

TRABAJO FIN DE GRADO

# Emociones en robots sociales

TRABAJO FIN DE GRADO PARA  
LA OBTENCIÓN DEL TÍTULO DE  
GRADUADO EN INGENIERÍA EN  
TECNOLOGÍAS INDUSTRIALES

JUNIO 2016

**Francisco Manuel Ruzafa Costas**

DIRECTOR DEL TRABAJO FIN DE GRADO:

**Ramón Galán López**  
**Oscar Reinoso García**



# Agradecimientos

---

*A mi familia, porque todo lo que tengo y lo que soy, se lo debo a ellos. El verdadero mérito de haber podido llegar hasta aquí es suyo.*

*A mi tutor Ramón Galán, por haber confiado en mí desde el principio, y por haberme aportado no sólo la oportunidad de trabajar en este proyecto, sino por devolverme la ilusión de seguir aprendiendo.*

La robótica social está avanzando rápidamente debido a la creciente demanda en un gran número de aplicaciones, a pesar de ser un campo de investigación reciente.

Por definición, un robot social es aquel que interactúa y se comunica con las personas siguiendo comportamientos, patrones o normas sociales, aunque dentro de esta definición existe un amplio abanico de niveles desarrollo, desde los sistemas más básicos de interacción hasta complejos sistema de inteligencia artificial.

Dentro de la interacción hombre-robot, uno de los puntos que causa un mayor número de problemas es la emulación de emociones. El desarrollo de soluciones específicas para cada aplicación queda en un segundo plano frente a la creación de un modelo emocional de referencia.

El objetivo principal de este proyecto ha sido el estudio, desarrollo e integración de un modelo emocional de forma que permita mejorar el comportamiento de los robots sociales. El modelo emocional está diseñado para trabajar en robots basados en agentes, y está formado por una serie de tareas que trabajan en paralelo.

Las ventajas que proporciona un modelo de este tipo frente a soluciones específicas son la adaptabilidad del sistema a un campo amplio de robots y su facilidad para ser ampliado en sistemas más complejos, aumentando el número de emociones y estímulos.

Por una parte, se recalculan constantemente los parámetros del modelo en función del estado emocional anterior y de los estímulos recibidos. Mientras tanto, el sistema obtiene una serie de modificadores de la comparación entre los parámetros actuales del modelo con los valores de referencia dados.

Esto habilita una fase de aprendizaje, en la que el sistema trata de alcanzar los objetivos modificando el modelo en base a los parámetros obtenidos en la comparación, lo que mejora la adaptación del robot frente a las interacciones, consiguiendo una imitación de las emociones cada vez más precisa.

**Palabras clave:** robot social, emociones, interacción hombre-robot, modelo emocional, inteligencia artificial

# Índice general

---

<b>1 INTRODUCCIÓN</b> .....	<b>1</b>
1.1. PREÁMBULO .....	1
1.2. OBJETIVOS DEL PROYECTO .....	3
1.3. MARCO DEL PROYECTO .....	3
1.4. ESTRUCTURA DEL TRABAJO .....	4
<b>2 ESTADO DEL ARTE</b> .....	<b>5</b>
2.1. ROBOTS SOCIALES .....	5
2.1.1. <i>Definición</i> .....	5
2.1.2. <i>Antecedentes</i> .....	5
2.1.3. <i>Morfología</i> .....	6
2.1.4. <i>Comportamiento</i> .....	8
2.1.5. <i>Emociones</i> .....	9
2.2. APLICACIONES DE LOS ROBOTS SOCIALES .....	11
2.2.1. <i>Robots de apoyo</i> .....	11
2.2.2. <i>Robots para la educación</i> .....	12
2.2.3. <i>Robots sociales con fines lúdicos</i> .....	13
2.3. ROBOTS SOCIALES UPM .....	14
2.3.1. <i>URBANO</i> .....	14
2.3.2. <i>Doris</i> .....	16
2.4. OTROS ROBOTS SOCIALES .....	17
2.4.1. <i>Maggie</i> .....	17
2.4.2. <i>REEM</i> .....	18
2.4.3. <i>Nao</i> .....	19
2.4.4. <i>ASIMO</i> .....	20
2.4.5. <i>Geminoid</i> .....	21

<b>3 DISEÑO Y DESARROLLO DEL PROGRAMA .....</b>	<b>22</b>
3.1. INTRODUCCIÓN .....	22
3.2. HERRAMIENTAS UTILIZADAS.....	22
3.2.1. <i>Visual Studio</i> .....	22
3.2.2. <i>Git</i> .....	23
3.3. METODOLOGÍA .....	24
3.3.1. <i>Espacio de estados</i> .....	24
3.3.2. <i>Multithreading</i> .....	26
3.3.3. <i>Clases</i> .....	29
<b>4 VERSIÓN ACTUALIZADA DEL PROGRAMA .....</b>	<b>30</b>
4.1 DESCRIPCIÓN DE CAMBIOS.....	30
<b>5 COMUNICACIÓN E INTEGRACIÓN.....</b>	<b>34</b>
5.1. INTRODUCCIÓN A LOS AGENTES .....	34
5.2. COMUNICACIÓN E INTEGRACIÓN DE AGENTES .....	38
<b>6 ENSAYOS Y RESULTADOS .....</b>	<b>42</b>
6.1. ENSAYOS REALIZADOS.....	42
6.2. RESULTADOS OBTENIDOS .....	43
<b>7 CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>45</b>
7.1 CONCLUSIONES.....	45
7.2 LÍNEAS FUTURAS .....	46
<b>8 PLANIFICACIÓN TEMPORAL Y PRESUPUESTO .....</b>	<b>49</b>
8.1. ESTRUCTURA DE DESCOMPOSICIÓN DEL PROYECTO (EDP).....	49
8.2. PLANIFICACIÓN TEMPORAL .....	51
8.3. PRESUPUESTO .....	53
<b>BIBLIOGRAFÍA .....</b>	<b>54</b>

# 1 INTRODUCCIÓN

---

Este capítulo consiste en una introducción a la robótica social , su campo de aplicación y las motivaciones y objetivos del proyecto, junto con el marco de el mismo, así como la estructura en la que ha sido basada la redacción del documento.

## 1.1. PREÁMBULO

---

La robótica consiste en el diseño y construcción de robots y aparatos que realizan operaciones o trabajos, generalmente en instalaciones industriales y en sustitución de la mano de obra humana.

Durante las últimas décadas, la robótica se ha integrado rápidamente en todos los campos del desarrollo humano, haciendo necesaria una evolución hacia sistemas cada vez más complejos capaces de satisfacer las necesidades del mundo actual.

Un claro ejemplo de esta evolución son los robots sociales (ver sección 2.1, que interactúan y se comunican con las personas siguiendo comportamientos, patrones y normas sociales.

Para facilitar las interacciones, además, el aspecto debe ser amigable, generalmente humanoide (Figura 1), ya que aunque en la actualidad se han desarrollado una gran variedad de robots sociales (ver secciones 2.3 y 2.4), este punto es común a ellos.

A pesar de que la robótica social es un campo reciente de investigación, muchos centros de investigación están haciendo grandes esfuerzos en conseguir avances en robótica social solucionando los requerimientos para este tipo de robots, más allá de dotarles de un aspecto humanoide.

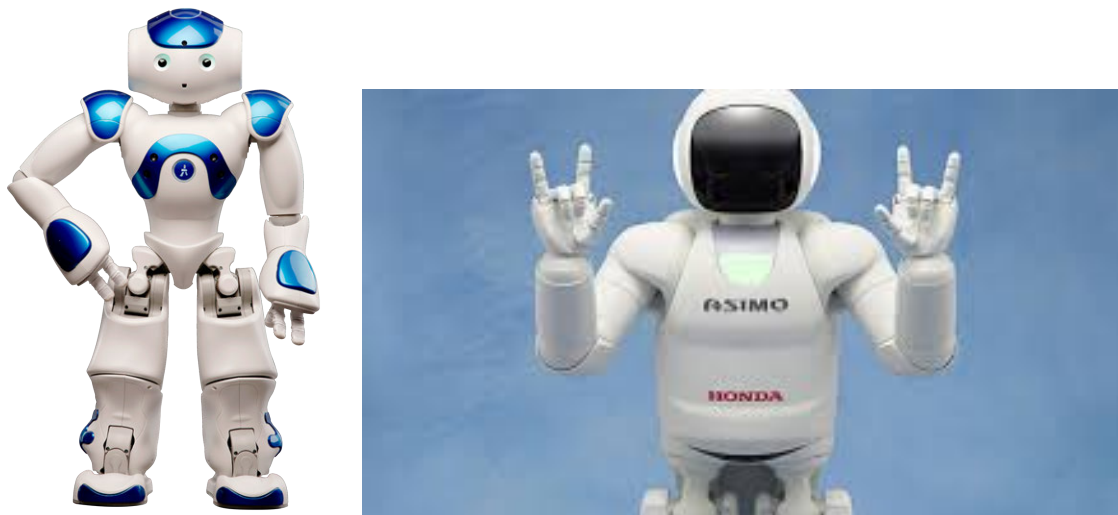


Figura 1 : Robots humanoides NAO (Aldebaran Robotics) y ASIMO (Honda)

Estas necesidades son las que marcan las líneas a seguir, entre las que se encuentra la demanda de robots capaces de mantener interacciones sociales, no sólo observando e interpretando el comportamiento humano, sino adaptándose a él de forma flexible y natural.

De esta adaptación nace la motivación de este proyecto, tratando de emular la forma en la que los estímulos afectan a las emociones, y las emociones al comportamiento, para mejorar el comportamiento de los robots frente a interacciones con humanos

Dentro de esta emulación se incluye el aprendizaje social, averiguando que acciones conducen a la obtención de estímulos que generen emociones positivas, y asignando el valor indicado a los estímulos para mantener los niveles emocionales deseados.

La solución es un modelo emocional capaz de mejorar la interacción entre humanos y máquinas, y que integrado junto a agentes ya existentes de comunicación, visión y movimiento, forme un sistema multidisciplinar completo, el robot social.

Entre las numerosas fuentes bibliográficas citadas a lo largo del trabajo, cabe destacar la tesis *“A Reference Model for Emotion in Artificial Systems”* (María Guadalupe Sánchez-Escribano, 2016), de la que se han obtenido desarrollos teóricos y conceptos fundamentales en el desarrollo del modelo, como el concepto de *universo emocional*.

## 1.2. OBJETIVOS DEL PROYECTO

---

El proyecto persigue el desarrollo de un modelo emocional que permite mejorar la interacción hombre-robot. Para ello, el modelo debe ser además, adaptable y ampliable.

El objetivo global se divide en otros objetivos concretos:

- Obtención, interpretación y tratamiento de los estímulos.
- Desarrollo de relaciones entre estímulos, emociones y acciones mediante el modelo de espacio de estados.
- Diseño e implementación de parámetros globales que actúen como modificadores lógicos del sistema completo basados en la comparación de las emociones con valores de referencia.
- Diseño e implementación de un protocolo comunicación que permita la integración con el resto de agentes que componen el sistema del robot social.
- Realización de ensayos de simulación para ajustar los parámetros de referencia a valores razonables que emulen un comportamiento real.

La obtención de los objetivos sigue un orden secuencial pero sin perder de vista el objetivo final de la integración en el robot para la mejora de sus habilidades de interacción social, trabajando el programa en conjunto con el resto de agentes que ya formaban parte del robot.

El objetivo global del proyecto es conseguir un robot social que interactúe con las personas, con un modelo de emociones robóticas que permita en un futuro seguir mejorándose para conseguir robots sociales más complejos.

## 1.3. MARCO DEL PROYECTO

---

Todo el trabajo se ha realizado dentro del grupo de control inteligente del departamento de automática, ingeniería electrónica e informática industrial del CAR (Centro de Automática y Robótica) en la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid.

El trabajo está estrechamente relacionado con el proyecto URBANO (ver sección 2.3.1), un robot social basado en agentes que encaja con el modelo emocional desarrollado.



## 1.4. ESTRUCTURA DEL TRABAJO

---

El documento está formado por el índice, ocho capítulos que serán resumidos a continuación, y la bibliografía.

- El capítulo 1, que sirve como introducción a la robótica social y al campo de desarrollo del proyecto, además de planteamiento de los objetivos perseguidos y la estructura del trabajo.
- En el capítulo 2 se describe el estado del arte en los robots sociales, ejemplificando las descripciones con proyectos de ámbito nacional e internacional, entre los que se incluyen los de la Universidad Politécnica de Madrid
- En el capítulo 3 se describen las características del modelo, desde la fase de diseño a la de integración, incluyendo las explicaciones que tiene cada uno de los elementos del programa
- El capítulo 4 enumera las características de la versión actual del programa respecto al diseño inicial, dejando lugar a futuras ampliaciones
- El capítulo 5 describe la comunicación entre el programa y el resto de agentes que constituyen el robot, y la integración con los mismos
- El capítulo 6 es una descripción de los resultados obtenidos, mediante los ensayos y simulaciones realizados a lo largo de todo el desarrollo del proyecto, y su comparación con los resultados esperados
- En el capítulo 7 se detallan las conclusiones obtenidas en base al trabajo y los resultados, así como las futuras ampliaciones.
- El capítulo 8 muestra la planificación temporal y el presupuesto sobre los que se ha desarrollado el proyecto

# 2 ESTADO DEL ARTE

---

En este capítulo se expone el estado del arte en robótica social, desde la definición de los mismos a una serie de ejemplos para exponer por completo cuales son las posibilidades, desde un enfoque práctico en sus aplicaciones.

Los modelos emocionales implicados en el comportamiento de los robots serán brevemente expuestos, más por su utilidad que su funcionamiento, ya que serán ampliados más tarde en el tercer capítulo.

También se hace una breve explicación de los proyectos del departamento de automática que tienen relación con el trabajo realizado.

## 2.1. Robots sociales

---

### 2.1.1. Definición

---

Un robot social es aquel que interactúa y se comunica con las personas siguiendo comportamientos, patrones y normas sociales. La robótica social se encarga de estudiar la integración de los robots en la sociedad.

### 2.1.2. Antecedentes

---

Las interacciones entre robots y con su entorno han sido uno de los campos de investigación que más han llamado la atención desde el principio de creación de vida artificial. Los investigadores decidieron aplicar el concepto de estigmergia, concepto que define la comunicación indirecta entre individuos mediante modificaciones en un entorno común [1].

En los años 90, Deneubourg y su equipo realizaron los primeros estudios en robots basados en la estigmergia [2] [3], dejando el camino abierto a nuevos investigadores para realizar estudios acerca del trabajo colectivo de los robots [4] [5].

El trabajo colectivo, da lugar a un comportamiento social, ya que permite a los robots realizar trabajos que serían imposibles de forma individual, por lo que los primeros estudios acerca de las relaciones y comportamientos sociales, constituyen un nexo entre el trabajo colectivo indirecto de robots (Figura 2) y los primeros robots sociales [6] [7] (Figura 3).

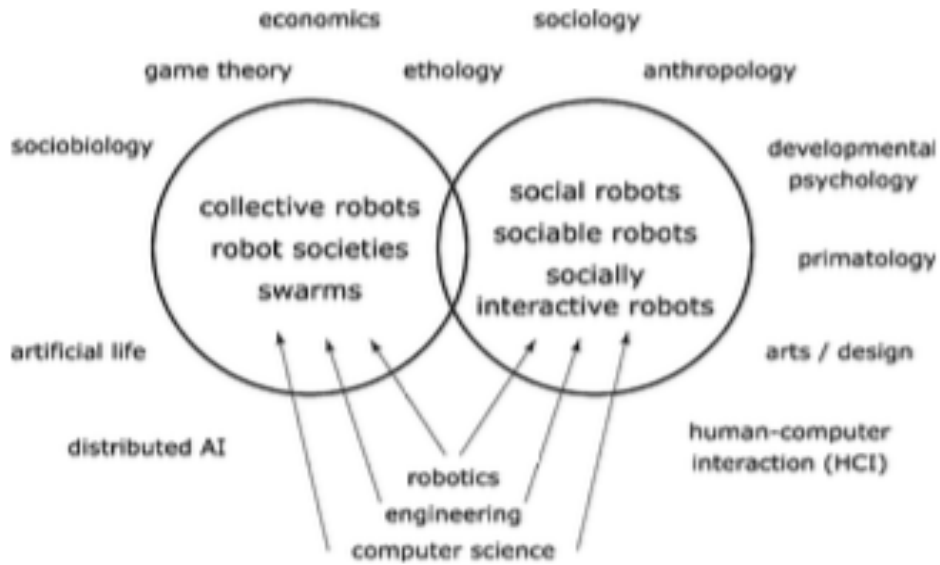


Figura 2 : Relación y campos de estudio de el trabajo colectivo y robots sociales

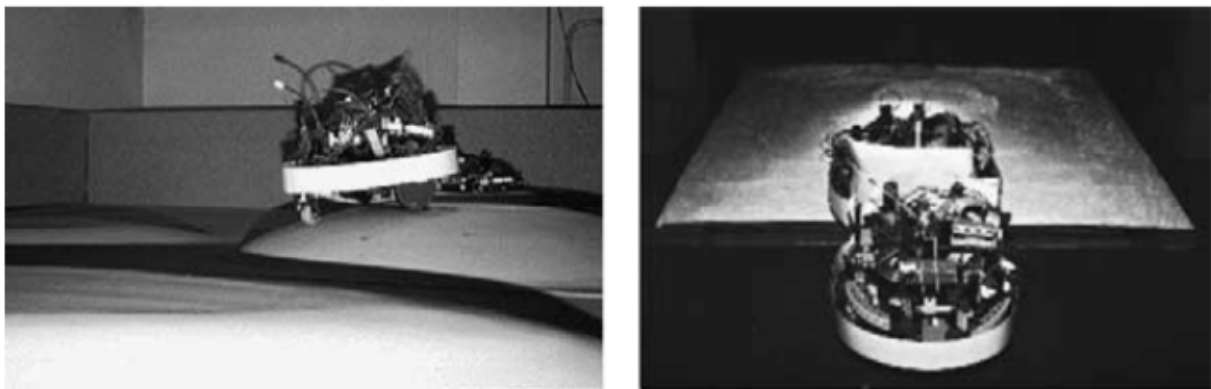


Figura 3 : Robots en estudios de interacción social entre robots

A lo largo del estudio de las interacciones realizadas por robots, se llega al punto de interacción hombre-robot, en el que hay que comenzar a trabajar en una serie de puntos para llegar a construir relaciones exitosas.

### 2.1.3. Morfología

La morfología de los robots debe ser consecuente con su función, y si desarrolla tareas en conjunto con los humanos debe favorecer la comodidad de éstos.

En el caso de los robots sociales, un aspecto humanoide resulta favorecedor a la hora de mantener interacciones exitosas, por lo que desde los proyectos más rudimentarios se ha intentado alcanzar este punto (Figura 4) [8], aunque aplicando ciertos límites.

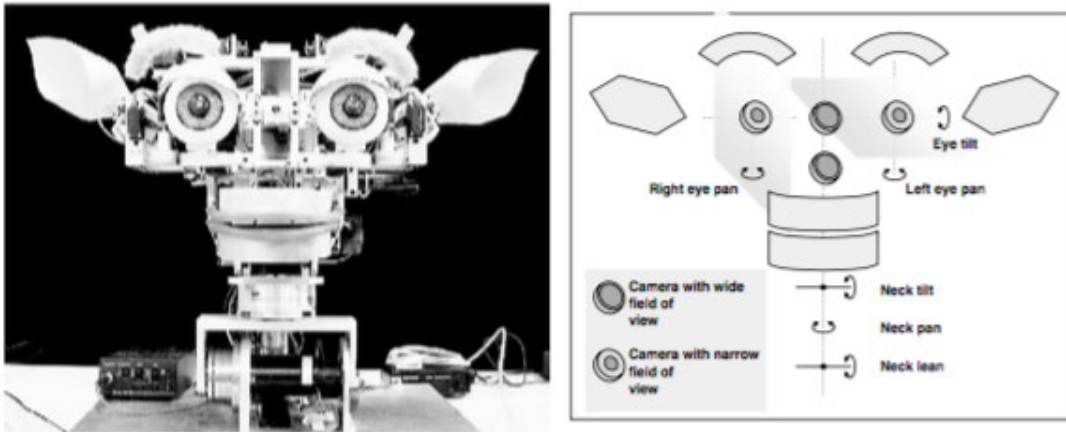


Figura 4: Robot social Kismet

El experto en robótica Masahiro Mori planteó la hipótesis de que cuando la apariencia de un robot es más humana, la respuesta emocional de un observador humano al robot se irá haciendo cada vez más positiva y empática, hasta cruzar un punto a partir del cual la respuesta se vuelve una fuerte repugnancia.

Sin embargo, cuando la apariencia del robot continua convirtiéndose menos distinguible de la de un ser humano, la respuesta emocional se vuelve positiva una vez más y se va aproximando a niveles de empatía como los que se dan entre humanos [9] [10].

A este concepto lo definió como valle inquietante (Figura 5), término que surge de la idea de que un robot que es casi humano es visto de forma general por un ser humano como extraño, y por esto resulta imposible alcanzar el requisito de una respuesta empática para la necesidad de una interacción humano-robot productiva.

En la gráfica se puede observar como los robots humanoides, con un parecido humano mayor que los robots industriales, resultan más familiares a la hora de ser percibidos por los humanos.

A pesar de ello, existe ese cierto punto en el que desciende radicalmente (valle inexplicable), hasta que se vuelve a mejorar el parecido humano llegando a límites realmente cercanos, como ocurre en la actualidad.

Este punto dentro del valle, indicado en la grafica como “muerto viviente”, implica una criatura con apariencia humana pero con ciertos aspectos que le diferencian de los mismos, los cuales a su vez crean una sensación de repugnancia-miedo a la persona que interactúa con este robot.

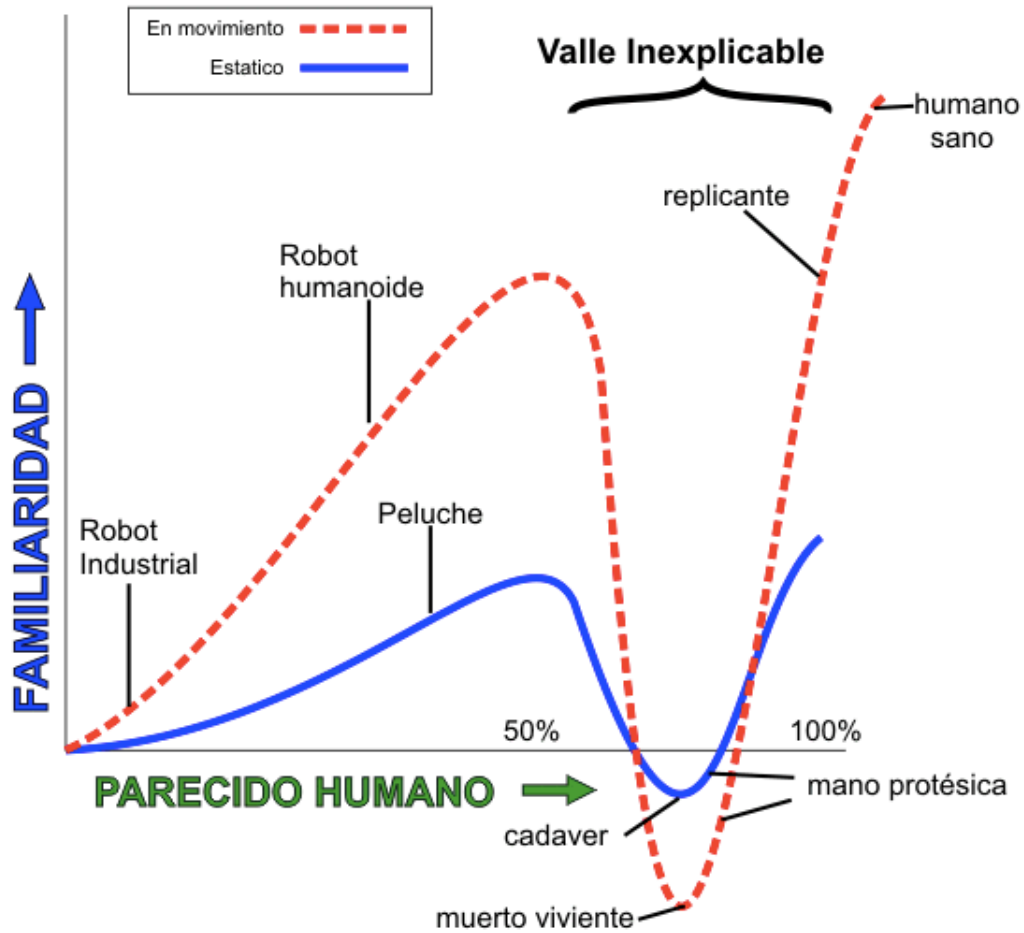


Figura 5 : Valle inquietante de Mori

#### 2.1.4. Comportamiento

Para llevar a cabo una interacción correcta, no basta sólo con que el aspecto resulte amigable, sino que debe cumplir una serie de requisitos.

El robot debe ser capaz de reconocer e interpretar en lenguaje, en distintos registros y tonos, para poder ser capaz de ajustar sus reacciones. Para ello, los robots también necesitan ser capaces de captar y analizar los movimientos del humano, así como la variación de sus expresiones faciales (Figura 6) [11].

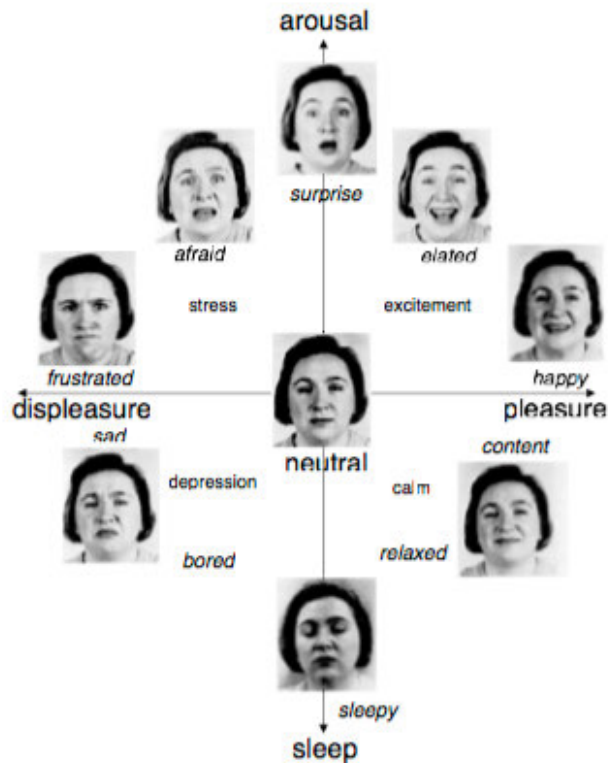


Figura 6 : Expresión facial de las emociones (Russell)

Por lo tanto, el robot debe estar dotado con Inteligencia Artificial (IA). Las primeras investigaciones sobre IA trataban de emular la inteligencia humana resolviendo situaciones de la forma más similar al comportamiento humano, pero más tarde se descubrió que a la hora de trasladar estas soluciones a los robots, algunas tareas relativamente sencillas tenían una complejidad desorbitada.

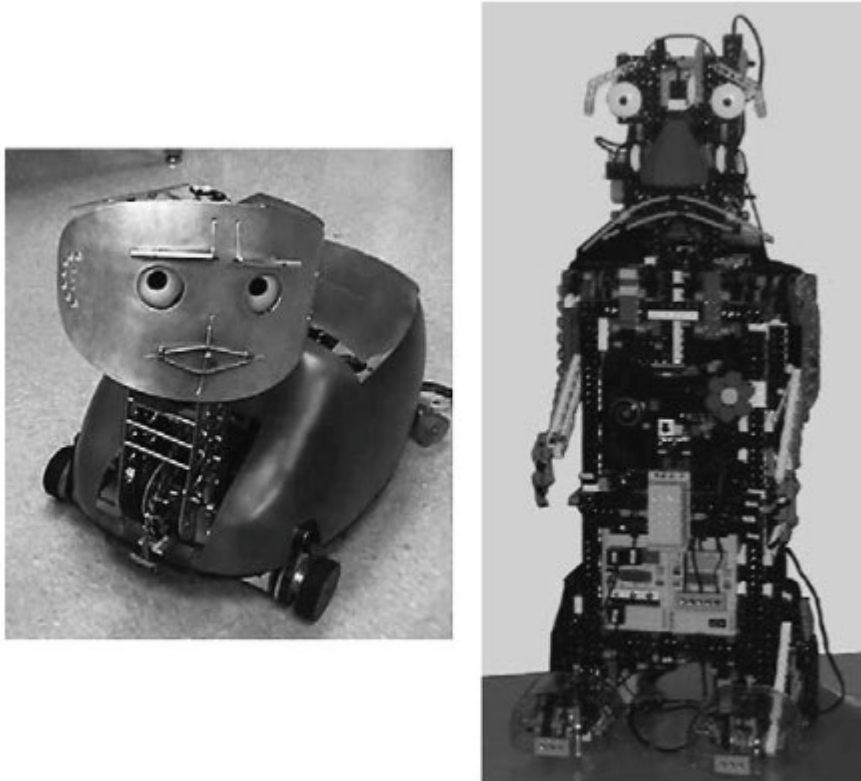
En un entorno dinámico e impredecible, se requiere un punto de equilibrio entre la complejidad del sistema sensorial del robot y la información de estos sensores que va a ser procesada, para dar una respuesta adecuada al entorno al que está sometido [12].

### 2.1.5. Emociones

Existe una gran cantidad de estudios que incluyen las emociones dentro del comportamiento, al estar éste condicionado por ellas, pero resulta necesario estudiarlas de forma individual ya que son el objeto de este trabajo.

La separación más lógica dentro de la IA son las actividades cognitivas, basadas en el razonamiento y desarrollo lógico, y las actividades afectivas, basadas en las emociones.

Las actividades afectivas son necesarias dado que influyen en un gran número de ámbitos, como puede ser el comportamiento, pero también el aprendizaje y la percepción de estímulos externos. En el caso de robots humanoides, las emociones ejercen una gran influencia en los gestos faciales, acción que resulta fundamental en el desarrollo de la empatía humana (Figura 7) [13] [14].



*Figura 7 : Robots con expresión Sparky y Felix*

Los robots pueden utilizar sus emociones para mejorar sus competencias en interacciones sociales, simulando la inteligencia humana [15], pero también existen robots cuyo objetivo mediante las interacciones es alcanzar unos objetivos sociales internos, es decir, modificar sus propias emociones [16], aunque para ello necesiten modelos cognitivos mucho más complejos [17] [18].

## 2.2. Aplicaciones de los robots sociales

### 2.2.1. Robots de apoyo

El objetivo de estos robots es ayuda a las personas en las actividades cotidianas [19], especialmente aquellas que no pueden valerse por sí mismas. Dentro de este grupo podemos diferenciar entre robots de ayuda y cuidados (Figura 8) y robots para facilitar la movilidad.

Entre los robots de ayuda, los que más reconocimiento reciben son los de asistencia sanitaria [20] [21] [22]. En esta categoría se incluyen también los robots dedicados a la cirugía que requieren de interacción social [23] [24].

Por su parte, los robots para facilitar la movilidad pueden ayudar frente a una inmovilidad total, como en una silla de ruedas (Figura 9) [25] [26], y también pueden ayudar a levantarse de una silla o una cama [27] [28].



Figura 8 : Robots Care-O-Bot II y Care-O-Bot 4 [29]

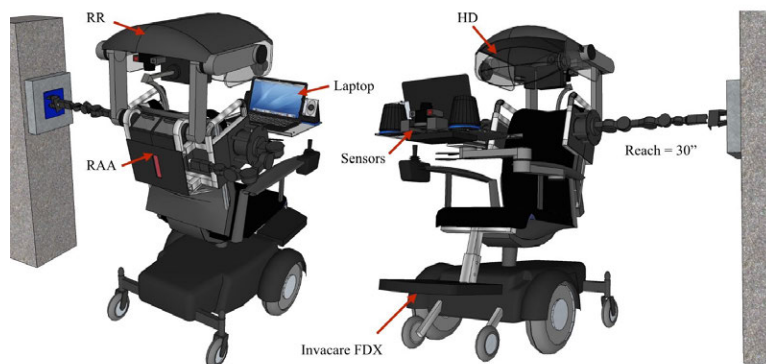


Figura 9 : Silla de ruedas inteligente



## 2.2.2. Robots para la educación

---

Dentro de este grupo debemos diferenciar entre los robots para la formación general y los robots dedicados a educar de una forma específica, por ejemplo en el caso de estudiantes con necesidades especiales.

En el primero grupo nos encontramos tanto con los robots de asistencia a profesores en las aulas de instituciones educativas [30] [31] [32] como con los robots guía de museos y actividades culturales [33] [34].

Dentro del grupo de estudiantes necesitados de ayuda, cabe destacar el caso de los niños autistas, cuya respuesta hacia el trabajo con robots sociales de asistencia a la educación es muy favorable [35] [36] [37] [38].

Para niños autistas, el robot más conocido es Nao (Ver sección 2.4.3), aunque también está extendido el uso de Milo (Figura 10) o el robot creado por la George Washington University (GWU) para ayudar a los niños autistas a interactuar con el resto de sus compañeros (Figura 11).

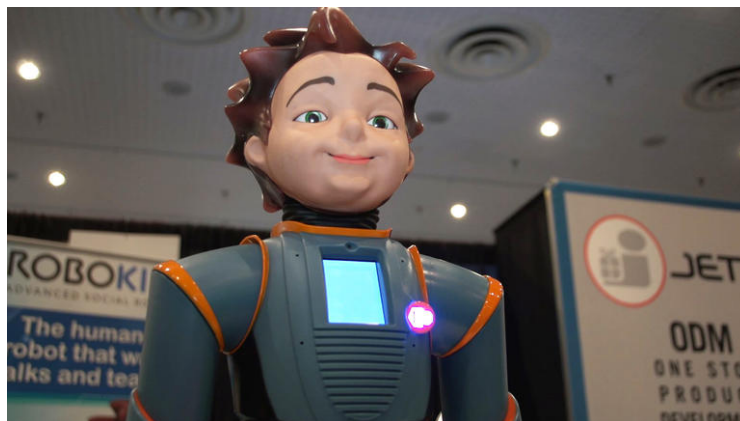


Figura 10 : Robot Milo para tratamiento de niños con autismo

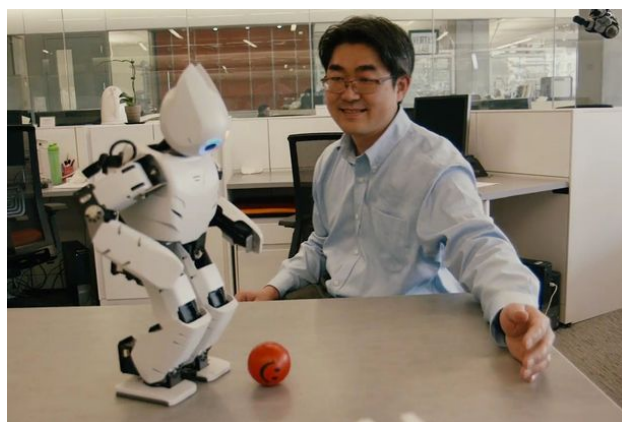


Figura 11 : Robot para tratar autismo en niños (GWU)

### 2.2.3. Robots sociales con fines lúdicos

---

En muchos casos no queda clara la división de los robots educativos frente a los lúdicos, ya que en general poseen bastantes características en común. Lo más destacable de estos robots es que a pesar de estar diseñados para el entretenimiento de los usuarios, no requieren de la participación del mismo, quedando relegado a un rol de expectación.

Los robots de este tipo requieren un nivel de interacción hombre-máquina muy complejo, ya que sino no podría satisfacer las necesidades de entretenimiento del usuario [39] [40] [41].

Constituyen un campo de investigación en auge, ya que se estima que en los próximos años pasarán a formar parte la vida cotidiana de las familias [42], ya sea como mascotas (Figura 12) [43], juegos [44] o a través de hobbies [45].



*Figura 12 : Perro robótico como mascota*

## 2.3. Robots sociales UPM

---

### 2.3.1. URBANO

---

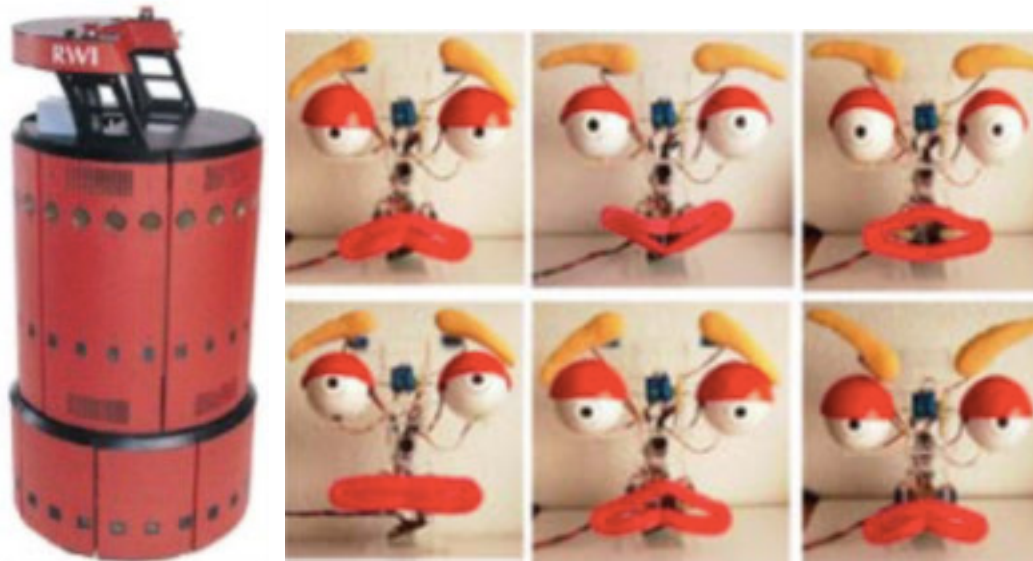
Este proyecto forma parte del grupo de Control Inteligente del Centro de Automática y Robótica de la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid.

El robot URBANO (Figura 13) es un robot utilizado para investigar en distintos campos de la robótica social desde la navegación autónoma [46] hasta modelos emocional [47], como el que se estudia en este trabajo de final de grado.



*Figura 13 : Robot URBANO*

Posee una plataforma con ruedas que habilita el movimiento (Figura 14) , una cara capaz de mostrar expresiones (Figura 15), y un brazo que permite realizar gestos.



Figuras 12 y 13 : Plataforma URBANO y cara URBANO

El robot urbano sigue una estructura de agentes en la que cada uno de ellos realiza una acción concreta y se integran para crear el comportamiento de robot social.

Entre estos agentes se encuentran:

- Programa de reconocimiento y tratamiento de imágenes obtenidas a través de una cámara.
- Programa de reconocimiento y tratamiento de voz obtenida mediante un micrófono.
- Capacidad de habla mediante un programa de traducción de texto a voz y reproducida mediante un altavoz.
- Movimiento mediante un programa de mapeo y movimientos autónomo, y realizado gracias a la plataforma con ruedas.
- Programa que codifica gestos faciales y movimientos con el brazo en función del estado del robot y la tarea a realizar.
- Una serie de protocolos de comunicación entre esos agentes y el programa principal del sistema.

### 2.3.2. Doris

---

El robot Doris es otro de los proyectos que forman parte del grupo de Control Inteligente del Centro de Automática y Robótica de la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid.

Doris es un robot social, humanoide, de características similares a URBANO pero dotado de una apariencia femenina, con una cara mecatrónica mejorada, que permite obtener variando los parámetros, una gran variedad de gestos y movimientos en los ojos (Figura 14).

Al igual que URBANO, también posee una plataforma con ruedas que le permite el movimiento, y un programa diseñado para que pueda desplazarse mediante navegación autónoma.



*Figura 14: Robot Doris*

## 2.4. Otros robots sociales

---

### 2.4.1. Maggie

---

Robot diseñado y construido en el Robotics Lab de la Universidad Carlos III de Madrid. El robot, de aspecto humanoide no realista, está formado por una carcasa de plástico y sostenido por una plataforma con ruedas [48].

En el proyecto colaboran profesores, doctorandos y algunos alumnos del departamento de Ingeniería de Sistemas y Automática.

El robot es capaz de moverse de forma autónoma, reconocer objetos etiquetados, caras, voces, contacto físico, y de proporcionar información solicitada mediante la conexión a Internet.

También cuenta con una pantalla táctil para poder manejarlo directamente.



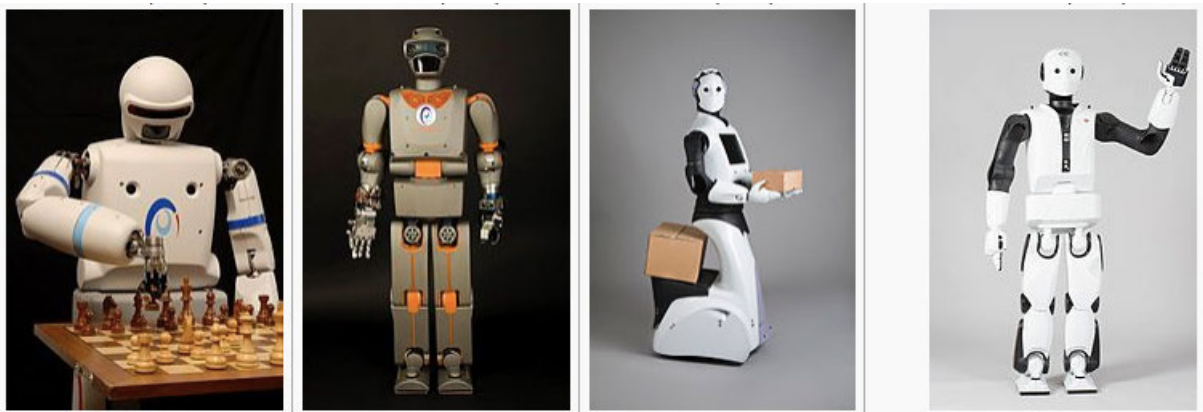
*Figura 15 : Robot Maggie*

## 2.4.2. REEM

---

Proyecto de robot humanoide desarrollado por la empresa Pal Robotics y que cuenta con la colaboración de la Universidad Politécnica de Cataluña y la Universidad Ramón Llull.

Este robot está capacitado para desplazarse, evitar obstáculos y orientarse de manera correcta, además del reconocimiento de caras y voces, y entre sus periféricos se encuentran una pantalla táctil para poder dar órdenes, todo tipo de sensores, un micrófono, un altavoz y una cámara [49].



*Figura 16 : REEM-A, REEM-B, REEM, REEM-C*

Las variaciones más notables entre versiones son la altura (1.40m para la versión frente el 1.60m de la versión C), la carga soportable (de 2kg a 10kg) y la capacidad de procesamiento (Intel Pentium M a Intel Core i7).

El único robot con varias particularidades fue la versión REEM, con ruedas en lugar de piernas, lo que disminuyó el número de grados de libertad pero aumentó la batería y la velocidad de desplazamiento.

### 2.4.3. Nao

---

Robot humanoide, autónomo y programable, desarrollado por Aldebaran Robotics.

Su principal característica es la empatía, que permite desarrollar emociones y formas lazos con los humanos con los que se relacionan.

Puede reconocer expresiones faciales, gestos corporales, y variaciones en el tono de voz, lo que lo habilita a averiguar el estado anímico de las personas.

Cuenta con un sistema multimedia basado en Linux, micrófonos para reconocimiento y detección de posición de los sonidos, altavoces y cámaras.

Una de las aplicaciones que más éxito ha tenido ha sido ayudar a la enseñanza en niños autistas, ya que algunos estudios demostraron que estos niños se encontraban más sencillo relacionarse con los robots que con los humanos.



*Figura 17 : Robot Nao*



#### 2.4.4. ASIMO

---

ASIMO (Advanced Step in Innovative Mobility) es un robot humanoide desarrollado por Honda en el año 2000.

Fue uno de los primeros robots bípedos con habilidad de movimiento.

El robot ASIMO está capacitado para reconocer objetos en movimientos, posturas corporales y gestos, así como detección del entorno, todo ello gracias a una cámara montada en su cabeza.

También cuenta de sistema de reconocimiento de voz y de caras, por lo que puede reaccionar adecuadamente ante estímulos proporcionados por humanos a los que ya ha conocido antes.

Está habilitado para proporcionar información gracias a su conexión a Internet.



*Figura 18: Robot ASIMO*

## 2.4.5. Geminoid

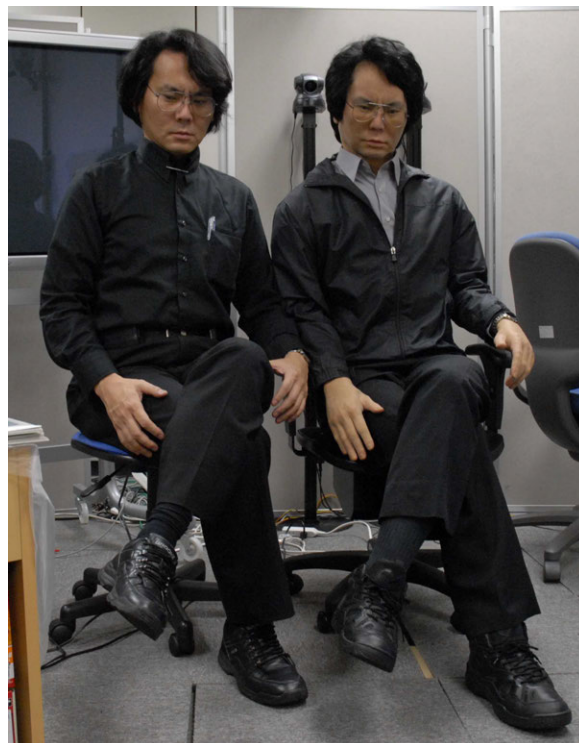
---

El proyecto Geminoid ha creado robots androides de características muy realistas. Está desarrollado en conjunto por el Advanced Telecommunications Research Institute y la compañía Kokoro. El director del proyecto es el profesor Hiroshi Isiguro.

El primer androide de la gama Geminoid era un clon del propio Isiguro, aunque actualmente hay otros prototipos como el Geminoid F, con aspecto de mujer y el Geminoid-DK, catalogado como el clon robótico más realista.

Su aplicación es la de un robot concebido para estudiar las expresiones y comportamientos humanos, con lo que se busca imitar al máximo la apariencia de los humanos.

Entre sus principales características se encuentra la teleoperación, la limitación de movimiento a cabeza y torso, y el control remoto mediante secuencias de movimiento preprogramadas. Posee un altavoz, con lo cual puede transmitir voz a través del operador que maneja el robot.



*Figura 19 : Primer robot Geminoid y su creador*

# 3 DISEÑO Y DESARROLLO DEL PROGRAMA

---

## 3.1. INTRODUCCIÓN

---

El objetivo del programa es la emulación de las emociones, mediante un modelo basado el espacio de estados. Para este fin, se ha hecho uso del lenguaje de programación C++, ya que debe poder integrarse con el resto de agentes del robot diseñados en C y C++, y además se hace uso de la programación orientada a objetos, mediante clases para los estados, estímulos y la comunicación.

Entre los objetivos del proyecto también se encuentra la mejora de las habilidades de programación, en concreto en el lenguaje C++ [50] [51] [52], punto que será ampliado en las conclusiones (ver sección 7.1).

En primer lugar, se repasarán las herramientas utilizadas, entre las que se encuentran el entorno de desarrollo Visual Studio y el software Git para el control de versiones durante la fase de desarrollo.

## 3.2. HERRAMIENTAS UTILIZADAS

---

### 3.2.1. Visual Studio

---

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django, etc., a lo cual sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas, etc.

Este proyecto comenzó su desarrollo en Visual Studio 2013, pero se continuó en la versión de 2010 (Figura 20) hasta solucionar algunas diferencias de adaptación a los programas ya hechos debido a modificaciones en las librerías.

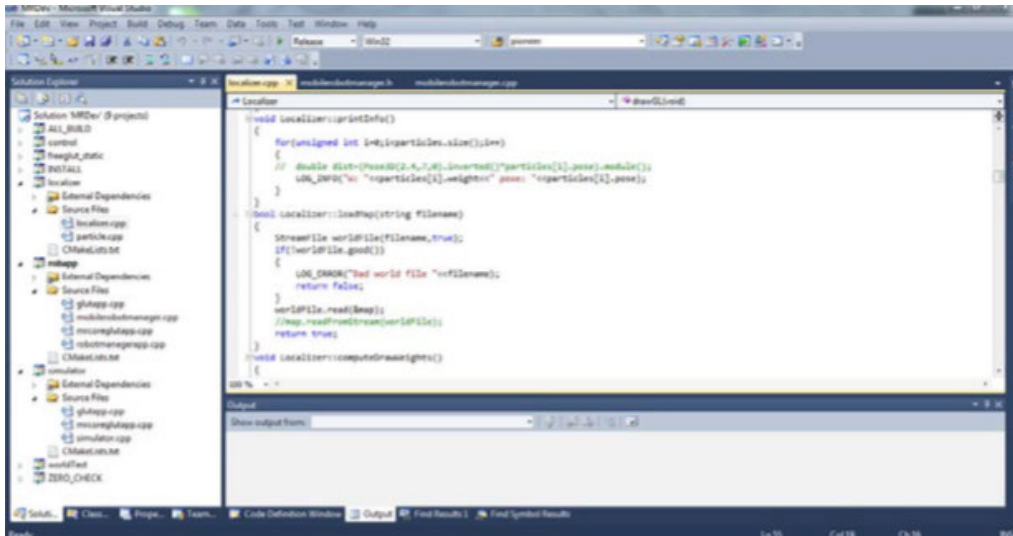


Figura 20 : Ventana de trabajo de Microsoft Visual C++ 2010

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo.

Además ofrece la posibilidad de crear aplicaciones para muchas plataformas de Microsoft como Windows, Azure, Windows Phone 7 o Sharepoint. Microsoft ha sido sensible a la nueva tendencia de las pantallas táctiles y con Visual Studio 2010 también es posible desarrollar aplicaciones para pantallas multitáctiles.

### 3.2.2. Git

Git (Figura 21) [53] es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

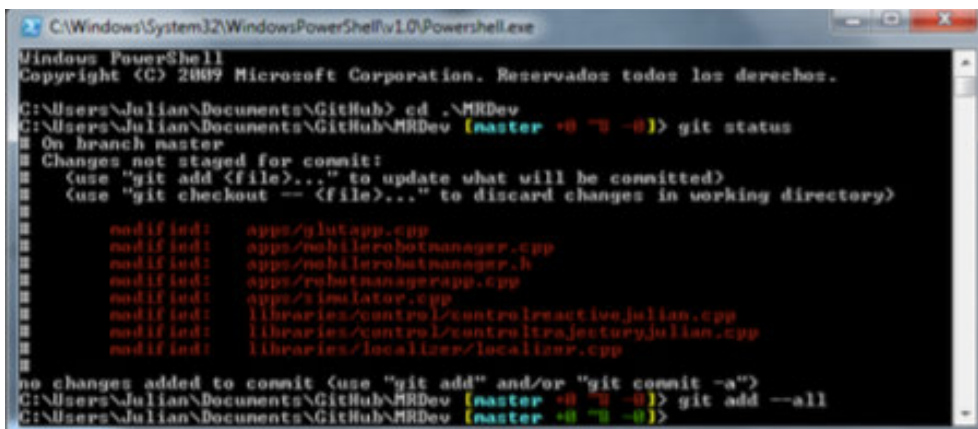


Figura 21 : Interfaz de comandos de Git

### 3.3. METODOLOGÍA

---

#### 3.3.1. Espacio de estados

---

La representación de espacio de estados (Figura 22) es un modelo matemático de un sistema físico descrito mediante un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales de primer orden que se combinan en una ecuación diferencial matricial de primer orden [54] [55].

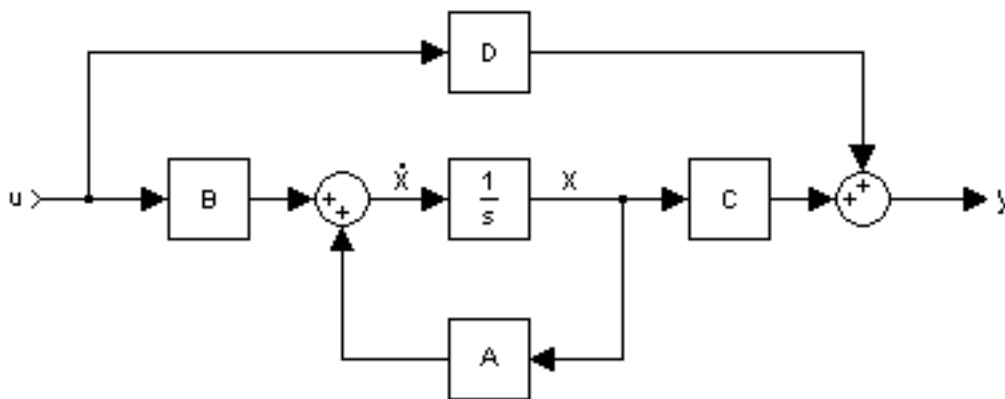


Figura 22 : Modelo de espacio de estado

Para prescindir del número de entradas, salidas y estados, las variables son expresadas como vectores y las ecuaciones algebraicas se escriben en forma matricial (esto último solo puede hacerse cuando el sistema dinámico es lineal e invariante en el tiempo). La representación de espacios de estado (también conocida como aproximación en el dominio del tiempo) provee un modo compacto y conveniente de modelar y analizar sistemas con múltiples entradas y salidas.

En la representación de espacio de estados es importante conocer el concepto de variables de estado, que representan el subconjunto más pequeño de variables de un sistema que pueden representar su estado dinámico completo en un determinado instante.

Estas variables de estado deben ser linealmente independientes, es decir, una variable de estado no puede ser combinación lineal de otras variables de estado.

El número mínimo de variables de estado necesarias para representar un sistema dado,  $n$ , es normalmente igual al orden de la ecuación diferencial que define al sistema con el que se está trabajando.

En el caso de sistemas lineales, la forma general de representación de espacios de estado de un sistema lineal con  $p$  entradas,  $q$  salidas y  $n$  variables de estado es el mostrado en la Figura 23.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) \\ \mathbf{y}(t) &= C(t)\mathbf{x}(t) + D(t)\mathbf{u}(t)\end{aligned}$$

donde

$$\begin{aligned}x(t) &\in \mathbb{R}^n; y(t) \in \mathbb{R}^q; u(t) \in \mathbb{R}^p; \\ \dim[A(\cdot)] &= n \times n, \\ \dim[B(\cdot)] &= n \times p, \\ \dim[C(\cdot)] &= q \times n, \\ \dim[D(\cdot)] &= q \times p, \\ \dot{\mathbf{x}}(t) &:= \frac{d\mathbf{x}(t)}{dt}.\end{aligned}$$

Figura 23 : Representación de espacios de estado de un sistema lineal

Donde  $x$  es el vector de estados,  $y$  es el vector de salidas,  $u$  es el vector de entradas o control,  $A$  es la matriz de estados,  $B$  es la matriz de entrada,  $C$  es la matriz de salida y  $D$  la matriz de transmisión directa.

En el modelo emocional objeto de este proyecto, el vector de estados estará formado por las emociones, el vector de salidas por los modificadores globales, y el vector de entradas o control, por los estímulos.

A la hora de implementar el programa o realizar ensayos de simulación con el, hay que definir una serie de parámetros en un fichero de configuración. Las variables a definir son las siguientes:

- $nVariablesEstado$ , que representa el número de variables de estado ( $n$ ), es decir, el número de emociones linealmente independientes que posee el modelo con el que trabajar.
- $nEntradas$ , que representa el número de entradas del sistema ( $p$ ), es decir, el número de estímulos diferentes que puede recibir el modelo.
- $nSalidas$ , que representa el número de salidas del sistema ( $q$ ), es decir, el número de parámetros modificadores globales del sistema

### 3.3.2. Multithreading

El procesamiento multithreading (multihilo) consiste en ejecutar al mismo tiempo dos o más hilos de ejecución. Al contrario que el multiprocessing, que requiere el uso de dos o más procesadores (CPU), el multithreading puede ser llevado a cabo por una sola unidad de procesamiento (Figura 24) [56] [57] [58] [59].

## Multiprocessing vs. Multithreading

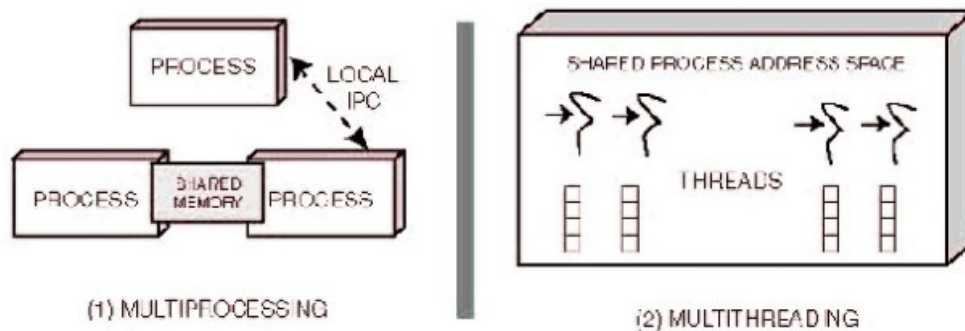


Figura 24 : Multiprocesamiento y multihilo

Este tipo de procesamiento tiene una serie de ventajas frente a la ejecución secuencial en un sistema como el estudiado [60] [61] [62], ya que existen parámetros que son recalculados constantemente durante toda la ejecución, y no siempre de forma secuencial, ya que dependen de factores externos como son los estímulos y la interacción con otros agentes.

El programa se divide en cinco hilos (Figura 25), distribuidos de la siguiente forma:

- threadEstimulos (Figura 26), que se encarga de leer y tratar a información recibida en forma de estímulos a través de métodos de la clase Estimulo (ver sección 3.3.3.).
- threadEmociones (Figura 27), que se encarga de recalculer el vector de estados (emociones), mediante el uso de métodos de la clase Estado (ver sección 3.3.3.)
- threadUniversoEmocional, (Figura 28) calcula el universo emocional (ver sección 1.1) mediante el error entre las emociones y los valores de referencia, y su derivada. Permite obtener los modificadores globales.

- threadObjetivos, en el que se realiza la comunicación con el resto de agentes (ver 5.2.), y define las acciones a seguir para cumplir los objetivos emocionales propuestos.
- threadAprendizaje, en el que se recalcula el efecto de los modificadores globales sobre el sistema, teniendo en cuenta el estado actual, el universo emocional y la aproximación a la consecución de objetivos.

```

DWORD threadEstimulosID, threadEmocionesID, threadUniversoEmocionalID, threadObjetivosID, threadAprendizajeID;
// Estructura de programa multithread, dividido en cinco hilos
HANDLE threadEstimulos = CreateThread(NULL, 0, estimulos, NULL, 0, &threadEstimulosID);
HANDLE threadEmociones = CreateThread(NULL, 0, emociones, NULL, 0, &threadEmocionesID);
HANDLE threadUniversoEmocional = CreateThread(NULL, 0, universoEmocional, NULL, 0, &threadUniversoEmocionalID);
HANDLE threadObjetivos = CreateThread(NULL, 0, objetivos, NULL, 0, &threadObjetivosID);
HANDLE threadAprendizaje = CreateThread(NULL, 0, aprendizaje, NULL, 0, &threadAprendizajeID);

```

*Figura 25 : Manejadores de hilos*

```

// La funcion de este thread es leer y tratar la información recibida en forma de estímulos
mutex.lock();

Estimulo.LeerEstimulo();
Estimulo.Calculo();
Estimulo.CalculoMatrizEntradas();

mutex.unlock();

```

*Figura 26 : threadEstimulos*

```

// La funcion de este thread es recalcular el vector de estados
mutex.lock();

Estado.Calculo();
CalculoVectorEstados();

mutex.unlock();

```

*Figura 27 : threadEmociones*



```

// La funcion de este thread es calcular el universo emocional (error y derivada)
// Permite obtener los modificadores GLOBALES

mutex.lock();

Estimulo.Mostrar();
Estado.Mostrar();
CalculoUniversoEmocional();

mutex.unlock();

```

*Figura 28 : threadUniversoEmocional*

Resulta necesaria la explicación de los métodos lock y unlock. Estos métodos de la clase mutex, se encargan de bloquear los elementos utilizados en un thread, de forma que si otro thread intenta acceder a ellos, se le deniegue el acceso hasta que vuelvan a ser desbloqueados. [63]

La utilidad de los métodos lock y unlock es máxima, ya que durante el recálculo de parámetros, el acceso de forma parcial o total a ellos, podría causar errores de cálculo o de transferencia de datos.

El inconveniente de los bloqueos es la existencia de desajustes en la velocidad de ejecución. Para obtener una ejecución eficiente, sin el derroche de recursos en cálculos inútiles (por ejemplo, recalcular dos veces la misma matriz antes de ser utilizada), se ajustan los tiempos de ejecución mediante paros temporales (Sleep), con tiempos obtenidos mediante ensayos (ver 6.1.) [64] [65].

Por lo tanto, los threads se encargan de calcular de forma individual los parámetros del sistema, actualizando el estado constantemente, mediante la evolución de los valores de las matrices (modificadores de vectores) y la obtención de modificadores globales.

Los modificadores globales son parámetros obtenidos a través del cálculo del universo emocional (comparación entre valores del vector de estado con los valores de referencia, y su derivada), que modifican el sistema por completo, con la intención de alcanzar los objetivos emocionales.

Para el aprendizaje, existe una matriz de ajuste frente a los modificadores globales, de forma que el modelo pueda darle un mayor o menor peso relativa a cada uno de los modificadores con el objetivo de conseguir los objetivos emocionales deseados.

### 3.3.3. Clases

---

En el programa se han usado clases propias, ya nombradas en la explicación de los threads, y este es su funcionamiento:

- Clase Estimulo : contiene las variables y métodos necesarios para inicializar la matriz de entradas y el vector de entradas del espacio de estados, así como tratar los nuevos valores de entrada e ir ajustando la matriz. Para facilitar los ensayos de simulación, también se encarga de mostrar por pantalla los valores de la matriz y el vector de entradas.
- Clase Estado : funcionamiento similar a Estimulo, pero en este caso con el vector de estados y la matriz de estados. Para facilitar los ensayos de simulación, también se encarga de mostrar la matriz y el vector por pantalla.
- Clase Tcp : se encarga de la comunicación del programa con el resto de agentes del robot. Hereda de la clase SocketNode (ver sección 5.2).

# 4 Versión actualizada del programa

---

Tras el desarrollo de una primera versión que incluía todo lo indicado anteriormente, y de forma posterior a los primeros ensayos, se observó que uno de los puntos para mejorar el funcionamiento del programa sería la borrosificación de parámetros.

Esta actualización del programa permite llevarlo a un grado de desarrollo en complejidad superior, con la vista puesta en seguir ampliándolo en el futuro (ver sección 7.2).

## 4.1 Descripción de cambios

---

Se crea una nueva clase EstadoFuzzy, que permite trabajar con parámetros borrosos, pero también obliga a modificar parte de la configuración para adaptarla a trabajar con estos parámetros.

Para este fin, se deberá realizar una configuración en un fichero externo, del que se leerán los datos. En este fichero se incluyen los términos borrosos, los estímulos y sentimientos, los coeficientes y modificadores, los efectos inversos y las reglas borrosas.

Lo primero que hay que definir son los términos y estímulos (Figura 29).

En relación a los términos, se configura alto como 2, medio como 1 y bajo como 0.

En cuanto a los estímulos se dividen en dos tipos:

- Los primeros, impulsionales, que se definen mediante un conjunto de reglas borrosas, por ejemplo gratificación, insulto o golpe.
- El otro tipo son los continuos, que están influenciados por una serie de coeficientes (también definidos en el archivo de configuración), y son, por ejemplo, la batería, la repetición o el biorritmo.

```

TERMINOS:
ALTO    2
MEDIO   1
BAJO    0

ESTIMULOS:
// Impulsos (Mediante reglas)
Gratificacion
Insulto
Golpe
// Continuos (Mediante coeficientes)
Bateria
Repeticion
Biorritmo

```

Figura 29 : Términos y estímulos en el fichero de configuración

También se incluyen en el fichero los objetivos emocionales, mediante estímulo, valor, tiempo y offset (Figura 30).

Esto significa que el objetivo del modelo será alcanzar estos valores para cada uno de los estímulos en un tiempo menor o igual al establecido, tras haberle aplicado un estímulo que haya modificado su estado a parámetros distintos a los de objetivo.

```

SENTIMIENTOS:
// Valor Tiempo Offset
Alegria      0    100    50
Disgusto     0    100     0
Enfado       0    100     0
Miedo        0     65     0

```

Figura 30 : Objetivos emocionales en el fichero de configuración

Como ya ha sido indicado, los estímulos deben responder de forma preprogramada mediante una serie de parámetros que ajusten su efecto. En el caso de los estímulos continuos, se realiza mediante el uso de coeficientes (Figura 31).

Esto significa que una determinada variación de uno de estos parámetros, afecta a las emociones en un valor preprogramado.

```
COEFICIENTES:
// Coef Offset (Sentimiento Valor)
Bateria 20 (Alegria -5) (Disgusto 5) (Enfado 5) (Miedo 5)
Biorritmo 50 (Alegria 5) (Disgusto -5) (Enfado -5) (Miedo -5)
Repeticion 30 (Alegria -5) (Disgusto 5) (Enfado 5)
```

Figura 31 : Coeficientes de estímulos continuos en el fichero de configuración

Los otros parámetros a tener en cuenta para la modificación del comportamiento, son como su nombre indica, los modificadores, que nos permiten ajustar todo el funcionamiento del sistema y también deben ser incluidos en el archivo de configuración (Figura 32).

```
MODIFICADORES:
// Offset
Información 40
Velocidad 60
Motivacion 20
```

Figura 32 : Modificadores en el fichero de configuración

Esto significa que una determinada variación de uno de estos parámetros, afecta a las emociones.

Otro de los puntos importantes a establecer en el fichero de configuración de esta versión del programa, son los efectos inversos. Estos efectos son los que producen los estímulos sobre los sentimientos, o los sentimientos sobre los modificadores.

Es necesario incluirlos en el fichero de configuración para un buen funcionamiento del programa (Figura 33).

```

EFFECTOS-INVERSOS:
// <Estímulo SOBRE Sentimiento> o
// <Sentimiento SOBRE Modificador>
Golpe SOBRE Alegria
Insulto SOBRE Alegria
Alegria SOBRE Velocidad
Miedo SOBRE Información

```

Figura 33 : Efectos inversos en el fichero de configuración

Finalmente, para asegurar un correcto funcionamiento del sistema frente a los estímulos impulsionales (como gratificación, insulto o golpe), es necesario definir las reglas.

Estas reglas establecer relaciones entre emociones, estímulos y modificadores.

En la Figura 34, se encuentra un ejemplo de reglas en el fichero de configuración, divididas en reglas para la dinámica de evolución, reglas de efecto de los estímulos y reglas de salida del sistema.

```

// Dinámica de evolución
REGLAS_A:
SI Alegria ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Alegria,Alegria) ES ALTO,MEDIO,BAJO
SI Miedo ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Miedo,Miedo) ES ALTO,MEDIO,ALTO
SI Disgusto ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Disgusto,Disgusto) ES ALTO,MEDIO,ALTO
SI Enfado ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Enfado,Enfado) ES ALTO,MEDIO,ALTO

// Efecto de los estímulos
REGLAS_B:
SI Miedo ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Golpe,Miedo) ES ALTO,ALTO,MEDIO
SI Alegria ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Gratificacion,Alegria) ES ALTO,MEDIO,BAJO
SI Alegria ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Insulto,Alegria) ES ALTO,MEDIO,BAJO

// Salida del sistema
REGLAS_C:
SI Miedo ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Miedo,Información) ES ALTO,MEDIO,BAJO
SI Miedo ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Miedo,Velocidad) ES ALTO,MEDIO,BAJO
SI Alegria ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Alegria,Velocidad) ES ALTO,MEDIO,BAJO
SI Alegria ES ALTO,MEDIO,BAJO ENTONCES EFFECTO(Alegria,Información) ES ALTO,MEDIO,BAJO

```

Figura 34 : Reglas en el fichero de configuración

# 5 COMUNICACIÓN E INTEGRACIÓN

## 5.1. Introducción a los agentes

El desarrollo de un robot social depende de muchas funciones distintas, desde la interacción directa en forma de conversación hasta funciones relacionados con el entretenimiento.

Una solución a este planteamiento es el de robot basado en agentes [66] [67] [68] [69] [70], que permiten realizar cada una de las funciones de forma individual e integrarlas mediante comunicación en un marco de trabajo (Figura 29).

Esta solución resulta óptima dada la necesidad de compartir recursos, ya que el control de motores, sensores y servos se realiza a través de un computador con sistema operativo Linux, mientras que la para la sintetización de voz del robot se utilizan librerías exclusivas de Windows.

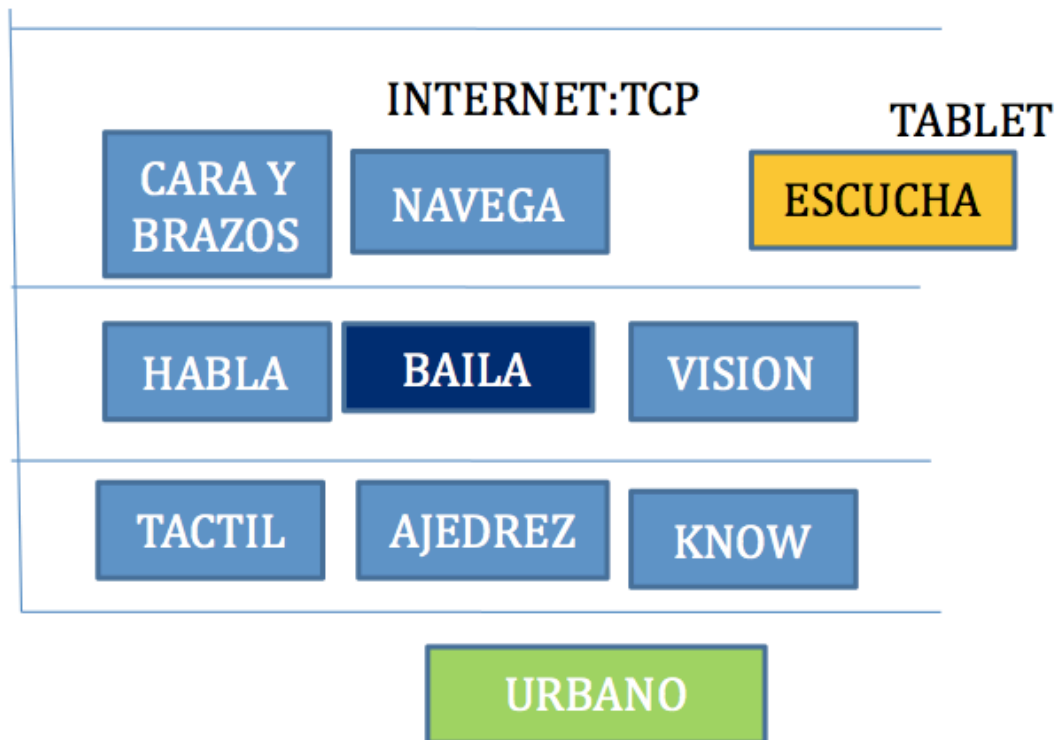


Figura 29 : Marco de trabajo robot URBANO

Dentro de este marco de trabajo podemos ver la distribución de programas, es decir, la arquitectura basada en agentes (Figura 30).



Figura 30 : Distribución basada en agentes URBANO 9.4

Dentro de este marco de trabajo podemos ver la distribución de programas, es decir, la arquitectura basada en agentes (Figura 30).

Algunos de los programas con los que hay que estar familiarizado para entender el funcionamiento del modelo emocional en relación al robot y son:

- Urbano.exe (Figura 31) : interfaz gráfica en la que se pueden observar todos los programas, direcciones y puertos, además de su estado de conexión. Desde esta interfaz se pueden enviar tareas, gestos y caras anteriormente codificadas en un archivo de configuración.
- Escucha.exe (Figura 32) : programa con interfaz diseñado para la probar la recepción de mensajes en Urbano.exe y permite simular el programa de reconocimiento de voz.



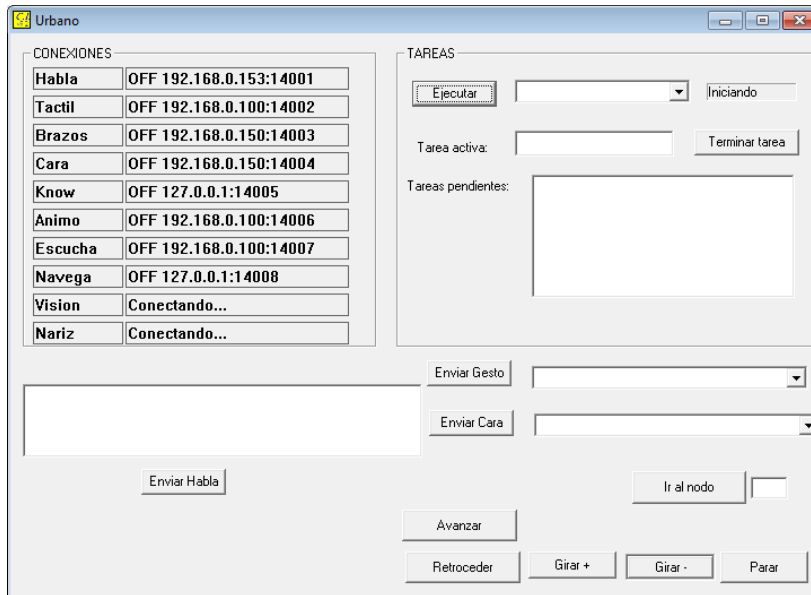


Figura 31 : Interfaz del programa Urbano.exe

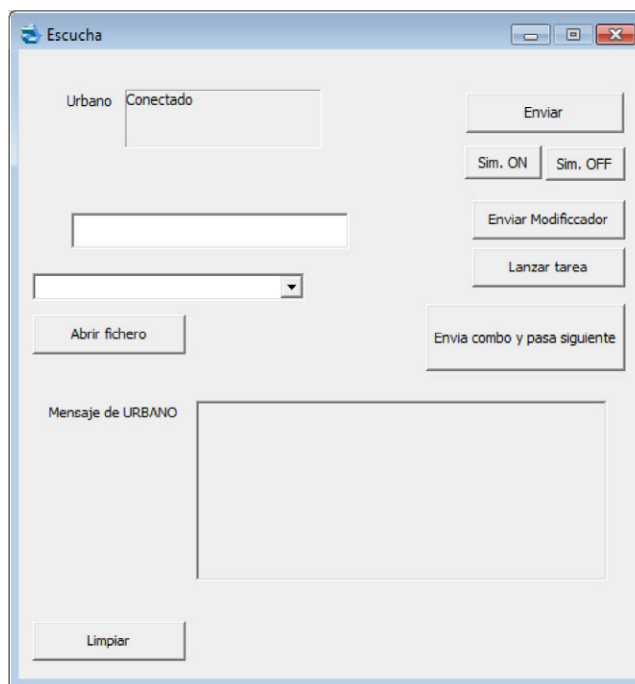
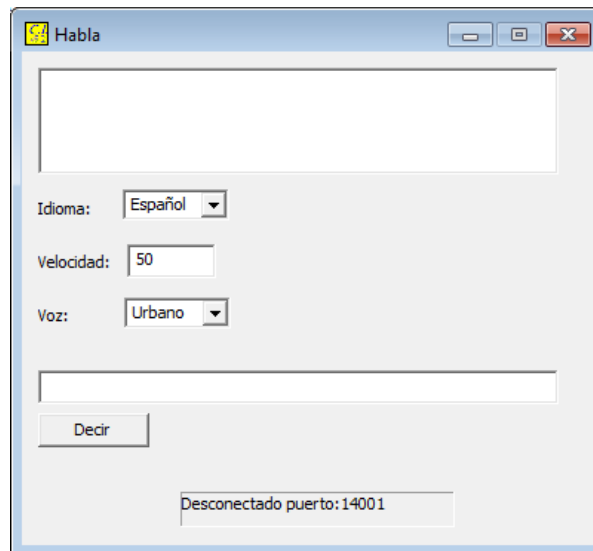


Figura 32 : Interfaz del programa Escucha.exe

- Habla.exe (Figura 33) : se ejecuta automáticamente en el PC Windows a bordo de URBANO, por lo que no la interfaz no es visible. Para tareas de simulación cuenta con la herramienta visible para sintetización de voz, así como unos parámetros ajustables.



*Figura 32 : Interfaz del programa Habla.exe*

Existen otros programas que integran el robot, aunque no imprescindibles para estudiar la relación con el modelo emocional, como:

- Navega.exe, para ayudar a la navegación segura del robot mediante el seguimiento de nodos definidos y transmitidos por Urbano.exe.
- Camara.exe, que transmite la imagen de una o varias cámaras conectadas por USB y usando librería de OpenCV.
- Vision.exe, que también transmite las imágenes vía UDP y además realiza reconocimiento y aprendizaje de caras.
- Baila.exe, que obtiene los parámetros característicos de una canción (mediante el programa PureData) y permite al robot moverse de acuerdo a esos parámetros.
- Tactil.exe, que permite la interacción del usuario mediante la selección de botones, imágenes, sonidos y vídeos.

## 5.2. Comunicación e integración de agentes

Para entender el funcionamiento de la comunicación en el robot, es importante definir como está diseñada la arquitectura de comunicaciones (Figura 33).

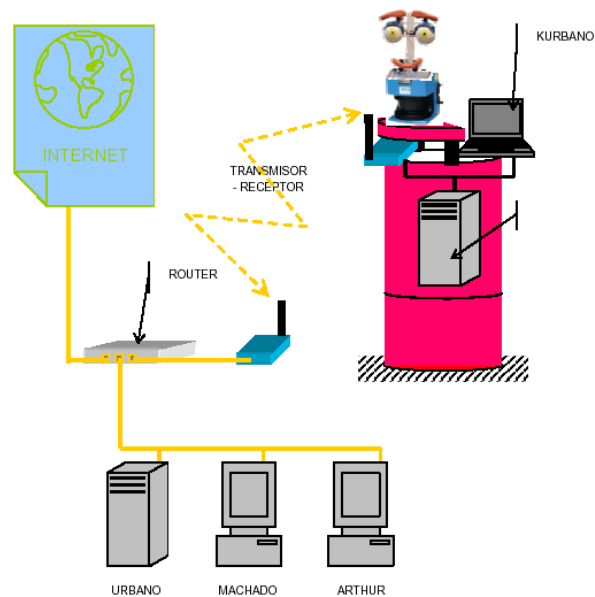


Figura 33 : Arquitectura de comunicaciones

La arquitectura de comunicaciones permite la comunicación entre las diferentes estaciones de trabajo, y se realiza vía red Ethernet. Se dispone de un transmisor receptor Ethernet vía radio para conectar los ordenadores a bordo de la plataforma robótica con el resto de la red.

Los protocolos de comunicación más utilizados en sistemas distribuidos son el protocolo OSI y el protocolo TCP/IP.

El protocolo OSI (Open System Interconnection) [71] [72] [73], fue un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones. Se compone de los siguientes niveles: nivel de aplicación, nivel de presentación, nivel de sesión, nivel de transporte, nivel de red, nivel de enlace de datos y nivel físico.

El protocolo TCP/IP (Figura 34) [74] [75] [76], cuya arquitectura se representa en la figura, está formado por cuatro niveles:

- Nivel de red. Se encarga de manejar el acceso al hardware de red y es específico de cada red en particular (Ethernet, ATM, etc.).
- Nivel de internet. Este nivel, también denominado IP, se encarga de transferir los distintos paquetes, denominados en este protocolo datagramas, entre diferentes computadores.

- Nivel de transporte. Este nivel se encarga de la transmisión de mensajes entre diferentes procesos. Existen dos protocolos de transporte:
  - Protocolo UDP. Es un protocolo basado en datagramas de tamaño máximo de 64 KB no orientado a conexión. Este protocolo no ofrece fiabilidad (no asegura que todos los datagramas lleguen al destinatario) ni ordenación en la entrega de los diferentes datagramas en el receptor.
  - Protocolo TCP. Se trata de un protocolo orientado a la conexión que garantiza que todos los datos se entregan en el orden en que se envían. Estas conexiones constituyen un flujo de bytes, asegurando la entrega ordenada de los mismos en el receptor.
- Nivel de aplicación. Está formado por las distintas aplicaciones que utilizan el nivel de transporte para comunicarse entre sí. Ejemplo de este tipo de aplicaciones son la transferencia de archivos (ftp), el protocolo http utilizado en los servidores Web y la conexión remota (telnet).

El tamaño del paquete suele estar limitado por restricciones hardware, los mensajes suelen tener un tamaño por lo general variable. En un protocolo no orientado a conexión no hay establecimiento de la conexión ni un cierre de la misma al final.

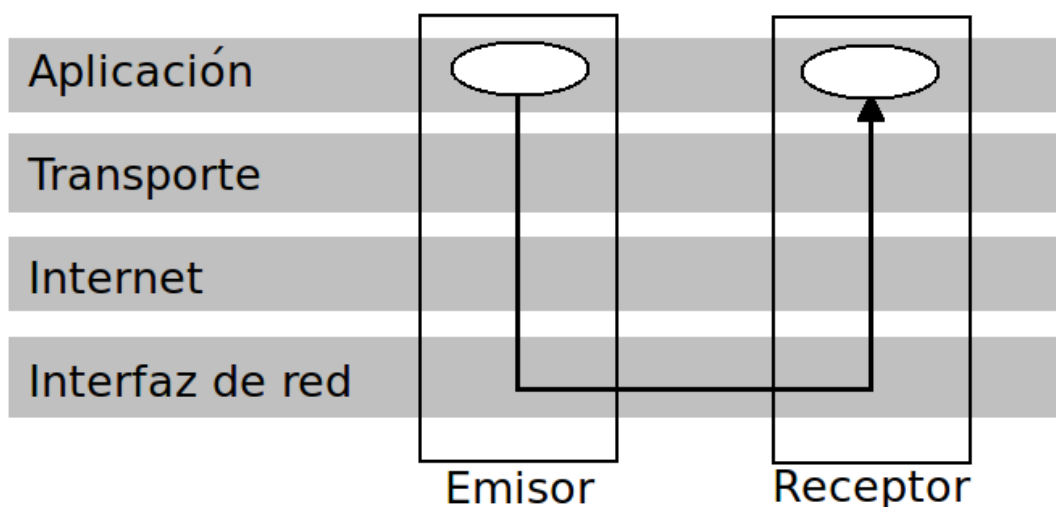


Figura 34: Protocolo TCP/IP

La arquitectura de comunicaciones permite la comunicación entre las diferentes estaciones de trabajo, y se realiza vía red Ethernet. Se dispone de un transmisor receptor Ethernet vía radio para conectar los ordenadores a bordo de la plataforma robótica con el resto de la red.

El último concepto que hay que definir es el de socket. Un socket es una abstracción que representa un extremo de la comunicación bidireccional entre dos procesos. Ofrece una interfaz para acceder a los servicios de red en el nivel de transporte de los protocolos TCP/IP.

Actualmente, la interfaz de sockets está siendo estandarizada dentro de POSIX y está disponible en prácticamente todos los sistemas UNIX. También existe una interfaz de sockets para entornos Windows, denominada WinSockets; y se encuentra disponible como una clase nativa en el lenguaje de programación Java.

Se puede decir que los sockets constituyen la interfaz de programación de aplicaciones más utilizada para el desarrollo de aplicaciones en Internet.

El modelo se conectará, por tanto, con el resto de agentes a través de sockets mediante el protocolo TCP/IP (Figura 35) con los métodos de la clase Tcp (que hereda de la clase SocketNode).

# TCP



Figura 35 : Comunicación TCP

El modelo se conectará, por tanto, con el resto de agentes a través de sockets mediante el protocolo TCP/IP (Figura 35) con los métodos de la clase Tcp (que hereda de la clase SocketNode).

La clase SocketNode, escrita en C++, permite una buena gestión de los sockets y favorece la modularidad. Su objetivo es el de simplificar la inicialización de las comunicaciones y ocultar muchas comprobaciones de errores que requiere un buen manejo de los sockets. Así, la comunicación entre los módulos se puede hacer de forma más eficiente.

Los métodos de SocketNode (heredados por Tcp) y métodos de Tcp que requieren explicación para este proyecto son:

- Init, para iniciar la comunicación por sockets, indicándoles la dirección, el puerto y si se llama a la función en modo cliente o en modo servidor.
- StartThread, que crea un thread que se ocupa de las conexiones. Este thread será un bucle infinito que realizará comprobaciones en la conexión, y manejará las conexiones y el envío y recibo de mensajes a través de los métodos indicados para ello.
- HandleConnection, llamado desde el thread al realizar comprobaciones, se encarga de restablecer la conexión en caso de desconexión.
- IsConnected, utilizado para saber el estado de la conexión, devolviendo 1 o 0 si existe o no existe la conexión.
- SendMsg, utilizado para el envío de mensajes, adaptándolos al formato establecido, de forma que el primer byte contiene la cabecera, el segundo la longitud de la cadena enviada, y el resto el mensaje propiamente dicho.
- RecvMsg, para recibir los mensajes enviados por SenMsg, analizando los distintos bytes según lo explicado.
- OnMsg, esta función se llama cada vez que se reciba un mensaje, para notificar al proceso y actuar en consecuencia.
- OnConnection, sirve para indicar en el hilo principal de la ejecución del programa que ha habido un cambio en el estado de la conexión. En conjunto con IsConnected, se puede saber si la conexión se ha establecido o por lo contrario, se ha perdido, y facilitar al programa que actúe en consecuencia.

# 6 ENSAYOS Y RESULTADOS

---

## 6.1. Ensayos realizados

---

Los ensayos realizados con el modelo siguen un procedimiento de toma de decisión de parámetros (valores de referencia y objetivos emocionales), aplicación de estímulos y observación de resultados. Posteriormente, se modifican los parámetros o se realizan ajustes y se vuelve a realizar el procedimiento.

Los objetivos de los ensayos son:

- Determinación de valores emocionales de referencia, con el objetivo de llevar a cabo el cálculo del universo emocional de forma lógica, y poder obtener modificadores en base a esos cálculos.
- Obtención y ajuste de los tiempos de procesamiento de los threads del programa principal, con el fin de ahorrar recursos eliminando cálculos redundantes.
- Evaluación de relaciones entre los modificadores globales y su efecto en la evolución de parámetros del sistema, mediante correcciones en la matriz de ajuste de modificadores globales.
- Análisis de la escalabilidad, adaptabilidad y robustez del modelo.

Para realizar estos ensayos, el programa incluye en su versión de test, la muestra por pantalla de los valores del modelo de espacio de estados, así como los modificadores globales y el cálculo del universo emocional.

También existe la posibilidad de guardar todos los datos en ficheros para su posterior evaluación, sirviendo así como memoria del historial de versiones.

En cuanto al ensayo físico integrado en el robot, requiere de desarrollo conjunto entre el modelo y el resto de agentes, para que un estado emocional determinado lleve asociado una expresión social determinada.

También es necesaria la coordinación entre acciones para alcanzar los objetivos emocionales y los programas de gestos de brazo, tratamiento del habla y reconocimiento de voz.

## 6.2. Resultados obtenidos

Para evaluar los resultados obtenidos, como se ha explicado en el apartado anterior, se han utilizado los datos mostrados pro pantalla y los guardados en registros en archivos de texto.

Pero también cabe destacar el uso de un programa creado anteriormente en el proyecto URBANO, que ha permitido el apoyo a la simulación, al mostrar de una forma gráfica la variación de algunas emociones respecto al tiempo a la hora de aplicar estímulos y modificadores.

El programa es una interfaz gráfica (Figura 36), comunicada con el modelo emocional, que mediante una serie de sliders permite modificar los estímulos y observar las evoluciones temporales mencionadas.

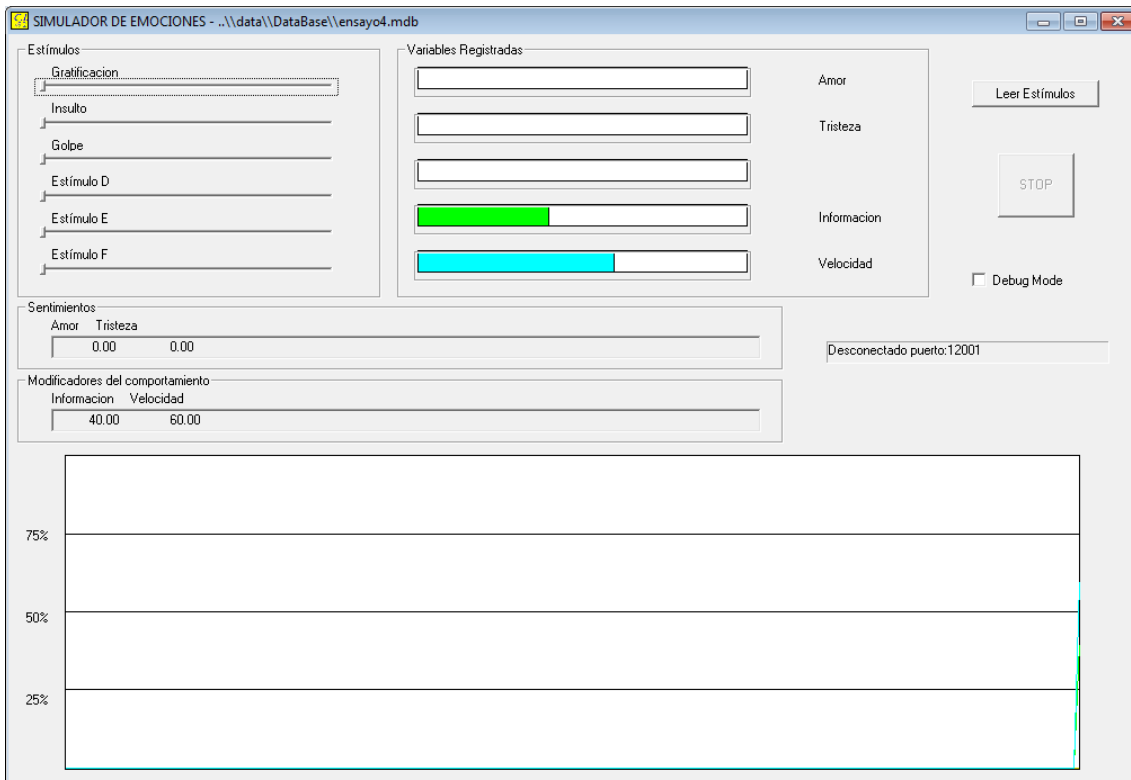


Figura 36 : Interfaz de usuario de programa de apoyo a la simulación



La simulación en conjunto con dicha interfaz tiene una serie de limitaciones, ya que el número de estímulos está limitado, de forma que para adaptarlo a nuevos sistemas habría que o bien adaptar este programa o bien realizar las simulaciones con otros métodos.

La principal ventaja de esta herramienta es la facilidad de manejo de parámetros para realizar pruebas aún sin estar familiarizado con el funcionamiento del modelo emocional, ya que prescindiendo de esta interfaz, resulta tedioso manejar los datos de entrada/salida por consola o ficheros en usuarios no experimentados.

En cuanto a la definición de resultados, se clasifican en valores de referencia, tiempos de procesamiento, modificadores globales y su efecto, y escalabilidad, adaptabilidad y robustez del modelo (los resultados respecto a este último punto se analizarán en las conclusiones, sección 7.1).

Los valores de referencia iniciales se han establecido en el punto medio del rango de valores que pueden alcanzar las emociones. Esto supone un punto a priori extraño ya que se estima el mismo valor para emociones que pueden estar relacionadas o no, incluso ser opuestas. Pero facilita el cálculo del universo emocional mediante la simplificación de esos valores un mismo valor común.

Hay que tener en cuenta que una vez que el robot tenga establecidos unos objetivos emocionales, estos marcarán los valores de referencia, para que en el cálculo del universo emocional se compare el error de los valores en cada momento con los objetivos, y su derivada.

Los tiempos de procesamiento han resultado sencillos de obtener, comprobando que se evitan duplicidades en el recálculo de las matrices y vectores del modelo de espacio de estado, calculando un único universo emocional en cada ocasión, y transmitiendo datos cada vez que se actualicen los modificadores globales.

Por tanto, la velocidad e ejecución de los hilos permite captar todos los estímulos recibidos, tratar esa serie de estímulos, usarlos en el cálculo y transmitir los resultados en forma de modificadores globales, sin que haya problemas de acceso a datos comunes por dos hilos (con el uso de lock y unlock) y ahorrando cálculos innecesarios.

Los modificadores globales se calculan en base al universo emocional, su efecto es notable ya que modifican el sistema completo, y cuando se establecen unos objetivos emocionales evolucionan correctamente para alcanzar esos objetivos.

Sin embargo, el sistema funciona de forma óptima con un solo modificador, ya que en el caso de dos o más, evolucionan de forma inversa, con cambios bruscos, provocando oscilaciones entorno a los valores objetivos, alcanzado finalmente los valores en tiempos similares a un solo modificador, pero con el peligro de inestabilidad y unos requerimientos mayores en capacidad de computación.

# 7 CONCLUSIONES Y LÍNEAS FUTURAS

---

## 7.1 Conclusiones

---

Sin embargo, el sistema funciona de forma óptima con un solo modificador, ya que en el caso de dos o más, evolucionan de forma inversa, con cambios bruscos, provocando oscilaciones entorno a los valores objetivos, alcanzado finalmente los valores en tiempos similares a un solo modificador, pero con el peligro de inestabilidad y unos requerimientos mayores en capacidad de computación.

Durante todo el periodo de desarrollo del trabajo se ha mantenido la vista en el objetivo de conseguir un modelo emocional robusto, fiable, adaptable y ampliable.

A pesar de la existencia de otros proyectos similares, incluidos algunos relacionados con el robot URBANO, el modelo emocional diseñado presenta diversas ventajas frente a otros programas, destacando por su efectividad a pesar de la simplicidad de manejo respecto al resto de opciones.

El modelo es fácilmente ampliable, tan sólo es necesario editar el archivo de configuración para incluir el número de estímulos, emociones y modificadores globales con los que sea necesario trabajar.

Por lo tanto, es innegable la sencillez de adaptabilidad en el caso de utilizar el programa para distintos robots sociales en nuevos proyectos.

Los parámetros incluidos en el modelo son fácilmente entendibles, ya que se basan en el conjunto estímulos (entradas), emociones (estados) y modificadores (salidas).

Esta sencillez relativa, provoca ventajas de cálculo respecto a otras opciones anteriores, ya que los modificadores se encargan de regular el comportamiento en base a los valores obtenidos en el modelo de espacio de estados y el error respecto a los objetivos emocionales.

En cuanto a la robustez y fiabilidad, el modelo responde correctamente a los ensayos de variación de parámetros realizado, por lo que no se consideran fallos de funcionamiento en usuarios inexpertos en el funcionamiento.

En el caso de usuarios que analicen el código del programa para realizar modificaciones de adaptación (aunque como ya se ha explicado, para adaptar a cada tipo de robot basta con modificar el número de parámetros), resulta cómoda la estructura multithread del programa, pudiendo modificar cada uno de los hilos de ejecución de forma específica.

## 7.2 Líneas futuras

---

Para analizar las líneas futuras de desarrollo que nacen de la realización de este trabajo, es necesario diferenciar entre las mejoras particulares en el modelo emocional y mejoras en la adaptación a distintos sistemas.

En cuanto a las mejoras del primer grupo, desde el inicio del proyecto se presentó la posibilidad futura de incluir algoritmos genéticos, para mejorar el aprendizaje del modelo emocional.

Estos algoritmos genéticos hacen evolucionar el sistema sometiéndolo a acciones aleatorias semejantes, seleccionando opciones de acuerdo a un criterio y combinando las soluciones posibles para alcanzar una solución muy cercana a la óptima a base de una serie de iteraciones.

Estos algoritmos presentan la ventaja de obtener una solución muy cercana a la óptima, a través de una serie de pasos organizados.

También tienen una serie de desventajas, siendo las más notables los grandes requerimientos de recursos en problemas complejos y la ambigüedad de la solución obtenida (la mejor respecto al resto de soluciones, no en base a una solución específica), lo que dificulta los criterios en cuanto a número de iteraciones.

Otra de las líneas de trabajo en relación a la de modelo es la realización de varios modos de funcionamiento (por ejemplo, modo trabajo y modo libre), que actúen sobre los parámetros.

Este punto requiere de un estudio en profundidad y de un trabajo en conjunto con modelos de comportamiento humano, para analizar exhaustivamente las diferencias entre las distintas vías de comportamiento en función de la situación en la que se encuentre el humano.

En relación al punto anterior, y también relacionado con distintas variaciones de parámetros en función de modos de funcionamiento, hay que trabajar en los distintos estímulos.

Una de las futuras incorporaciones al modelo es la forma de tratar los estímulos (actualmente se tratan todos de la misma forma) en función de si son visuales, auditivos, físicos, o internos (por ejemplo, mediante la integración de recuerdos en el robot, que puedan modificar el estado de ánimo).

En cuanto a la mejora de adaptación a sistemas, el primer punto obvio es mantener la adaptabilidad y ampliabilidad del modelo aún integrando las mejoras anteriormente citadas.

El primer punto importante de adaptación es la actualización, tanto del modelo emocional como de otros agentes, a versiones modernas que trabajen en sinergia con las nuevas versiones de software (por ejemplo, con Visual Studio 2015).

Este tipo de actualización es importante de cara a la integración en proyectos modernos, que vayan surgiendo, y la utilización de nuevas versiones de sistemas operativos.

El otro punto importante de mejora en la integración es la comunicación. Actualmente usa un protocolo de comunicación muy específico, mediante el uso de clases explicadas a lo largo del proyecto, y con una codificación muy específica (ver sección 5.2).

Una de las líneas futuras puede ser la adaptación de el modelo emocional a distintos tipos de protocolos de comunicación, sin necesidad de que el usuario trabajador de un proyecto robótico tenga que editar gran parte del thread de comunicaciones, o incluso crear e integrar clases propias para poder adaptar la comunicación a su robot.

En cuanto a las líneas de investigación llevadas a cabo por el grupo de control inteligente, en relación al estudio de las emociones robóticas, uno de los puntos más importantes es el control del tiempo libre.

En este estudio, habrá que reanalizar los puntos en común entre la psicología de las emociones humanas y las emociones robóticas.

Por ejemplo, cuando el robot en su modo tiempo libre no llega a los niveles de felicidad objetivo, puede tomar decisiones como pasear, poner música, pensar o aprender algo nuevo. Estas decisiones deben ser tomadas cuando se pasa en un mismo estado demasiado tiempo, sin recibir feedback positivo.

A priori es un campo que denota sencillez de acción-reacción (no soy feliz, hago algo), pero requiere un estudio profundo de las motivaciones que llevan a realizar esas acciones.

Por otra parte, la implementación de un sistema de este tipo no es para nada trivial, ya que requerirá de un sistema de selección entre un campo de acciones para recuperar el estado de felicidad.

Una de las formas de implementación será mediante la modificación en tiempo real de los parámetros de la matriz de ajuste para volver a ese estado de felicidad, aunque no es la única de la solución.

Otra solución de mayor complejidad, es la introducción de nuevos estímulos durante el modo funcionamiento del robot, es decir, mientras está operativo, el robot asume nuevos estímulos para intentar alcanzar los objetivos emocionales.

Finalmente, una posible futura solución sería el ensayo y error, de forma que cuando el robot no está emocionalmente en su objetivo, prueba a realizar una serie de acciones para intentar mejorar ese estado. Estos datos se pueden almacenar para que el robot tenga un histórico de acciones y consecuencias, con el fin de que sus pruebas tengan cada vez consecuencias más positivas.

En resumen, las líneas futuras, aún clasificadas en estas dos consideraciones son muy cuantiosas, amplias y diversas, dado que la robótica social es un campo en auge que necesitará muchos proyectos de investigación y desarrollo durante los próximos años, entre los que se incluyen los modelos emocionales cada vez más avanzados para conseguir interacciones hombre-robot exitosas.

# 8 PLANIFICACIÓN TEMPORAL Y PRESUPUESTO

---

En este apartado se analiza la gestión y desarrollo del proyecto, tanto de manera temporal como económica. Este capítulo está dividido en la estructura de descomposición del proyecto (EDP), la planificación temporal del mismo y el presupuesto utilizado para su desarrollo.

Se utiliza la herramienta de la EDP, diagrama de Gantt y presupuesto dividido en tablas para facilitar la explicación y comprensión a grandes rasgos del alcance del proyecto.

## 8.1. Estructura de descomposición del proyecto (EDP)

---

Para la realización del proyecto, la estructura de descomposición del proyecto está dividida en 6 grupos de tareas, desde la fase planificación a la de resultados, numerados del 0 al 5.

0. Planificación y gestión
1. Estudios previos
2. Diseño del programa
3. Desarrollo y mejora del programa
4. Implementación y comunicación
5. Ensayos del modelo

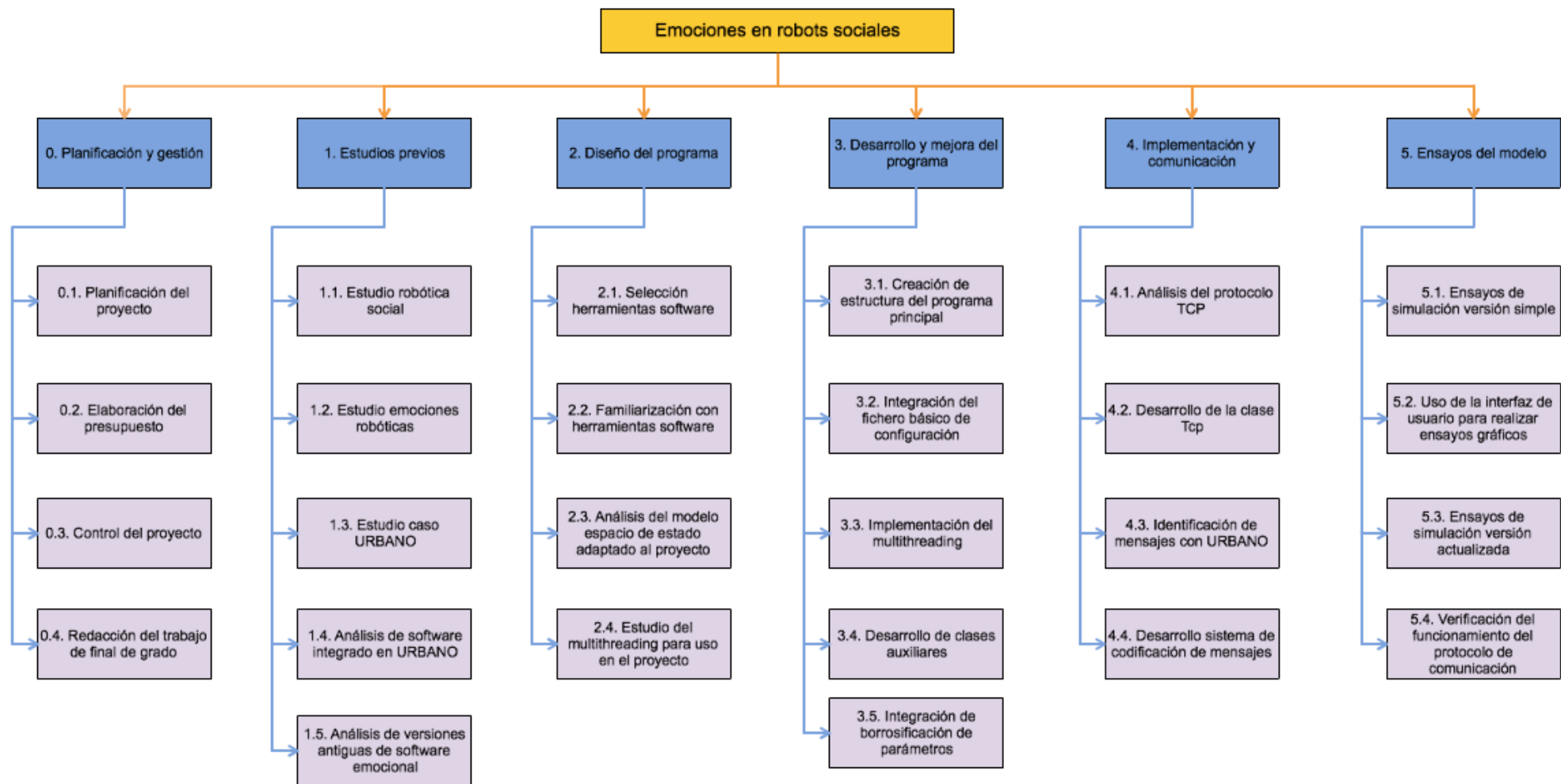


Figura 37 : Estructura de descomposición del proyecto

## 8.2. Planificación temporal

---

A partir de la estructura de descomposición del proyecto, se desarrolla una distribución temporal de las tareas que aparecen en la EDP.

La programación temporal del proyecto está realizada mediante un diagrama de Gantt, en el que se muestran todos los grupos de tareas (y las tareas incluidas en cada grupo) a lo largo de toda la distribución temporal del proyecto.

El desarrollo del proyecto se distribuye a lo largo de siete meses, en el período de Diciembre-Junio, ambos inclusive.

Dada la naturaleza del trabajo, se dedican los primeros meses a los estudios previos, con el fin de conocer a fondo el tema a trabajar, así como a familiarizarse con algunas de las herramientas ya existente en proyectos anteriores.

En el diagrama de Gantt se puede ver sombreado cada grupo de tareas en un periodo específico de tiempo, que corresponde a la focalización en dicho objetivo. También sombreado de forma menos intensa el período de realización de cada una de las tareas dentro de los grupos.

A pesar de ello, y de nuevo, dada la naturaleza del trabajo, este hecho no significa que cada tarea se ha realizado de forma exclusiva en dicho periodo, ya que resulta necesario por ejemplo, continuar con el estudio a pesar de estar focalizado en la fase de desarrollo.

Finalmente, se han coloreado de forma leve los dos periodos de exámenes, durante la segunda y tercera semana de enero, y durante la última semana de mayo y la primera de junio, ya que se considera que durante esos días los exámenes se llevan la mayor parte del esfuerzo frente al desarrollo del proyecto.



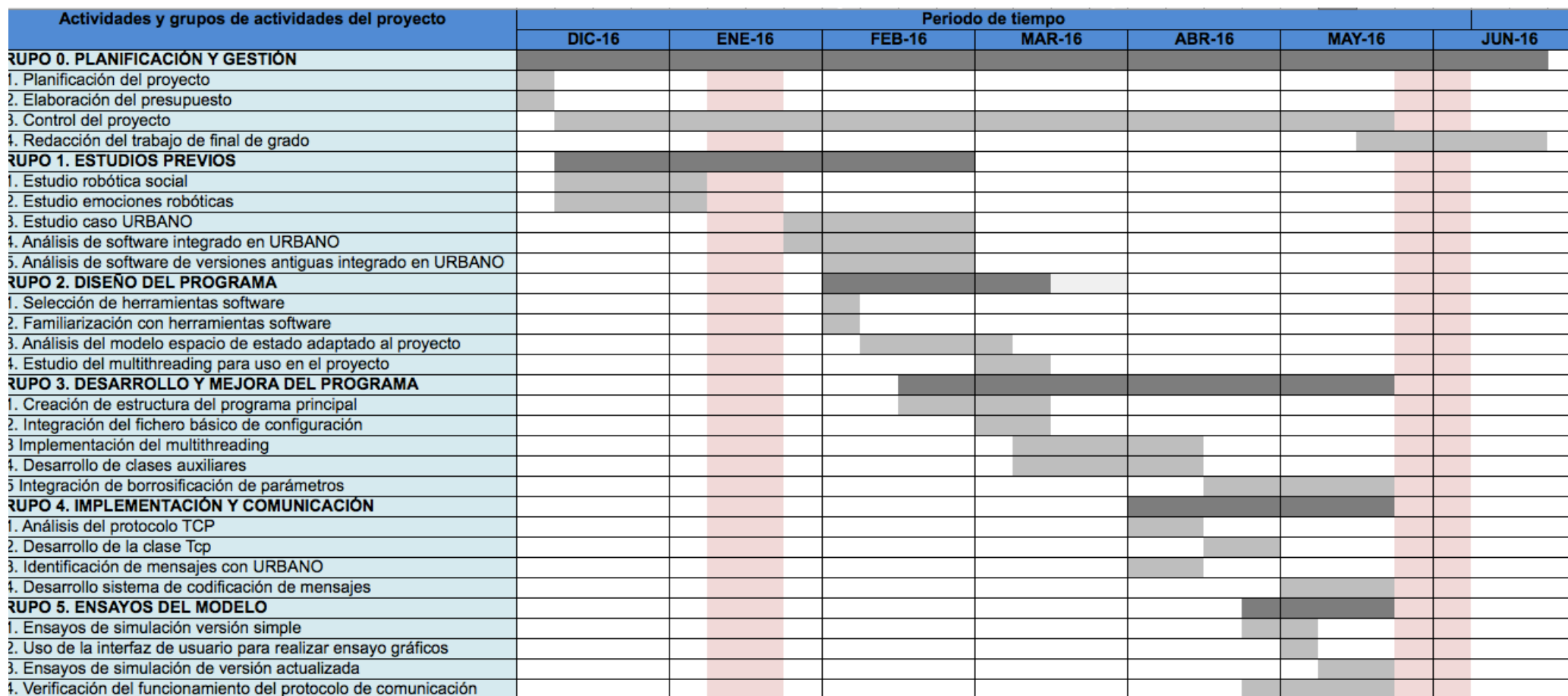


Figura 38 : Diagrama de Gantt del proyecto

### 8.3. Presupuesto

Para el cálculo del presupuesto se han considerado individualmente los gastos de personal y los gastos de material, entre los que se incluye el equipo de trabajo y el software utilizado.

Concepto	Cuantía	Duración	Total
<b>Gastos de personal</b>	300,00€	7 meses	2100,00€

Figura 39: Gastos de personal

Concepto	Coste unitario	Unidades	Amortización (%)	Total
<b>Lenovo G-50 i7-5500U 16Gb RAM R5 M330 2GB</b>	799,00€	1	10	79,90€
<b>Visual Studio 2013</b>	0,00€	1	100	0,00€
<b>Git</b>	0,00€	1	100	0,00€
<b>TOTAL MATERIAL</b>				79,90€

Figura 40: Gastos de material

<b>GASTOS PERSONAL</b>	2100,00€
<b>GASTOS DE MATERIAL</b>	79,90€
<b>TOTAL</b>	<b>2179,90€</b>

Figura 41 : Gastos totales

# BIBLIOGRAFÍA

---

- [1] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Oxford, 1999.
  
- [2] R. Beckers, et al., From local actions to global tasks: Stigmergy and collective robotics, in: *Proceedings of Artificial Life IV*, 1996.
  
- [3] J.-L. Deneubourg, et al., The dynamic of collective sorting robot-like ants and ant-like robots, in: *Proceedings of the International Conference on Simulation of Adaptive Behavior*, 2000.
  
- [4] C. Kube, E. Bonabeau, Cooperative transport by ants and robots, *Robotics and Autonomous Systems* 30 (2000) 85–101.
  
- [5] C. Melhuish, O. Holland, S. Hoddell, Collective sorting and segregation in robots with minimal sensing, in: *Proceedings of the International Conference on Simulation of Adaptive Behavior*, 1998.
  
- [6] A. Billard, K. Dautenhahn, Grounding communication in situated, social robots, in: *Proceedings Towards Intelligent Mobile Robots Conference*, Report No. UMCS-97-9-1, Department of Computer Science, Manchester University, 1997.
  
- [7] A. Billard, K. Dautenhahn, Grounding communication in autonomous robots: An experimental study, *Robotics and Autonomous Systems* 24 (1–2) (1998) 71–81.
  
- [8] Breazeal, C., 2002a. *Designing Sociable Robots*. MIT Press, Cambridge, MA.
  
- [9] B. Duffy, Anthropomorphism and the social robot, *Robotics and Autonomous Systems* 42 (2003) 177–190 (this issue).

- [10] J. Reichard, *Robotics: Fact, Fiction, and Prediction*, Viking Press, 1978.
- [11] Russell, J., 1997. Reading emotions from and into faces: resurrecting a dimensional–contextual perspective. In: Russell, J., Fernandez-Dols, J. (Eds.), *The Psychology of Facial Expression*. Cambridge University Press, Cambridge, UK, pp. 295–320.
- [12] R.A. Brooks, *Cambrian intelligence*, MIT Press, Cambridge, MA; 1999.
- [13] M. Scheeff, et al., Experiences with Sparky: A social robot, in: *Proceedings of the Workshop on Interactive Robot Entertainment*, 2000.
- [14] L. Cañamero, J. Fredslund, I show you how I like you— can you read it in my face? *IEEE Transactions on Systems, Man and Cybernetics* 31 (5) (2001).
- [15] K. Dautenhahn, “The art of designing socially intelligent agents- science, fiction, and the human in the loop” *Applications of Artificial Intelligence* 12, 1998, pp. 573–617.
- [16] C. Breazeal, “Social interaction in HRI: the robot view” *IEEE Transactions on System, Man and Cybernetics: Part C* 34, 2004, pp. 181–186.
- [17] C. Breazeal, *Designing sociable robots*, MIT Press, Cambridge, MA; 2002
- [18] C. Breazeal, “Towards sociable robots” *Robotics and Autonomous Systems* 42, 2003, pp. 167–175

- [19] Gates, Bill (2007): Un robot en cada casa. Revista Investigación y Ciencia, marzo, N° 366, p. 38-45. Barcelona, Editorial Prensa Científica.
- [20] T. Mukai, S. Hirano, H. Nakashima, Y. Kato, Y. Sakaida, S. Guo, and S. Hosoe, "Development of a nursing- Care Assistant Robot RIBA That Can Lift a Human in Its Arms", IEEE/RSJ International Conference on Intelligent Robots and Systems, Taiwan, pp. 5996-6001, October 2010.
- [21] Torbjørn S. Dahl ;Maged N. Kamel Boulos ; Robots in Health and Social Care: A Complementary Technology to Home Care and Telehealthcare?
- [22] Frennert, S.A.; Forsberg, A.; Östlund, B. Elderly people's perceptions of a telehealthcare system: Relative advantage, compability, complexity and observability. J. Technol. Hum. Serv. 2013, 31, 218–237.
- [23] Kroh, M.; El-Hayek, K.; Rosenblatt, S.; Chand, B.; Escobar, P.; Kaouk, J.; Chalikonda, S. First human surgery with a novel single-port robotic system: Cholecystectomy using the da Vinci single-site platform. Surg. Endosc. 2011, 25, 3566–3573.
- [24] Tabatabaei, S.N.; Lapointe, J.; Martel, S. Shrinkable hydrogel-based magnetic microrobots for interventions in the vascular network. Adv. Robot. 2011, 25, 1049–1067
- [25] Galindo C, Cruz-Martin A, Blanco J.L, Fernández- Madrigal J.A, Gonzalez J (2006) A multi-agent control architecture for a robotic wheelchair. Applied Bionics & Biomechanics. 3: 179-189.
- [26] Er-Hsu P,Liang-Hsu Y, Chang K., Geiser C (2012) Mobility Assistance Design of the Intelligent Robotic Wheelchair
- [27] G. Lacey and K. M. Dawson-Howe, "The Application of Robotics to a Mobility Aid for the Elderly Blind," Robotics and Auton. Systems, no. 23, pp. 245-252, 1998.

- [28] H. Mori and S. Kotani, "A Robotic Travel Aid for the Blind – Attention and Custom for Safe Behavior," *Int. Symp. Rob. Research*, Springer-Verlag, pp.237-245, 1998.
- [29] R.D. Schraft, C. Schaeffer and T. May, "Care-O- bot(tm): The Concept of a System for Assisting Elderly or Disabled Persons in Home Environments," *IECON: Proc. of the IEEE 24th Annual Conf.*, V.4, 1998, pp.2476-2481
- [30] Dillenbourg, P.: Design for classroom orchestration. *Computers and Education* 69, 485–492 (2013)
- [31] Kanda, T., Hirano, T., Eaton, D., Ishiguro, H.: Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial. *Human-Computer Interaction* 19(1), 61–84 (Jun 2004)
- [32] Kanda,T.,Sato,R.,Saiwaki,N.,Ishiguro,H; A two-month field trial in an elementary school for long-term human-robot interaction. *IEEE Transactions on Robotics* 23(5), 962–971 (2007)
- [33] W. Burgard, A.B., Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. *AAAI-98*.
- [34] Schulte, C. Rosenberg, and S. Thrun. Spontaneous short-term interaction with mobile robots in public places. Same volume.
- [35] Costa S, Santos C, Soares F, Ferreira M, Moreira F. Promoting interaction amongst autistic adolescents using robots; 32nd Annual International Conference of the IEEE/EMBS; Buenos Aires, Argentina. 2010. pp. 3856–3859
- [36] Dautenhahn K. Roles and functions of robots in human society: Implications from research in autism therapy. *Robotica*. 2003;21:443–452.

- [37] Dautenhahn K, Werry I. Towards interactive robots in autism therapy: Background motivation, and challenges. *Pragmatics & Cognition*. 2004;12:1–35.
- [38] De Silva PRS, Tadano K, Saito A, Lambacher SG, Higashi M. Therapeutic-assisted robot for children with autism; *IEEE/RSJ International Conference on Intelligent Robots and Systems*; New York, NY: ACM Press; 2009. pp. 3561–3567.
- [39] W.W. Gaver, J. Bowers, A. Boucher, H. Gellerson, S. Pennington, A. Schmidt, A. Steed, N. Villars, and B. Walker, “The drift table: designing for ludic engagement,”
- [40] L.A. Suchman, *Plans and Situated Actions: The Problem of Human- Machine Communication*, Cambridge University Press, 1987.
- [41] M. Jacobsson, Y. Fernaeus, and L.E. Holmquist, “GlowBots: Designing and Implementing Engaging Human-Robot Interaction,” *Journal of Physical Agents*, vol. 2, 2008.
- [42] F. Kaplan, “Everyday robotics: robots as everyday objects,” *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*
- [43] .F. Melson, J. Peter H. Kahn, A.M. Beck, B. Friedman, T. Roberts, and E. Garrett, “Robots as dogs?: children's interactions with the robotic dog AIBO and a live australian shepherd”
- [44] A.G. Brooks, J. Gray, G. Hoffman, A. Lockerd, H. Lee, and C. Breazeal, “Robot's play: interactive games with sociable machines,” *Comput. Entertain.*, vol. 2, 2004, pp. 10-10.

- [45] Y. Fernaeus and M. Jacobsson, "Comics, Robots, Fashion and Programming: outlining the concept of actDresses," *TEI'09*, Cambridge, UK: 2009.
- [46] Rodríguez-Losada, Diego. SLAM geométrico en tiempo real para robots móviles en interiores basado en EKF. Madrid, España, 2004.
- [47] Modelo Emocional para Robot Guía. Madrid, España, 2010.
- [48] Miguel A. Salichs, Ramón Barber, Alaa M. Khamis, María Malfaz, Javier F. Gorostiza, Rakel Pacheco, Rafael Rivas, Ana Corrales, Elena Delgado, David García; Maggie: A Robotic Platform for Human-Robot Social Interaction
- [49] Francesco Ferro, Sergio Garcia, Esteban Gomez. REEM-B, An autonomous lightweight human-size humanoid robot
- [50] Chapman, Davis. *Teach yourself Visual C++ 6 in 21 days*. Indiana: Macmillan Computer Publishing, 1998.
- [51] Bustamante, Paul, Iker Aguinaga, Miguel Aybar, Luis Olaizola, y Iñigo Lazacano. *Aprenda C++ Básico como si estuviera en primero*. Navarra, 2004.
- [52] Bustamante, Paul, Iker Aguinaga, Miguel Aybar, Luis Olaizola, y Iñigo Lazacano. *Aprenda C++ avanzado como si estuviera en primero*. Navarra, 2004
- [53] Mini-Introducción a Git. 2007. <http://geneura.ugr.es/~jmerelo/tutoriales/git/>.
- [54] C. Zambrano J., I. González A, Implementación de un algoritmo de control predictivo en espacio de estados sobre una plataforma de simulación desarrollada en Matlab



- [55] C. R. Cutler and B. Ramaker, "Dynamic matrix control—a computer control algorithm," in *Proceedings of the joint automatic control conference*, vol. 1. University of Michigan Ann Arbor, MI, 1980, pp. Wp5–B.
- [56] T. Bergan, O. Anderson, J. Devietti, L. Ceze, and D. Grossman. CoreDet: a compiler and runtime system for deterministic multithreaded execution. In *Proceedings of the fifteenth edition of ASPLOS on Architectural support for programming languages and operating systems*
- [57] E. D. Berger, T. Yang, T. Liu, and G. Novark. Grace: safe multithreaded programming for C/C++. In *OOPSLA '09: Proceeding of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications*
- [58] C. Bienia and K. Li. Parsec 2.0: A new benchmark suite for chip multiprocessors. In *Proceedings of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, June 2009.
- [59] H. Cui, J. Wu, C. Tsa, and J. Yang. Stable deterministic multithreaded through schedule memoization. In *OSDI'10: Proceedings of the 9th Conference on Symposium on Operating Systems Design & Implementation*
- [60] T. J. LeBlanc and J. M. Mellor-Crummey. Debugging parallel programs with instant replay. *IEEE Trans. Comput.*, 36:471–482, April 1987.
- [61] C. E. McDowell and D. P. Helmbold. Debugging concurrent programs. *ACM Comput. Surv.*, 21(4):593–622, 1989.
- [62] M. Olszewski, J. Ansel, and S. Amarasinghe. Kendo: efficient deterministic multithreading in software. In *ASPLOS '09: Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 97–108, New York, NY, USA, 2009. ACM.

- [63] J. W. Havender. Avoiding deadlock in multitasking systems. *IBM Systems Journal*, 7(2):74–84, 1968.
- [64] J. Pool, I. Sin, and D. Lie. Relaxed determinism: Making redundant execution on multiprocessors practical. In *Proceedings of the 11th Workshop on Hot Topics in Operating Systems (HotOS 2007)*, May 2007.
- [65] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis. Evaluating MapReduce for multi-core and multi-processor systems. In *HPCA '07: Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture*, pages 13–24, Washington, DC, USA, 2007. IEEE Computer Society.
- [66] T. Finin, Y. Labrou, J. Mayfield, “KQML as an Agent Communication Language,” in *Software Agents*, ed. J. M. Bradshaw, AAAI Press/The MIT Press, pp. 291–316, 1997.
- [67] M. R. Genesereth, S. P. Ketchpel, “Software Agents,” *Communications of the ACM*, vol. 37, no. 7, pp. 48–53 and 147, 1994.
- [68] M. W. Gertz, D. B. Stewart, P. K. Khosla, “A Software Architecture-Based Human-Machine Interface for Reconfigurable Sensor-Based Control Systems,” *Proc. of the IEEE Int. Symposium on Intelligent Control*, Chicago, Illinois, USA, 1993, pp. 75–80.
- [69] Arai Tamio, Pagello Enrico and Parker Lynne E, "Editorial: Advances in Multi-Robot Systems", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655-661, 2002
- [70] C.H. Lin and K.T. Song, "Flexible Real-Time Control of Home Robots Using a Multi-Agent Based Approach", *Proc. 2004 IEEE/RSJ Int. Conf on Intelligent Robots and Systems*, pp. 3092

- [71] Anderson, Christa. "Mapping Practice to Theory: NT Networking and the OSI Model." *Windows 2000 Magazine*. 1 March 1999.
- [72] Tan Teng Hong, Andrew; Chee Meng, Mah; Yew Wai, Chee; Yoke Chuan, Tan; Kim Ming, Cheong; "Comparing OSI and TCP/IP."
- [73] Rohlin, Robert W. "OSI Model: Upper Layers". (31 July 2001).
- [74] M. Leech. Key management considerations for the TCP MD5 signature option. RFC 3562, Internet Engineering Task Force, July 2003.
- [75] W. R. Stevens, *TCP/IP Illustrated Vol. 1 – The Protocols*, Addison-Wesley, 1994.
- [76] D. E. Comer, *Internetworking with TCP/IP: Vol. 1 – Principles, Protocols and Architecture*, Third Edition, Prentice Hall, 1995.