



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

**Modelado de emociones**  
**en el**  
**robot bípedo NAO**

Trabajo Fin de Grado  
**Grado en Ingeniería Informática**

**Autor:** Carrasquer Garcia, Rosa

**Tutor:** Julián Inglada, Vicente Javier

**Cotutor:** Carrascosa Casamayor, Carlos

2015-2016



## **Resumen**

En la actualidad cada vez se investiga más en el ámbito de la robótica y la Inteligencia Artificial. Se desarrollan dispositivos capaces de cocinar, limpiar una casa o incluso supervisar a personas mayores o niños.

En esta última área, una de las cosas a tener en cuenta para la toma de decisiones del sistema de Inteligencia Artificial debería ser el estado de ánimo de la persona con la que se trata, así pues el presente proyecto tratará de desarrollar un sistema software que pueda evaluar el estado de ánimo de las distintas personas con las que interaccione.

Para este proyecto se trabajará con un robot humanoide llamado NAO, desarrollado por la empresa francesa Aldebaran Robotics que será el soporte físico de nuestro software.

**Palabras clave:** robótica, Inteligencia Artificial, estado de ánimo, sistema software

---

## **Resum**

En l'actualitat cada vegada s'investiga més en l'àmbit de la robòtica i la Intel·ligència Artificial. Es desenvolupen dispositius capaços de cuinar, netejar una casa o inclòs supervisar a persones majors o xiquets.

En aquesta última àrea, una de les coses a tenir en compte per a la presa de decisions del sistema de Intel·ligència Artificial haurien de ser l'estat d'ànim de la persona amb qui tracta, així, el present projecte tractarà de desenvolupar un sistema de programari que pugui avaluar l'estat d'ànim de les distintes persones amb qui interaccione.

Per a aquest projecte es treballarà amb un robot humanoide anomenat NAO, desenvolupat per l'empresa francesa Aldebaran Robotics que serà el suport físic del nostre programari.

**Paraules clau:** robòtica, Intel·ligència Artificial, estat d'ànim, sistema de programari

---

## **Abstract**

At present each time it investigates more in the field of robotics and Artificial Intelligence. Develop devices capable of cooking, cleaning a house or even monitor to elderly people or children.

In this last area, one of the things that have to be taken into account for the decisions making of the Artificial Intelligence system should be the state of mind of the person concerned, thus the present project will seek to develop a software system that can assess the state of mind of the different people with whom you interact.

For this project will work with a humanoid robot called NAO, developed by the French company Aldebaran Robotics that will be the physical support of our software.

**Key words:** robotics, Artificial Intelligence, state of mind, software system

---



## Índice general

<b>Índice de tablas</b> .....	8
<b>Índice de Figuras</b> .....	8
<b>Introducción</b> .....	10
Motivación.....	10
Objetivos.....	11
Estructura de la memoria.....	12
<b>Estado del arte</b> .....	14
Modelos emocionales.....	14
Reconocimiento de emociones mediante imágenes.....	16
Robot NAO.....	17
Choreographe.....	18
Python.....	20
Herramientas del Proyecto Oxford de Microsoft.....	21
<b>Descripción del problema</b> .....	25
Problemas propuestos.....	25
<b>Diseño del sistema</b> .....	28
FSM Maquina de estados finitos.....	28
Versiones.....	37
<b>Evaluación</b> .....	40
Sistema de reconocimiento facial.....	40
Sistema de reconocimiento de emociones.....	43
<b>Conclusiones</b> .....	49
Trabajo futuro.....	49
<b>Referencias</b> .....	52



## **Índice de tablas**

1	Conversión de las emociones al modelo PAD.....	16
2	Prueba de felicidad.....	44
3	Prueba de tristeza.....	44
4	Prueba de enfado.....	45
5	Prueba de sorpresa.....	45

---

## **Índice de figuras**

1	Escalas de PAD.....	15
2	Robot NAO.....	18
3	Interfaz Choreographe.....	19
4	Robot limpiador inteligente.....	25
5	Maquina de estados finitos.....	28
6	Versión 1.....	38
7	Versión 2.....	38
8	Prueba de funcionamiento 1.....	41
9	Prueba de funcionamiento 2.....	42
10	Prueba de funcionamiento 3.....	43

---



# **Introducción**

## **Motivación**

La informática se divide en muchas áreas. Las dos que interesa introducir para este proyecto son la robótica y la inteligencia artificial.

La robótica se ocupa del diseño y construcción de los robots, éstos son dispositivos físicos capaces de realizar tareas. Los robots se pueden manejar directamente por los humanos o pueden tener un sistema software que les permita realizar acciones automáticamente. Los robots se usan para realizar tareas que para un ser humano son peligrosas o tediosas.

La inteligencia artificial es un ámbito del software que se ocupa de desarrollar sistemas capaces de tomar decisiones y aprender. Dichos sistemas tienen diversas acciones implementadas pero su trabajo es elegir en cada momento cuál de esas acciones es la más adecuada para cada situación. Además estos sistemas aprenden durante el tiempo, almacenan las consecuencias de sus acciones en cada situación para en el futuro incorporar estas posibles consecuencias al cálculo de la mejor acción a realizar.

En este proyecto se trabaja con estas dos áreas de la informática, se utilizará el robot humanoide NAO que es programable y autónomo. Se usará como agente en el que se desarrollará nuestro software.

En la actualidad la robótica y la Inteligencia Artificial están en proceso de crecimiento y muchas empresas invierten en este área. Uno de los ámbitos que abarca este crecimiento son los sistemas sociales, es decir, que interaccionan con personas. Un ejemplo de esto podría ser un robot que cuida de una persona mayor ya sea trayéndole sus medicinas, vasos de agua o comida, como tomándole la temperatura o controlando las constantes vitales, para esto el robot debe de saber que es lo que necesita la persona en todo momento y además saber de que manera debería proceder teniendo en cuenta como se siente dicha persona mayor.

La posibilidad de que un robot esté constantemente recopilando información emocional de

una misma persona abre muchas puertas para la evaluación psicológica de las personas. Además estos sistemas ayudarán a hacer más preciso el reconocimiento emocional por los gestos que hacemos.

Cada vez más se introducen sistemas inteligentes en ámbitos sociales en los que tienen trato con personas, ya sea un sistema que juegue al ajedrez contra personas, como un robot que se encargue de la venta en un supermercado. A causa de este creciente trato con las personas en situaciones sociales es importante que en la toma de decisiones de estos sistemas se incorporen las emociones.

## **Objetivos**

El objetivo central de este proyecto es desarrollar un sistema para el robot bípedo NAO que reconozca a las personas con las que interactúa y a su vez reconozca las emociones que muestran dichas personas mediante el análisis de sus gestos faciales. Este proyecto se ha distribuido en distintos subobjetivos. A continuación se detallan estructurados por orden de desarrollo.

### **Subobjetivos**

En primer lugar, se estudiarán las distintas aplicaciones existentes que permiten realizar reconocimiento facial y análisis emocional, que utilizaría nuestro sistema para recopilar la información necesaria de las personas con las que interactúe. Se deberán estudiar dichas aplicaciones y al final decidir cuál de ellas se adapta mejor a lo que queremos hacer.

A continuación se desarrollará una herramienta que permita reconocer a las personas con las que interactúa. Este proceso se realizará mediante una foto que el robot tomará en tiempo real y utilizando la aplicación que hayamos seleccionado recuperará los datos de identificación de todas las personas que estén interactuando con él. También se encargará de crear nuevos registros en caso de que alguna persona no se reconozca en la base de datos.

Seguidamente se desarrollará una herramienta que mediante la misma foto con la que se

ha reconocido a las personas que interaccionen con el robot recupere las emociones asociadas a cada una de estas personas. También tendrá un registro de tiempo donde almacenará la evolución emocional de dichas personas a lo largo del tiempo.

Una vez elaboradas todas las herramientas necesarias se creará el sistema objeto de este proyecto mediante la herramienta *Choreographe*. Dicho sistema realizará las fotos oportunas e incorporará las dos herramientas citadas anteriormente para la recuperación tanto de la información de la identidad de las personas como de sus emociones y posteriormente utilizar una herramienta de visualización para analizar los resultados. La herramienta *Choreographe* de que se hará uso cuenta con diversas tareas prediseñadas. Algunas de estas permiten al robot NAO tomar fotografías y comunicarse con las personas que le rodean.

Por último se realizarán las pruebas necesarias para validar el sistema y evaluar su buen funcionamiento. En concreto se realizarán múltiples pruebas a lo largo del tiempo para ver la evolución de un mismo interlocutor, con varios interlocutores, pruebas con personas nuevas y comprobar que se añaden bien al sistema de reconocimiento o incluso pruebas simples para garantizar que el sistema reconoce a tantas personas como haya en la estancia.

## **Explicación de la memoria**

La estructura de la presente memoria es la siguiente. En el apartado del estado del arte se expondrá todo aquello que será necesario para implementar el sistema, además se comentarán algunos modelos emocionales así como herramientas que existen para el reconocimiento de emociones mediante imágenes.

Posteriormente en el apartado de diseño del sistema se explicará todo el proceso para implementar el sistema que nos ocupa, así como la creación de las distintas herramientas que se han nombrado en los subobjetivos y como se relacionarán entre ellas.

Finalmente la parte correspondiente a las validaciones se explicará más extensamente en el apartado de evaluación.



## Estado del arte

### **Modelos emocionales**

La emoción se define como un estado de ánimo caracterizado por una conmoción orgánica provocadas por impresiones de los sentidos, ideas o recuerdos, la cual produce fenómenos viscerales (cambios en nuestro cuerpo como puede ser la aceleración del pulso) que percibe el sujeto emocionado, y con frecuencia se traduce en gestos, actitudes u otras formas de expresión<sup>[1]</sup>. Estos estados de ánimos nos permiten tomar decisiones, tendiendo a acercarnos a las cosas que nos producen emociones positivas y alejarnos de aquello que nos provoca emociones negativas.

En nuestro caso buscamos poder dar nombre a distintas emociones y no quedarnos solo con si son alegres o tristes, vamos a intentar definir los sentimientos que aparecen como resultado de las emociones. Existen distintos sentimientos tales como enfado, felicidad, tristeza, sorpresa, disgusto... Como existen muchos y al ser algo difícil de enmarcar y calcular buscaremos algún modelo emocional con el que trabajar.

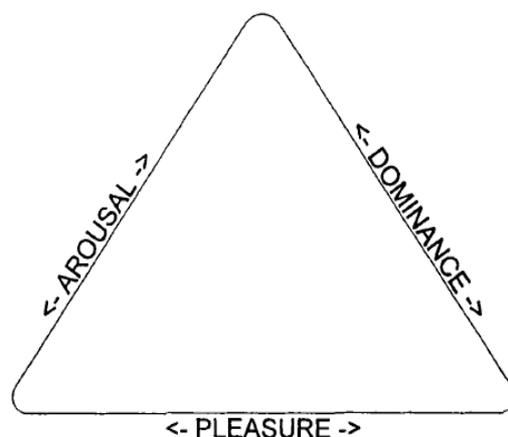
Los modelos emocionales son herramientas que se usan para la identificación de emociones. Al ser una ciencia inexacta existen modelos emocionales distintos que tratan los sentimientos de diferentes modos para determinar que emoción predomina en situaciones dadas. A continuación vamos a explicar dos modelos emocionales concretos como son el modelo emocional que utiliza Microsoft en sus sistemas y el modelo emocional *PAD*.

Microsoft utiliza un modelo emocional propio para enmarcar los sentimientos que siente alguien en un instante dado que se basa en dar unos valores a distintos sentimientos determinados. Este modelo emocional está basado en el *modelo OCC* (toma el nombre por la teoría de Ortony, Clore y Collins)<sup>[2]</sup> que funciona del mismo modo pero incluye muchos más sentimientos para darles valor. Estos sentimientos son tristeza, desprecio, neutral, disgusto, enfado, sorpresa, miedo y felicidad.

Cuando alguna herramienta de Microsoft analiza el estado emocional de alguien da un

valor numérico entre 0 y 1 a cada una de dichos sentimientos, siendo el valor más próximo a 1 el correspondiente al sentimiento más predominante y el valor más cercano a 0 el menos predominante. Esto se debe a que el cálculo de sentimientos en la actualidad no puede ser exacta puesto que no se pueden especificar científicamente los sentimientos y por tanto todos los cálculos que podamos realizar serán aproximaciones. Por otro lado las personas no sentimos de manera absoluta, esto quiere decir, en un mismo instante de tiempo podemos tener distintos sentimientos, unos más intensos y otros menos, por tanto en este modelo emocional se le puede dar valores a distintos sentimientos a la vez.

El modelo que se va a utilizar en el presente proyecto es el modelo emocional Placer-Excitación-Dominancia (Pleasure-Arousal-Dominance en inglés), al que nos referiremos a partir de ahora como el modelo *PAD*. Fue desarrollado por Albert Mehrabian y James A. Russell para poder describir y también medir las emociones<sup>[3]</sup>. Este modelo usa tres dimensiones numéricas que se corresponden con las tres palabras que componen la nomenclatura del modelo (ver figura 1).



**Figura 1.** Escalas de PAD

El modelo *PAD* utiliza escalas tridimensionales, cada escala mide características de los sentimientos asociadas a ella, de tal modo que la escala del placer mide el agrado hacia algo, por ejemplo el miedo es un sentimiento desagradable por tanto se posiciona a un extremo de la escala mientras que la alegría es agradable y se posicionará en el extremo opuesto. La escala de la excitación mide cuán intensa es una emoción, por ejemplo dos emociones como el aburrimiento o la ira son desagradables pero el primero tiene un valor de excitación más bajo que el otro. Por último la escala de dominancia se encarga de medir

el grado de sumisión que representa un sentimiento, por ejemplo la ira es un sentimiento dominante mientras que el miedo es sumiso, por tanto se encontrarán en lados opuestos de la escala<sup>[4]</sup>. Con estas tres escalas cada sentimiento se posicionará en el lugar tridimensional que le corresponda.

Puesto que en el sistema que se quiere diseñar se pretende guardar la información referente a las emociones en el modelo emocional *PAD* se ha obtenido una tabla de conversión<sup>[5][6]</sup> que se puede observar a continuación:

	P	A	D
Tristeza	-0,18	0,03	-0,14
Desprecio	-0,3	0,1	-0,4
Neutral	0,01	0,01	0,01
Disgusto	-0,59	-0,01	0,4
Enfado	-0,51	0,59	0,25
Sorpresa	0,41	0,55	0,19
Miedo	-0,64	0,6	-0,43
Felicidad	0,4	0,2	0,2

**Tabla 1.** Conversión de las emociones al modelo *PAD*

## Reconocimiento de emociones mediante imágenes

Una persona expresa sus emociones de maneras muy distintas, por el tono de voz, por el lenguaje personal o por los gestos faciales. Esta última es la que se va a explicar a continuación.

Si en una imagen aparece una cara se podría deducir cual era el estado de ánimo que tenía la persona que aparece en el fotograma analizando algunos detalles. Lo primero que un algoritmo debería hacer es delimitar el cuadro que ocupa la cara en la imagen para poder analizar únicamente esa parte de la imagen. A continuación se debería analizar la posición de las cejas, los ojos los labios y la mandíbula<sup>[6]</sup>.

En las cejas se puede diferenciar si están caídas o levantadas, si están tensas o relajadas y

también la forma que tienen en ese momento, si en forma de uve o al revés. Teniendo todo esto en cuenta se le daría distintos valores para poder valorar a que emoción corresponde.

Por otro lado se pueden analizar los ojos, tanto en lo correspondiente a si se tienen los ojos abiertos o cerrados como la orientación a la que apunta el globo ocular, con el mismo objetivo de aproximar que emoción denota ese gesto.

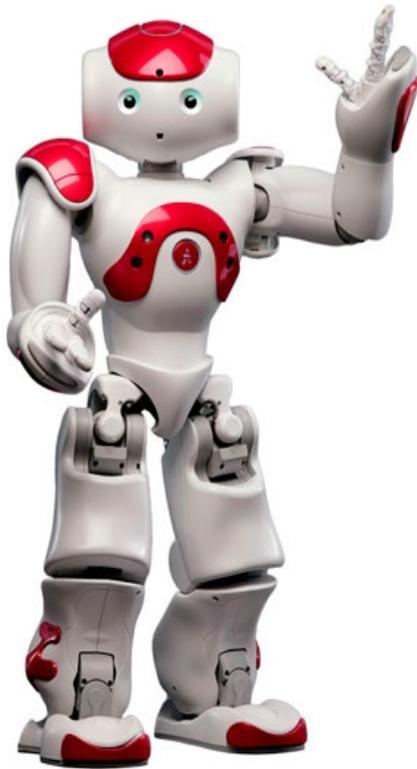
También se analizaría la boca, teniendo en cuenta si está abierta o cerrada, si las comisuras de los labios están hacia arriba o hacia abajo, si los labios están hacia afuera o sobresaliendo. Estas gesticulaciones también aportarán información relevante para el cálculo de la emoción.

La última de las regiones faciales que se debe analizar para averiguar cual es la emoción presente en la cara que se está interpretando es la mandíbula. En este caso se observa si la mandíbula está hacia arriba (boca cerrada) o hacia abajo (boca abierta) y también si está desplazada hacia la derecha o la izquierda. Estas dos observaciones a respecto de la mandíbula sirven para la aproximación de la emoción que representan.

Con todo lo anterior se puede ver que mediante una imagen que contenga una cara se consigue analizar esa cara para aproximar la emoción que la persona que aparece en dicho fotograma sentía en el momento en que se tomó. Existen diversas herramientas que implementan estos análisis para detectar las emociones como por ejemplo la herramienta *Emotion* del proyecto Oxford de Microsoft.

## **Robot NAO**

En este proyecto se ha optado por utilizar como soporte para nuestro sistema el robot humanoide NAO (ver figura 2). Cuando se habla de un robot humanoide hace referencia a un robot que imita la anatomía de los humanos, es decir, suelen tener dos brazos, dos piernas y una cabeza. Este robot tiene una altura de 58 cm y un peso de 4.3 kg y además tiene diversos sensores, visuales y de sonido, de los que se harán uso en el proyecto<sup>[7]</sup>.



**Figura 2. Robot NAO**

como lenguajes de programación el *C++*, *Python*, *Java*, *MATLAB*, *Urbi*, *C* y *.Net*, además es compatible con los sistemas operativos Windows, Mac OS y Linux, por tanto se podrá trabajar desde el dispositivo que sea más cómodo sin ningún inconveniente.

Se ha elegido este dispositivo como base para nuestro sistema de reconocimiento de emociones porque consta de elementos tales como cámara, altavoces y micrófonos que serán útiles en el desarrollo del proyecto, además la gran cantidad de lenguajes de programación que admite y la compatibilidad con distintos sistemas operativos facilita la adaptación.

### **Choreographe<sup>[7]</sup>**

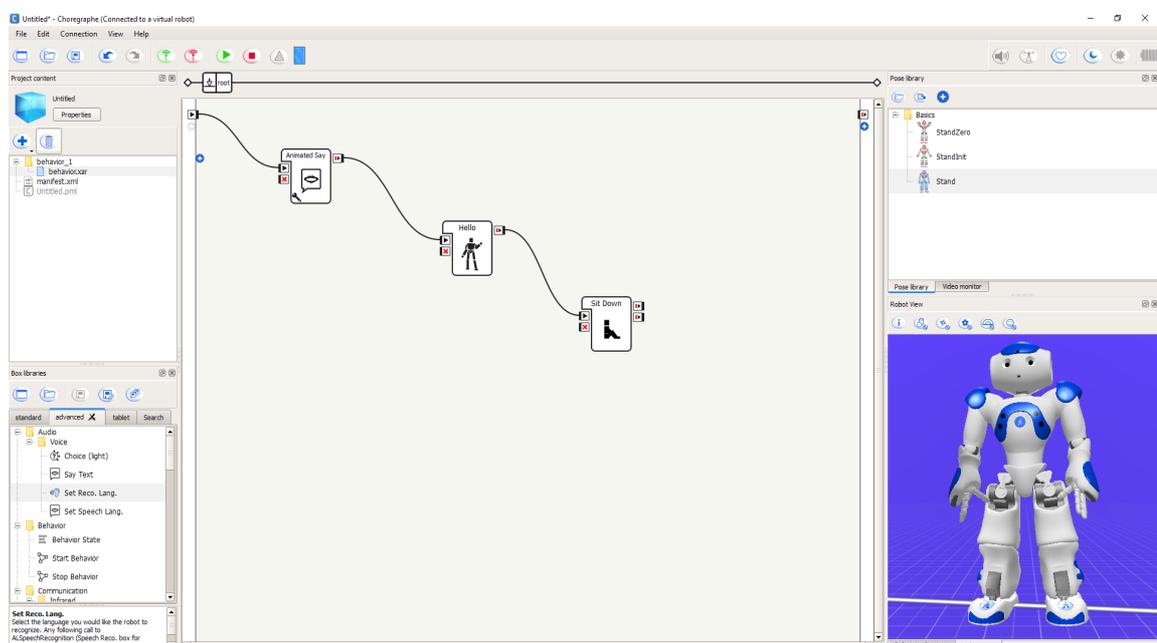
Para trabajar cómodamente con el robot NAO la empresa *Aldebaran Robotics* ha desarrollado un simulador llamado *Choreographe*. Este simulador se puede usar solo o vinculado a un robot real. En el primer caso ofrece un robot virtual que se puede observar en una ventana de la interfaz del propio programa y ver como ejecuta las rutinas que se le

El robot consta de dos cámaras HD, cuatro micrófonos y altavoces direccionales, un sonar, dos emisores y receptores de infrarrojos, una unidad de medición inercial, nueve sensores táctiles y ocho sensores de presión, además es capaz de conectarse a Internet tanto por conexión *WiFi* como mediante *Ethernet*. Cuenta además de una autonomía de 90 minutos con un uso normal.

El robot se controla mediante un sistema operativo basado en Linux llamado *NAOqi* el cual puede acceder y controlar todos los recursos de que dispone el robot mencionados en el párrafo anterior. Para trabajar con él, su sistema operativo admite

programan. En el segundo caso mediante la dirección IP del robot se puede conectar el simulador con el robot real y programarlo para que realice tareas o acceder a los múltiples recursos que tiene el robot como acceder a las cámaras. Con esta herramienta se puede trabajar independientemente con cada elemento del robot (sensores o incluso articulaciones) y además se puede estar al tanto del estado del robot.

Lo que hace más atractivo el uso de esta herramienta es su interfaz (ver Figura 3), ya que es muy clara e intuitiva. En la parte izquierda presenta dos ventanas que muestran arriba el proyecto sobre el que estamos trabajando y abajo todas las tareas predefinidas que puede realizar el robot NAO. Contiene muchas y muy variadas rutinas predefinidas que se pueden usar simplemente arrastrando a la ventana central uniendo dicha tarea con el hilo de ejecución y pulsando el botón de *Run* situado en la barra de tareas superior. En la parte derecha de la interfaz tiene dos posibles vistas, en caso de que estemos trabajando con un robot virtual tiene dos ventanas, una con las posiciones predeterminadas del robot y otra que nos muestra el robot virtual, el cual realizará todo aquello que se le programe, mientras que si está conectado a un robot real mostrará una ventana con la imagen correspondiente a una de las cámaras que el robot tiene en la cabeza.



**Figura 3.** Interfaz *Choregraphe*

La herramienta también consta de un menú en la parte superior que permite realizar tareas desde guardar el proyecto actual o cargar uno nuevo hasta conectar con un robot real. Además se puede acceder al resto de información que *Choreographe* proporciona. Para aprender a trabajar con esta herramienta, la empresa *Aldebaran Robotics* pone a disposición de los usuarios una guía con tutoriales y explicaciones.

## Python

El lenguaje que se ha elegido para trabajar en el proyecto y para desarrollar los programas que constituirán el sistema de reconocimiento de emociones es el lenguaje de programación llamado *Python*.

*Python* es un lenguaje interpretado, lo que quiere decir que está diseñado para ser ejecutado por medio de un intérprete sin necesidad de haber compilado antes el código, orientado a objetos, es decir, en este lenguaje siempre se trabaja sobre objetos, y que no tiene declaración explícita de constantes ni variables, por lo que no hay que especificar que tipo de información va a contener.

*Python* es un lenguaje de programación de propósito general y además es bastante simple y fácil de aprender ya que la sintaxis de este lenguaje es muy clara y fácil de leer. También se debería añadir que *Python* es un lenguaje de programación con una verificación de errores mejor que el de otros lenguajes ya que detecta muchos errores de programación que escapan a la detección de los compiladores.

*Python* pertenece a un proyecto de código abierto, administrado por la *Python Software Foundation* y por ello está en continua evolución. Este lenguaje de programación tiene una gran cantidad de librerías que aumenta constantemente, lo que permite la creación de múltiples aplicaciones. En la actualidad este lenguaje de programación es usado por empresas como Google, NASA, Yahoo o Walt Disney entre otros<sup>[8]</sup>.

*Choreographe* permite la creación de tareas mediante *scripts* de *Python*, por eso y por las características citadas anteriormente se ha elegido como lenguaje a utilizar en el presente proyecto. Los *scripts* de *Python* en *Choreographe* tienen una estructura muy concreta

donde se puede modificar cada instante de ejecución ya que las tareas en esta herramienta constan de distintos momentos de ejecución. Generalmente constan de la estructura siguiente:

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #Pon el código de inicialización
        pass

    def onUnload(self):
        #Pon aquí el código de resolución
        pass

    def onInput_onStart(self):
        #self.onStopped() #Activa la salida de la caja
        pass

    def onInput_onStop(self):
        self.onUnload() #se recomienda volver a utilizar el método de resolución cuando se detiene
        self.onStopped() #Activa la salida de la caja
```

En esta estructura el segundo método *onLoad* inicializa todo aquello que hará falta para la tarea que se quiera llevar a cabo. El tercer método *onUnload* se encargará al finalizar la ejecución de terminar, por ejemplo, todas las conexiones que se hayan hecho. El cuarto método *onInput\_onStart* estará implementado todo aquello que deba llevar a cabo el *script*. Por último el método *onInput\_onStop* se encargará de terminar la ejecución y pasar a la siguiente caja de tarea de la ejecución de la herramienta *Choreographe*.

## Herramientas del Proyecto Oxford de Microsoft

Microsoft ha desarrollado recientemente en el proyecto Oxford varias herramientas, de las cuales se va a hacer uso en este proyecto de dos, la herramienta de reconocimiento de emociones que toma el nombre de “*Emotion APP*” y la herramienta de reconocimiento facial de las personas que se llama “*Face APP*”.

La herramienta *Face API* ofrece diversas funcionalidades entre las que se encuentran la detección de caras, verificación facial, identificación facial, búsqueda de caras similares y la agrupación facial<sup>[9]</sup>.

La detección de caras encuentra las caras que se encuentren en una imagen y devuelve los rectángulos donde se encuentran dichas caras en la fotografía así como algunas características de cada cara basadas en aprendizaje automático de los rasgos faciales. Las características que esta aplicación proporciona son el género, el sexo, la edad, la sonrisa el vello facial entre otros.

La herramienta de verificación facial se utiliza para calcular la probabilidad de que dos caras pertenezcan a la misma persona. Esta herramienta utiliza la anterior para detectar las dos caras y poder hacer la comparación, además Microsoft recomienda que para utilizar esta funcionalidad solo haya una cara en las imágenes que se envíen para comparar.

La aplicación que se usa para identificar a personas por la cara funciona teniendo en cuenta la cara que esté intentando de identificar y las personas que se encuentren registradas en una base de datos que la herramienta define como un grupo de personas que tiene que haber sido creado antes de intentar identificar a alguien y que se puede ir modificando con el tiempo. Cada grupo puede contener hasta 1000 personas distintas con las que comparar después, así mismo cada persona que pertenezca a un grupo puede tener varias caras registradas. Cuando un grupo se crea se debe entrenar para que en el futuro pueda identificar nuevas caras. Cuando se envía una cara para la identificación, si esta es positiva la herramienta devuelve un identificador de la persona a la cual pertenece esa cara.

La búsqueda de caras similares se centra en la detección de una cara y la obtención de caras de características similares en la base de datos de Microsoft. Esta herramienta puede funcionar con dos modos. El primero "*matchPerson*" devuelve una serie de caras similares después de aplicar un umbral de coincidencia usando la aplicación explicada anteriormente de verificación. El segundo modo "*matchFace*" ignora el umbral que se aplica mediante la utilización de la verificación facial y devuelve caras similares pero que pueden pertenecer a distintas personas. Para una misma cara de búsqueda el primer método encontrará menos o, como mucho, el mismo número de coincidencias que el segundo y a su vez el segundo

incluirá entre sus respuestas el resultado del primer método.

Por último la herramienta de agrupación facial, para un grupo dado de caras desconocidas, se encarga de dividir dichas caras en conjuntos basados en la similitud facial. Puede darse el caso de que las caras pertenecientes a un mismo grupo creado como resultado de esa asociación pertenezcan a la misma persona.

La herramienta *Emotion API* ofrece tanto la funcionalidad de reconocimiento de emociones en fotografía como en vídeo, aunque esta última no es relevante para el presente proyecto. La herramienta para el reconocimiento de emociones mediante imagen funciona de la siguiente manera<sup>[10]</sup>:

En primer lugar la herramienta toma una imagen como entrada y envía al usuario un conjunto de emociones para cada cara que haya detectado en dicha imagen. El conjunto de emociones que devuelve (que ya se ha mencionado en el presente documento en el apartado de modelos emocionales) es tristeza, desprecio, neutral, disgusto, enfado, sorpresa, miedo y felicidad. En la respuesta de la herramienta cada emoción se corresponde con una puntuación. La emoción que mayor puntuación haya obtenido es la predominante, además todas las puntuaciones suman uno. Una funcionalidad que da la herramienta es la posibilidad de establecer un umbral de confianza.



## **Descripción del problema**

En este apartado se va a presentar una descripción del problema que se plantea para el desarrollo del proyecto además de aportar ejemplos de uso.

### **Problemas propuestos**

Como ya se ha comentado anteriormente en la actualidad el ámbito de la informática está en auge y cada vez se presentan y desarrollan dispositivos más complejos. En nuestro caso nos centramos más en el avance en robótica e Inteligencia Artificial que es un campo donde cada vez se investiga más. Existen robots capaces de adaptarse a una casa para limpiarla, de poder cocinar por sí mismos e incluso robots que pueden estar en una empresa cara el público.



**Figura 4.** Robot limpiador inteligente

Cuando una máquina ha de comunicarse con una persona siempre resulta tedioso que sea por ejemplo una grabación que se repite y no tenga ninguna evolución como podría tener una persona. Puesto que cada vez los sistemas inteligentes y los robots se introducen más en nuestro día a día conseguir que sean lo más humanos posibles es un punto a su favor, ya que un sistema que pueda modificar sus respuestas o su interacción teniendo en cuenta las respuestas que la persona a la que atiende le dé aumenta su funcionalidad y su calidad.

Con todo esto se quiere decir que cuanto más se ajuste un sistema inteligente o un robot a la persona con la que interacciona más fácil es que la persona se sienta bien atendida. Todo esto está bien cuando se trata de un sistema sencillo, que no necesita un aprendizaje, de trato al público como podría ser un sistema encargado de atención al cliente, por ejemplo en que únicamente ha de responder a dudas concretas o un sistema de limpieza de casa en el que se le puede decir que zonas se quiere que limpie, pero lo que nos ocupa en el presente proyecto es algo más complejo.

El escenario que hemos seleccionado para enmarcar este trabajo es una situación donde

una persona necesita ayuda continuada, como por ejemplo una persona mayor que se encuentra sola en casa y que pueda tener algún tipo de discapacidad motora o por el contrario un niño pequeño al que hay que tener vigilado continuamente.

En el primer caso podríamos suponer que es un hombre de edad avanzada con algunos problemas de salud que le impiden andar distancias largas o al que hay que hacerle análisis o pruebas médicas sencillas como podrían ser un control de azúcar o tomarle la tensión. En esta situación puede que este hombre no pueda vivir solo, pero ahora entra nuestra propuesta de un robot inteligente con movilidad suficiente como para poder llevar al hombre un vaso de agua, su medicación o acercarle un utensilio para que pueda andar bien. En nuestro caso el robot de asistencia incluye un sistema que identifica el estado de ánimo de este hombre y puede actuar teniendo en cuenta estos sentimientos, también tendremos en cuenta que el robot debería aprender a reaccionar a dichos sentimientos con el tiempo, ya que cada persona reacciona diferente a los sentimientos. Una vez que el robot aprendiera como afectan los cambios de humor al hombre que tiene a su cuidado podría adaptar sus rutinas para mejorar el humor de esta persona y así cuidarle mejor.

En el segundo caso nuestro proyecto puede que sea más útil, ya que los niños tienen más cambios de humor y son más dinámicos con ellos. Pongamos el ejemplo de un niño de 5 años que de vez en cuando tiene pataletas porque la comida no le gusta, en este caso nuestro sistema podría aprender a identificar cuando al niño le va a dar la pataleta y prevenirlo de algún modo. También el sistema de identificación de emociones puede cambiar el trato al niño dependiendo de como se sienta el niño en ese momento, si está demasiado alterado por algo cambiar el trato para que se pueda tranquilizar, si en cambio está triste o aburrido el robot puede intentar animarle o instarle a que juegue a la pelota para hacer que esté más alegre.

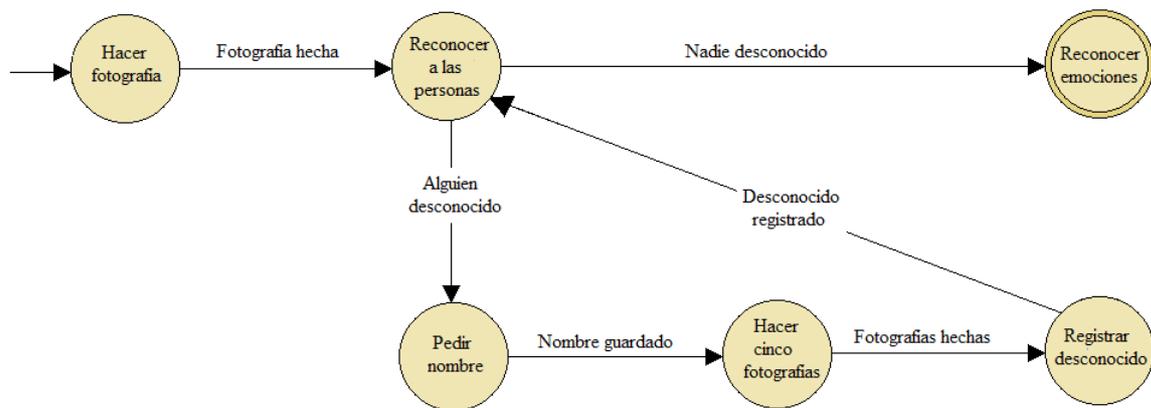
Teniendo en cuenta todo esto, podemos afirmar que un robot cuyo sistema inteligente tenga en cuenta los sentimientos de la persona con quien interacciona es mucho mejor en situaciones en que el trato va a ser continuado ya que al poder adaptarse a la situación facilita mucho las cosas, porque el trato, a pesar de ser con un elemento mecánico, se vuelve más humano.



## Diseño del sistema

En este apartado se va a explicar como se ha desarrollado el sistema que es objeto de este proyecto. El desarrollo se ha dividido en diferentes fases, las cuales se corresponden con los estados de una máquina de estados finitos. Una máquina de estados finitos es un modelo computacional que realiza operaciones sobre un estado inicial para llegar a un estado final.

La máquina de estados diseñada para el sistema de este proyecto contiene seis estados de los que solo el último es un estado final. Esta máquina de estados es la mostrada en la figura 5. Los estados y las transiciones se explicarán a continuación.



**Figura 5.** Máquina de estados finitos

### **Máquina de estados finitos (FSM)**

#### Estado 1: Captura de imagen

Este es el primer estado de la FSM y por el que el sistema siempre pasará. Consiste en una tarea que se encarga de hacer una imagen de aquello que se encuentre enfrente del robot. En esta imagen se encontrarán las personas a las que se les quiere reconocer y registrar sus emociones. En este caso, *Choreographe* tiene una tarea predefinida que se puede utilizar llamada *Take Picture*. Solamente se deben cambiar en el código la ruta donde se guardará la imagen y el nombre que se le dará. Mantenemos todas las preferencias de tamaño, resolución y formato. Este estado, puesto que no cambia durante el tiempo crea una imagen temporal que se sustituirá por una nueva cada vez que el sistema se ponga en funcionamiento.

## Estado 2: Reconocer a las personas que hay en la imagen

El segundo estado de la FSM se encargará de reconocer a las personas que aparecen en la imagen que se ha tomado anteriormente y se comprobará si hay algún registro anterior en el robot, es decir, si el robot conoce su nombre y alguna vez anterior ha guardado información sobre dicha persona.

En caso de que el robot reconozca a todas las personas que aparecen en la imagen pasará directamente al sexto estado que se corresponde con el reconocimiento de las emociones.

En caso contrario, es el que exista alguna persona a la que el robot no reconoce se necesitará registrar a esa persona guardando su nombre y creando una entrada nueva en la base de datos del sistema.

Esto se hará haciendo uso de la herramienta *Face API* de Microsoft que se detalló en el apartado del estado del arte. Como se expuso anteriormente es necesario crear un grupo de personas e introducir las personas de la base de datos del sistema para proceder al reconocimiento facial, así pues los siguientes métodos se encargarán de ponerse en contacto con la *API* de Microsoft y llevar a cabo estas tareas:

```
1. 1 def createPersonGroup(body, namePersonGroup):
1. 2     conn = httpLib.HTTPSConnection("api.projectoxford.ai")
1. 3     conn.request("PUT", "/face/v1.0/persongroups/" + namePersonGroup, body, headersFace)
1. 4     response = conn.getResponse("")
1. 5     data = response.read()
1. 6     conn.close()
1. 7     return data
```

Este primer método se encarga de crear un grupo donde se basará el reconocimiento facial de las personas que aparezcan en las imágenes. A dicho grupo se le ha dado el nombre de "gtiia" La primera línea es la definición del método que se llama *createPersonGroup* y consta de dos atributos de entrada. Dichos atributos de entrada se corresponden con una descripción opcional que se le quiera dar (*body*) y a el nombre que se quiera dar al grupo (*namePersonGroup*). La segunda y tercera línea se encargan respectivamente de conectar con la *API* de Microsoft y de conectar exactamente con la *Face API* y con la herramienta que gestiona los grupos de personas así como de elegir lo que se quiere hacer con la tarea *PUT* (primer atributo) que aparece, con lo que la herramienta interpreta que se quiere crear un nuevo grupo. La cuarta y quinta línea se encargan de recuperar la respuesta de la herramienta y almacenar los datos. La sexta línea se encarga de cerrar y dar por terminada

la conexión y la última devuelve a la ejecución principal los datos que confirman la creación del grupo o informan de un error de dicha creación.

```
1. 1 def createPerson(body, namePersonGroup):
1. 2     conn = httplib.HTTPSConnection("api.projectoxford.ai")
1. 3     conn.request("POST", "/face/v1.0/persongroups/" + namePersonGroup + "/persons", body,
headersFace)
1. 4     response = conn.getresponse()
1. 5     data = response.read()
1. 6     conn.close()
1. 7     return data
```

Este segundo método sigue la misma estructura que el primero y algunas de las líneas sirven para lo mismo. Las líneas dos, cuatro, cinco, seis y siete son idénticas a las del método anterior y sirven para lo mismo. La primera línea es la definición del método que se llama *CreatePerson* y además incluye dos atributos de entrada. Estos atributos de entrada son el nombre y el identificador que se le asignará a la persona que se quiere incluir en un grupo (*body*) y el nombre identificador del grupo al que se quiere incluir la nueva persona (*namePersonGroup*). La tercera línea conecta con la herramienta característica de los grupos de personas y además con el grupo que nos ocupa (segundo atributo), además le da la información a la herramienta de que lo que se quiere hacer es introducir una nueva persona al grupo con el primer atributo (*POST*), el tercer atributo se corresponde con el atributo de entrada *body*, mientras que el último atributo, *headersFace*, incluye toda la información necesaria para acceder a la herramienta. Por último cabe destacar que la respuesta de la herramienta incluirá la confirmación de que la persona se ha creado correctamente o la información del error que no ha permitido llevarlo a cabo.

```
1. 1 def addPersonFace(body, personGroupId, personId):
1. 2     conn = httplib.HTTPSConnection("api.projectoxford.ai")
1. 3     conn.request("POST", "/face/v1.0/persongroups/" + personGroupId + "/persons/" +
personId + "/persistedFaces", body, headersLandMark)
1. 4     response = conn.getresponse()
1. 5     data = response.read()
1. 6     conn.close()
```

Este método es necesario para, una vez creado un objeto persona en el sistema, asignar al menos cinco caras de la persona en cuestión a la persona creada para así poder, tanto entrenar el grupo de personas para poder, posteriormente, reconocer nuevas fotografías,

como a la hora de identificar a personas en imágenes tener caras con las que comparar para que la herramienta calcule si pertenecen a la misma persona o no. La primera línea se corresponde con la definición del método llamado *addPersonFace* y que tiene tres atributos de entrada, los cuales se corresponden respectivamente a la cara que se quiere asociar, el nombre del grupo en que se encuentra el objeto persona al que queremos asignarle una cara y por último el identificador de dicha persona dentro del grupo. La segunda y tercera línea se corresponden con la conexión con la aplicación de Microsoft. En la tercera indica que la tarea que se quiere llevar a cabo es la de asociar una cara a una persona (primer atributo) indicando la ruta hasta el grupo y la persona deseados (segundo atributo), la imagen de la cara (tercer atributo) y, por supuesto, la información necesaria para poder acceder a la herramienta (cuarto atributo). Por otro lado las cuarta y quinta línea recuperan el resultado de la operación y la última línea termina la conexión.

```
1. 1 def trainPersonGroup(namePersonGroup):
1. 2     conn = httplib.HTTPSConnection("api.projectoxford.ai")
1. 3     conn.request("POST", "/face/v1.0/persongroups/" + namePersonGroup + "/train", "",
headersTrain)
1. 4     response = conn.getresponse()
1. 5     data = response.read()
1. 6     conn.close()
1. 7     return data
```

Este tercer método sirve para entrenar al grupo que hemos creado y poder identificar en el futuro a personas que se encuentren en una imagen que se le envíe. La mayoría del método actúa de la misma manera que los métodos explicados anteriormente. El cambio se reduce a la definición del método ya que toma un nombre distinto (*trainPersonGroup*) y solamente tiene un atributo de entrada que se corresponde con el nombre identificador del grupo que se quieren entrenar. Por otro lado la tercera línea del método conecta con la herramienta que gestiona los grupos de personas y los entrena (segundo atributo) además de que le dice cual es la tarea que se quiere llevar a cabo con la orden *POST* (primer atributo).

```
1. 1 def identificationFace(body):
1. 2     conn = httplib.HTTPSConnection("api.projectoxford.ai")
1. 3     conn.request("POST", "/face/v1.0/identify", body, headersFace)
1. 4     response = conn.getresponse("")
1. 5     data1 = response.read()
1. 6     conn.close()
1. 7     data = json.loads(data1)
1. 8     return data
```

Este es el último método de que se hace uso para el reconocimiento de personas. En la primera línea se define el método con el nombre *identificationFace* el cual tiene un atributo de entrada que incluirá la información referente a las caras que se quiere identificar en el grupo de personas, el nombre identificador del grupo donde se debe intentar identificar a las personas y además el número máximo de candidatos que queremos que la herramienta devuelva. Esto último quiere decir que la herramienta devolverá tantos identificadores como se le diga. También se le podría dar un umbral de coincidencia que se quiera respetar al devolver a los candidatos. La tercera línea coincide con la conexión que se hace con la herramienta de la *Face API* que se encarga de la identificación, también envía la información necesaria mediante el atributo *body* (tercer atributo). Por último se debe explicar la séptima línea ya que es la encargada de convertir la información que la herramienta devuelve a un formato que se pueda tratar fácilmente, con esta línea la información se convierte en una lista de identificadores que podremos tratar.

Por último se puede observar que en la llamada de la tercera línea de todos los métodos el último atributo se corresponde con el nombre *headersFace* o *headersTrain* o *headersLandMark* este atributo contiene la información necesaria para que la *Face API* nos permita trabajar con ella. Las declaraciones de estas cabeceras son las siguientes:

```
1. 1 headersFace = {
1. 2   # Request headers
1. 3   'Content-Type': 'application/json',
1. 4   'Ocp-Apim-Subscription-Key': '{subscription key}',
1. 5 }
```

```
1. 1 headersTrain = {
1. 2   'Ocp-Apim-Subscription-Key': '{subscription key}',
1. 3 }
```

```
1. 1 headersLandMark = {
1. 2   # Request headers
1. 3   'Content-type': 'application/octet-stream',
1. 4   'Ocp-Apim-Subscription-Key': '{subscription key}',
1. 5 }
```

Estas cabeceras incluyen en primer lugar (excepto en el segundo caso que no es necesario) el tipo de contenido multimedia. Se puede corresponder con “*application/json*” que permite una comunicación en que el contenido del atributo *body* sea a partir de una conexión http o puede corresponder a una configuración *application/octet-stream* la cual permite el envío de datos locales. Dependiendo del método interesa usar una configuración o otra. La cuarta (o segunda en el segundo caso) línea incluye la clave necesaria para que Microsoft de acceso a la *Face API*, ya que estas herramientas requieren estar suscrito al proyecto Oxford de Microsoft.

### Estado 3: Pedir el nombre a aquella persona que no tenga registro anterior

Este es un estado que no se ejecutará en todas las ocasiones en que el sistema esté en funcionamiento, solo se ejecutará en caso de que en la imagen inicial haya alguien que no figure en la base de datos del sistema. En este caso se ha creado una tarea mediante *Python* que accede a los altavoces y micrófonos del robot NAO, haciendo así que el robot pregunte por el nombre de la persona desconocida y espere escuchando unos segundos a la respuesta del individuo, para así poder almacenar esta información para futuras interacciones. A continuación se analizará algún extracto del *script*:

```

1. 4 def __init__(self):
1. 5     GeneratedClass.__init__(self, False)
1. 6     try :
1. 7         self.asr = ALProxy("ALSpeechRecognition")
1. 8     except Exception as e:
1. 9         self.asr = None
1.10         self.logger.error(e)

```

Este trozo de código lo que hace es intentar conectar los recursos de reconocimiento de sonido. En caso de no poder conectar devolvería un error.

```

1.20 def onInput_onStart(self, p):
1.21     name = ""
1.22     self.tts.say("Dime tu nombre para que te pueda reconocer, por favor")
1.23     asr.setLanguage("Spanish")
1.24     asr.setVocabulary(vocabulary, True)
1.25     memory = ALProxy("ALMemory")
1.26     asr.subscribe("Test_ASR")
1.27     time.sleep(15)
1.28     name = memory.getData("WordRecognized")
1.29     asr.unsubscribe("Test_ASR")
1.30     self.onStopped(name)

```

Este trozo de código se encarga en primer lugar de que el robot, mediante altavoces, diga lo que se especifica en la línea 22, mientras que a continuación se especifica el lenguaje con el que se debe reconocer la voz (línea 23) y se permite la introducción de nuevas palabras al vocabulario que el robot reconoce. Puesto que lo que se quiere es reconocer un nombre no se puede prediseñar un vocabulario específico, si fuese un caso en que lo único que se debiera reconocer fuese “sí” o “no” sí que se recomendaría tener un vocabulario de las palabras que se deberán reconocer. Conecta con la memoria interna del robot desde la cual se deberá recuperar la información del reconocimiento de voz (línea 25), se arranca el motor de reconocimiento de voz con el usuario *Test\_ASR* (línea 26) y se dispone a esperar durante 15 segundos para que la persona tenga tiempo de decir su nombre(línea 29). Una vez ha pasado este tiempo se accede a la memoria para recuperar la palabra que se haya reconocido y se interpretará como el nombre de la persona desconocida (línea 28). Por último se desactiva el motor de reconocimiento de voz (línea 29) y se llamará al método de cierre del *script* dando como salida de la tarea el nombre reconocido (línea 30).

#### Estado 4: Tomas 5 imágenes a la persona nueva

La herramienta de Microsoft que se utiliza en este proyecto necesita de cinco imágenes para poder dar un identificador a una persona que no existe en su registro, así pues el sistema, en caso de que haya alguien que no tenga un registro en la base de datos, ejecutará esta tarea que comparte el mismo código del primer estado pero hace cinco fotografías de la misma persona a la que no conoce. Como este estado viene del anterior, recoge la información que la anterior tarea ha recogido y así da el nombre de la persona desconocida a las nuevas imágenes para mantener el flujo de información a través de la máquina de estados.

#### Estado 5: Registrar a la persona nueva en la base de datos

Este quinto estado, como los dos anteriores, solo se ejecuta en el caso de que en la primera imagen que toma el sistema haya alguna persona que sea desconocida para el mismo. Este estado hace uso de los métodos mencionados y explicados en el segundo estado. Añade a la persona desconocida en el grupo que se hace uso durante toda la ejecución, “*gtia*”, mediante el método *createPerson* y a continuación se hace uso del método *trainPersonGroup* para entrenar el grupo de personas para poder identificar personas en el futuro incluyendo a la persona desconocida. Por último se vuelve a reconocer la imagen inicial para asegurarnos de que no queda nadie desconocido en la escena, en caso de que haya alguien más desconocido se volverán a ejecutar el tercer, cuarto y quinto estados del sistema.

#### Estado 6: Reconocer las emociones de las personas de la primera imagen

Este último estado se ejecuta en todas las ocasiones en que el sistema se pone en funcionamiento. En el momento en que todas las personas de la imagen inicial tienen un registro en la base de datos del sistema se lanza la ejecución de este estado. Se trata de una tarea que se encarga de conectarse con la herramienta de Microsoft encargada del reconocimiento de emociones y una vez recibe una respuesta, transforma la información recibida al modelo emocional *PAD* mediante la conversión explicada en el segundo capítulo de este documento. Al finalizar almacena la información en la base de datos del sistema como un nuevo instante de tiempo y guarda en el registro de cada persona el estado emocional en ese instante de tiempo.

Este estado hace uso de un nuevo método que necesita otra cabecera que es igual a la

explicada anteriormente para los métodos de *Face API*, pero como este método utiliza *Emotion API* también es necesaria la declaración de una nueva cabecera que incluya la clave específica de esta herramienta de Microsoft, ya que también requiere de una suscripción. La cabecera toma la siguiente forma:

```
1.1 headersEmotion = {  
1.2   'Content-Type': 'application/octet-stream',  
1.3   'Ocp-Apim-Subscription-Key': '{subscription key}',  
1.4 }
```

Por otro lado se va a proceder a la explicación del nuevo método que se encarga de la recuperación de los datos que contendrán la información referente al estado emocional de las personas que se encuentren en la imagen. El método es el siguiente:

```
1.1 def getEmotionalState(body):  
1.2   conn = httplib.HTTPSConnection("api.projectoxford.ai")  
1.3   conn.request("POST", "/emotion/v1.0/recognize", body, headersEmotion)  
1.4   response = conn.getresponse("")  
1.5   data1 = response.read()  
1.6   conn.close()  
1.7   data = json.loads(data1)  
1.8   return data
```

Este método se encarga del reconocimiento de emociones. En la primera línea se encarga de la definición del método llamado *getEmotionalState* además de incluir el atributo de entrada (*body*) que se corresponde con la imagen que se quiere reconocer. La siguiente línea realiza la conexión con el proyecto de Microsoft y la tercera se encarga de enviar la información necesaria para la conexión con la herramienta deseada, la imagen que se quiere reconocer y la cabecera necesaria para que se de el servicio solicitado. Las líneas cuarta y quinta recuperan la información que devuelve la herramienta. La siguiente línea cierra la conexión con Microsoft. La séptima línea convierte la información recibida en datos que se pueden manipular, es decir, convierte la respuesta en una lista de diccionarios, en los que cada elemento de la lista contiene los valores emocionales de cada persona que se encontraba en la imagen. Por último, la octava línea termina el método y devuelve como salida la lista de diccionarios.

## Versiones

Se han hecho dos versiones del sistema, la segunda versión es más eficiente en tiempo de ejecución. Esto ha sido así porque la primera intención ha sido hacer un sistema que funcionase e hiciese todas las tareas que se propusieron al inicio del proyecto. Cuando se ha terminado esta versión se ha hecho una valoración de las funcionalidades y el rendimiento, y se ha observado que había aspectos que se podían mejorar y se ha implementado la segunda versión. A continuación se explicarán ambas.

La ejecución de ambos sistemas coinciden en la mayor parte de la ejecución. Primero se entrena el grupo de personas y a continuación se reconoce a las personas que hay en la fotografía temporal que se ha hecho. Si en este punto hay alguien desconocido el robot informa de ello y añade a la nueva persona al grupo de personas asignándole además cinco fotografías de su cara para poder reconocerlo en el futuro. En este punto se vuelve a entrenar el grupo de personas y se reconoce de nuevo la imagen, en caso de que hubiese alguien mas desconocido se repetiría el proceso. En caso de que el sistema reconozca a todas las personas pasa a reconocer los estados de ánimo de las distintas personas. Una vez la herramienta devuelve la información solicitada se utiliza la conversión al modelo emocional *PAD* y por último se almacena por un lado la información referente a los identificadores y los nombres de todas las personas existentes en el sistema y por otro lado se almacena la nueva información de los estados de ánimo para cada persona reconocida.

Una vez explicados los puntos que coinciden, se va a proceder a exponer los cambios que hay de una versión a otra:

La primera versión del sistema (ver prueba de ejecución en la figura 7) realiza todo el proceso creando un grupo de personas en la parte correspondiente al segundo estado de la FSM, y por tanto, se deben crear de nuevo todos los objetos persona que pertenecen al grupo y que existen en la base de datos local del sistema, además de deber asignar de nuevo las caras de cada persona a su objeto asociado. Esto, en ejecución, se traduce en que cada vez que se lanza el sistema se ejecutan los métodos *createPersonGroup*, *createPerson* y *addPersonFace*. Además, al finalizar el sistema siempre hay que borrar el grupo creado además de que no se puede mantener una base de datos local de una

manera cómoda, ya que cada vez que el sistema se ejecuta en esta versión las personas, a pesar de existir ya en el sistema, al crear de nuevo el grupo y los objetos persona, los identificadores varían y se debe adaptar todo a los nuevos identificadores (ver Figura 6).

La segunda versión del sistema que se ha implementado (ver prueba de ejecución en la figura 7) no crea ni borra en cada ejecución el grupo de personas, esto hace que tampoco sea necesario el añadir a las personas existentes en la base de datos ni

asignarles caras, ya que todo esto ya existe en el proyecto Oxford de Microsoft. Esto facilita mucho el tratamiento de la información almacenada localmente.

```
Arranca el sistema
-----
Grupo gtiia creado
-----

Anyadiendo personas existentes al grupo...
Identificador de la Persona...
Nombre: Rosa
Id: 69ff049c-0f3d-40d7-8f6c-5faeb22d5360

Entrenando el grupo de personas: gtiia...
Entrenamiento terminado con exito

Cargando la imagen...
Imagen cargada con exito

Eres: Rosa

Realizando el reconocimiento emocional...
Reconocimiento emocional terminado con exito

Guardando la informacion obtenida...
Informacion guardada con exito

Cerrando el sistema...
('Se ha eliminado el grupo de personas: ', 'gtiia')
```

Figura 6. Versión 1

```
Arranca el sistema
-----

Entrenando el grupo de personas: gtiia
Entrenamiento terminado con exito

Cargando la imagen...
Imagen cargada con exito

Eres: Rosa

Realizando el reconocimiento emocional...
Reconocimiento emocional terminado con exito

Guardando la informacion obtenida...
Informacion guardada con exito
```

Figura 7. Versión 2

Por otro lado, en esta versión la primera vez que se lanza el sistema se ha de crear el grupo de personas de la misma manera que en la primera versión, esto solo ocurre una vez así que hay un ahorro considerable de tiempo para el resto de ejecuciones.



## **Evaluación**

Se van a realizar una serie de pruebas para comprobar el buen funcionamiento del proyecto. Primero se comprobará el funcionamiento de la parte del reconocimiento facial, es decir si reconoce bien a las personas que el sistema ya conoce, si identifica a las personas desconocidas y es capaz de introducirlos correctamente al sistema. A continuación se harán una serie de pruebas para comprobar el correcto funcionamiento del reconocimiento de emociones, es decir, con las personas que el sistema reconozca se enviarán una serie de imágenes tratando de interpretar emociones dadas y se comprobará si las emociones que se detectan coinciden en cada caso.

A parte de la interacción real del robot NAO, el sistema se ha preparado para que comunique por consola que está haciendo en todo momento, así se puede aportar esta información al presente trabajo.

### **Sistema de reconocimiento facial**

Se ha contado con tres modelos para hacer las pruebas de reconocimiento facial. Se trata de dos mujeres y un hombre que tienen edades comprendidas entre los 20 y los 60 años. En un primer momento solo una de estas personas está introducida en el sistema por lo que esto se utilizará para probar los tres puntos críticos de esta parte del sistema: que sea capaz de identificar correctamente a la persona que sí se conoce y que pueda identificar a las personas desconocidas y sean correctamente introducidas al sistema.

La primera prueba de evaluación que se ha hecho ha sido comprobar que el sistema identificaba correctamente al modelo que sí estaba registrado. El robot toma una imagen de la persona que debe reconocer y a continuación la envía a la herramienta de reconocimiento indicándole en que grupo de personas debe buscar una coincidencia. La herramienta compara la cara que aparece en la imagen que recibe con las caras almacenadas en el grupo de personas y en caso de encontrar una coincidencia toma el identificador de la persona que coincide y lo devuelve al robot. Este comprueba si tiene registrado en su base de datos el identificador que le ha devuelto la herramienta y, como en este caso lo tiene registrado, indica de quien se trata y procede al análisis emocional. El resultado obtenido en esta prueba ha sido la siguiente:

```
Arranca el sistema
-----
Entrenando el grupo de personas: gtiia
Entrenamiento terminado con exito

Cargando la imagen...
Imagen cargada con exito

Eres: Rosa

Realizando el reconocimiento emocional...
Reconocimiento emocional terminado con exito

Guardando la informacion obtenida...
Informacion guardada con exito
```

**Figura 8.** Prueba de funcionamiento 1

Se puede observar que ha reconocido correctamente a la persona que ya existía en el sistema. Esta prueba se ha realizado varias veces cambiando la distancia desde la que se toma la imagen a analizar y ha funcionado correctamente en todas las repeticiones, por tanto podemos afirmar que esta funcionalidad actúa satisfactoriamente.

La siguiente prueba que se va a realizar es enviar una fotografía de alguien desconocido y que el sistema lo detecte y lo introduzca en el sistema. Para ello utilizaremos una fotografía de un modelo distinto al de la prueba anterior. (El resultado de la ejecución se puede ver en la figura 9).

Se puede observar que el sistema detecta correctamente que en la imagen introducida hay alguien a quien no reconoce y procede a registrarla mediante las cinco fotografías que ha hecho en el proceso. Se ha introducido correctamente en la base de datos y ha procedido al reconocimiento emocional. Una vez ha terminado la ejecución se ha comprobado la información en los archivos donde el sistema almacena los datos de la base de datos. El archivo que recoge el identificador y el nombre está correctamente almacenado con la nueva persona y la existente. El archivo donde se guarda la información de los estados de ánimo ha conservado la información que ya había registrada y ha guardado correctamente la nueva información. Como en la prueba anterior se han realizado varias repeticiones y todas han tenido una ejecución satisfactoria. Así, se puede afirmar que el sistema ha realizado la

ejecución correctamente.

```
Arranca el sistema
-----

Entrenando el grupo de personas: gtiia
Entrenamiento terminado con exito

Cargando la imagen...
Imagen cargada con exito

Hay alguien a quien no conozco
Hola Maribel, voy a proceder a registrarte en mi base de datos

Identificador de la Persona...
Nombre: Maribel
Id: 779165e2-09d2-444e-aaa5-70672ca3ca47

Maribel, ya perteneces a mi base de datos

Entrenando el grupo de personas: gtiia
Entrenamiento terminado con exito

Eres: Maribel

Realizando el reconocimiento emocional...
Reconocimiento emocional terminado con exito

Guardando la informacion obtenida...
Informacion guardada con exito
```

**Figura 9.** Prueba de funcionamiento 2

La siguiente y última prueba que se va a llevar a cabo consiste en introducir en el sistema una imagen en la que aparecerán dos personas, una conocida y otra desconocida. Se va a comprobar que el sistema introduce correctamente a la persona desconocida y, además, trata y almacena los estados de ánimo de las dos personas.

En esta prueba se ha podido comprobar que cuando hay una persona desconocida en la imagen y otra conocida actúa como se esperaba. En primer lugar introduce a la nueva persona en la base de datos y en el grupo de personas para poder identificarla más adelante. A continuación vuelve a entrenar el grupo con la nueva persona registrada y vuelve a analizar la imagen inicial para identificar a las personas que hay en ella. Una vez esto se ha realizado pasa al reconocimiento emocional y almacena los datos correctamente. Esta prueba se ha realizado diversas veces cambiando la imagen de entrada y en todas las repeticiones se ha ejecutado correctamente. El resultado de la ejecución del sistema se puede observar en la

siguiente imagen:

```
Arranca el sistema
-----
Entrenando el grupo de personas: gtiia
Entrenamiento terminado con exito

Cargando la imagen...
Imagen cargada con exito

Hay alguien a quien no conozco
Hola Roberto, voy a proceder a registrarte en mi base de datos

Identificador de la Persona...
Nombre: Roberto
Id: 58ca57b8-be5e-4848-80c7-93dbd467edec

Roberto, ya perteneces a mi base de datos

Entrenando el grupo de personas: gtiia
Entrenamiento terminado con exito

Eres: Roberto

Eres: Maribel

Realizando el reconocimiento emocional...
Reconocimiento emocional terminado con exito

Guardando la informacion obtenida...
Informacion guardada con exito
```

**Figura 10.** Prueba de funcionamiento 3

## **Sistema de reconocimiento de emociones**

En este apartado se van a relatar las distintas pruebas que se han hecho con varios sujetos para comprobar el análisis de las emociones. En estas pruebas se pedirá a los sujetos que hagan caracterizaciones de distintas emociones y, una vez se haya analizado en el sistema, se comprobará que las emociones se han reconocido correctamente. Se realizarán diversas repeticiones para cada emoción y para cada sujeto. En la presente memoria se incluirán dos o tres ejemplos para cada emoción.

En el primer caso se ha solicitado a los sujetos de la prueba que hagan una mueca de *felicidad* al interactuar con el robot, así el sistema ha analizado las expresiones. A continuación se muestran dos de los resultados, en el modelo PAD, obtenidos en las distintas repeticiones que se han realizado:

	P	A	D
Sujeto 1	0,39	0,2	0,19
Sujeto 2	0,4	0,2	0,19

**Tabla 2.** Prueba de felicidad

Cuando utilizamos la escala de conversión asociamos unos valores PAD a cada emoción, y por tanto, podemos hacer la misma conversión a la inversa para poder averiguar a que emoción corresponden. En este caso la felicidad se correspondía con los valores  $P=0,4$ ,  $A=0,2$  y  $D=0,2$ . Los valores obtenidos en la ejecución se aproximan mucho a los deseados, por tanto, se puede afirmar que el resultado es satisfactorio.

A continuación se ha pedido a los sujetos que realizasen una mueca de *tristeza* para su análisis. El resultado ha sido el siguiente:

	P	A	D
Sujeto 1	-0,12	0,03	-0,1
Sujeto 2	-0,16	0,03	-0,13
Sujeto 3	0,04	0,06	0,03

**Tabla 3.** Prueba de tristeza

Como se ha hecho en la prueba anterior, observamos cuales son los valores de PAD para la tristeza  $P=-0,18$ ,  $A=0,03$ ,  $D=-0,14$ . Los dos primeros sujetos han sabido interpretar la emoción y sus resultados se han aproximado bastante a lo esperado, mientras que el tercer sujeto ha obtenido unos resultados que se acercan más a una emoción neutral ( $P=0,01$ ,  $A=0,01$ ,  $D=0,01$ ) que a la deseada. Esto puede deberse a que la mueca del sujeto no tenia suficiente caracterización. Esto reafirma que la detección de emociones es complicada ya que depende de cada persona y como exteriorice con sus gestos faciales dichas emociones.

El siguiente sentimiento que se ha pedido interpretar es el *enfado*, así pues los sujetos han intentado poner una mueca de enfado obteniendo los siguientes resultados:

	P	A	D
Sujeto 1	-0,32	0,5	0,24
Sujeto 2	-0,52	0,35	0,27

**Tabla 4.** Prueba de enfado

Los valores esperados de PAD son  $P=-0,51$ ,  $A=0,59$ ,  $D=0,25$ . Si analizamos los resultados obtenidos podemos ver que los resultados obtenidos se acercan a los esperados y, por tanto, el reconocimiento de emociones ha sido satisfactoria.

La última emoción que se ha pedido interpretar ha sido la *sorpresa*. Los resultados obtenidos de esta prueba han sido los siguientes:

	P	A	D
Sujeto 1	0,41	0,55	0,19
Sujeto 2	0,37	0,41	0,18
Sujeto 3	0,4	0,38	0,19

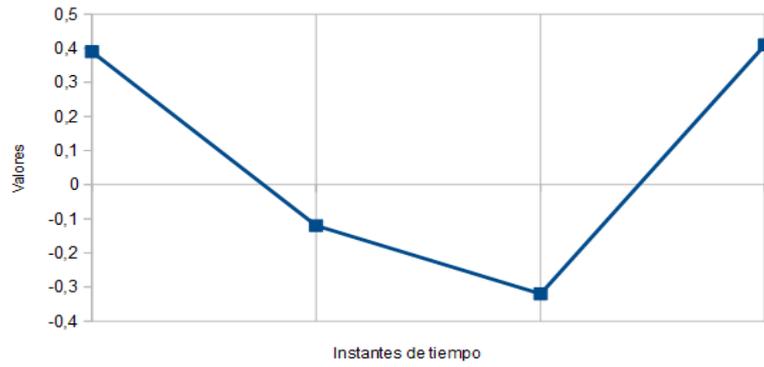
**Tabla 5.** Prueba de sorpresa

Una vez más, se comparan los resultados con los valores PAD asociados con la emoción de la sorpresa. Estos valores son  $P=0,41$ ,  $A=0,55$ ,  $D=0,19$ . En esta prueba los resultados obtenidos se han acercado bastante a los valores esperados, por esto podemos decir que la emoción se ha identificado correctamente.

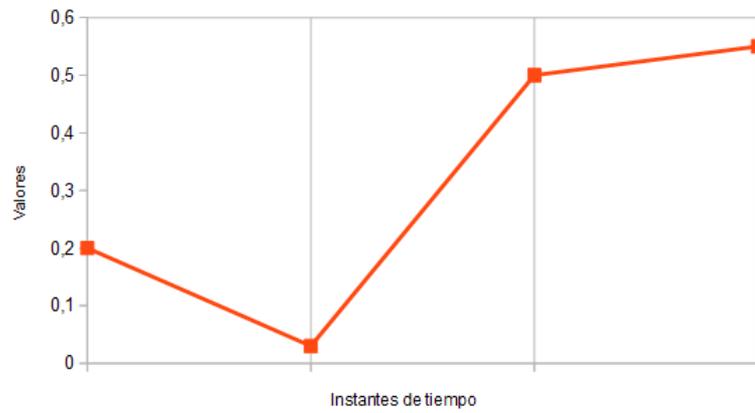
Los resultados obtenidos de dos de los sujetos (los cuales han participado en las cuatro pruebas) se presentan en las dos siguientes páginas en gráficos. Hay un gráfico por cada valor PAD y se muestran los distintos valores en cada instante de tiempo. Estos valores se corresponden, en cada uno de los cuatro instantes de tiempo mostrados, con una emoción concreta. Puesto que los resultados mostrados son la representación gráfica de los resultados obtenidos de las pruebas redactadas anteriormente las emociones de cada instante de tiempo son, respectivamente, felicidad, tristeza, enfado y sorpresa. Se han obtenido las siguientes gráficas para mostrar los diferentes valores obtenidos para cada emoción de una manera más visual.

## Sujeto 1

P



A

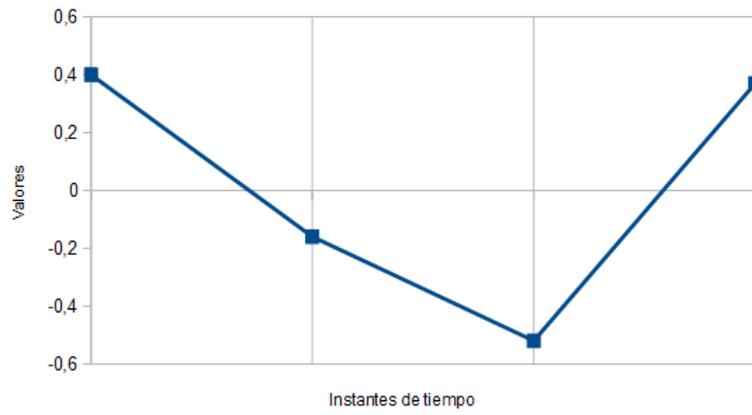


D

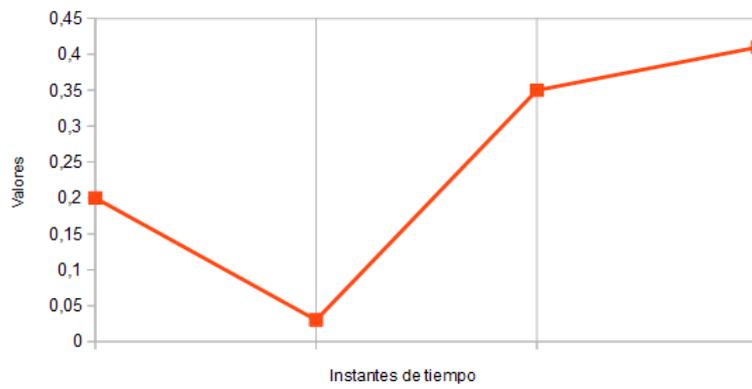


## Sujeto 2

P



A



D





## **Conclusiones**

Los objetivos del proyecto incluían el desarrollo de un sistema que reconociese a las personas que se detecten en una imagen y pudiese introducir personas nuevas en la base de datos, e incorporar dicha funcionalidad al robot bípedo NAO. Una vez todas las personas de la imagen fueran conocidas el sistema debía hacer un reconocimiento del estado de ánimo de las personas presentes en una fotografía y, por último, debía almacenar la información obtenida.

La parte de la implementación de la herramienta que reconociese a las personas de una imagen se desarrolló con éxito. Se desarrolló correctamente la ampliación de la herramienta para incluir a personas nuevas al sistema.

Se ha llevado a cabo la implementación satisfactoriamente de una herramienta para el reconocimiento de emociones de distintas personas que aparezcan en una imagen dada. Además se ha desarrollado con éxito una herramienta para guardar la información obtenida.

Por último se han unido todas las herramientas creadas para construir el sistema. Además se ha podido optimizar el sistema en cuanto a rendimiento y tratamiento de datos. Una vez se ha terminado todo el proceso de desarrollo del sistema se ha procedido a incorporarlo al robot bípedo NAO mediante la herramienta *Choreographe*, con el que se han hecho las pruebas que han comprobado el buen funcionamiento del sistema y del robot. Así pues se puede afirmar que al término de este proyecto se han cumplido todos los objetivos que se habían marcado al inicio. Por esta razón se da el proyecto por terminado y realizado con éxito.

## **Trabajo futuro**

Este proyecto puede servir como base para diversos propósitos en el futuro. El sistema se podría añadir, como ha sido el caso, en un robot para hacer que su toma de decisiones tenga en cuenta los estados emocionales de las personas con las que trata, así se podrían implementar rutinas que el robot lleve a cabo dependiendo de dicho estado de ánimo. Esto, como se comentó al inicio del proyecto, puede ser útil en muchas áreas. En este proyecto se ha trabajado con el robot bípedo NAO, pero el sistema diseñado es fácil de adaptar para poder utilizarlo en otro tipo de robot o incorporar el desarrollo realizado sobre una aplicación real, como por ejemplo una aplicación orientada a la supervisión de una persona con algún tipo de necesidad especial.

También se podría utilizar para realizar diversos estudios, tanto científicos como comercial. Por ejemplo, se podría disponer de un grupo de personas a las que se les analizara la evolución emocional en un periodo de tiempo. Esto podría servir para múltiples propósitos, por ejemplo el estudio de como una persona reacciona a diversas imágenes. También se podría utilizar con un objetivo comercial, haciendo uso del sistema diseñado para ver la evolución emocional de una persona que visualiza un vídeo publicitario y así determinar si la publicidad cumple su función o se debería considerar cambiar algún aspecto para hacer más adecuada.



## **Referencias**

- [1] *Diccionario Marín de la lengua Española*. Vol. 1. Editorial Marín, S.A., 1982
- [2] A. L. Laureano, D. E. Hernández, M. Mora, J. Ramírez. *Aplicación de un modelo cognitivo de valoración emotiva a la función de evaluación de tableros de un programa que juega al ajedrez*. Revista de Matemática: Teoría y aplicaciones. 2012
- [3] A. Mehrabian. *Pleasure-Arousal-Dominance: A general Framework for Describing and Measuring Individual Differences in Temperament*. Universidad de California, Los Angeles. 1996
- [4] A. Mehrabian. *Basic dimensions for a general psychological theory*. Oelgeschager, Gunn & Hain. 1980
- [5] D. A. Barteneva. *Computational mind models for emotional behavioral multi-agent systems*. Universidad de Oporto. 2006
- [6] Shen Zhang, Zhiyong Wu, Helen M. Meng, Lianhong Cai. *Facial expression synthesis using PAD emotional parameters for a chinese expressive avatar*. Universidad Tsinghua, Beijing y Universidad de Hong Kong, HKSAR.
- [7] Aldebaran Robotics. *Documentación del robot NAO*. 2014
- [8] Asignatura “Sistemas de almacenamiento y recuperación de información”. *Introducción a Python*. Escuela técnica superior de ingeniería informática, Universidad Politécnica de Valencia.
- [9] Microsoft Cognitive Services. *Documentación de Face API*. 2016
- [10] Microsoft Cognitive Services. *Documentación de Emotion API*. 2016