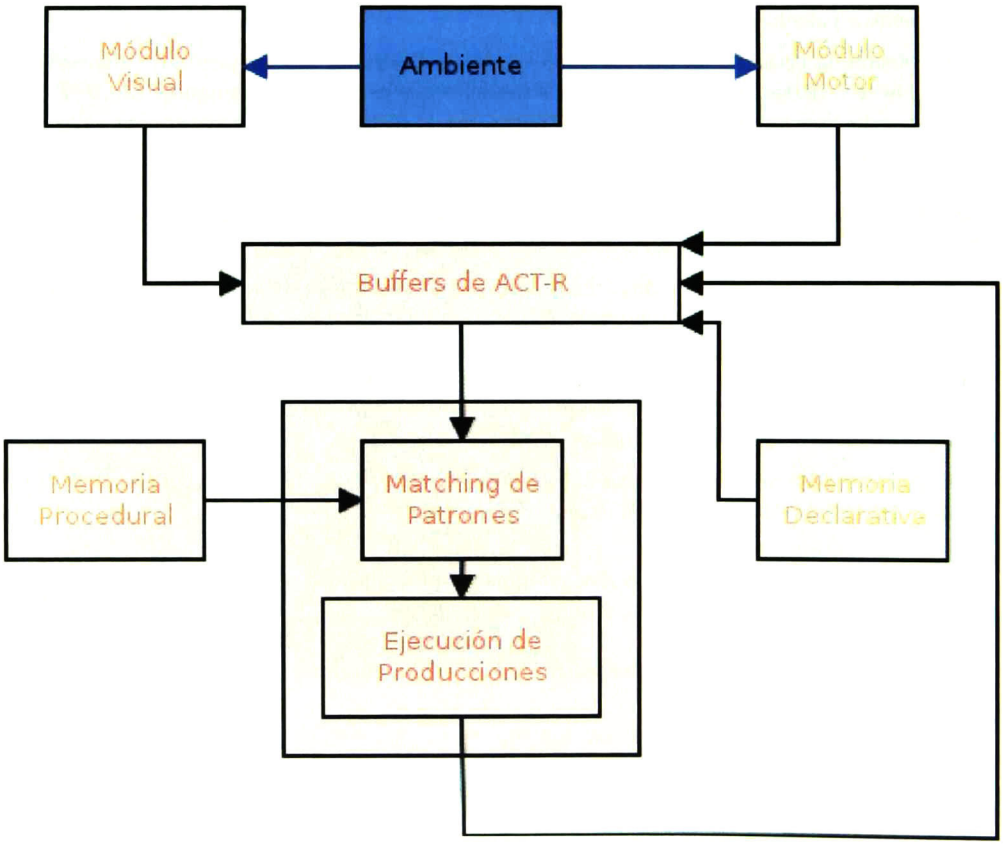


Figura 2.3: **ACT-R** Estructura de ACT-R.

El módulo perceptual mayormente desarrollado es el de visión, mediante ellos se obtienen los estímulos del ambiente que son codificados en "chunks" unidades declarativas de conocimiento. Los códigos son transportados a través de los buffers dedicados hasta el "matcher" de patrones, al cual le llegan (también, a través de los buffers dedicados) chunks provenientes del módulo de memoria declarativa, y conjuntos de producciones desde la memoria procedural. Finalmente, se aplican las producciones a los chunks y se genera un comportamiento, que eventualmente resolverá la meta actual del sistema, mediante los módulos motores, principalmente, manuales.

Memoria Declarativa

El módulo consiste de chunks, y conjuntos de chunks. Ésta información denota hechos conocidos por la arquitectura, tales como sumas, hechos históricos, valores, etcétera. El comportamiento del módulo de memoria es controlado por un conjunto de ecuaciones y parámetros, por ejemplo, la activación de un chunk i (A_i) se define, en base a teorías de activación, como:

$$A_i = B_i + \sum_j W_j S_{ji},$$

donde B_i es la activación base del chunk i, W_j los pesos atencionales de los elementos que constituyen la meta actual, y los S_{ji} las fuerzas de asociación entre los elementos j y el chunk i. La activación de un chunk controla la probabilidad y a su vez la velocidad de recuperación. La Figura 2.4 muestra la codificación para la suma $8 + 4 = 12$.. En la aplicación, $W_j = 1/n$ y $S_{ji} = S - \ln(fan_j)$, donde n es el total de fuentes de activación, fan_j el total de hechos asociados al término j, y S una constante dependiente al modelo. La activación de base se comporta con la práctica (crecimiento) y el retraso (caída) de acuerdo a:

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right)$$

donde t_j es el tiempo desde que la $j-$ésima práctica de un elemento.

Los chunks son recuperados sólo si su activación es mayor a un umbral. La probabilidad de que esto suceda, cuando el umbral es τ dado:

$$P_i = \frac{1}{1 + e^{(A_i - \tau)/s}},$$

donde s controla el ruido en los niveles de activación. Si un chunk será recuperado, la velocidad de recuperación esta dada por:

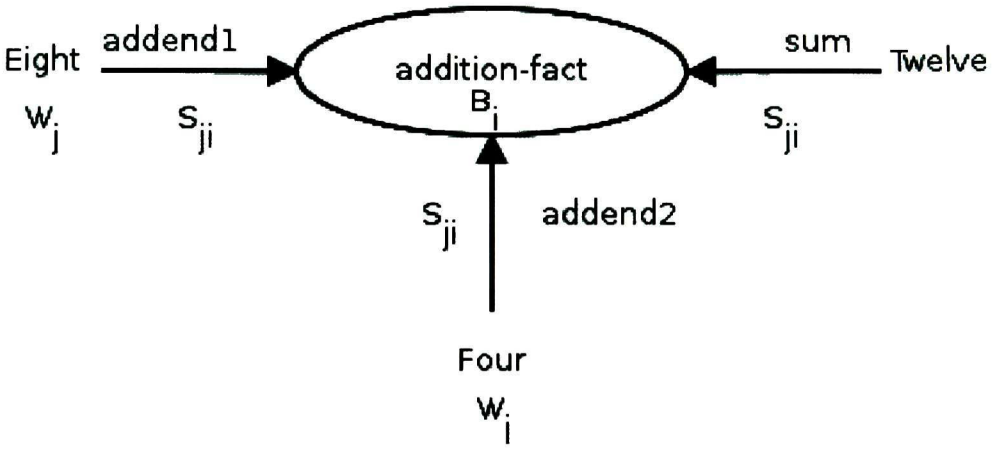


Figura 2.4: Codificación para la suma $8 + 4 = 12$.

$$T_i = Fe^{A_i}.$$

donde F es el factor de latencia, comúnmente denotado como $F \approx 0{,}35e^{\tau}$

La teoría anterior es utilizada en modelos de recuperación en la memoria declarativa. Se supone que dicha teoría toma en cuenta naturalmente una distinción entre memoria explicita e implícita. Las memorias explicitas son los chunks declarativos específicos que pueden ser recuperados e inspeccionados. La memoria implícita se observa en los procesos de activación que controlan la disponibilidad de las memorias explicitas.

Memoria Procedural

El sistema de producción, alimentado por el módulo de memoria procedural, puede detectar patrones, que aparecen en los buffers, y decidir que hacer con ellos para alcanzar un comportamiento coherente. La idea principal es que se apliquen reglas en cualquier momento de la simulación, pero la ejecución de las mismas es serial, por ello se selecciona una, la que tiene mayor utilidad. Las utilidades de las reglas de producción son ruidosas y varian continuamente al igual que las activaciones declarativas, y tienen un rol en la selección así como las activaciones lo tienen hacía la selección de chunks. La utilidad de una producción i se define como:

$$U = P_i G - C_i,$$

Cuando una producción nueva Z fue compuesta por 'X y Y cualquiera de las tres es aplicable. Como se explicó, la selección de alguna de ellas está determinada por las utilidades. En el caso de Z sus probabilidades iniciales son asignadas mediante los "priors" Bayesianos.

2.3.2. Ventajas

Entre las ventajas que se pueden observar en la arquitectura se listan las siguientes:

Los chunks permiten la generación de modelos con gran cantidad de conocimiento declarativo.

La conceptualización de la memoria declarativa implícita y explícita como se explicó arriba está basada en el modelo de memoria más aceptado actualmente.

Las salidas del sistema son medidas comparables directamente con aquellas obtenidas en experimentos de psicología cognitiva.

2.3.3. Desventajas

Las desventajas de ACT-R que se observan son las siguientes:

No se encontró un módulo de aprendizaje de chunks, parece que los modelos deben tener toda la memoria necesaria para resolver una meta.

- Tener todo el conocimiento en los modelos supone un gran trabajo en la aplicación de producciones a pesar de la teoría de la memoria declarativa.

Al igual que SOAR, ACT-R tiene fundamentos psicológicos, por lo que se basa en conocimientos adquiridos por observaciones los cuales no explican las bases al origen de dicho comportamiento.

2.3.4. Conclusión o Discusión

ACT-R es otra arquitectura cognitiva que tiene fundamento en las teorías de Newell acerca de la cognición humana. Su utilidad se ha probado en distintas áreas de aplicación, por ejemplo, en educación con el Tutor Cognitivo de Matemáticas [30] que, en general, "adivina" las dificultades que los estudiantes tienen sobre distintos temas del álgebra, y con ello provee ayuda enfocada. También, se han probado algunos ejemplos para observar el funcionamiento de la arquitectura.

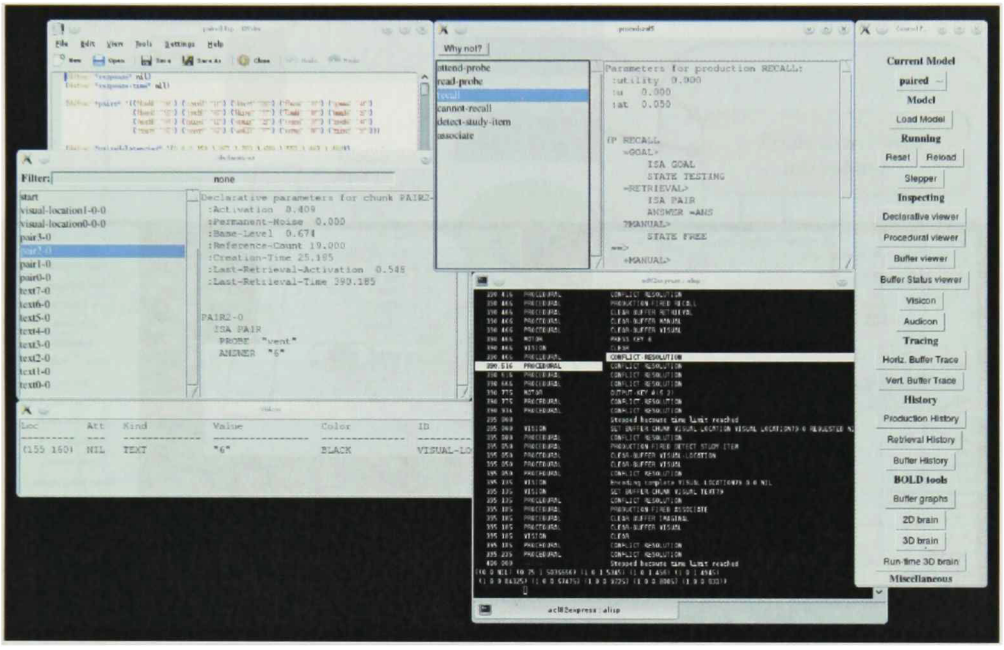


Figura 2.5: **Ejemplo de ACT-R** Las distintas ventanas presentan los módulos utilizados en el experimento: La lista de pares se da como base de conocimiento; se generan los chunks en el módulo declarativo; se presentan las producciones en el módulo procedural; y se describe el objeto de entrada en el módulo perceptual de visión. En la terminal se describen las acciones hechas por ACT-R para la recuperación de los pares.

Entre los ejemplos que probaron se presentan dos aquí, que muestran la tarea de pares asociados. La Figura 2.5 es la simulación del experimento de pares sin aprendizaje, y la Figura 2.6 es sobre el mismo experimento con aprendizaje.

2.4. EPIC

EPIC (Executive Process-Interactive Control) es una arquitectura cognitiva utilizada para modelar comportamiento multimodal y desempeño de tareas múltiples. EPIC fue desarrollado modelar la cognición humana. La arquitectura EPIC esta conformado por procesadores perceptuales, procesadores cognitivos y procesadores motores circundado un procesador de producción de reglas y es utilizado para construir modelos computacionales precisos para una

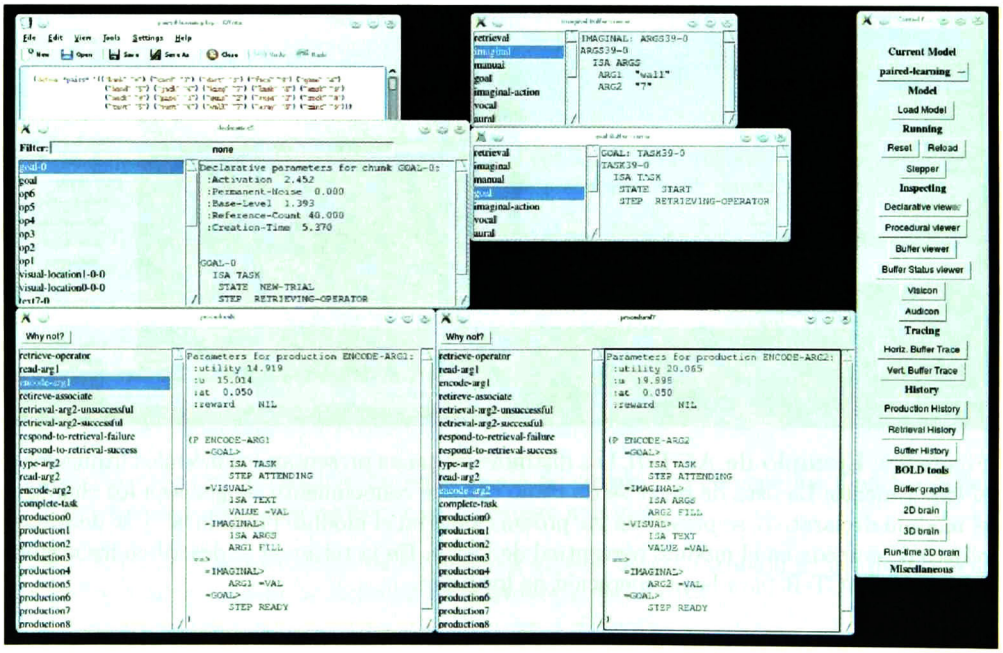


Figura 2.6: **Módulos usados en el experimento con aprendizaje** La base de conocimiento contiene los pares asociados; la tarea tiene dos metas, la recuperación y la codificación; el buffer imaginal almacena el estímulo actual sobre el que hace la codificación; y el módulo procedural tiene las producciones codificadas como se explica en el ejemplo del texto.

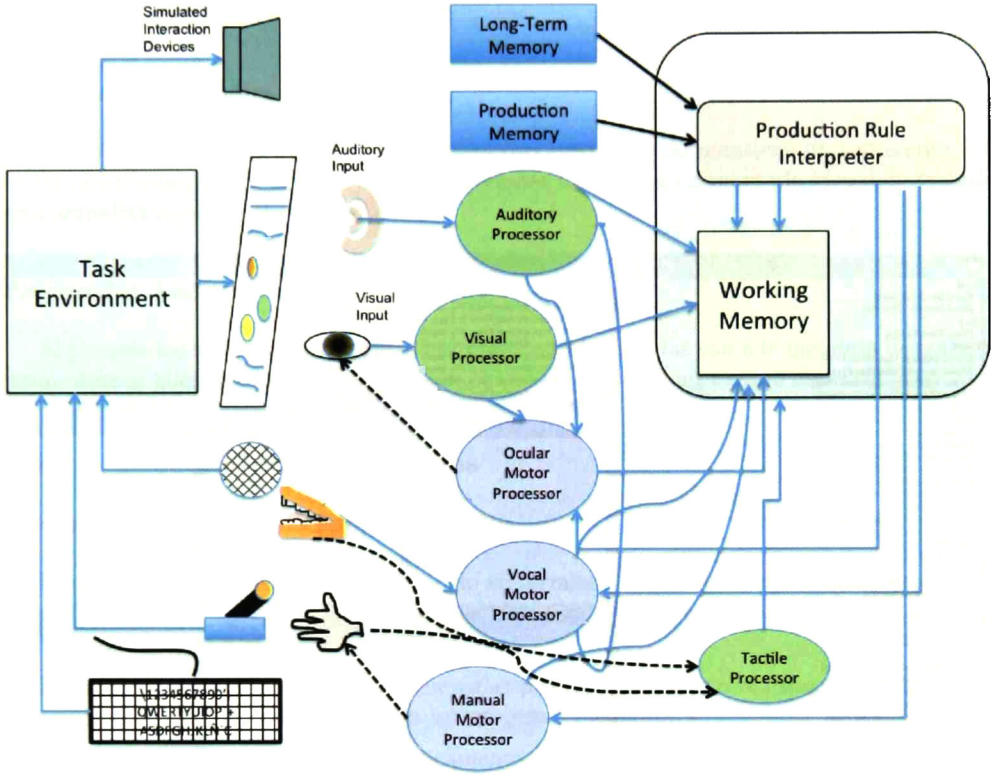


Figura 2.7: **EPIC** Estructura de la arquitectura EPIC

variedad de situaciones en las que interactúan el humano y la computadora. [23].

La 2.7 muestra un ambiente de tarea simulado y un humano simulado. El modulo *de tareas del ambiente* asigna una ubicación física con los objetos interfase y genera simulaciones visuales de eventos y sonidos que la computadora u otra entidad en el ambiente producen en respuesta a la simulación del comportamiento humano.

2.4.1. Procesadores Perceptuales

Un estímulo simple que entra a los procesadores perceptuales produce múltiples salidas, depositadas en la memoria de trabajo. Los tipos de procesadores con los que cuenta EPIC

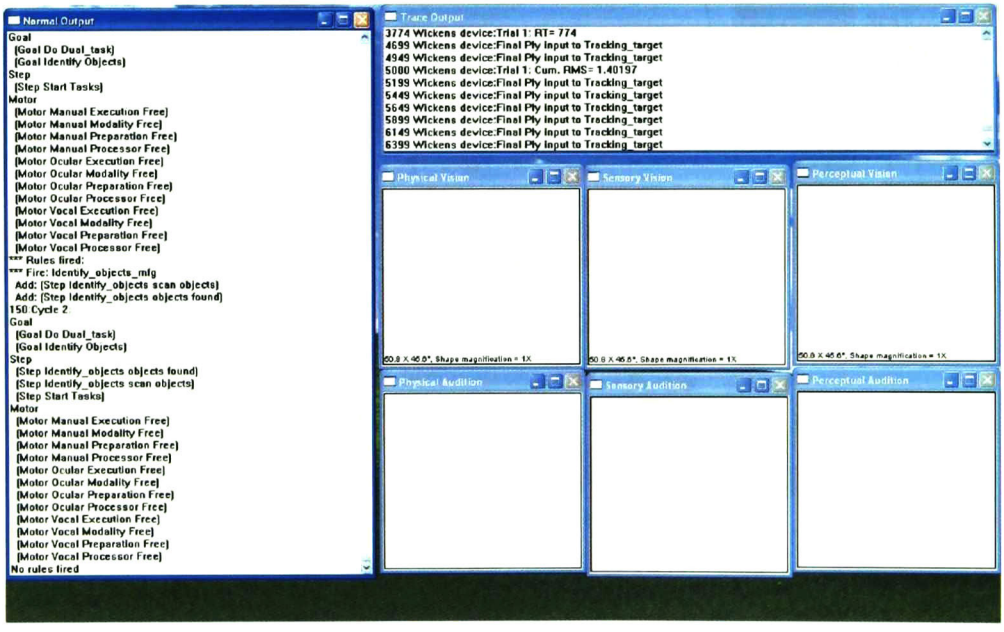


Figura 2.8: **Ejemplo de EPIC** Se puede observar la manera de trabajo de la arquitectura EPIC.

son procesadores visuales procesadores auditivos.

Procesadores Visuales

En EPIC la memoria visual de trabajo, el procesador visual mantiene un representación de los objetos que están visibles y que propiedades tienen. La La memoria visual de trabajo esta actualizándose constantemente.

Procesador Auditivo

El procesador auditivo acepta entradas auditivas y las salidas van a la memoria de trabaja como eventos auditivos. Estos desaparecen cuando transcurre un tiempo determinado.

2.4.2. Procesadores Cognitivos

Producción de reglas y Tiempo del ciclo

El procesador cognitivo esta programado en términos de producción de reglas, por lo que el modelo de EPIC para resolver una tarea debe incluir un conjunto de reglas y acciones relacionadas a estas reglas.

Las acciones asociadas al conjunto de reglas pueden actualizar o remover elementos de la memoria de trabajo y mandar tareas a los procesadores motores.

El procesador cognitivo trabaja ciclicamente, al comenzar un ciclo actualiza la memoria de trabajo.

Paralelismo cognitivo

A diferencia de otras arquitecturas cognitivas como SOAR [23], el procesador cognitivo de EPIC puede realizar varias acciones simultáneamente cuando varias reglas son disparadas. La realización de estas tareas en paralelo depende de los recursos que se necesite para resolverlas, por ejemplo, el humano no puede manejar dos automóviles al mismo tiempo porque no puede estar en dos lugares al mismo tiempo.

Memoria de Trabajo

El sistema de producción de memoria de trabajo en EPIC esta dividido en varias memorias: las memorias visual, auditiva y táctil contiene la información actual proveniente de estos

procesadores. La memoria motora de trabajo almacena el estado actual de los procesadores motores.

Existen dos tipos mas de memoria de trabajo:

1. *Control de almacenamiento* Contiene elementos que representan los objetivos actuales y los pasos para llegar a conseguir los objetivos.
2. *Memoria de trabajo general* Almacena el resto de la información.

2.4.3. Procesadores Motores

Estos procesadores son los encargados de mover los las manos, los ojos y la boca con respecto a los objetivos generados.

2.4.4. Ejemplo de EPIC

En la Figura 2.8 podemos observar como EPIC utiliza los tres tipos de procesadores con los que trabaja.

En la ventana *Normal Output* podemos ver lo que esta almacenado en la memoria de trabajo; objetivos, paso necesarios para lograr los objetivos, reglas disparadas o utilizadas, estado de los motores, numero de ciclo, actualizaciones de la memoria.

En la ventana *Trace Output* se observa el registro de las acciones realizadas.

En el resto de las ventanas se observa lo que la información que el sistema esta obteniendo del ambiente.

2.4.5. Ventajas

Entre las ventajas del sistema de memoria de EPIC se pueden observar las siguientes:

El uso de dos tipos de memoria: memoria de trabajo y memoria a largo plazo.

Tener la capacidad de procesar información provenientes de diferentes sentidos: auditivo, visual.

Utilizar la memoria de trabajo para almacenar planes y objetivos.

Actualiza la memoria de trabajo constantemente.

SOAR, EPIC y ACT-R
La mayor debilidad de estas aproximaciones es el origen de sus modelos. Las tres tienen un fundamento psicológico lo que las aleja de la forma en que lo hace el cerebro humano.
EPIC, SOAR y ACT-R cuando reciben un estímulo tratan de asignarle un patrón, si no lo tiene almacenado utilizan herramientas evolutivas combinando los patrones que ya tienen para asignarle un nuevo patrón.
EPIC utiliza información proveniente de diferentes sentidos que almacena en su memoria y utiliza para conseguir objetivos (Visual y Auditivo).
Representan el conocimiento en base a reglas de inferencia.
Tienen memoria de trabajo y memoria a largo plazo.
Almacenan la información de forma centralizada.
Como SOAR, ACT-R se constituye de reglas de producción y "chunks".

Tabla 2.1: **Resumen de las arquitecturas**

2.4.6. Desventajas

Se encontraron distintas desventajas en el sistema de memoria:

- EPIC tiene un fundamento psicológico, por lo que se basa en conocimientos adquiridos por observaciones los cuales no explican las bases al origen de dicho comportamiento.
- Para ser utilizado, el ambiente debe ser modelado con reglas-acciones.

2.5. Conclusiones

En la Tabla 2.1 se hace un resumen de las arquitecturas aquí descritas usando características que serán usadas para compararlas con nuestra propuesta. Sin embargo ya que en este punto podemos decir que la principal ventaja que buscamos es proponer modelo de memoria que este sustentado en el conocimiento generado por la neurociencia y que se adapte a la evolución de esta ciencia.

Capítulo 3

Modelo de Memoria

3.1. Introducción

El objetivo de esta propuesta es generar un modelo de memoria, capaz de brindarle a las criaturas virtuales la capacidad de identificar objetos en su ambiente. Para poder alcanzar este objetivo necesitamos desarrollar la memoria semántica la cual nos ayuda a identificar objetos en base a sus atributos y la relación que tienen con el ambiente. En este capítulo se describe el fundamento neurocientifico y el diseño del modelo. En el capítulo se define la propuesta desde un punto de vista neurocientifico, lista cada una de las áreas cerebrales involucradas en la generación de memorias para el reconocimiento de objetos, detalla la función de cada una de ellas, señala la información que reciben de otras áreas cerebrales (*aferencias*), muestra la información que envían a otras áreas cerebrales (*eferencias*) y describe como estas áreas cerebrales en conjunto logran generar las tres etapas de la memoria (codificación, almacenamiento y recuperación) para la identificación de objetos.

3.2. Descripción Neurocientifica

El modelo propuesto está basado en los resultados que se han obtenido por la neurociencia a través de los años, estos nos indican que la memoria se encuentra distribuida en toda la corteza cerebral [32], por lo que el proceso de memoria involucra el trabajo paralelo y/o concurrente de diferentes áreas cerebrales. A continuación se listan las áreas cerebrales que intervienen en la generación de memorias para la identificación de objetos (Figura 3.1).

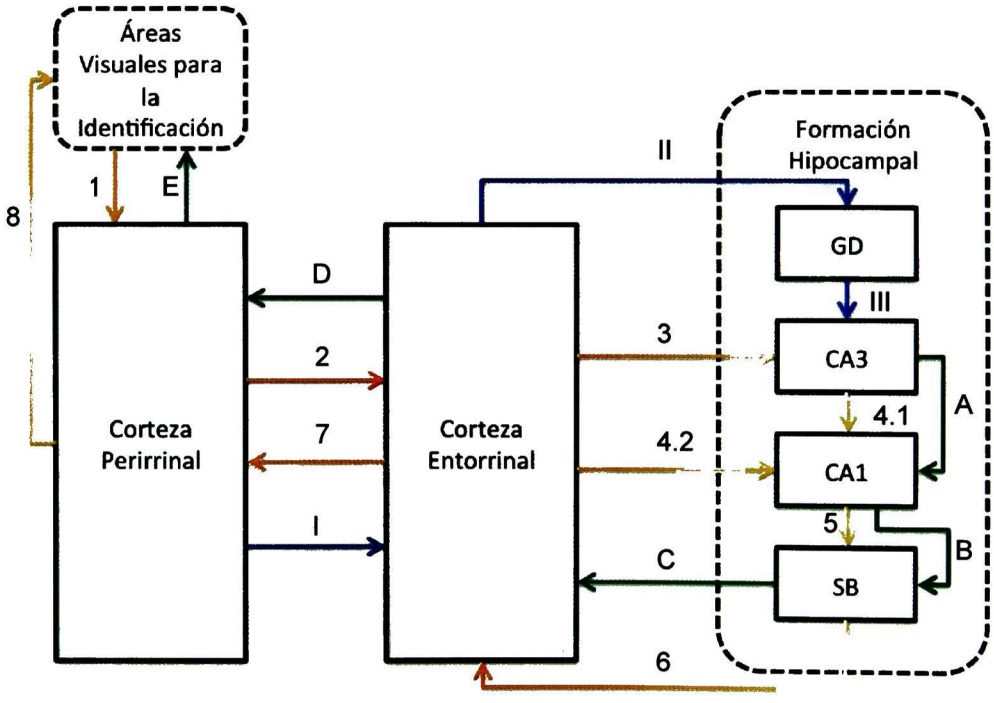


Figura 3.1: **Flujo para la generación de memoria** El camino **rojo** (números del 1 al 8) representa el flujo de información necesario para la *RECUPERACIÓN* de memorias, el camino **verde** (letras de la A a la E) representan el flujo de información necesaria para el *ALMACENAMIENTO* de memorias y el camino **azul** (numero del I al III) representa el flujo de información necesario para la *CODIFICACIÓN* de la información.

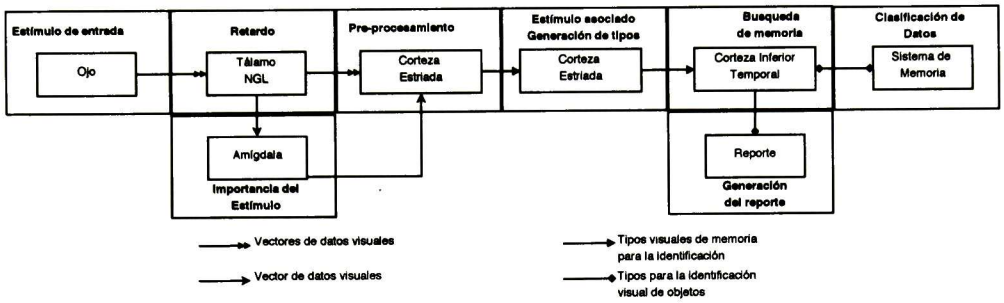


Figura 3.2: **Arquitectura general para la identificación de objetos** Este diagrama muestra los módulos involucrados, funciones y el flujo de la información para la tarea de identificación de objetos.

3.2.1. Áreas

Áreas Visuales para la Identificación

La identificación visual de objetos es un proceso importante para la generación de comportamiento. Se ha diseñado e implementado con anterioridad una arquitectura que realiza este tarea [35].

El modelo está compuesto de distintas áreas cerebrales que constituyen la parte ventral del sistema visual para la identificación. Este modelo (Figura 3.2) se explica a continuación:

El modelo esta dividido en los módulos siguientes:

el ojo es donde la identificación comienza, captura los datos visuales del ambiente, los datos que recibe del ambiente son imágenes que contienen figuras en blanco y negro como se muestra en la Figura 3.3, la matriz entrante es dividida en ventanas para su procesamiento;

núcleo geniculado lateral (NGL) es el siguiente paso para el procesamiento visual, donde se realiza un relevo y filtrado de los datos a través de dos grandes rutas, una de ellas, P-patway, determina los datos visuales utilizados para la identificación, reenvía la información que recibe del ojo a la amígdala y a la corteza estriada;

la amígdala recibe el estímulo y le da un valor en base al nivel de atención, el cual envía a la corteza estriada para determinar si este estímulo es procesado;

corteza visual primaria es donde los datos visuales pre-procesados se producen, una primera segmentación de los objetos de la escena se realiza, trata de encontrar objetos en

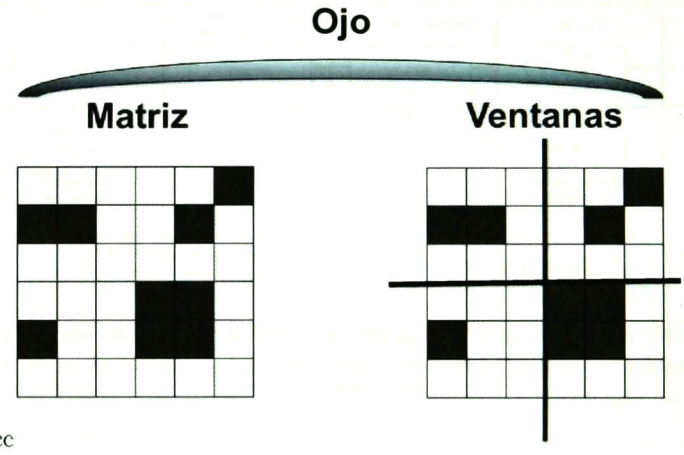


Figura 3.3: **El ojo** La entrada que recibe el ojo es una matriz que contiene unos(negros) y ceros(blancos), esta matriz es segmentada en ventanas para su posterior procesamiento.

cada una de las ventanas ya definidas;

corteza visual secundaria son separados para su identificación posterior;

corteza inferior temporal es el ultimo paso para el procesamiento visual, transporta los objetos a la corteza perirrinal, la principal entrada al sistema de memoria;

el ultimo paso de la identificación es la selección y el reporte de regreso en la corteza inferior temporal, un listado de clases a las que puede pertenecer el objeto llega, la corteza inferior temporal se encarga de seleccionar la clase a la cual pertenece el objeto.

Las *aferencias* a las áreas visuales para la identificación son las siguientes:

- La *corteza perirrinal* le envía la lista de clases a las cuales puede pertenecer el objeto y el identificador de la nueva clase.

Las *funciones* de las áreas visuales para la identificación son:

- Es la principal entrada al sistema de memoria. Los datos que envía consisten de *tipos singulares* definidos como representaciones codificadas de objetos en el escenario, transportados por el sistema de memoria. También recibe el nombre de los identificadores de las nuevas clases. Como ultimo paso de la simulación, una lista de clases arriban del

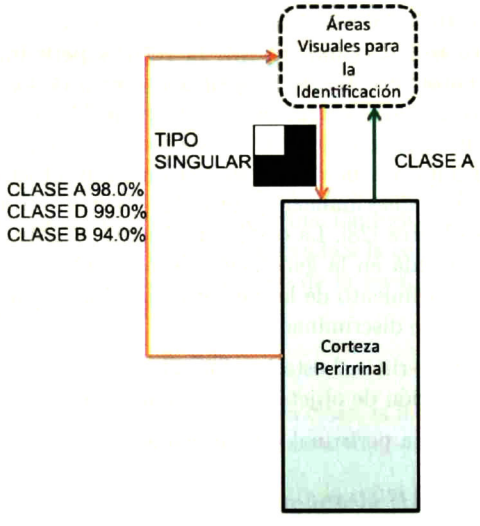


Figura 3.4: **Áreas Visuales para la Identificación** Las áreas visuales para la identificación envía un tipo singular (cuadro en blanco y negro) a la corteza perirrinal, recibe el identificador de una nueva clase (cuando se identifica una nueva clase de objeto) y un listado de posibles clases a las cuales puede pertenecer el objeto (ademas de un valor de certidumbre, el cual indica que tanto se parece este objeto a la clase mencionada)

sistema de memoria, se realiza una selección en base a ellas. Las clases seleccionadas son reportadas al usuario (Figura 3.4).

La *eferencia* de las áreas visuales para la identificación es:

Le envía a la *corteza perirrinal* un tipo singular.

Corteza Perirrinal

La corteza perirrinal tiene un papel importante en el reconocimiento e identificación de objetos, porque estudios recientes indican que la corteza perirrinal en conjunto con otras áreas, como la corteza entorrinal, recibe información acerca de los objetos, mantiene representaciones de los objetos activos en la memoria de trabajo [34]. Asocia diferentes perspectivas y atributos no visibles para cada objeto; esto es ayuda a distinguir un objeto en base a su olor, textura y sabor. También ayuda en la asociación de un objeto con otros para alcanzar un objetivo, interviene en la discriminación entre dos objetos y en la generación de la representación interna de los objetos [28]. La corteza perirrinal es activada significativamente con estímulos nuevos lo que ayuda en la generación de nuevas memorias [37], [8]. Esta involucrada en la tarea de reconocimiento de la memoria visual [36], [15]. Ayuda en el proceso de familiaridad y en sistemas de discriminación [8].

En resumen, la corteza perirrinal esta involucrada en el reconocimiento de objetos, identificación de objetos, asociación de objetos y representación de objetos.

Las *aferencias* a la corteza perirrinal son las siguientes:

Las *áreas visuales para la identificación* le envían un tipo singular.

La *corteza entorrinal* le envía el identificador de la nueva clase, la lista de clases a las cuales puede pertenecer el objeto y la señal de estímulo desconocido.

Las *funciones* de la corteza perirrinal son:

- Reenvía el tipo singular a la corteza entorrinal.

  Envía la señal para comenzar los procesos de codificación y almacenamiento a la corteza entorrinal.

  Reenvía la lista de posibles clases a las cuales pertenece un objeto a las áreas visuales para la identificación.

  Reenvía el identificador de una nueva clase detectada o reconocida a las áreas visuales para la identificación.

Las *eferencias* de la corteza perirrinal son:

Le envía a las *áreas visuales para la identificación un identificador* de la nueva clase y la lista de clases a las cuales puede pertenecer el objeto.

- Le envía a la *corteza entorrinal* un tipo singular y la señal de inicio del proceso de codificación.

Corteza Entorrinal

La corteza entorrinal ayuda en la tarea de discriminación entre objetos diferentes y la retención de memoria. Responde diferente a diferentes estímulos, sugiriendo memoria, también actúa como puente entre estructuras subsecuentes, como el hipocampo y la corteza perirrinal [34], [8].

La corteza entorrinal es la puerta principal a la formación hipocampal. Lo que significa que el mayor flujo de información que alcanza la formación hipocampal proviene de la corteza entorrinal [15]. La formación hipocampal esta conectada a la corteza entorrinal principalmente por el subículo; esto es, la información que sale de la formación hipocampal pasa por el subículo para llegar a la corteza entorrinal [36].

Las *aferencias* a la corteza entorrinal son:

El *subículo* le envía el identificador de la nueva clase, la lista de clases a las cuales puede pertenecer el objeto y la señal de estímulo desconocido.

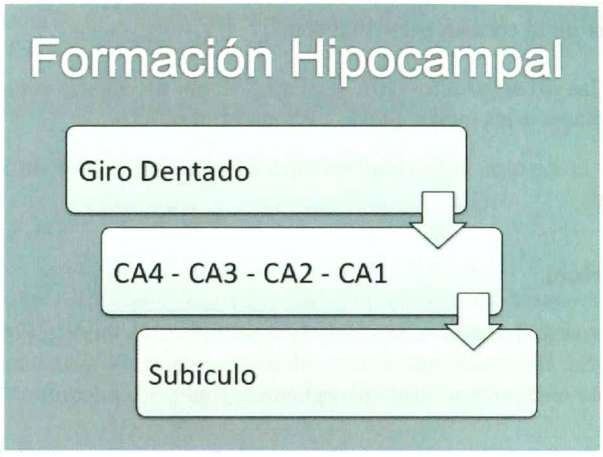
La *corteza perirrinal* le envía un tipo singular y la señal para comenzar la codificación.

Las *funciones* de la corteza entorrinal son:

La corteza entorrinal reenvía el tipo singular que recibe de la corteza perirrinal. La corteza perirrinal le indica si el tipo singular es conocido o desconocido. Si es un estímulo conocido lo envía a CA3, de lo contrario se lo envía al giro dentado para posteriormente codificarlo y almacenarlo. También envía el tipo singular a CA1 para realizar una comparación entre la predicción de CA3 y su percepción.

Las *eferencias* de la corteza entorrinal son:

- El *giro dentado* le envía un tipo singular.
- Le envía a *CA3*un tipo singular.
- Le envía a *CA1* un tipo singular.



cc

Figura 3.5: **Formación Hipocampal** Esta compuesta por el giro dentado, el cornu ammonis, que a su vez esta divido en cuatro secciones llamadas CA1, CA2, CA3 y CA4

Formación Hipocampal

La formación hipocampal actúa como un almacén de memoria temporal [36], esta involucrada en el reconocimiento de objetos [37] y la codificación de información para la generación de nuevas memorias [8].

La formación hipocampal esta constituida por el giro dentado (GD), el cornu ammonis y el subículo. El cornu ammonis esta dividido en cuatro áreas conocidas como CA1, CA2, CA3 y CA4 (Figura 3.5), para la generación de memorias e identificación de objetos solo intervienen CA1 y CA3. La información que alcanza la formación hipocampal viaja hacia las áreas subsecuentes, esto es, la información entra por el giro dentado ó CA3 y sigue su camino hacia CA1 para salir por el subículo para alcanzar otras áreas cerebrales adyacentes [15]. La siguiente sección describe cada una de ellas.

Giro Dentado

El giro dentado esta involucrado en la generación de nuevas memorias, genera patrones de disparo, los cuales envía a CA3 cuando se obtiene un nuevo estímulo [36], estos patrones de disparo son utilizados para realizar búsquedas en la memoria. El estímulo desconocido llega a través de la corteza entorrinal [36].

La *aferencia* al giro dentado es:

La *corteza entorrinal* le envía un tipo singular.

Las *funciones* del giro dentado son:

- Cuando el giro dentado recibe un estímulo desconocido, le asigna una representación y un identificador. Estos dos datos son enviados a CA3. CA3 los agrega a su base de datos y puede utilizarlos posteriormente para hacer predicciones acerca del ambiente.

Las *eferencias* del giro dentado son:

Le envía a *CA3* un identificador de la nueva clase y la representación de la nueva clase.

CA3

CA3 actúa como una memoria asociativa almacenando nuevas memorias, recuperando información almacenada con fragmentos de la información, haciendo una especie de predicción para recuperar memorias, reuniendo información para diferentes áreas corticales, formando representaciones del ambiente [36].

Las *aferencias* a CA3 son:

El *giro dentado* le envía un identificador de la nueva clase y la representación de la nueva clase.

La *corteza entorrinal* le envía un tipo singular.

Las *funciones* de CA3 son:

Cuando recibe un tipo singular, de la corteza entorrinal, trata de realizar una predicción en base a el. Si el estímulo es conocido realiza una predicción y la envía a CA1, la predicción consta de una lista de clases a las cuales puede pertenecer el estímulo. Si el estímulo es desconocido lo informa a CA1.

Las *eferencias* de CA3 son:

- Le envía a *CA1* un identificador de la nueva clase, la lista de clases a las cuales puede pertenecer el objeto y la señal de estímulo desconocido.

CA1

CA1 actúa como un comparador, esto es, compara la predicción de CA3 y la información de la corteza entorrinal, CA1 tiene una activación significativa con los estímulos nuevos, pero estudios muestran que las señales de activación ocurren en etapas anteriores del procesamiento, lo que significa que la señal de activación por estímulo desconocido es enviada por áreas cerebrales anteriores [27].

Las *aferencias* a CA1 son:

- *CA3* le envía la lista de clases a las cuales puede pertenecer el objeto.

  La *corteza entorrinal* un tipo singular.

Las *funciones* de CA1 son:

Si recibe un identificador de una clase nueva, reenvía la información al subículo.

- Si recibe la señal de que el estímulo es desconocido o nuevo envía la señal para comenzar el proceso de codificación y almacenamiento.

  Si recibe la predicción de CA3, lista de clases a las cuales puede pertenecer el objeto, realiza una comparación entre lo que predice CA3 y el estímulo proveniente del ambiente enviado por la corteza entorrinal.

Las *eferencias* de CA1 son:

Le envía al *subículo* la lista de clases a las cuales puede pertenecer el objeto, un identificador de la nueva clase y la señal para comenzar el proceso de codificación.

Subículo

El subículo toma la salida de CA1 y la envía a la corteza entorrinal, lo que significa es la principal salida de información de la formación hipocampal, principalmente a la corteza entorrinal. Ayuda en la formación de nuevos episodios, ya que es la principal salida de información de la formación hipocampal [36].

Las *aferencias* al subículo son:

CA1 le envía la lista de clases a las cuales puede pertenecer el objeto, un identificador de la nueva clase y la señal para comenzar el proceso de codificación.

La *función* del subículo es:

Toda la información que recibe de CA1 la reenvía a la corteza entorrinal para subsecuente procesamiento.

Las *eferencias* del subículo son:

Le envía a la *corteza entorrinal* la lista de clases a las cuales puede pertenecer el objeto, el identificador de la nueva clase y la señal para comenzar el proceso de codificación.

3.2.2. Etapas de la memoria

A continuación se describe como las áreas cerebrales mencionadas en conjunto logran realizar las tres etapas de la memoria. Cada una de estas etapas (codificación, almacenamiento y recuperación) son desglosadas en subfunciones. Como se muestra en la Figura 3.1 estas subfunciones son realizadas por las áreas cerebrales mencionadas, mediante el paso de información y el trabajo en conjunto.

Codificación

El proceso de *codificación* (Figura 3.6) crea una representación interna de la información. Para el proceso de reconocimiento de objetos se puede listar a las área cerebrales involucradas, las cuales son: la corteza perirrinal, la corteza entorrinal, el giro dentado y CA3. La codificación es realizada siguiendo los siguientes pasos (camino azul de la Figura 3.1).

I. La corteza perirrinal es activada indicando que un estímulo desconocido ha llegado. Entonces procede a reenviar el tipo singular a la corteza entorrinal, indicando le que debe comenzar el proceso de codificación.

II. Cuando la corteza perirrinal le informa a la corteza entorrinal que codifique el estímulo desconocido, esta lo reenvía al giro dentado.

III. El giro dentado genera un patrón de disparo y le asigna un identificador único, basando se en el tipo singular que recibió. Posteriormente el giro envía el patrón de disparo y el identificador a CA3 quien lo utiliza para realizar predicciones.

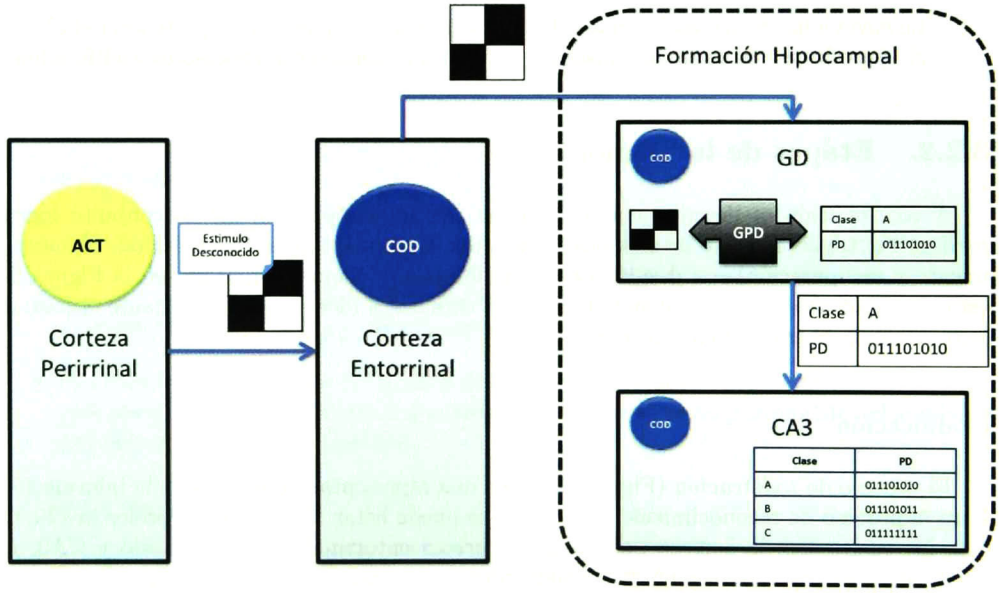


Figura 3.6: **Codificación** Podemos ver como durante el proceso de codificación; la corteza perirrinal se actíva (ACT); la corteza entorrinal se pone en modalidad codificación (COD); el giro dentado genera (GPD) una patrón de disparo (PD) y un identificador de clase; y finalmente CA3 almacena los nuevos datos.

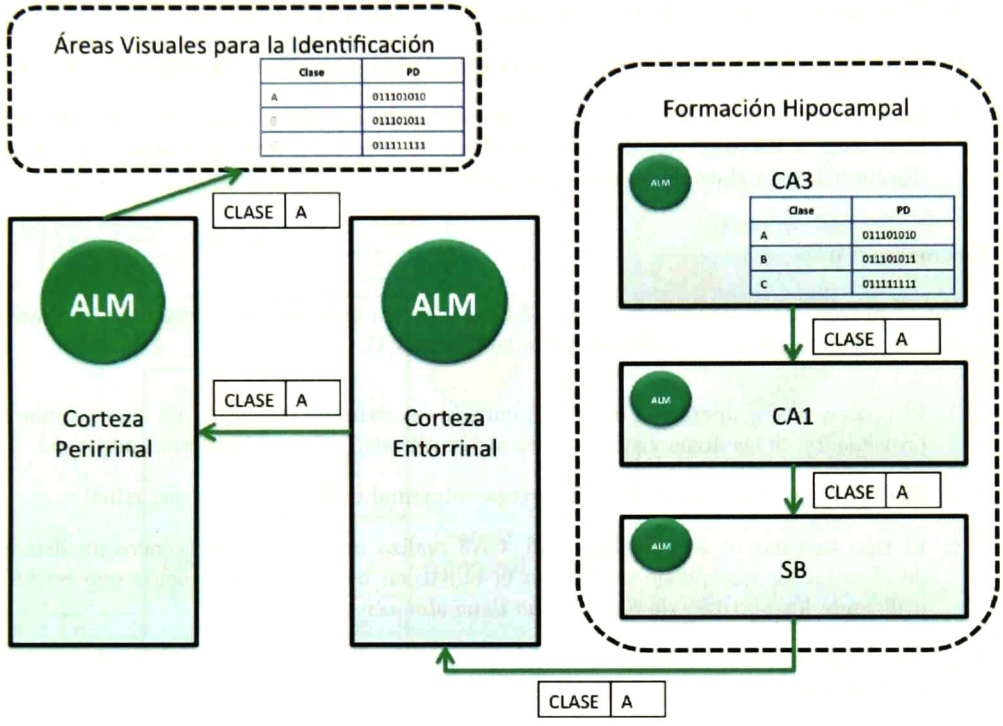


Figura 3.7: **Almacenamiento** Durante el almacenamiento; CA3 y las áreas visuales para la identificación almacenan los datos y se pone en modalidad de almacenamiento (ALM); mientras que CA1, el subículo, la corteza entorrinal, la corteza perirrinal solo reenvían los datos que reciben.

Almacenamiento

El proceso de *almacenamiento* (Figura 3.7) mantiene la información a través del tiempo. Para el proceso de reconocimiento de objetos es posible listar las áreas cerebrales involucradas, las cuales son: CA3, CA1, subículo, la corteza entorrinal, la corteza perirrinal y las área visuales para la identificación (camino verde de la Figura 3.1).

A. CA3 almacena el nuevo patrón de disparo, posteriormente envia a CA1 el identificador de la nueva clase.

B. CA1 reenvía el identificador de la nueva clase a el subículo.

C. El subículo reenvía el identificador de la nueva clase a la corteza entorrinal.

D. La corteza entorrinal reenvía el identificador de la nueva clase a la corteza perirrinal.

E. La corteza perirrinal detecta que el identificador de una nueva clase a llegado, entonces lo reenvia a las área visuales para la identificación, para que estas sean capaces de discriminar esta clase de objetos entre otras.

Recuperación

En el proceso de *recuperación* (Figura 3.8)se accede a información almacenada mediante fragmentos de información (camino rojo de la Figura 3.1).

1. El proceso de recuperación comienza cuando un estímulo (en forma de tipo singular), proveniente de las áreas visuales, para la identificación llega a la corteza perirrinal.
2. El tipo singular es reenviado a la corteza entorrinal desde la corteza perirrinal.
3. El tipo singular es reenviado a CA3. CA3 realiza una predicción (genera un listado de clases a las que puede pertenecer el objeto)en base a la información que recibió, utilizando los patrones de disparo que tiene almacenados.
4. *Ambiente contra predicciones* CA1 compara la predicción de CA3 contra el estímulo original que le envía la corteza entorrinal.
    1. Las predicciones de CA3 son enviadas a CA1 para compararlas con el estímulo original.
    2. La corteza entorrinal reenvía el estímulo original a CA1.
5. El resultado que se obtiene de realizar la comparación entre la predicción de CA3 y el estímulo original, es enviado al subículo junto con las predicciones de CA3 (lista de clases).
6. El subículo envía las predicciones de CA3 a la corteza entorrinal.
7. La corteza entorrinal envía las predicciones de CA3 a la corteza perirrinal.
8. La corteza perirrinal envía las predicciones de CA3 a las áreas visuales para la identificación, estas a su vez eligen la clase a la cual el estímulo corresponde.

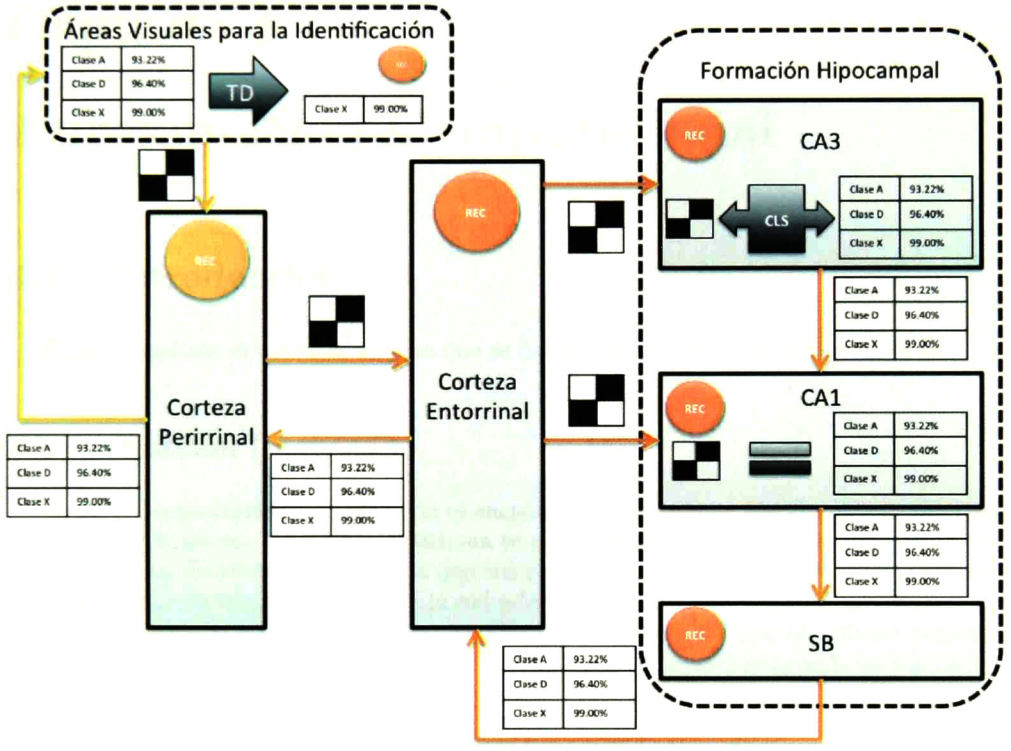


Figura 3.8: **Recuperación** Durante la recuperación (REC); CA3, utilizando un clasificador, reúne un conjunto de posibles clases a las que puede pertenece el objeto, este listado de clases pasa por todas las áreas subsecuentes (CA1, subículo, corteza entorrinal, corteza perirrinal y áreas visuales para la identificación, en esta ultimas se realiza un proceso de selección de la clase a la que mas se parece el objeto.

Capítulo 4

Aproximación Computacional

4.1. Introducción

En este capítulo se define el sistema que se desarrollo en base al modelo propuesto.

4.1.1. Sistema Distribuido

Como se ha mencionado, la memoria se encuentra distribuida en toda la corteza cerebral, por lo que al momento de diseñar el sistema se opto por implementar un sistema distribuido. Un sistema distribuido es aquel en que sus componentes están localizados en diferentes computadoras, conectadas a través de la red además se comunican y coordinan a través del paso de mensajes [10]. Entre las razones por las que elegimos este tipo de sistema podemos encontrar las siguientes [2]:

- incremento del rendimiento
- incremento de la confiabilidad y disponibilidad

  flexibilidad

  facilitar la expansibilidad

  modularidad
- reducción de costos

4.1.2. Middleware

El corazón (core) del sistema es un *middleware*. Un middleware (MW) es un software que provee una abstracción de programación, así como un enmascaramiento de la heterogeneidad subyacente de las redes, hardware, sistemas operativos y lenguajes de programación [10].

Las propiedades del MW nos permiten distribuir el sistema en un conjunto de maquinas, que se comunican a través de la red, dando la sensación de tener un sistema centralizado, así las peticiones se realizan de forma local pero son resueltas por algún elemento del sistema de manera transparente.

4.2. Descripción de la descomposición

4.2.1. Descomposición modular

El sistema esta compuesto por dos tipos de estructuras (Figura 4.1): MW y nodos. Los nodos representan áreas del cerebro, los cuales simulan el comportamiento que tendrían estas áreas en el proceso de memoria. Los MW representan secciones del cerebro donde se encuentran las áreas representadas por los nodos, por ejemplo, la corteza entorrinal y la corteza perirrinal se encuentran en el giro parahipocampal. Por lo que el MW llamado giro parahipocampal se encarga de la comunicación de la corteza entorrinal y la corteza perirrinal con el resto del sistema.

Middleware Principal

Es el encargado de registrar y buscar todos los MW que se encuentran activos en el sistema. Cada MW que se encuentra en el sistema debe de registrarse con el, para que el resto de los componentes del sistema lo puedan encontrar.

La Formación Hipocampal

Es la encargada de registrar nodos (giro dentado, subículo y el cornu ammonis), encontrar nodos en diferentes MW, además de informarle a el resto del sistema donde se encuentran los nodos que administra.

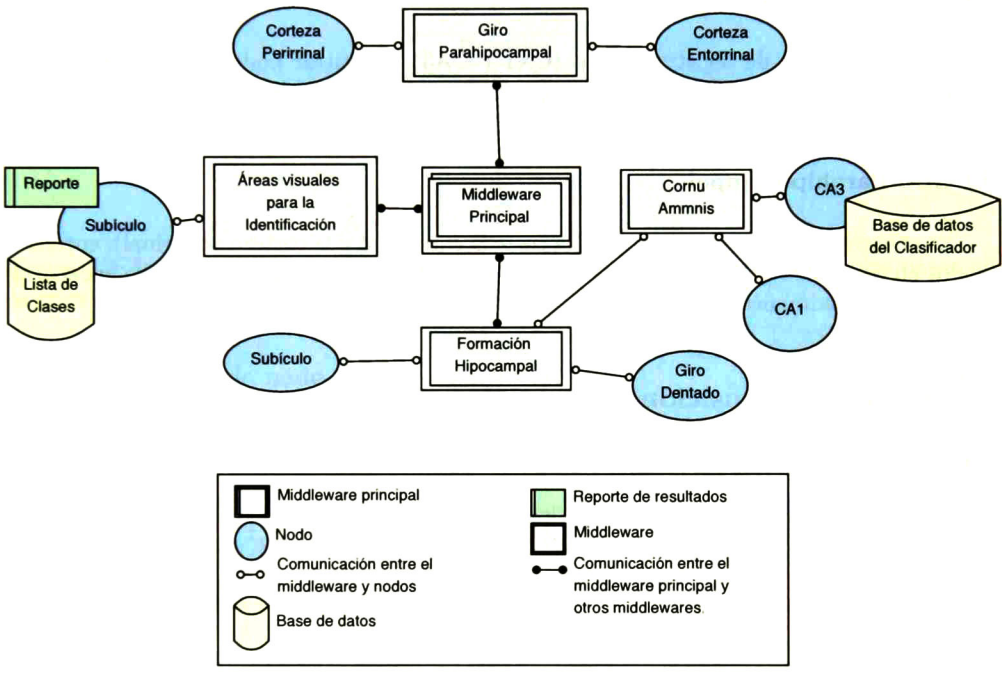


Figura 4.1: **Modelo de Subsistemas** El modelo de memoria esta compuesto por cinco grandes subsistemas; el MW principal, el Giro Parahipocampal, las Áreas Visuales para la Identificación, el Cornu Ammonis y la Formación Hipocampal

Las Áreas Visuales para la Identificación

Son las encargadas de registrar nodos (nodo áreas visuales para la identificación), encontrar nodos en diferentes MW, además de informarle a el resto del sistema donde se encuentran los nodos que administra.

El Cornu Ammonis

Es el encargado de registrar nodos (CA1 y CA3), encontrar nodos en diferentes MWs, además de informarle a el resto del sistema donde se encuentran los nodos que administra.

El Giro Parahipocampal

Es el encargado de registrar nodos (la corteza entorrinal y la corteza perirrinal), encontrar nodos en diferentes MWs, además de informarle a el resto del sistema donde se encuentran los nodos que administra.

4.2.2. Descomposición actual del proceso

Proceso de inicio de middleware principal

En el proceso de inicio de MW principal, como se muestra en la Figura 4.2, primero este debe de abrir un socket para reservar un puerto, después inicia un hilo para que este revisando las peticiones de la red y así empezar a resolver peticiones como: registro y búsqueda de MW.

Proceso de inicio de middleware

Los MW definidos (la formación parahipocampal, las áreas visuales para la identificación, el cornu ammonis y el giro parahipocampal) están estructurados idénticamente,tienen las mismas funciones y las mismas estructuras de datos, lo que cambia entre ellas son los nodos que administra. Por ello, todos inician siguiendo los mismos pasos (Figura 4.3).

Cuando inicia un MW primero abre un socket para poderse comunicar con el exterior, inicia un hilo para escuchar peticiones de la red, genera un mensaje para registrarse en el MW principal, envía el mensaje, cuando recibe el mensaje de que ha sido registrado empieza a aceptar peticiones, continua aceptando peticiones hasta que detiene el hilo y termina su ejecución.

CC

Figura 4.2: **Inicio middleware principal** Diagrama de estados que muestra los pasos necesarios para que el MW principal inicie.

Proceso de inicio de nodo

Todos los nodos del sistema tienen en mismo proceso de inicio, como se muestra en la Figura 4.4.

Un nodo comienza con abrir un socket, iniciar el hilo que reciba peticiones de la red, envía una solicitud para registrarse en el middeware que le corresponde, espera la respuesta del MW, cuando la recibe inicia su ejecución; resuelve peticiones, si las tiene, si no genera paquetes para solicitar información o enviar resultados obtenidos, espera respuestas.

Proceso de intercambio de datos

Para enviar datos de un nodo a otro, como se ve en la Figura 4.5 se deben seguir los siguientes pasos.

El nodo A comienza buscando en sus registros internos la dirección del *nodoB*; si lo tiene registrado; le envía directamente el mensaje; si no lo tiene; le pregunta la dirección a su MWC y el MWC busca si tiene registrado el *nodoB*; si lo tiene; le envía los datos al *nodoA*; si no lo tiene; busca en sus registros al MWD que administra el *nodosB* y después le pregunta al *MWprincipal* la dirección del MWD; si el *MWprincipal* encuentra la dirección del MWD; le envía los datos al MWC y el MWC le pregunta al MWD la ubicación del *nodoB*; si lo tiene registrado; le envía los datos al MWC, el MWC le informa al *nodoA* y el *nodoA* le

cc

Figura 4.3: **Inicio de un MW** Diagrama de estados que muestra los pasos necesarios para que un MW inicie.

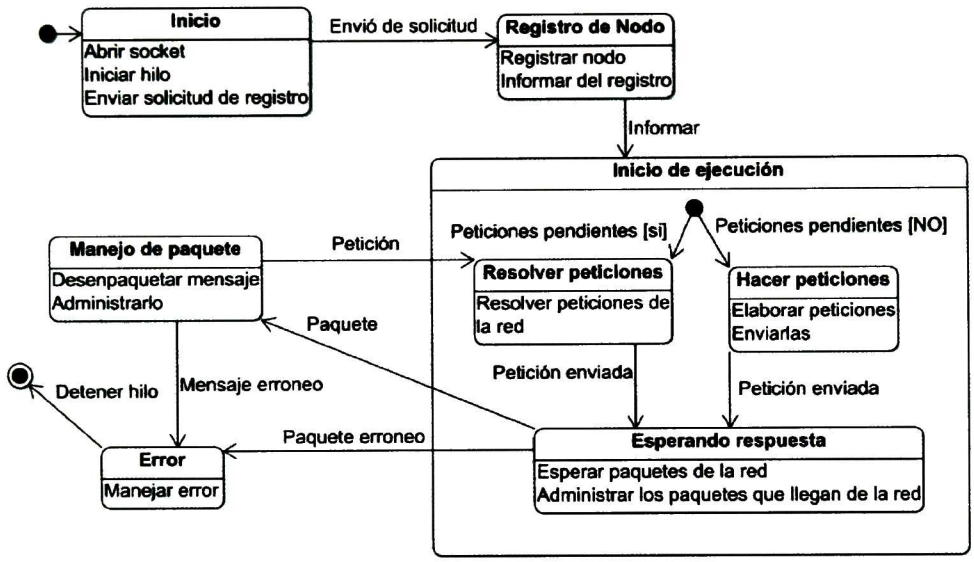


Figura 4.4: **Inicio del nodo** Diagrama de estados que muestra los pasos necesarios para que un nodo inicie.

Tabla de Patrones	
1	000000***1
2	010101****
3	1111**1100
4	101011111*
5	000111***0
	...
	...
n	**11**11**

Tabla 4.1: **Tabla de patrones** muestra la estructura del listado de patrones con su identificador.

envía la información al *nodoB*; si no lo tiene; el *MWC* le informa al *nodoA* y termina el proceso; si no lo tiene; se le informa al *MWC*, el *MWC* le informa al *nodoA* y termina el proceso.

4.2.3. Descomposición de los datos

El sistema cuenta con dos tablas de datos. Una contiene el identificador de todas las clases de objetos que se han reconocido, la otra contiene un listado de los identificadores de objetos junto con su patrón de disparo (regla) que lo representa en el sistema.

Lista de clases

La lista de clases es una tabla, que contiene los identificadores de todos los objetos que se han reconocido en el sistema. Esta tabla se encuentra en el nodo de las áreas visuales para la identificación.

Registro del clasificador

El sistema cuenta con un clasificador, el cual ayuda en la etapa de memoria *recuperación*. Este clasificador como se explica en las siguientes secciones [14], trata de asociar el estímulo entrante con objetos ya almacenados, para lograr esto el estímulo se transforma en una representación clara para el clasificador (cadenas de ceros y unos como se ve en el Tabla 4.1). El listado de los identificadores asociados con un patrón es lo que se almacena en esta tabla.

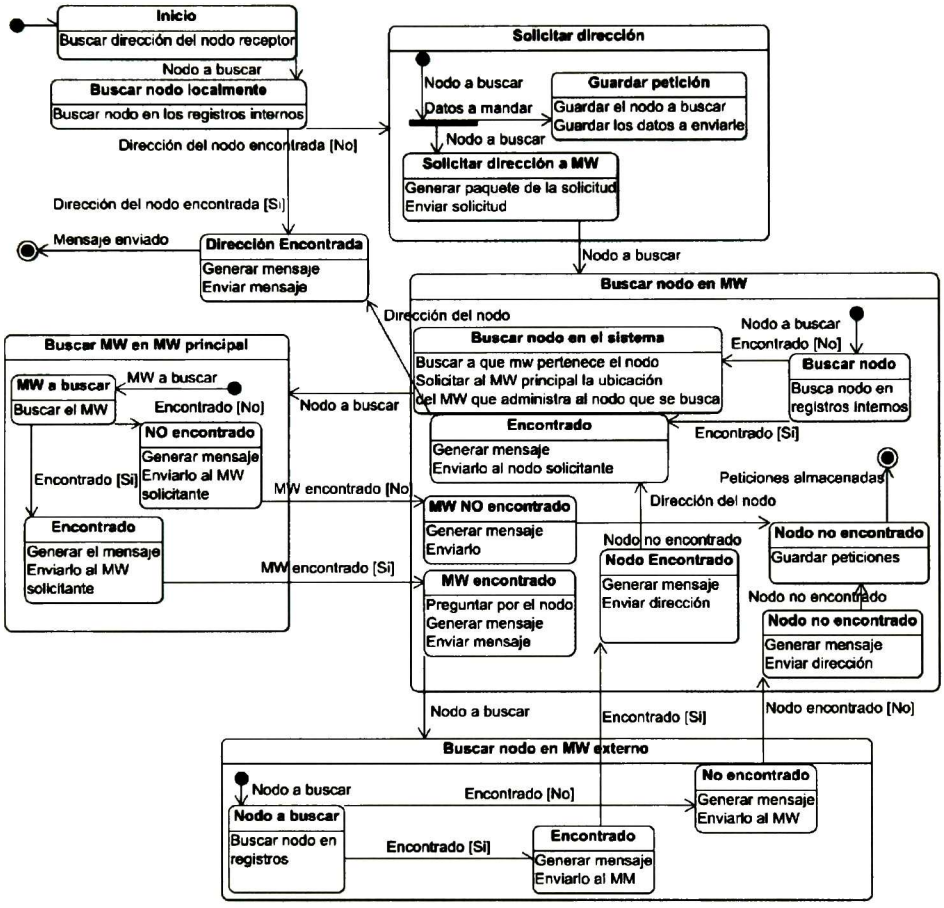


Figura 4.5: **Intercambio de datos** Diagrama de estados que muestra el proceso necesario para que un nodo se comunique con otro a través de la red.

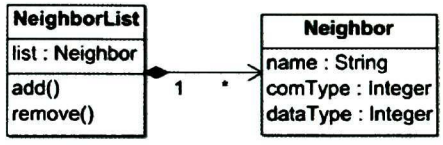


Figura 4.6: **NeighborList** Diagrama de dependencias de las clases NeighborList y Neighbor.

4.3. Descripción de dependencias

En esta sección se definen las clases que conforman el sistema, las cuales están divididas en cinco secciones; *Clases de utilidades* que ayudan al resto de las clases a trabajar en conjunto; *Interfaces* que ayudan a las demás clases a tener una comunicación bidireccional entre dos clases; las estructuras que conforman el *Middleware principal*; las estructuras que conforman un *Middleware*; las estructuras que conforman un *Nodo*.

4.3.1. Clases de utilidades

En esta sección se listan las dependencias de las clases que generan parte de las funciones de las estructuras principales (MW principal, MW y nodo).

NeighborList

La Figura 4.6 muestra a las clases NeighborList y Neighbor. La clase Neighbor es utilizada para representar los nodos con los que se va a comunicar otro nodo. Contiene el tipo de información que se va a intercambiar. NeighborList es un arreglo de objetos de la clase Neighbor, ya que cada nodo puede comunicarse con mas de un nodo.

RequestList

La Figura 4.7 muestra a las clases RequestList y RequestNode. La clase RequestNode es utilizada para almacenar peticiones, sin resolver, que se han realizado. Por ejemplo: cuando un *nodoA* le pregunta algo al *nodoB*, ese ultimo almacena sus datos de manera temporal mientras resuelve su petición. RequestList es un arreglo de objetos de la clase RequestNode, se encuentra en todos los nodos para poder manejar todas las peticiones que le lleguen.

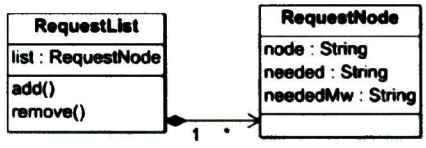


Figura 4.7: **RequestList** Diagrama de dependencias de las clases RequestList y RequestNode.

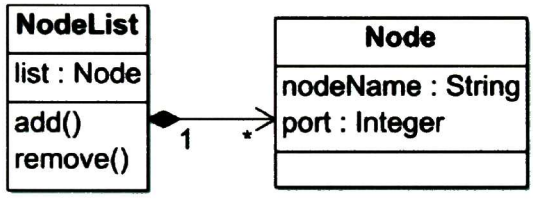


Figura 4.8: **NodeList** Diagrama de dependencias de la clase NodeList y Node.

NodeList

La Figura 4.8 muestra a las clases NodeList y Node. La clase Node tiene la información que identifica un nodo, como su nombre y el puerto donde se encuentra. Es utilizada para almacenar la información de todos los nodos, para poder comunicarse con ellos. NodeList es un arreglo de objetos de la clase Node.

WhitePage

La Figura 4.9 muestra a las clases WhitePage y Tuple. La clase Tuple es utilizada para almacenar a que MW pertenece cada nodo. Por ejemplo: (perirrinal parahipocampal).WhitePage es un arreglo de objetos de clase Tuple. Cada MW tiene la misma copia de la clase WhitePage.

4.3.2. Interfaces

En la Figura 4.10 podemos ver las interfaces que se declararon para permitir la comunicación bidireccional ente las clases.

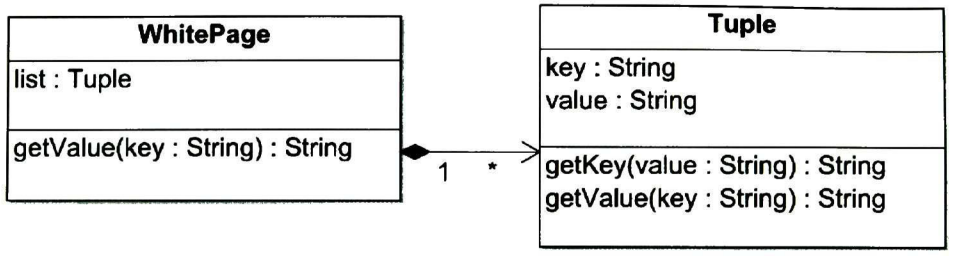
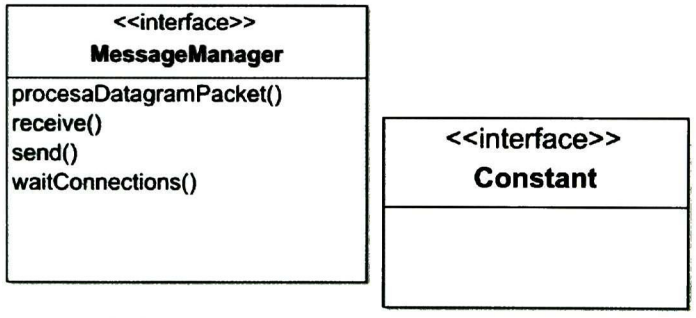


Figura 4.9: **WhitePage** Diagrama de dependencias de las clases WhitePage y Tuple.



(a) MessageManager (b) Constant

Figura 4.10: **Interfaces Utilizadas**

MessageManager

La imagen (a) de la Figura 4.10 muestra la interfaz MessageManager. MessageManager permite la comunicación bidireccional entre las clases que contienen sockets y las que contienen hilos. Así cuando un paquete llega a la clase que contiene el hilo este se lo envía a la clase que contiene el socket a través de la interfaz.

Constant

La imagen (b) de la Figura 4.10 muestra la clase Constant. Constan contiene las constantes que que utilizan los nodos del sistema.

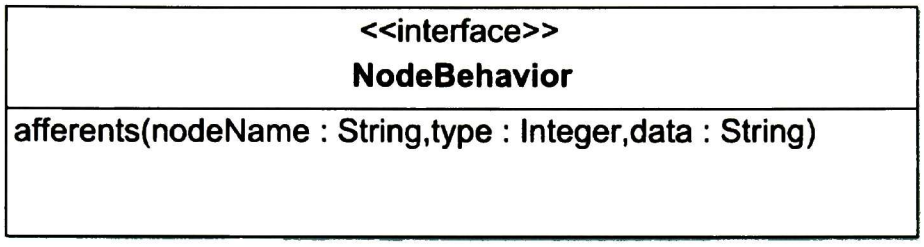


Figura 4.11: **NodeBehavior**

NodeBehavior

La Figura 4.11 muestra a la interfaz NodeBehavior. NodeBehavior ayuda a entablar la comunicación bidireccional entre las clases que representan un área cerebral y aquellas que se encargan la comunicación (NodeSocket).

4.3.3. Middleware principal

En la Figura 4.12 se muestra el diagrama de dependencias del MW principal. El MW principal esta compuesto por dos clases; MainMiddlewareSocket maneja las peticiones y la comunicación con el resto del sistema; MainMiddlewareThread espera las peticiones entrantes.

MainMiddleware

La imagen 4.13b de la Figura 4.13 muestra la clase MainMiddleware. Esta solo contiene un objeto de la clase MainMiddlewareSocket.

MainMiddlewareSocket

La Figura 4.14 muestra a la clase MainMiddlewareSocket. MainMiddlewareSocket es la encargada de manejar la comunicación del MW con la red. Registra nodos, busca nodos, transporta información, etcétera.

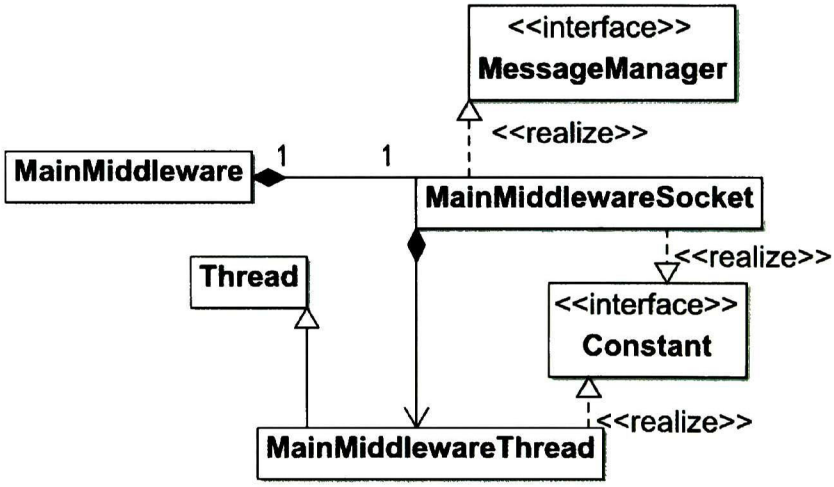
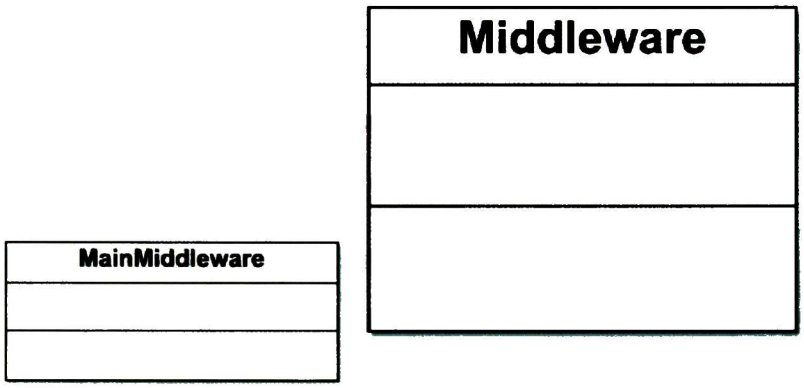


Figura 4.12: **Diagrama de dependencias del Middleware Principal**



(a) MainMiddleware (b) Middleware

Figura 4.13: **Clases MainMiddleware y Middleware**

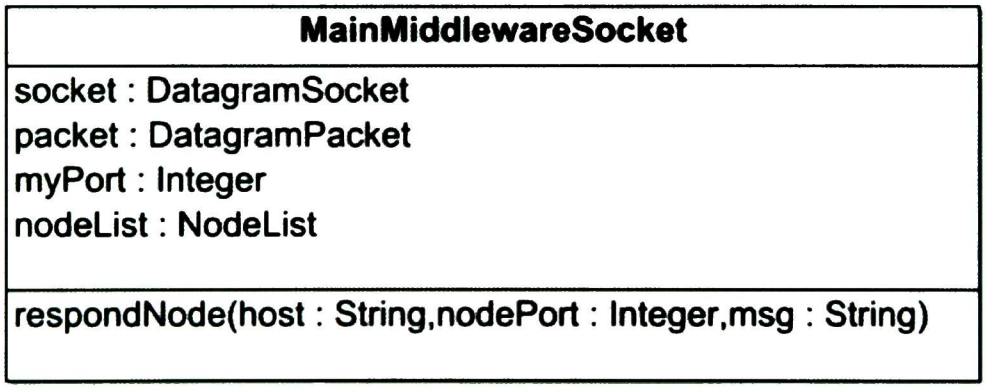


Figura 4.14: **MainMiddlewareSocket** Diagrama de la clase MainMiddlewareSocket

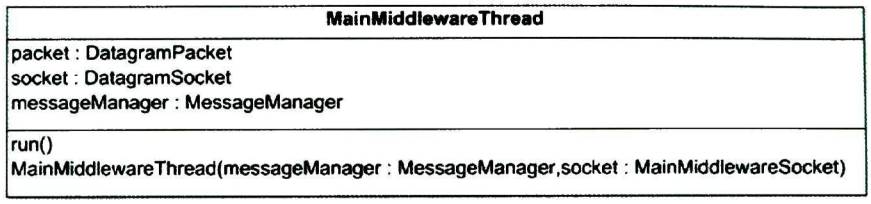


Figura 4.15: **MainMiddlewareSocket** Diagrama de la clase MainMiddlewareThread

MainMiddlewareThread

La Figura 4.15 muestra a la clase MainMiddlewareThread. MainMiddlewareThread contiene el hilo que espera las peticiones de la red. Los paquetes que le llegan los reenvía a la clase MainMiddlewareSocket a través de la interfaz MessageManager.

4.3.4. Middleware

En la Figura 4.16 se muestra el diagrama de dependencias de un MW. El MW esta compuesto por dos clases; MiddlewareSocket maneja las peticiones y la comunicación con el resto del sistema; MiddlewareSocket espera las peticiones entrantes.

Middleware

imagen (a) de la Figura 4.13 muestra la clase Middleware. Esta solo contiene un objeto de la clase MiddlewareSocket.

MiddlewareSocket

La Figura 4.17 muestra a la clase MiddlewareSocket. MiddlewareSocket es la encargada de manejar la comunicación del MW con la red. Registra nodos, busca nodos, transporta información, etcétera.

MiddlewareThread

La Figura 4.18 muestra a la clase MiddlewareThread. MiddlewareThread contiene el hilo que espera las peticiones de la red. Los paquetes que le llegan los reenvía a la clase MiddlewareSocket a través de la interfaz MessageManager.

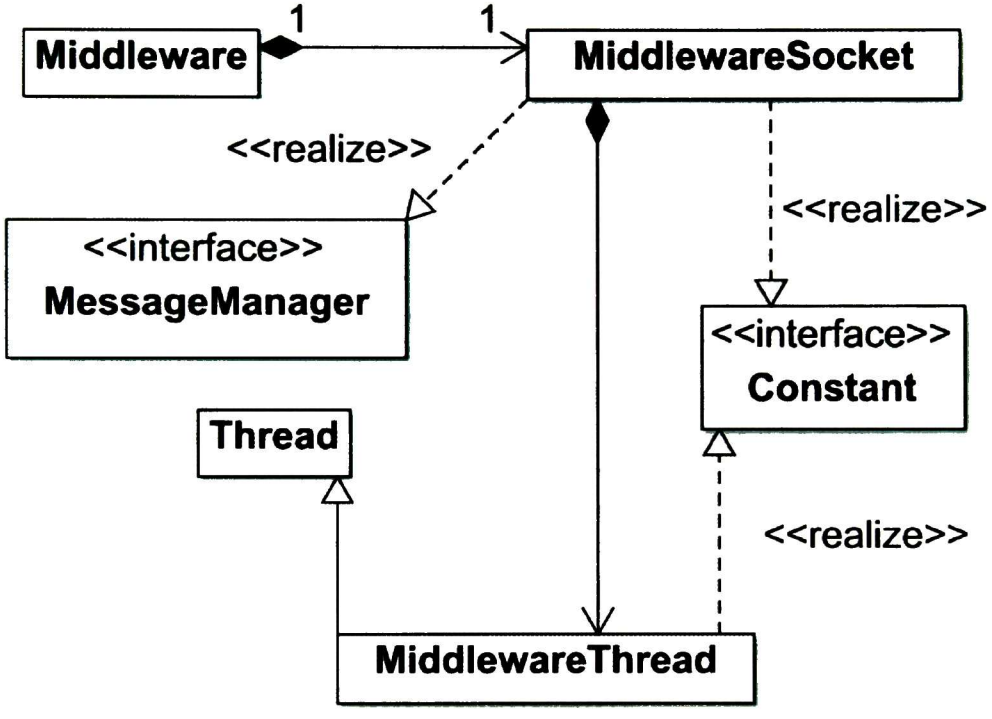
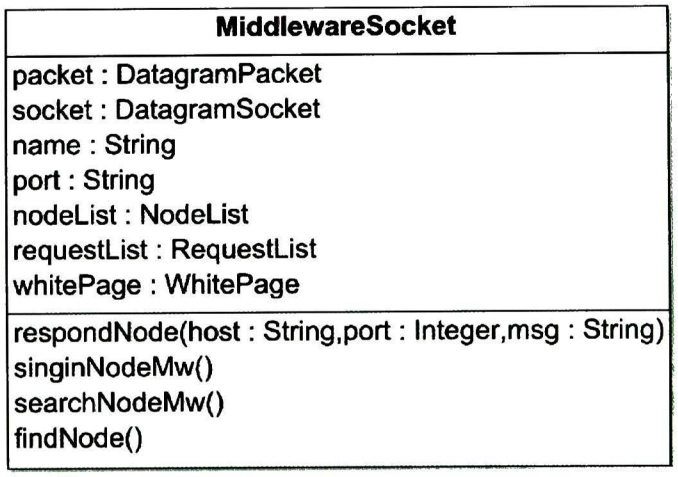


Figura 4.16: **Diagrama de dependencias del Middleware**



cc

Figura 4.17: **MiddlewareSocket** Diagrama de la clase MiddlewareSocket

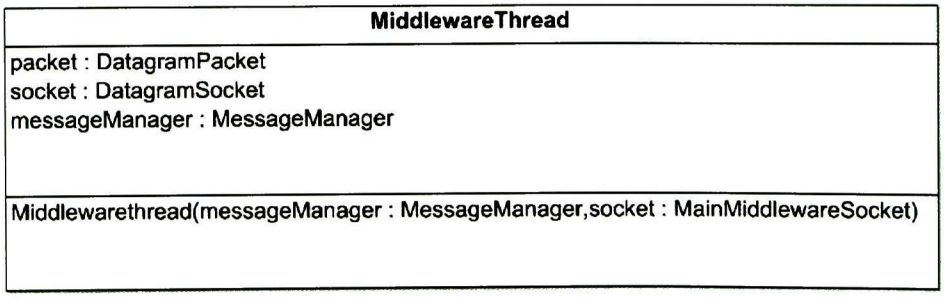


Figura 4.18: **MiddlewareThread** Diagrama de la clase MiddlewareThread

4.3.5. Nodos

En 4.19 se muestra el diagrama de dependencias de un nodo.Un nodo esta compuesto por dos clases; NodeSocket maneja las peticiones y la comunicación con el resto del sistema; NodeThread espera las peticiones entrantes.

Estructura de los nodos

La Figura 4.20 muestra la clase CPerrinal. CPerrinal representa a la corteza perirrinal. Como ya se menciono cada área cerebral que interviene en la generación de memorias es representada por un nodo. Cada nodo contiene un objeto de NodoSocket (es el encargado de la comunicación) además de las funciones propias del área que representa. En el caso de la corteza perirrinal las funciones que tiene son las de reenviar mensajes en base a su contenido y su origen.

NodeSocket

La Figura 4.21 muestra a la clase NodeSocket. NodeSocket es la encargada de manejar la comunicación de un nodo con la red. Registra nodos, pregunta por la ubicación de nodos, transporta información, etcétera.

NodeThread

La Figura 4.18 muestra a la clase NodeThread. NodeThread contiene el hilo que espera las peticiones de la red. Los paquetes que le llegan los reenvía a la clase NodeSocket a través de la interfaz MessageManager. 4.22

4.4. Detallado del funcionamiento

A continuación se detallan las tres funciones principales del sistema. En conjunto generan el comportamiento que esperamos del sistema.

4.4.1. Clasificador de CA3

CA3 es el encargado de realizar predicciones, las predicciones consisten en tratar de asignarle una o varias categorías al estímulo. Para eso CA3 revisa en todas las categorías

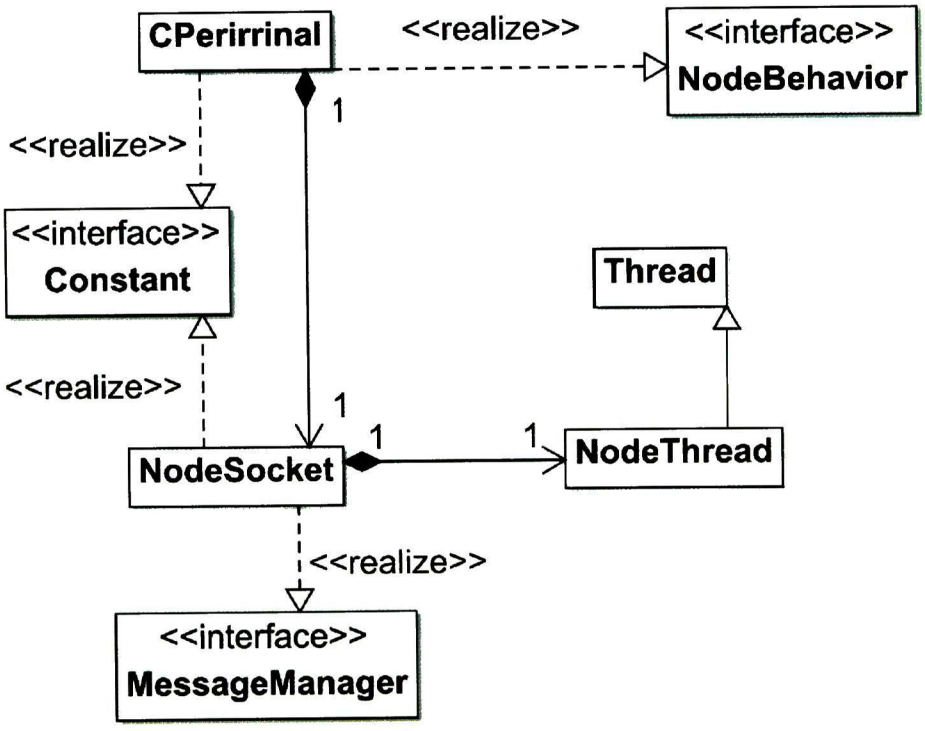


Figura 4.19: **Nodo** Diagrama de dependencias del Nodo.

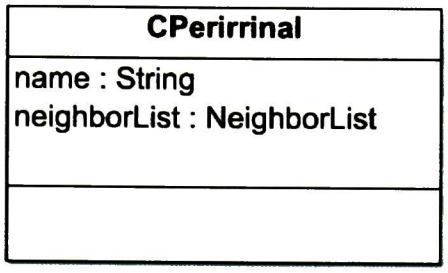
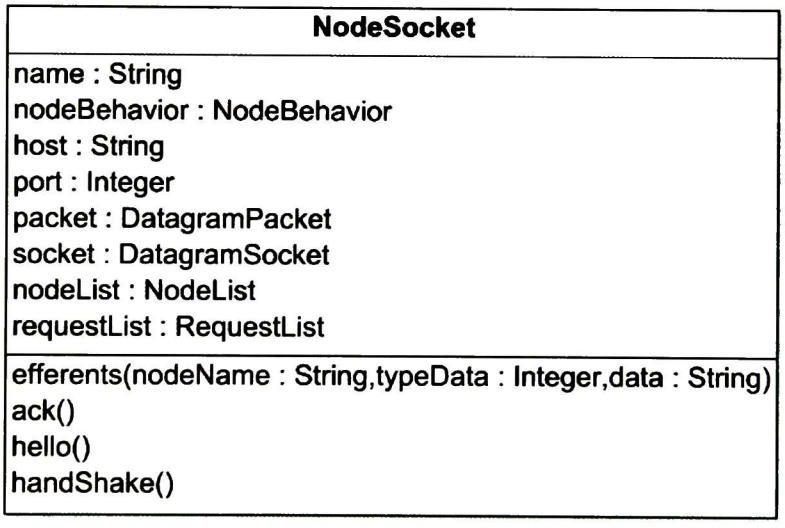


Figura 4.20: **Nodo Ejemplo** Diagrama de la clase CPr (Corteza Perirrinal) como ejemplo de las estructuras de los nodos.



cc

Figura 4.21: **NodeSocket** Diagrama de la clase NodeSocket.

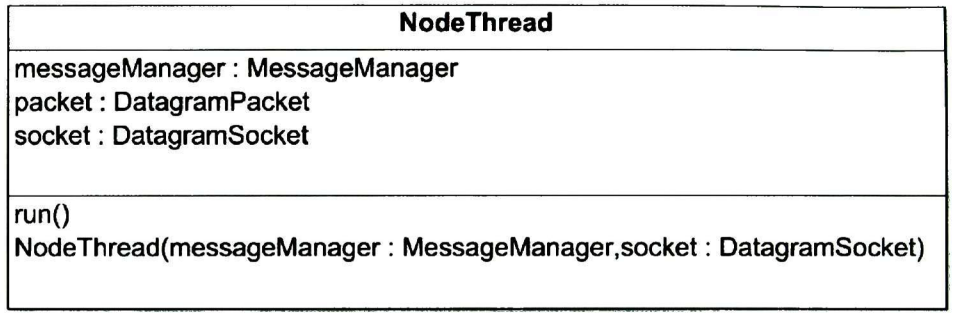


Figura 4.22: **NodeThread** Diagrama de la clase NodeThread.

que tiene almacenadas, trata de empatar al estímulo con ellas. CA3 lista las mas parecidas (las que tienen mayor porcentaje de certidumbre) y las envía a CA1.

Para lograr este comportamiento, el nodo que representa a CA3 contiene un clasificador LCS (*Learning Classifier Systems*). Un clasificador LCS esta formado por una población de reglas (también llamadas clasificadores) formados por una condición y una acción, que compiten y cooperan para dar una solución [14]. Las reglas cuya condición *empaten* con el estado actual son activas y sus acciones ejecutadas. La Figura 4.23 muestra el funcionamiento del clasificador [14].

La manera en que CA3 lo utiliza es la siguiente: CA3 toma el estímulo, se lo manda al clasificador, este le informa a que clasificadores o reglas puede pertenecer junto con un porcentaje de certidumbre). Estos clasificadores son utilizados por las áreas visuales para decir a que clase pertenece un objeto.

4.4.2. Comparador de CA1

CA1 es el encargado de comparar el estímulo con las predicciones de CA3. Para cumplir con esta función CA1 realizar las siguientes funciones (como se puede ver en la Figura 4.24 y en el pseudocódico que se encuentra en la parte inferior): toma el estímulo que le llega de la corteza entorrinal y lo compara con todos los patrones de disparo que CA3 mando. Los patrones de disparo que envía CA1 contienen caracteres *no importa*, lo que significa que no importa el valor que tenga el patrón en esa posición, así cuando se realiza la comparación de el estímulo con un patrón no se toma en cuenta esa posición para decidir si el estímulo puede pertenecer a ese patrón.

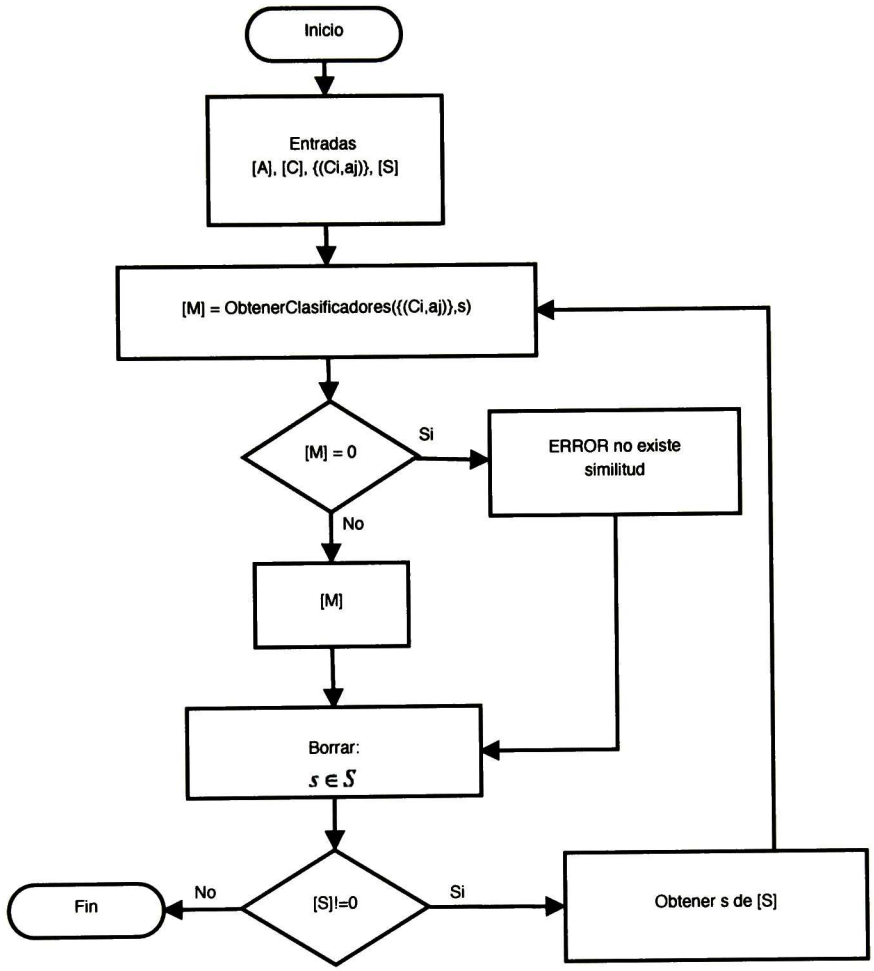


Figura 4.23: **LCS** Diagrama de flujo del clasificador LCS; $[A]$ es un conjunto de acciones, $[C]$ es un conjunto de condiciones, (C_i, a_i) es un conjunto de Condiciones-Acciones, $[S]$ es un el conjunto de clasificadores del LCS y $[M]$ Es el subconjunto de Condiciones-Acciones (subconjunto de (C_i, a_i)) que empataron con el elemento que se trata de clasificar.

```
begin
  p := generaPatrones(tipoSingular)
  msg := ""
  valido := verdadero
  for i := 0 to i < dimension(listaPatrones) ∧ valido step 1 do
      if !patronValido(p, listaPatrones[i]) then valido := false fi;
  od
  if valido then generaMensaje(VALIDO, listaPatrones)
            else msg := generaMensaje(INVALIDO, listaPatrones)
  fi;
  envia(SB, msg)
end
```

4.4.3. Toma de decisiones en las áreas visuales para la identificación

Las área visuales para la identificación son las encargadas de tomar la decisión de elegir a que clase pertenece el objeto. El proceso que sigue solo consiste en seleccionar la clase que cuente con el mayor grado de certidumbre como se muestra en la Figura 4.25.

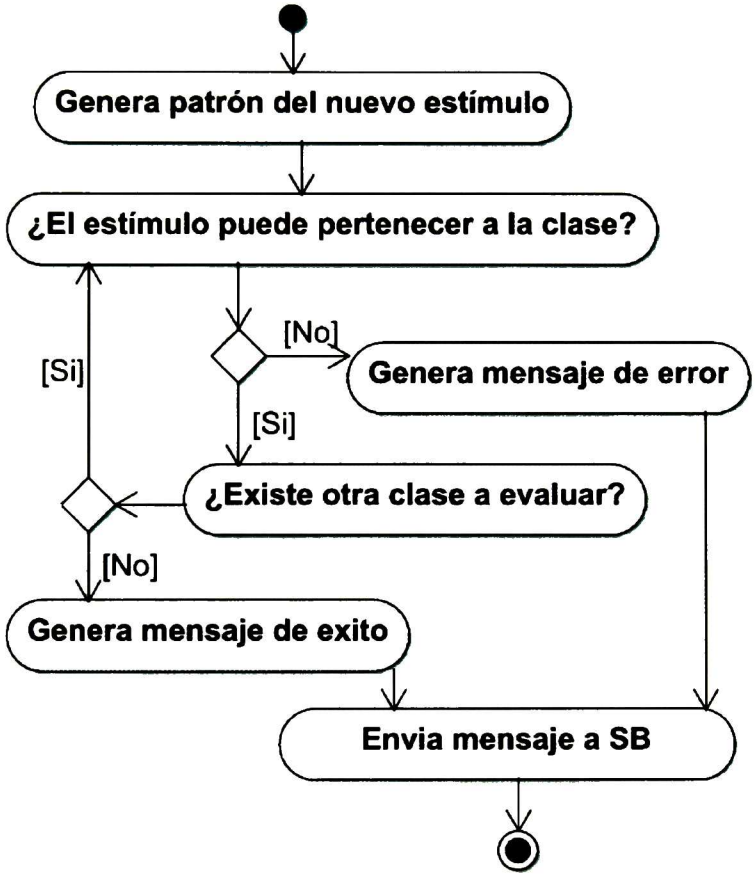


Figura 4.24: **Comparador de Patrones**

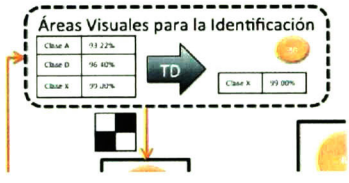


Figura 4.25: **Toma de decisiones** Las áreas visuales para la identificación son las encargadas de elegir a que clase pertenece el estímulo procesado.

Capítulo 5

Implementación y Resultados

5.1. Introducción

Para la implementación se utilizó una computadora de escritorio con procesador *Intel Core*^{TM} i7 3.07GHz y 12GBs de memoria RAM, cuenta con dos tarjetas de vídeo nVIDIA GeForce GTX 480 con 1546 MB memoria y 480 CUDA cores. Cuenta con conexión SLI entre las tarjetas de vídeo. La aplicación se ejecuta sobre el sistema operativo Debian GNU/Linux versión wheezy amd64 kernel 2.6.38

La implementación se realizó con el lenguaje de programación Java.

En las siguientes secciones se describe la implementación y resultados del modelo presentado en los capítulos 4 y 5.

Se describen los resultados de dos casos de estudio basados en las secciones anteriores. El primero caso de estudio surge de la pregunta ¿ qué pasa cuando llega un *estímulo conocido* al sistema ? y el segundo caso de estudio surge de la pregunta ¿ qué pasa cuando llega un *estímulo desconocido* en el sistema?.

Para desarrollar estos casos de estudio se tienen algunas restricciones iniciales a considerar:

- Cada estímulo que procesa el sistema contiene solo un objeto a la vez.
    - Cuando se deben de identificar varios objetos a la vez, la activación de las áreas cerebrales es diferente a la propuesta en el modelo, la activación depende del acomodo de los objetos en la escena[37].
    - Cuando se identifican objetos nuevos, son aprendidos (codificados y almacenados) uno a la vez para poder recuperarlos después.
- Los estímulos utilizados cumplen con las características de los objetos utilizados en la arquitectura para la identificación [35].

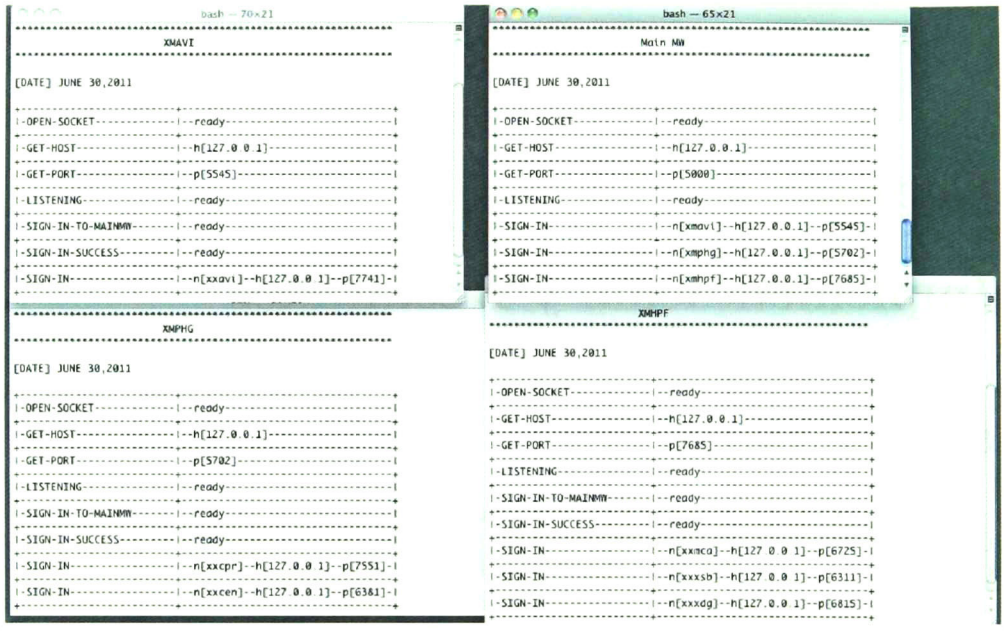


Figura 5.1: **Inicio del sistema** Pantallas tomadas del inicio del sistema, donde se ve la comunicación que existe entre los MW y los nodos al inicio de la comunicación.

- Consisten en un arreglo de unos y ceros, que determinan las orillas de los objetos y la presencia o ausencia de un objeto respectivamente.
- No existe limite en el tamaño del objeto.

La comunicación es a través de la red, por medio de paquetes cuyos encabezados contienen un código de operación que indica la operación a realizar con la información contenida en el resto del mensaje.

- La simulación comienza cuando todos los nodos fueron levantados y registrados (Figura 5.1).
  - Cada nodo es iniciado (primero MW y luego nodos).
  - Cada nodo se registra en su respectivo MW.
  - Cada nodo solicita la ubicación de los nodos con los que trabajara.
  - Cuando los nodos tienen la dirección del resto de los nodos, le envían un mensaje para entablar comunicación.

CODOP	INFORMACIÓN	TAREA
50	Tipo singular	Enviar y analizar
51	Identificador de clases, patrones y porcentaje de certidumbre	Enviar y analizar
52	Tipo singular con patrones incorrectos	Buscar nuevamente los patrones
53	Lista de identificadores de los clases	Enviar y analizar
54	Tipo singular no encontrado	Enviar
55	Tipo singular a codificar	Empezar a codificar el tipo singular
56	Identificador de la nueva clase y tipo singular	Enviar y almacenar
57	Identificador de la nueva clase	Enviar y almacenar

Tabla 5.1: **Lista de códigos de operaciones** Lista de códigos de operaciones, información enviada y tareas realizadas.

- Estos responden con otro mensaje avisando que ya pueden comunicarse con ellos.

- Los códigos de operación utilizados, indican la información que contiene el paquete y que tarea se debe realizar con ella (Figura 5.1):

Tomando estas restricciones en cuenta, a continuación se describen los dos casos de estudio.

5.2. Caso de Estudio: Estímulo Conocido

Cuando un estímulo es conocido el sistema realiza el proceso de recuperación y regresara las clases a las que puede pertenecer el objeto en cuestión. Para las pruebas de este caso, las clases que pueden representan el estímulo, además de otras, fueron pre-almacenadas en el nodo CA3. La simulación del sistema completo es como sigue:

1. El nodo AVI (áreas visuales para la identificación) envía el tipo singular al nodo CPR (corteza perirrinal). La estructura del mensaje sera como se muestra en la Tabla 5.2.

050	011011

Tabla 5.2: **Paquete TIPOSINGULAR** los tres primeros caracteres representan el código de operación, el resto es el tipo singular en cadena de bits.

051	3	A	93.76%	011***	B	98.77%	****11	C	99.00%	011*11

Tabla 5.3: **Paquete POSIBLESPATRONES** los tres primeros caracteres representan el código de operación, el siguiente caracter dice el numero de clases encontradas, el resto del mensaje contiene el identificador de la clase, el porcentaje de certidumbre y el patrón de disparo para todas las clases encontradas.

2. El nodo CPR recibe un mensaje de AVI, lo interpreta y lo reenvía al nodo CEN (corteza entorrinal).

3. El nodo CEN recibe un mensaje de CPR, lo interpreta y lo reenvía a los nodos CA3 y CA1.

4. CA3 recibe un mensaje del nodo CEN:

    1. Obtienen el tipo singular del mensaje que recibió.
    2. Utilizando el clasificador, obtiene un grupo de clases a las cuales puede pertenecer el objeto contenido en el tipo singular.
    3. Empaqueta las clases (Tabla 5.3) y se las envía al nodo CA1.

5. CA1 recibe un mensaje del nodo CA3:

    1. Obtiene las clases, junto con sus atributos, del mensaje.
    2. Obtiene el tipo singular del mensaje que recibió del nodo CEN.
    3. Compara cada patrón de disparo, enviado por el nodo CA3, con el tipo singular.
        1) Si existen patrones a los que no pueda pertenecer el tipo singular, le envía al nodo CA3 un mensaje (Tabla 5.4) donde le informa que los patrones que genero no son correctos.
        2) En Caso contrarió, envía un mensaje al nodo SB (subículo) con los identificadores de las clases y los porcentaje de certidumbre (Tabla 5.5).

6. El nodo SB envía el conjunto de clases probables al nodo CEN.

7. El nodo CEN envía el conjunto de clases probables al nodo CPR.

052	011011

Tabla 5.4: **Paquete PATRONESINCORRECTOS** los tres primeros caracteres representan el código de operación, después viene el tipo singular para volver a buscar las clases probables.

053	A	93.76	B	98.77	C	99.00

Tabla 5.5: **Paquete POSIBLESCLASES** los tres primeros caracteres representan el código de operación, después vienen el identificador de la clase con el porcentaje de certidumbre.

8. El nodo CPR envía el conjunto de clases probables al nodo AVI.
9. El nodo AVI recibe un mensaje del nodo CPR:
    1. Recupera las clases junto con sus porcentajes de certidumbre.
    2. Selecciona la clase que tiene un porcentaje de certidumbre mas alto.
    3. Muestra en pantalla la clase que selecciono.

5.2.1. Resultados

A continuación se muestran la salida que tiene los nodos del sistema durante la ejecución del sistema.

AVI

La Figura 5.2 muestra los eventos del nodo AVI cuando arriba al sistema un estímulo conocido:

Envía al nodo CPR, el tipo singular 011011 al nodo AVI.

Recibe del nodo CPR, las posibles clases: {(A:93.76 %), (B:98.77 %), (C:99.00 %)}.

Selecciona la clase A, porque tiene el porcentaje 99.00 %, y la muestra.

El resultado fue el esperado en el nodo AVI por las siguientes razones: fue capaz de enviar el estímulo inicial para comenzar la ejecución y selecciono correctamente la clase con el porcentaje de certidumbre mayor.

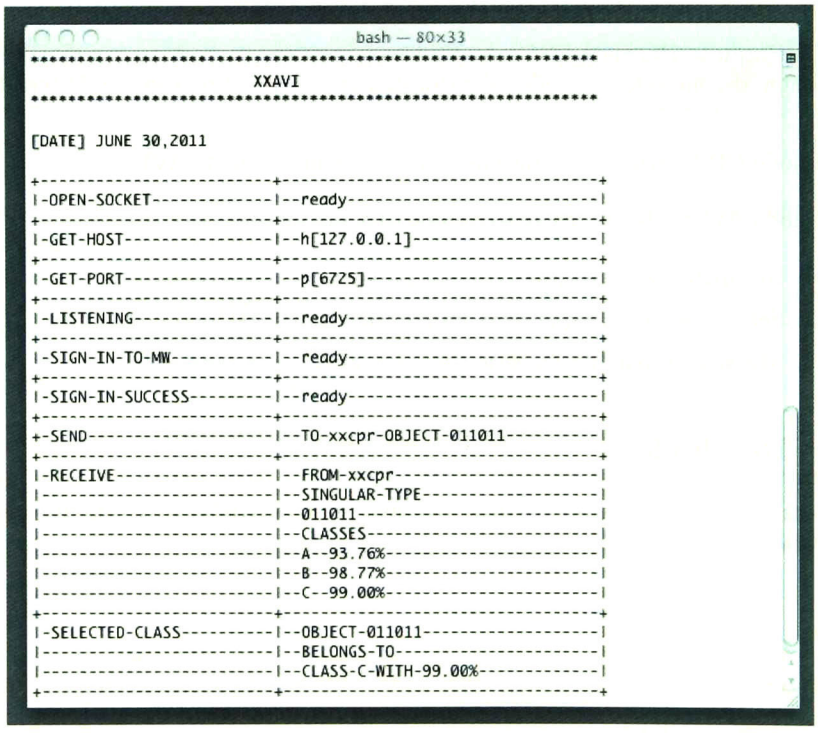


Figura 5.2: **Nodo AVI** Resultados obtenidos en el nodo AVI durante el análisis de un estímulo conocido.

CPR

La Figura 5.3 muestra los eventos del nodo CPR cuando arriba al sistema un estímulo conocido:

- Recibe del nodo AVI, tipo singular 011011.

  Envía al nodo CEN, tipo singular 011011.

  Recibe del nodo CEN, la lista de posibles clases:
  {(A:93.76 %), (B:98.77 %), (C:99.00 %)}.

  Envía al nodo AVI, las clase junto con su porcentaje de certidumbre.

El resultado fue el esperado en el nodo CPR por las siguientes razones: recibió e interpreto todos los mensajes que le llegaron, además de enviarles el tipo singular al nodo CEN y las posibles clases al nodo AVI.

CEN

La Figura 5.4 muestra los eventos del nodo CEN cuando arriba al sistema un estímulo conocido:

- Recibe del nodo CPR, el tipo singular 011011.

  Envía al nodo CA3, el tipo 011011.

  Envía al nodo CA1, el tipo 011011.

  Recibe del nodo SB, la lista de las posibles clases junto con el porcentaje de certidumbre.

- Envía al nodo CPR, la lista de las posibles clases junto con el porcentaje de certidumbre.

El resultado fue el esperado en el nodo CEN por las siguientes razones: recibió e interpreto todos los mensajes que le llegaron; envió el tipo singular a CA3 y a CA1 como debía y envió la lista de posibles clases al nodo CPR.

CA3

La Figura 5.5 muestra los eventos del nodo CA3 cuando arriba al sistema un estímulo conocido:

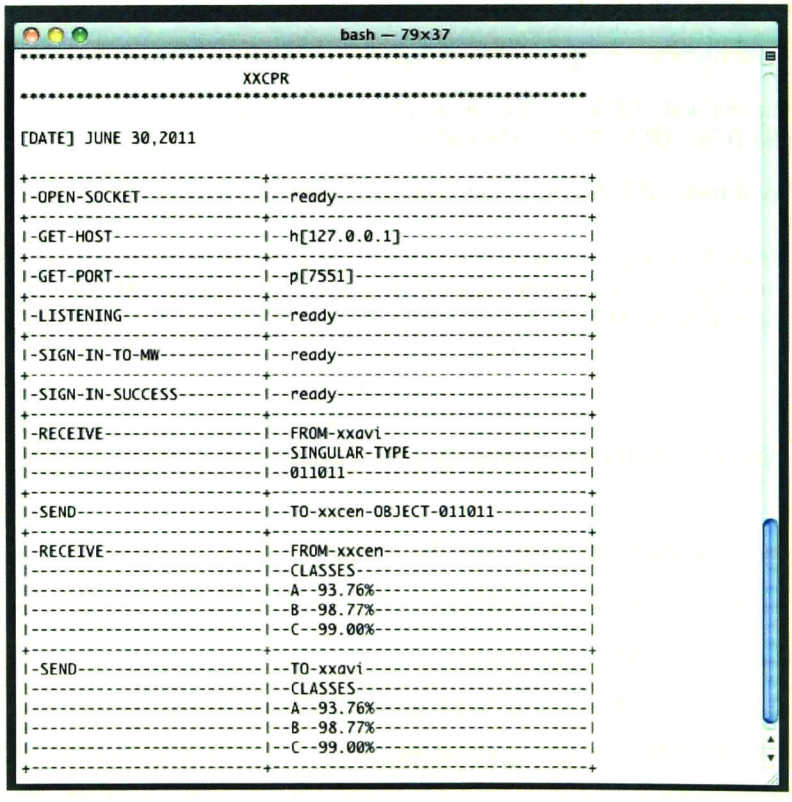


Figura 5.3: **Nodo CPR** Resultados obtenidos en el nodo CPR durante el análisis de un estímulo conocido.

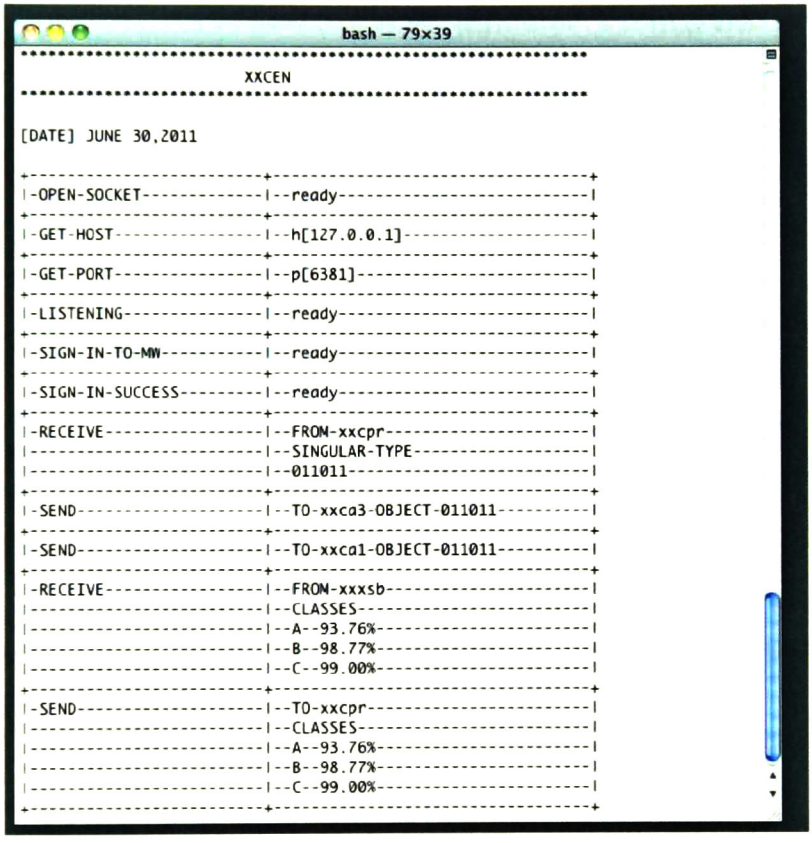


Figura 5.4: **Nodo CEN** Resultados obtenidos en el nodo CEN durante el análisis de un estímulo conocido.

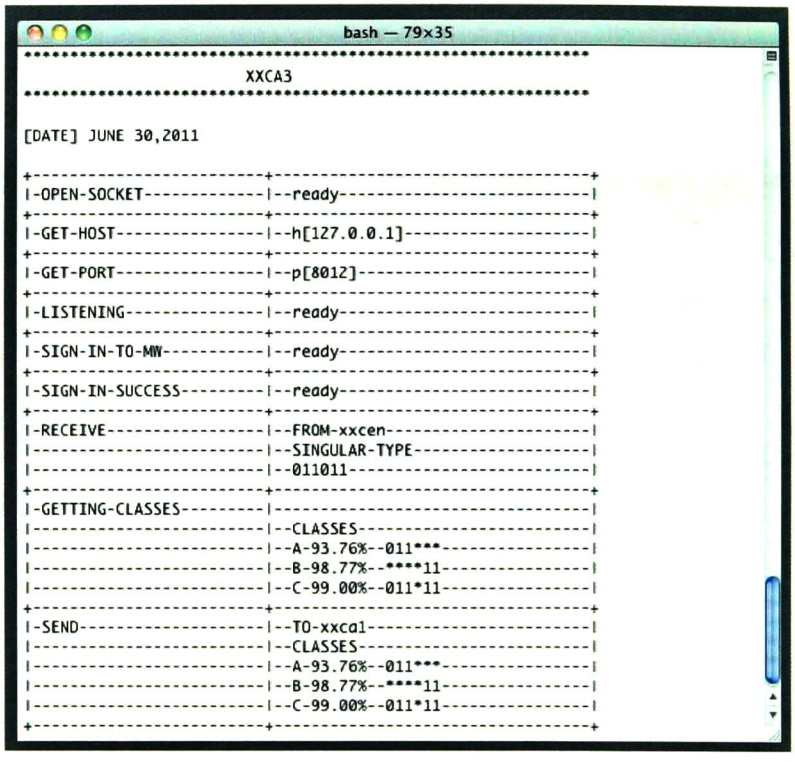


Figura 5.5: **Nodo CA3** Resultados obtenidos en el nodo CA3 durante el análisis de un estímulo conocido.

- Recibe del nodo CEN, el tipo singular 011011.

- Obtiene las clases y los patrones a las que puede pertenecer el tipo singular: {(A:93.76 %:011***), (B:98.77 %:****11), (C:99.00 %:011*11)}.

- Envía al nodo CA1, las clases y los patrones.

El resultado fue el esperado en el nodo CA3: obtuvo el tipo encontró la lista de posibles clases de un objeto, las clases que obtuvo son correctas de acuerdo a lo que esperábamos.

CA1

La Figura 5.6 muestra los eventos del nodo CA1 cuando arriba al sistema un estímulo conocido:

Recibe del nodo CA3, las clases posibles del tipo singular, el porcentaje de certidumbre y los patrones:
{(A:93.76 %:011***), (B:98.77 %:****11), (C:99.00 %:011*11)}.

Recibe del nodo CEN, el tipo singular 011011.

Compara las clases enviadas por CA3 y el tipo singular enviado por el nodo CEN.

- Envía al nodo SB, las clases junto con los porcentajes de certidumbres:
  {(A:93.76 %), (B:98.77 %), (C:99.00 %)}.

El resultado fue el esperado en el nodo CA1 por la siguiente razón: CA1 comparo los patrones que recibió de CA3 con el tipo singular que recibió de CEN y el resultado fue como lo esperábamos.

SB

La Figura 5.7 muestra los eventos del nodo SB cuando arriba al sistema un estímulo conocido:

- Recibe del nodo CA1, las posibles clases a las que puede pertenecer el objeto junto con sus porcentajes de certidumbre:
  {(A:93.76 %), (B:98.77 %), (C:99.00 %)}.

El resultado fue el esperado en el nodo SB por la siguiente razón: envió todos los mensajes en el orden que se esperaba y con la información adecuada.

5.3. Caso de Estudio: Estímulo Desconocido

Cuando un estímulo es desconocido, el proceso de recuperación falla entonces el proceso de codificación y almacenamiento comienza. Para las pruebas de este caso de estudio, la clase a la que puede pertenecer el estímulo no existe, tampoco existen clases parecidas en el nodo CA3. La simulación es como sigue:

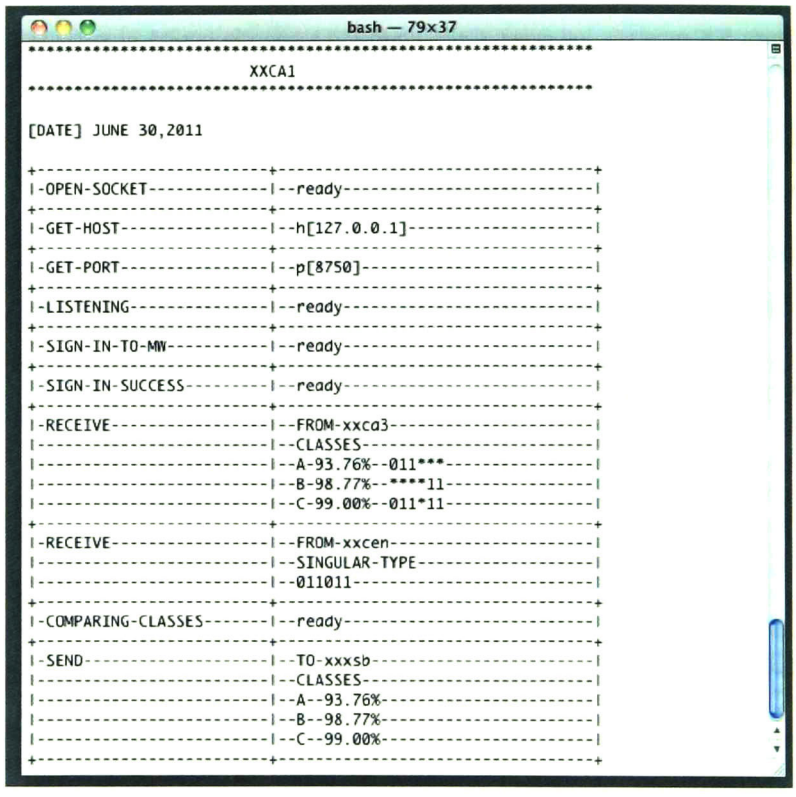


Figura 5.6: **Nodo CA1** Resultados obtenidos en el nodo CA1 durante el análisis de un estímulo conocido.

050	111111

Tabla 5.6: **Paquete TIPOSINGULAR** Paquete utilizado en el caso de estudio Estímulo Desconocido

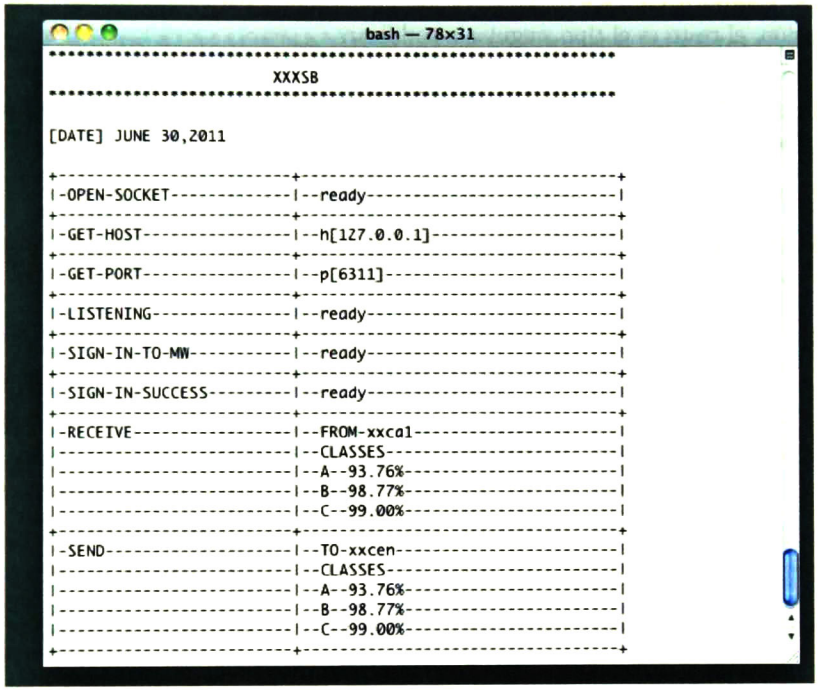


Figura 5.7: **Nodo SB** Resultados obtenidos en el nodo SB durante el análisis de un estímulo conocido.

054	111111

Tabla 5.7: **Paquete NOENCONTRADO** los tres primeros caracteres representan el código de operación, el resto es el tipo singular no encontrado en el nodo CA3

055	111111

Tabla 5.8: **Paquete CODIFICACION** los tres primeros caracteres representan el código de operación, el resto es el tipo singular a codificar.

1. El nodo AVI envía el tipo singular al nodo CPR. La estructura del mensaje sera como el que se muestra en la Tabla 5.6.
2. El nodo CPR recibe el mensaje de AVI, lo interpreta y lo reenvía al nodo CEN.
3. El nodo CEN recibe el mensaje de CPR, lo interpreta y lo reenvía a los nodos CA3 y CA1.
4. CA3 recibe el mensaje del nodo CEN
    1. Obtienen el tipo singular del mensaje que recibió.
    2. Utilizando el clasificador no encuentra alguna clase a la que pueda pertenecer el tipo singular.
    3. Informa al nodo CA1 que no encontró alguna clase a la que pueda pertenecer el tipo singular, como se observa en la Tabla 5.7.
5. El nodo CA1 envía el mensaje de NOENCONTRADO al nodo SB.
6. El nodo SB envía el mensaje de NOENCONTRADO al nodo CEN.
7. El nodo CEN envía el mensaje de NOENCONTRADO al nodo CPR.
8. En el nodo CPR:
    1. Recibe el mensaje del nodo CEN.
    2. Identifica que no encontraron al tipo singular que envío.
    3. Genera un mensaje para comenzar el proceso de CODIFICACION.
    4. Le envía el mensaje de CODIFICACION al nodo CEN, como se ve en la Tabla 5.8.
9. El nodo CEN envía el mensaje de CODIFICACION al nodo GR (giro dentado).

056	A	111111

Tabla 5.9: **Paquete NUEVACLASE** los tres primeros caracteres representan el código de operación, después esta el identificador de la nueva clase, el mensaje termina con el nuevo patrón generado.

057	A

Tabla 5.10: **Paquete ALMACENAMIENTO** los tres primeros caracteres representan el código de operación, después esta el identificador de la nueva clase.

10. En el nodo GR:
    1. El nodo GR genera un nuevo patrón de disparo en base al tipo singular que recibió.
    2. El nodo GR envía el nuevo patrón de disparo al nodo CA3, como se muestra en la Tabla 5.9.
11. En el nodo CA3:
    1. Se almacena el nuevo patrón.
    2. Le envía un mensaje al nodo CA1, el mensaje contiene una señal para comenzar el proceso de almacenamiento y el identificador de la nueva clase. Como se ve en la Tabla 5.10.
12. El nodo CA1 envía el mensaje para comenzar el proceso de ALMACENAMIENTO al nodo SB.
13. El nodo SB envía el mensaje para comenzar el proceso de ALMACENAMIENTO al nodo CEN.
14. El nodo CEN envía el mensaje para comenzar el proceso de ALMACENAMIENTO al nodo CPR.
15. El nodo CPR envía el mensaje para comenzar el proceso de ALMACENAMIENTO al nodo AVI y envía el mensaje con el tipo singular al nodo CEN para comenzar el proceso de recuperación.
16. El nodo AVI almacena el identificador de la nueva clase.
17. El nodo CEN recibe el mensaje de CPR, lo interpreta y lo reenvía a los nodos CA3 y CA1.

18. CA3 recibe el mensaje de CEN,

    1. Obtienen el tipo singular del mensaje que recibió.
    2. Utilizando el clasificador, obtiene un grupo de clases a las cuales puede pertenecer el objeto contenido en el tipo singular.
    3. Empaqueta las clases y se las envía al nodo CA1.

19. En el nodo CA1 recibe el mensaje de CA3:

    1. Como el mensaje le indica que le envía las posibles clases.
    2. Obtiene las clases, junto con sus atributos, del mensaje.
    3. Obtiene el tipo singular que venia en el mensaje que ya había recibido de CEN.
    4. Compara cada patrón de disparo con el tipo singular.
        1) Si existen patrones a los que no pueda pertenecer el tipo singular, le envía a CA3 un mensaje donde le informa que los patrones que genero no son correctos.
        2) En Caso contrarió, envía un mensaje al nodo SB con los identificadores de las clases con el porcentaje de certidumbre.

20. El nodo SB envía el conjunto de clases probables al nodo CEN.

21. El nodo CEN envía el conjunto de clases probables al nodo CPR.

22. El nodo CPR envía el conjunto de clases probables al nodo AVI.

23. En el nodo AVI:

    1. Recupera las clases junto con sus porcentajes de certidumbre.
    2. El nodo AVI selecciona la clase que tiene un porcentaje de certidumbre mas alto.
    3. AVI muestra en pantalla la clase que selecciono.

5.3.1. Resultados

A continuación se muestra la salida que tiene los nodos del sistema durante la ejecución del sistema.

AVI

La Figura 5.8 muestra los eventos del nodo AVI cuando arriba al sistema un estímulo desconocido.

Envía al nodo CPR, el tipo singular 111111.

Recibe del nodo CPR, el identificador de la nueva clase A.

Almacena la nueva clase A.

- Recibe del nodo CPR, la clase a la que puede pertenecer el estímulo: {(A:100.00 %)}.

Selecciona a la clase A con el 100.00 % de certidumbre.

Los resultados fueron los esperados en el nodo AVI por las siguientes razones: envió el tipo singular, almaceno el identificador de la nueva clase y selecciono la clase con mayor porcentaje de certidumbre correctamente.

CPR

La Figura 5.9 muestra los eventos del nodo CPR cuando arriba al sistema un estímulo desconocido.

Recibe del nodo AVI, el tipo singular 111111.

- Envía al nodo CEN, el tipo singular 111111.

Recibe del nodo CEN, el tipo singular no encontrado 111111.

Envía al nodo CEN, la señal para comenzar a codificar el tipo singular 111111.

Recibe del nodo CEN, la señal para comenzar a almacenar la clase A.

- Envía al nodo AVI, la señal para almacenar la clase A - 111111.

Envía al nodo CEN, el tipo singular 111111.

Recibe del nodo CEN, la clase a la que posiblemente pertenece el tipo singular 111111.

- Envía al nodo AVi, la clase a la que posiblemente pertenece el tipo singular 111111. {(A:100.00 %:111111)}.

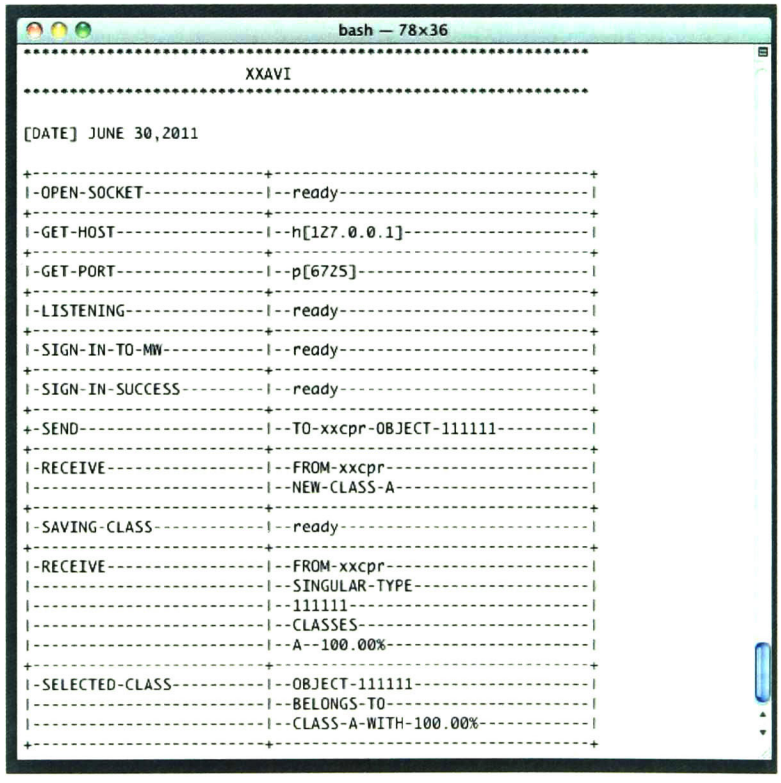


Figura 5.8: **Nodo AVI** Resultados obtenidos en el nodo AVI durante el análisis de un estímulo desconocido.

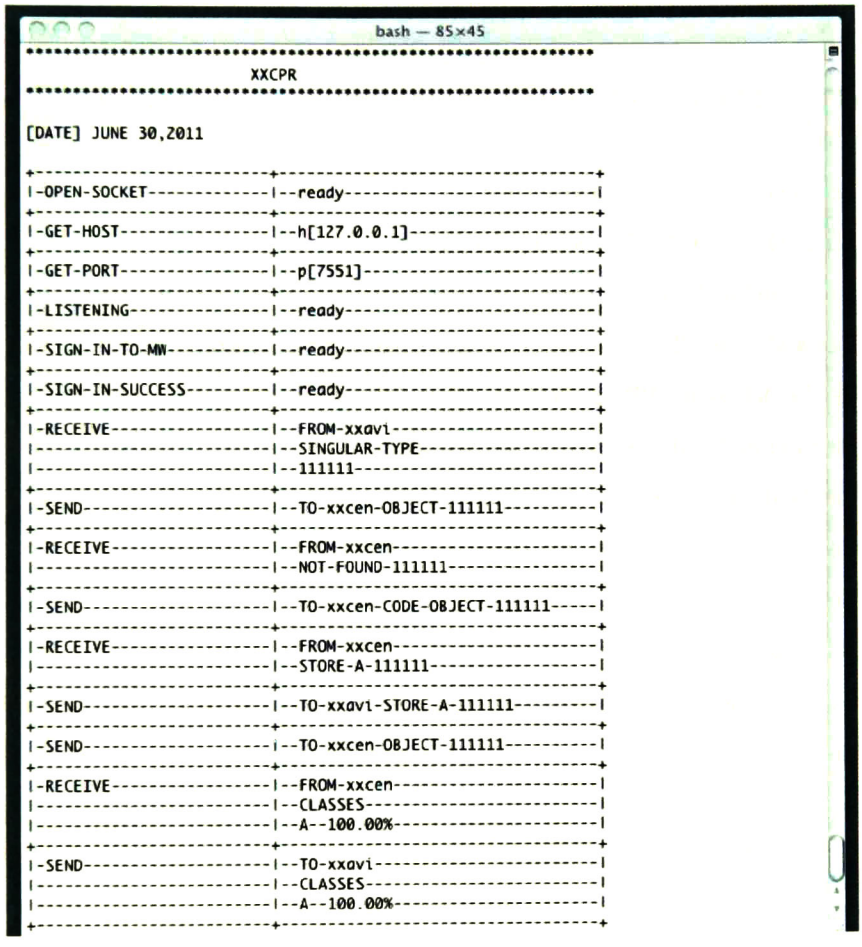


Figura 5.9: **Nodo CPR** Resultados obtenidos en el nodo CPR durante el análisis de un estímulo desconocido.

Los resultados fueron los esperados en el nodo CPR por las siguientes razones: envió el tipo singular, envió la señal para comenzar el proceso de codificación/almacenamiento y envió la señal para almacenar la nueva clase.

CEN

La Figura 5.10 muestra los eventos del nodo CEN cuando arriba al sistema un estímulo desconocido.

- Envía al nodo CA3, el tipo singular 111111.

  Envía al nodo CA1, el tipo singular 111111.

  Recibe del nodo SB, el mensaje de tipo singular no encontrado 111111.

  Envía al nodo CPR, el mensaje de tipo singular no encontrado 111111.

  Recibe del nodo CPR, la señal de codificación del tipo singular 111111.

  Envía al nodo GD, la señal de codificación.

  Recibe del nodo SB, la señal de almacenar.

  Recibe del nodo CPR, el tipo singular 111111.

  Envía al nodo CA3, el tipo 111111.

  Envía al nodo CA1, el tipo 111111.

  Recibe del nodo SB, la lista de las posibles clases junto con el porcentaje de certidumbre.

- Envía al nodo CPR, la lista de las posibles clases junto con el porcentaje de certidumbre.

Los resultados fueron los esperados en el nodo CEN por las siguientes razones: envió el tipo singular, envió la señal para comenzar el proceso de codificación/almacenamiento y envió las clases probables al nodo CPR.

CA3

La Figura 5.11 muestra los eventos del nodo CA3 cuando arriba al sistema un estímulo desconocido.

- Recibe del nodo CEN, el tipo singular 111111.

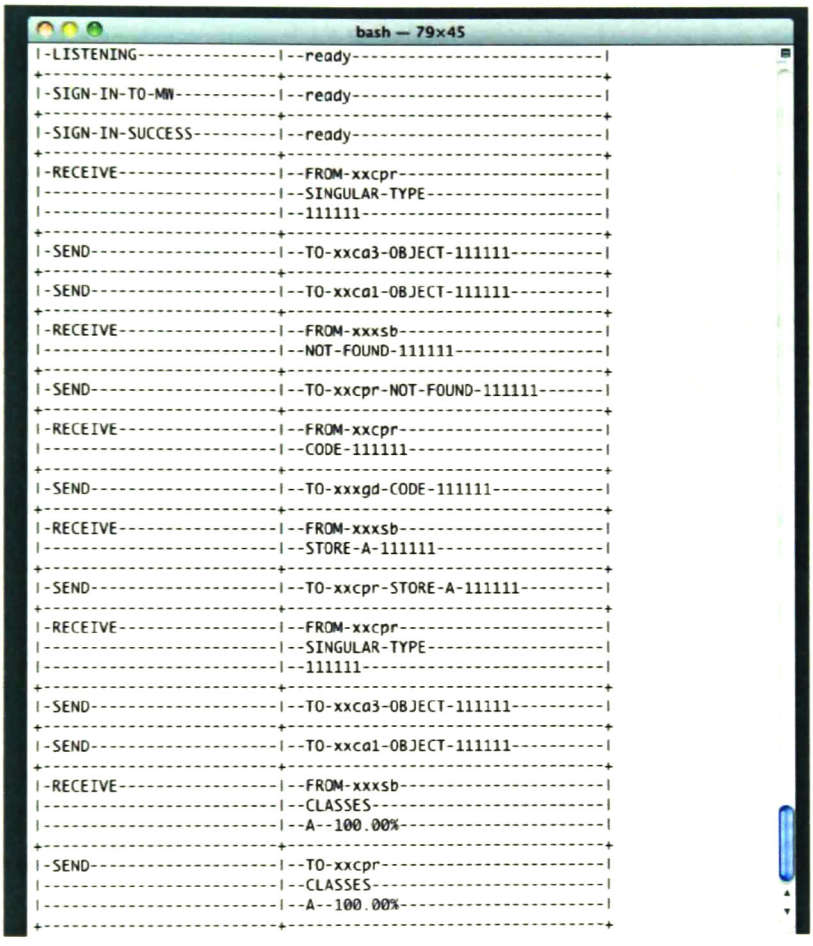


Figura 5.10: **Nodo CEN** Resultados obtenidos en el nodo CEN durante el análisis de un estímulo desconocido.

No puede encontrar alguna clase similar ala del tipo singular.

Envía al nodo CA1, un mensaje informando que no encontró las clases.

Recibe del nodo GD, el identificador de la nueva clase.

Envía al nodo CA1, el identificador de la nueva clase para comenzar a almacenar.

Recibe del nodo CEN, el tipo singular 111111.

Obtiene las clases y los patrones a las que puede pertenecer el tipo singular:
{(A:100.00 %:111111)}.

Envía al nodo CA1, las clases y los patrones.

Los resultados fueron los esperados del nodo CA3 por las siguientes razones: reporto que el tipo singular era desconocido, almaceno el nuevo patrón generado por el nodo GD, cuando recibió el tipo singular encontró el patrón al que pertenece y lo envió a CA1.

CA1

La Figura 5.12 muestra los eventos del nodo CA1 cuando arriaba al sistema un estímulo desconocido.

Recibe del nodo CA3, objeto no encontrado.

Envía al nodo SB, objeto no encontrado.

Recibe de nodo CA3, la nueva clase.
{A:111111}.

Envía al nodo SB, la señal para almacenar la nueva clase.

Recibe del nodo CA3, las clases posibles del tipo singular, el porcentaje de certidumbre y los patrones:
{(A:100.00 %:111111)}.

- Recibe del nodo CEN, el tipo singular 111111.
- Compara las clases enviadas por CA3 y el tipo singular enviado por el nodo CEN.
- Envía al nodo SB, las clases junto con los porcentajes de certidumbres:
{(A:100.00 %)}.

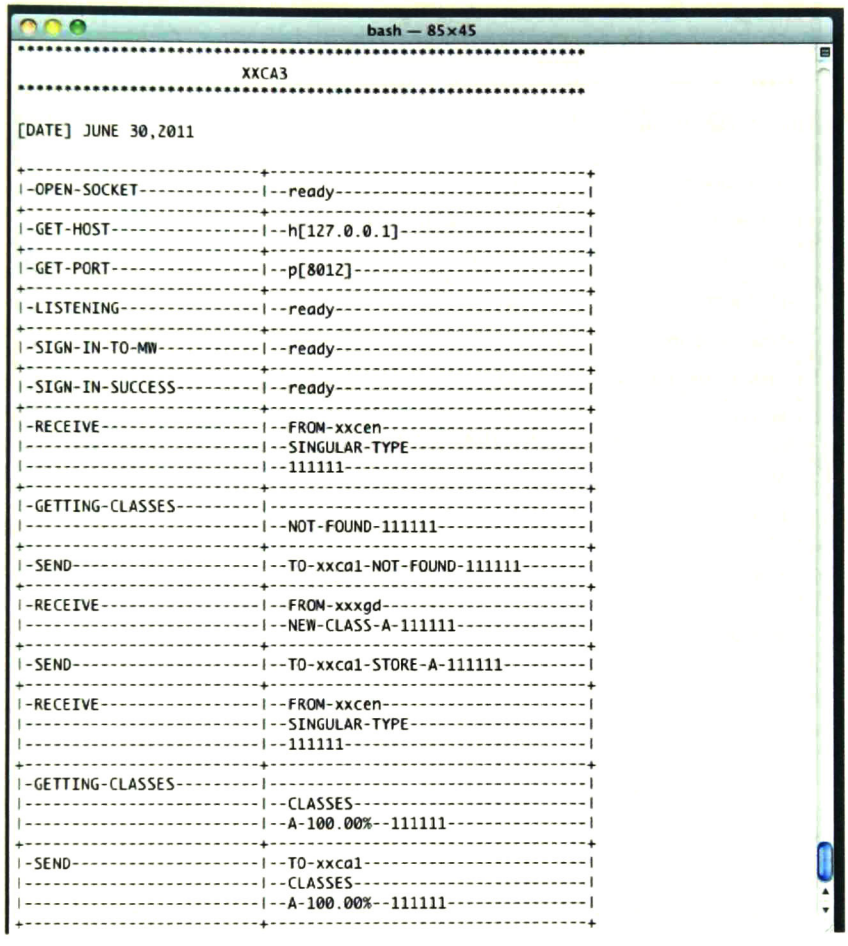


Figura 5.11: **Nodo CA3** Resultados obtenidos en el nodo CA3 durante el análisis de un estímulo desconocido.

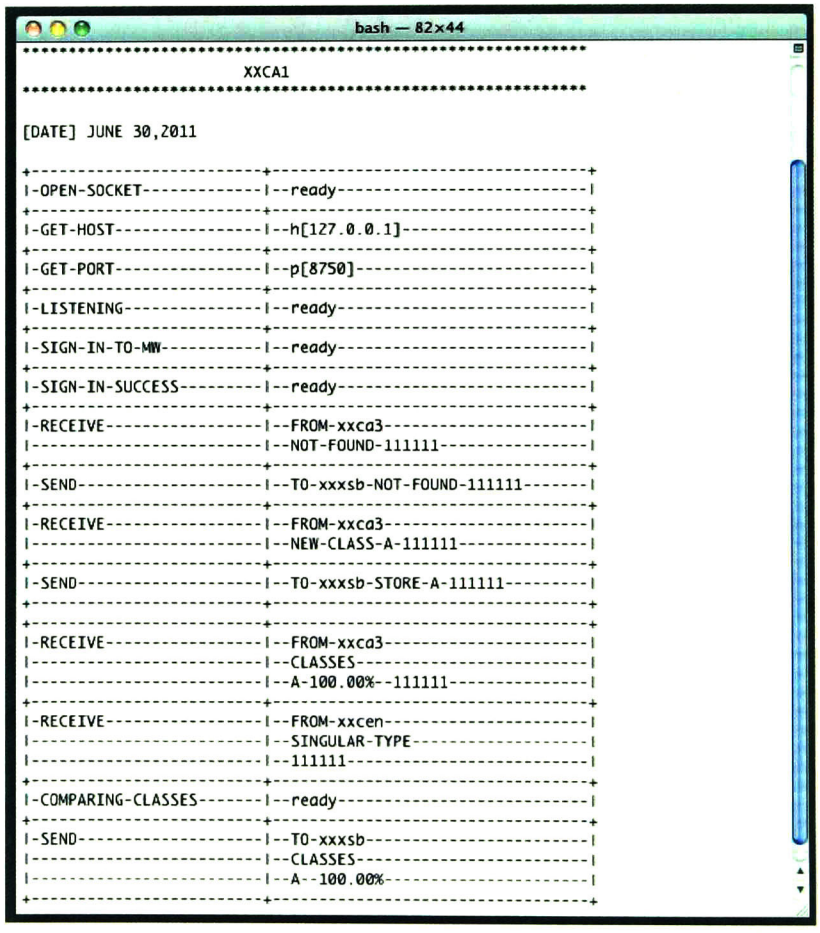


Figura 5.12: **Nodo CA1** Resultados obtenidos en el nodo CA1 durante el análisis de un estímulo desconocido.

Los resultados fueron los esperados en el nodo CA1 por las siguientes razones: envió el mensaje de que no se encontró el tipo singular, comparo el patrón que envió CA3 con el tipo singular de CA3 e identifico que era un patrón probable.

SB

La Figura 5.13 muestra los eventos del nodo SB cuando arriba al sistema un estímulo desconocido.

- Recibe del nodo CA1, objeto no encontrado.
- Envía al nodo CEN, objeto no encontrado.
- Recibe del nodo CA1, la señal para almacenar la clase A 1111.
- Envía al nodo CEN, la señal para almacenar la clase A 111111.
- Recibe del nodo CA1, las lista de clases a la que puede pertenecer el objeto: {(A:100.00 %)}.

  Envía al nodo CEN, las lista de clases a la que puede pertenecer el objeto: {(A:100.00 %)}.

Los resultados fueron los esperados en el nodo SB por las siguiente razones: el nodo SB reenvió los mensajes que le llegaron al nodo CEN en el orden adecuado.

GD

La Figura 5.14 muestra los eventos del nodo GD cuando arriaba al sistema un estímulo desconocido.

- Recibe del nodo CEN, señal para codificar el tipo singular 111111.
- Genera el nuevo patrón.

  Envía al nodo CA3, la nueva clase junto con el patrón de disparo A - 111111.

Los resultados fueron los esperados en el nodo GD por las siguientes razones: genero el patrón de disparo para el clasificador de CA3 en base al tipo singular y lo reenvió al nodo CA3.

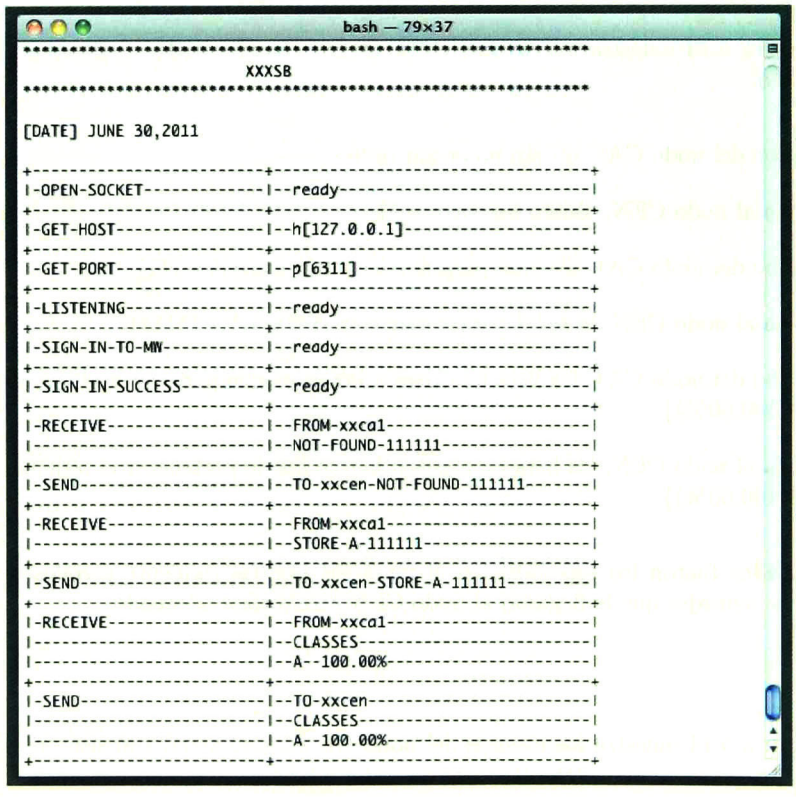


Figura 5.13: **Nodo SB** Resultados obtenidos en el nodo SB durante el análisis de un estímulo desconocido.

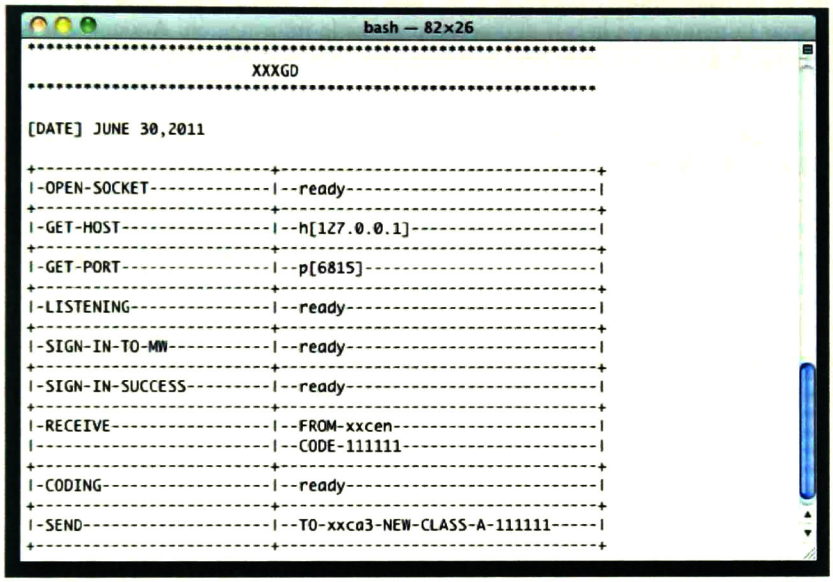


Figura 5.14: **Nodo GD** Resultados obtenidos en el nodo GD durante el análisis de un estímulo desconocido.

5.4. Conclusiones

Cuando un estímulo conocido arriba al sistema este fue capaz de asignarle la clase a la que pertenece. Cada uno de los nodos envío y recibió la información como se esperaba.

Cuando un estímulo desconocido arriba al sistema este fue capaz de reconocer que no conocía el estímulo, mando las señales para codificar, codifico el objeto, lo almaceno y finalmente lo recupero.

Durante la simulación del modelo, se puede observar que existen mensajes de mas, mensajes que se podrían ahorrar o enviar directamente. La razón para dejar estos mensajes es para respetar la manera en la que el cerebro procesa la información. Además estos mensajes, pueden contener mayor información en etapas posteriores del modelo.

Capítulo 6

Trabajo Futuro y Conclusiones

En este documento se expuso un modelo de memoria semántica. A continuación se discuten las conclusiones a las que se llegaron, finalmente se hace un listado del trabajo futuro.

6.1. Conclusiones

En este trabajo se presentan un modelo de memoria semántica basada en los resultados de la neurociencia y un sistema distribuido que implementa nuestro modelo propuesto. En la Tabla 6.1 se observa los logros alcanzados en comparación con las arquitecturas existentes y algunos metas pendientes.

En el caso del modelo de memoria semántica,

- Se logro obtener un modelo que esta integrado por las áreas cerebrales que intervienen en le proceso de memoria semántica.

  Se modelo cada área cerebral

  - Se definieron funciones para cada una de ellas así como la información que intercambian.

  El modelo es capaz de codificar, almacenar y recuperar objetos contenidos en los estímulos que procesa.

En el caso del sistema distribuido,

El sistema permite la comunicación transparente entre procesos ubicados en diferentes plataformas.

El sistema implementado es abierto, es decir, el permite agregar, modificar o eliminar nodos fácilmente, puesto que estos son administrados por el middleware.

Es flexible en cuanto al conocimiento. Es decir, el conocimiento que tiene todos los nodos sobre el resto del sistema puede ser modificado entre una ejecución y otra.

La saturación de los MW es mínima ya que los nodos preguntan por la ubicación de otros solo una vez durante cada ejecución.

Para desarrollar un nuevo nodo, solo se necesita implementar funciones, ya que la estructura que deben tener los nodos para comunicarse con el resto del sistema ya están definidos.

6.2. Trabajo Futuro

El modelo propuesto nos brinda solo una función de la memoria. Para considerar que el modelo de memoria esta concluido esta debe contar con memoria episódica, memoria procedural, memoria de trabajo y hacer uso de la información provista por otros sentidos como el oído o el tacto.

La forma en la que se diseño el sistema donde se simula el modelo permite agregar nuevos módulos y nuevas funciones de manera fácil. Por lo que incrementar el sistema implica solo declarar nuevos servios e iniciar los nuevos nodos.

6.2.1. Incluir otros sentidos

En este punto, la identificación de objetos es en base a la información que recibe de los estímulos visuales. Se considera los siguientes trabajos posteriores para incluir otros sentidos en nuestra arquitectura y emular mejor el comportamiento de nuestro modelo de memoria:

agregar estímulos auditivos

agregar estímulos provenientes del tacto

conjuntar la información provenientes de los tres sentidos

proponer mecanismos de codificación y almacenamiento de tal forma que el sistema sea capaz de reconocer un objeto en base a cualquier tipo de estímulo

SOAR, EPIC y ACT-R	El modelo aquí presentado
La mayor debilidad de estas aproximaciones es el origen de sus modelos. Las tres tienen un fundamento psicológico lo que las aleja de la forma en que lo hace el cerebro humano.	La mayor fortaleza de el modelo aquí presentado, a diferencia de los otros modelos, es el origen de los fundamentos en los que se basa. El modelo esta basado en los resultados obtenidos por la neurociencia. Esto significa que estos resultados no cambiaran en futuras investigaciones neurocientitificas.
EPIC, SOAR y ACT-R cuando reciben un estímulo tratan de asignarle un patrón, si no lo tiene almacenado utilizan herramientas evolutivas combinando los patrones que ya tienen para asignarle un nuevo patrón.	El modelo al recibir un estímulo trata de clasificar lo para asignarle un categoría y un porcentaje de certidumbre, este porcentaje de certidumbre nos dice la probabilidad de que el estímulo pertenezca a dicha categoría.
EPIC utiliza información proveniente de diferentes sentidos que almacena en su memoria y utiliza para conseguir objetivos (Visual y Auditivo).	Solo utiliza información visual.
Representan el conocimiento en base a reglas de inferencia.	Realiza codificaciones del conocimiento adquirido, dependiendo del tipo de información adquirida.
Tienen memoria de trabajo y memoria a largo plazo.	Tiene un modulo de la memoria a largo plazo; memoria semántica.
Almacenan la información de forma centralizada.	La información esta distribuida en diferentes módulos.
Como SOAR, ACT-R se constituye de reglas de producción y "chunks"	Cada etapa del proceso de memoria tiene su propia representación de la información.

Tabla 6.1: **Tabla comparativa** En esta tabla se comparan las arquitecturas antes mencionadas con el modelo propuesto en este documento

6.2.2. Memoria Episódica

Como ya se menciono, la memoria episódica almacena todos los eventos vividos de manera secuencial. Por lo cual es importante agregarla al modelo, para realizar está integración se necesitan realizar las siguientes tareas:

identificar las área cerebrales que intervienen en la memoria episódica

identificar la función de cada una de ellas

proponer y desarrollar algoritmos que codifiquen los eventos

proponer y desarrollar algoritmos capaces de almacenar los eventos con la menor perdida de información y lo mas apegados a como lo haría el cerebro

proponer y desarrollar mecanismos para recuperar los eventos con la menor distorsión

desarrollar nodos que las representen y agregarlos al sistema

6.2.3. Memoria Procedural

Como ya se dijo, la memoria procedural almacena habilidades y hábitos tales como habilidades motoras o respuestas emocionales. Para incluirla en el modelo necesitamos:

identificar las áreas cerebrales que intervienen en la memoria procedural

identificar las funciones de cada una de ellas

- proponer y desarrollar algoritmos que codifiquen la información genera una respuesta emotiva o motora

proponer y desarrollar algoritmos que recuperen las memorias motoras

proponer y desarrollar mecanismos para recuperar información de la memoria procedural con la menor distorsión

- desarrollar nodos que las representen y agregarlos al sistema

6.2.4. Memoria de Trabajo

La memoria de trabajo es un sistema cognitivo que temporalmente mantiene una cantidad limitada de información en un estado activo para que pueda ser utilizada en cuento se necesita, integrada con otra información o modificada [13]. La memoria ayuda a realizar un amplio rango de tareas básicas, como mover objetos, asignar significados y crear relaciones entre objetos [13]. Para conseguir agregarla al modelo debemos:

investigar como diferentes memorias son utilizadas para generar la memoria de trabajo

proponer algoritmos que la generen haciendo uso de toda la memoria

- investigar como la información es almacenada temporalmente en la memoria de trabajo
- identificar las áreas cerebrales que intervienen

desarrolla nodos que las representen y agregarlos al sistema

6.2.5. Aportaciones a la neurociencia

Estamos estudiando como esta arquitectura puede ayudar a los neurocientificos a estudiar daños específicos del cerebro. Esta ayuda puede consistir en:

Probar hipótesis en la arquitectura.

Generar hipótesis.

Bibliografía

[1] Claude Alain, David L Woods, and Robert T Knight. A distributed cortical network for auditory sensory memory in humans. *Brain Research*, 812(1-2):23–37, November 1998.

[2] Akkihebbal L. Ananda and B. Srinvasan. *Distributed Computing Systems: Concepts and Structures*, chapter Definition, Motivation, Concepts of Distributed Systems, pages 1–52. Number 1. IEEE Computer Society Press, 1990.

[3] Per Andersen, Richard Morris, David Amaral, Tim Blis, and John O'Keefe. *The Hippocampus Book*, chapter The Hippocampal Formation, pages 3–8. Number 1. Oxford University Press, 2007.

[4] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglas, C. Lebiere, and Y. Qin. An integrated theory of mind. *Psychological Review*, 111(4):1036–1060, 2004.

[5] R. C. Atkinson and R. M. Shiffrin. *The psychology of learning and motivation*, chapter Human Memory: A Proposed System and Its Control Processes. New York: Academic Press, 1968.

[6] A. D. Baddeley. *The handbook of memory disordes.*, chapter The Psychology of Memory. Number 1. John Wiley and Sons, Inc, 2002.

[7] Alan Baddeley. Working memory: The interface between memory and cognition. *Journal of Cognitive Neuroscience*, 4(3):281–288, 1992.

[8] Malcolm W Brown and John P Aggleton. Recognition memory: what are the roles of the perirhinal cortex and hippocampus? *Nature Reviews Neuroscience*, 2(1):51–61, 2001.

[9] Lee Brownston, Robert Farrell, Elaine Kant, and Nancy Martin. *Programming Expert Systems in OPS5.* Addison Wesley., 1985.

[10] George Coulouris, Jean Dollimore, and Tim Kindberg. *Sistemas Distribuidos Conceptos y Diseño*, chapter Categorización de los Sistemas Distribuidos, pages 1–26. Number 1. PEARSON Educación, 3 edition, 2001.

[11] N. Derbinsky and N. A. Gorsky. Exploring the space of computational memory models. In *Symposium on Human Memory for Artificial Agents*, 2010.

[12] N. Derbinsky and J. E Laird. Extending soar with dissociated symbolic memories. In *Symposium on Human Memory for Artificial Agents*, 2010.

[13] T Drew and E K Vogel. Working memory: Capacity limitations. In Larry R. Squire, editor, *Encyclopedia of Neuroscience.* Academic Press, 2009.

[14] Jan Drugowitsch. *Desing and Analysis of Learning Classifier Systems*, chapter Introduction, pages 1–12. Number 1. Springer-Verlag Berlin Heidelberg, 2008.

[15] Henri M. Duvernoy. *The Human Hippocampus Functional Anatomy, Vascularization and Serial Sections with MRI*, chapter Structure, Functions, and Connections, pages 5–37. Number 3. Springer, 3 edition, 2005.

[16] Real Academia Española. *Diccionario de la Lengua Española.* Real Academia Española, 22 edition, 2001.

[17] Society for Neuroscience. What is neuroscience?, 2011.

[18] ACT-R Research Group. About act-r, Julio 2011.

[19] G W Van Hoesen and A Solodkin. Perirhinal cortex. In Larry R. Squire, editor, *Encyclopedia of Neuroscience.* Academic Press, 2009.

[20] A. Hollingwood and A. J. Luck. *Visual memory systems*, pages 3–9. Visual Memory. Oxford University Press, 2008.

[21] Eric R. Kandel and Irving Kupfermann. *Principles of Neural Science*, chapter Learning and Memory, pages 1228–1247. Number 62. McGraw-Hill, 4 edition, 2000.

[22] Stamations V. Kartalopoulos. *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*, chapter Biological Neural Networks, pages 1–36. Number 1. IEEE Press Understanding Science and Technology Series, 1996.

[23] David E. Kieras and David E Meyer. An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Hum.-Comput. Interact.*, 12:391–438, 1997.

[24] J. E Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[25] J. E Laird, P. S. Rosenbloom, and A. Newell. Chunking in soar: The anatomy of a general learning mechanism. *Machine Learning*, 1(1):11–46, 1986.

[26] P. Langley and J. E. Laird. Cognitive architectures: Research issues and challenges. Technical report, Institute for the Study of Learning and Expertise, Palo Alto, CA, 2002.

[27] John E. Lisman and Anthony A. Grace. The hippocampal-vta loop: Controlling the entry of information into long-term memory. *Neuron*, 46(5):703–713, 2005.

[28] Elisabeth A. Murray and Barry J Richmond. Role of perirhinal cortex in object perception, memory, and associations. 2001.

[29] Allen Newell. *Unified Theories of Cognition.* Harvard University Press, 1990.

[30] Steven Ritter, John R. Andreson, Kenneth R. Koedinger, and Albert Corbert. Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin and Review.*, 14(2):249–255, 2007.

[31] Larry R. Squire. *Memory and Brain*, chapter Divisions of Long-Term Memory, pages 151–174. Number 11. Oxford University Press, 1987.

[32] Larry R. Squire. *Memory and Brain*, chapter Localized and Distributed Memory Storage, pages 56–74. Number 5. Oxford University Press, 1987.

[33] Larry R. Squire. Declarative and nondeclarative memory: Multiple brain systems supporting learning and memory. *Journal of Cognitive Neuroscience*, 4(3):232–243, 1992.

[34] Wendy A. Suzuki, Earl K. Miller, and Robert Desimone. Object and place memory in the macaque entorhinal cortex. *Journal of Neurophysiology*, 78(2):1062–1081, 1997.

[35] Gustavo Torres, Karina Jaime, Felix Ramos, and Gregorio Garcia. Brain architecture for visual object identification. 10th IEEE International Conference on Cognitive Informatics and Cognitive Computing, August 2011.

[36] Alessandro Treves and Edmund T. Rolls. Computatonal analysis of the role of the hippocampus in memory. *HIPPOCAMPUS*, 4(3):374–391, June 1994.

[37] H Wan, J P Aggleton, and M W Brown. Different contributions of the hippocampus and perirhinal cortex to recognition memory. *J. Neurosci*, (18):1142–1148, 1999.

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

"2011, Año del Turismo en México".

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Modelo de Memoria Semántica para Criaturas Virtuales Basado en Neurociencias

del (la) C.

Ana Karina JAIME OLIVER

el día 25 de Agosto de 2011.

Dr. Juan Manuel Ramírez Arredondo
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Juan Luis Del Valle Padilla
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara