Universidad Zaragoza
1542

Proyecto Fin de Carrera

Desarrollo de un agente virtual multimodal para dispositivos móviles

Autor

Jorge Barriendos Sanz

Directores

Dra. Sandra Baldassarri
Dra. Eva Cerezo Bagdasari

Escuela de Ingeniería y Arquitectura
Septiembre 2014

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Departamento de Informática e Ingeniería de Sistemas Universidad Zaragoza

DEPARTAMENTO DE INFORMÁTICA E INGENIERÍA DE SISTEMAS

GIGA Affective Lab

GRUPO DE INFORMÁTICA GRÁFICA AVANZADA AFFECTIVE LAB

Ficha Técnica

Proyecto Fin de Carrera

Título:	Desarrollo de un agente virtual para dispositivos móviles
Autor:	D. Jorge Barriendos Sanz
DNI:	73009624-B
Promoción:	2007-2012
Titulación:	Ingeniería Informática
Directoras:	Dra. Sandra Baldassarri Dra. Eva Cerezo Bagdasari
Departamento:	Informática e Ingeniería de Sistemas
Centro:	Escuela de Ingeniería y Arquitectura
Universidad:	Universidad de Zaragoza
Fecha:	Septiembre 2014

Derechos de Autor

Desarrollo de un agente virtual multimodal para dispositivos móviles

RESUMEN

La finalidad de este Proyecto Fin de Carrera, llevado a cabo en el seno del grupo GIGA Affective Lab, ha sido desarrollar un agente virtual para dispositivos móviles que permita una interacción multimodal y lo más natural posible con el usuario, teniendo en consideración las limitaciones de procesado intrínsecas a estos dispositivos.

Para lograr dicho objetivo, las tareas que se han abordado a lo largo de la realización del proyecto son las siguientes:

- Se ha llevado a cabo la elección de las herramientas con las que implementar el agente virtual, específico para dispositivos móviles, tanto desde el punto de vista gráfico como para desarrollar cada uno de sus modos de interacción a incorporar al propio agente. Con este fin se ha realizado un exhaustivo análisis de las distintas opciones existentes así como de las utilizadas en otros proyectos de índole similar.

- Se han desarrollado los módulos necesarios para hacer posible la comunicación oral entre el agente virtual y el usuario. En este sentido, se han implementado un sintetizador de voz o TTS, que permite reproducir oralmente el discurso del agente virtual, y un reconocedor de discurso o ASR, que permite al usuario dirigirse al agente de forma oral.

- Se han desarrollado dos módulos que permiten la comunicación escrita entre el agente virtual y el usuario. En primer lugar, se ha implementado un panel deslizable, el cual reproduce textualmente los mensajes provenientes del agente virtual en el interior de la interfaz gráfica del sistema. En segunda instancia, se ha implementado un área de texto, situada dentro de la interfaz, la cual se encarga de gestionar el proceso de escritura del mensaje por parte del usuario y de enviar dicho mensaje al agente virtual.

- Se ha dotado al sistema de una interfaz gráfica en la que el agente virtual es el elemento central de la misma, permitiendo al agente expresarse y comunicarse de forma visual con el usuario. Además, se ha desarrollado un módulo motor encargado de gestionar las animaciones, tanto corporales como faciales, que incorpora el agente virtual.

- Se ha desarrollado también un módulo Gestor de Diálogo que permite programar al agente virtual de forma que sea capaz de mantener una conversación fluida y coherente con el usuario. Este módulo Gestor de Diálogo, basado en el Programa AB, utiliza ficheros AIML para reconocer los mensajes del usuario y llevar a cabo la búsqueda de las respuestas más adecuadas a cada uno de estos mensajes.

- Se ha dotado de aspectos emocionales a todas las modalidades de interacción entre el agente virtual y el usuario. En este sentido, se han generado distintas voces emocionales para cada uno de los estados emocionales en los que se puede encontrar el agente, se han seleccionado distintos colores de fuente y texturas para denotar diversas emociones a través del panel deslizable y se han utilizado personajes tridimensionales que incorporan animaciones que se corresponden con cada uno de los estados emocionales del agente virtual a representar.

- Se han llevado a cabo pruebas exhaustivas y sistemáticas con usuarios finales para evaluar las voces emocionales que incorpora el sistema, así como la influencia del contenido semántico de las frases y la imagen en la percepción emocional del usuario.

El trabajo se ha centrado en un subconjunto de los dispositivos móviles existentes, los *smartphones* Android, y se ha hecho uso de la plataforma de desarrollo comercial Unity3D puesto que es la más utilizada hoy en día para desarrollar aplicaciones 3D sobre estos dispositivos.

Agradecimientos

Quiero dar las gracias a mis directoras, las Dras. Sandra Baldassarri y Eva Cerezo, quienes me han ayudado en todo momento, me han enseñado a afrontar el problema paso a paso y con las que he aprendido a explicar de forma detallada un trabajo como el aquí realizado.

Agradecer también al Dr. Javier Marco su interés por el desarrollo de este proyecto, habiéndome prestado ayuda y consejo siempre que los he requerido.

Y gracias a mi familia y amigos por haber aceptado realizar cuantas pruebas han sido necesarias tanto en la fase de generación de voces emocionales realistas como en la fase de evaluación de este proyecto.

Índice general

1. Introducción

En este primer capítulo de la memoria se introduce el Proyecto Fin de Carrera, describiendo el contexto en el que se ha desarrollado y los objetivos concretos que se han perseguido con su realización. Por último, se presenta la estructura de este documento, dando a conocer tanto los capítulos que conforman la memoria principal como los anexos.

1.1 Contexto

El grupo GIGA AffectiveLab, perteneciente al Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, es un grupo de investigación centrado en el ámbito de la interacción persona-ordenador. A lo largo de los últimos años, su trabajo se ha focalizado en cuatro cuestiones principales: el desarrollo de interfaces de usuario tangibles, la consideración de los aspectos afectivos propios de la interacción con el usuario, el desarrollo de interfaces accesibles para todo tipo de usuarios y la gestión de agentes virtuales que puedan ser utilizados como interfaces multimodales en aplicaciones de tiempo real.

El esfuerzo del grupo en el ámbito de los agentes virtuales se ha centrado en el desarrollo del sistema Maxine [Baldassarri et al, 2008].

Maxine es un potente motor gráfico, dirigido por *scripts*, que permite la representación y gestión en tiempo real de entornos y agentes virtuales tridimensionales. Gracias a este sistema es posible desarrollar agentes virtuales que soporten una interacción multimodal en tiempo real, permitiendo al usuario interactuar con el propio agente a través de múltiples vías, a saber: texto, voz, imágenes y movimiento.

Los agentes de Maxine poseen la capacidad de variar su estado emocional a lo largo de la interacción con el usuario, haciendo conocedor al mismo de dichos cambios a través de sus reacciones, el tono de voz utilizado y el tipo de respuesta dado. Para ello, los agentes virtuales incorporan múltiples animaciones, tanto faciales como corporales, con los que expresan en todo momento su estado de ánimo. Además, el proceso de generación de voz sintética es parametrizable, siendo posible variar aspectos como la velocidad, el tono y/o el volumen del discurso, lo que permite corresponder el estado emocional mostrado por el agente a través de las animaciones anteriores con el tipo de discurso usado por el mismo para comunicarse con el usuario. Todo ello, unido a la sincronización labial de la cual están dotados los agentes, otorga un alto grado de naturalidad a la interacción entre el usuario y el agente. Estos agentes han sido ya utilizados con éxito como presentadores virtuales (véase Figura 1.1.1), asistentes para el control de entornos domóticos [Baldassarri et al, 2007], intérpretes virtuales [Baldassarri et al, 2009] y agentes pedagógicos encargados de enseñar distintas materias [Serón et al, 2008]. Pero su utilización siempre ha estado sujeta al uso de máquinas con una alta capacidad de procesado que pudieran soportar los distintos módulos que conforman el sistema.

No obstante, el continuo desarrollo y mejora en prestaciones de dispositivos móviles como las *tablets* y los *smartphones*, unido a su enorme proliferación en los últimos años, ha abierto un nuevo campo sobre el que trabajar. De este modo, surge la motivación de desarrollar un agente virtual para dispositivos móviles de características similares a los desarrollados con Maxine, siendo conscientes en todo momento de que la capacidad de procesado de estos dispositivos es inferior a la de los ordenadores en los que normalmente se trabaja con Maxine.

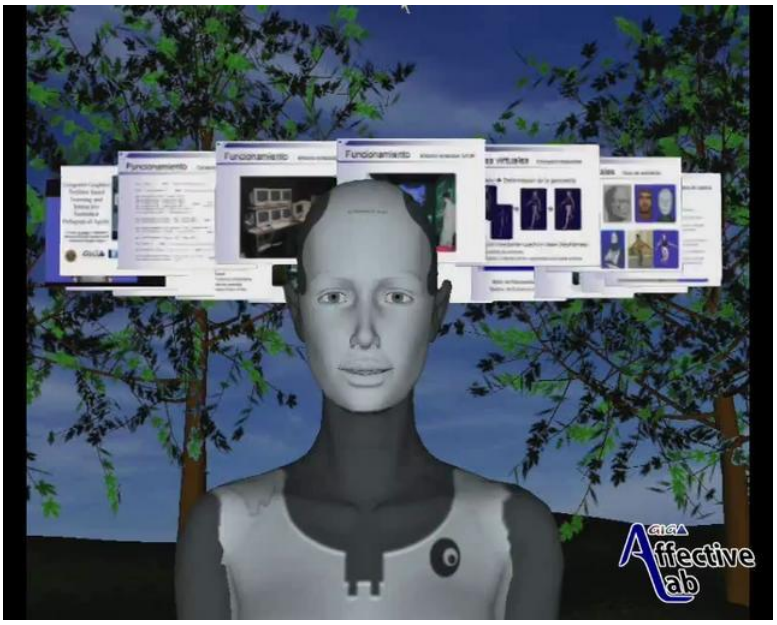


Figura 1.1.1: *Captura de pantalla de una presentación virtual*

1.2 Objetivos del proyecto

El objetivo principal de este Proyecto Fin de Carrera es desarrollar un sistema basado en un agente virtual para dispositivos móviles, que permita una interacción multimodal y lo más natural posible con el usuario, teniendo en consideración las limitaciones de procesado intrínsecas a estos dispositivos. Dentro de la amplia gama de dispositivos móviles existentes, este proyecto se centra en un subconjunto de los mismos, más concretamente, en los *smartphones* Android. La elección de este subconjunto se debe a su amplia difusión en el mercado [CM ANDROID web].

Este objetivo define la meta última perseguida, pero para alcanzarla es necesario abordar una serie de objetivos intermedios. A continuación se hace un desglose detallado de los objetivos secundarios o aspectos que será necesario contemplar:

- Elección del entorno de trabajo y de las herramientas con las que implementar el agente virtual, específico para dispositivos móviles, tanto desde el punto de vista gráfico como para desarrollar cada uno de sus modos de interacción a incorporar al propio agente. Con este fin se llevará a cabo un exhaustivo análisis de las distintas opciones existentes así como de las utilizadas en otros proyectos de índole similar.

- Desarrollo de los módulos que hagan posible la comunicación oral entre el agente virtual y el usuario. En este sentido, se precisarán tanto un módulo sintetizador de voz como un módulo reconocedor de discurso.

- Desarrollo de un mecanismo que permita la comunicación escrita entre el agente virtual y el usuario a través de campos de texto.

- Desarrollo de una interfaz gráfica en la que el agente sea capaz de expresarse de forma visual. Con este objetivo se precisará gestionar las animaciones que incorpore el agente virtual, tanto corporales como faciales, permitiendo la mezcla o reproducción simultánea de varias de estas animaciones.

- Consideración, en la medida de lo posible, de los aspectos emocionales en todas las modalidades de interacción entre el agente virtual y el usuario.

- Implementación de un prototipo que permita, en un escenario sencillo de aplicación, valorar con usuarios los modos de interacción y las prestaciones del sistema.

1.3 Estructura de la memoria

El presente documento se encuentra dividido en dos partes bien diferenciadas: una primera parte que contiene la memoria principal, donde se explica el trabajo realizado a lo largo del proyecto; y una segunda que la conforman los anexos, donde se profundiza acerca de algunos aspectos de la memoria.

La memoria principal consta de los siguientes capítulos:

- **Capítulo 1 –** *Introducción:* describe el contexto de desarrollo del proyecto, los objetivos a alcanzar y la estructura de la memoria.

- **Capítulo 2 –** *Análisis:* se da a conocer una breve reseña del estado del arte, se explica la plataforma de desarrollo utilizada y se describe un conjunto de requisitos funcionales y no funcionales que deberá cumplir el sistema a implementar.

- **Capítulo 3 –** *Diseño de la interacción con el agente virtual:* se presenta la propuesta de diseño del sistema, se explican los módulos que intervienen tanto en la comunicación oral como escrita del agente virtual, se detalla la gestión de diálogo llevada a cabo durante la interacción, se explica el módulo motor y se describe brevemente la interfaz gráfica del sistema.

- **Capítulo 4 –** *Pruebas con usuarios: generación y evaluación de las voces preliminares:* se presentan los métodos de evaluación seguidos a lo largo de las distintas fases de pruebas con usuarios, se explican brevemente las pruebas preliminares llevadas a cabo para la generación de las voces emocionales y se detallan cada una de las pruebas realizadas para la evaluación de dichas voces y otros aspectos emocionales del sistema.

- **Capítulo 5 –** *Resultados de las pruebas con usuarios:* se da a conocer la caracterización de las distintas voces emocionales generadas y se describen los resultados más relevantes de cada una de las pruebas realizadas para evaluar dichas voces emocionales junto a otros aspectos emocionales del sistema.

- **Capítulo 6 –** *Conclusiones y trabajo futuro:* se analiza el cumplimiento de los objetivos, se proponen posibles ideas para un trabajo futuro sobre el sistema desarrollado y se muestra la opinión personal del autor sobre el trabajo realizado a lo largo del PFC.

La memoria principal viene acompañada por lo siguientes anexos:

- **Anexo A** – *Estudios previos relacionados:* incluye un amplio resumen de toda la información recolectada en el proceso de documentación acerca del desarrollo de agentes virtuales sobre dispositivos móviles. Además, se explican los ejemplos de agentes virtuales sobre dispositivos móviles ya existentes.

- **Anexo B** – *Proceso de ingeniería del software:* se explica el modelo de proceso seguido a lo largo de este Proyecto Fin de Carrera junto con sus ventajas e inconvenientes.

- **Anexo C** – *Documentación del desarrollo del software:* se describen los procesos de análisis y diseño del sistema, así como se muestra toda la documentación asociada a los mismos.

- **Anexo D** – *Implementación de la interacción con el agente virtual:* se describen los aspectos más importantes de la implementación llevada a cabo para dotar al sistema de la capacidad de interactuar con el usuario de una forma multimodal.

- **Anexo E** – *Prototipo funcional de la interfaz gráfica:* Se explica en detalle el diseño y funcionalidad de la primera versión de la interfaz gráfica desarrollada.

- **Anexo F** – *Generación de las voces emocionales:* Se describen en detalle tanto la metodología seguida como las distintas fases llevadas a cabo durante el proceso de generación de las voces emocionales.

- **Anexo G** – *Resultados adicionales:* se detallan los resultados obtenidos en cada una de las pruebas realizadas con usuarios para evaluar las voces emocionales generadas junto a otros aspectos emocionales del sistema.

2. Análisis

Este capítulo presenta el análisis previo al desarrollo del nuevo sistema y se encuentra dividido en dos apartados. En el primero de ellos se realiza una breve introducción al desarrollo de agentes virtuales para dispositivos móviles, mientras que en el segundo se describen los requisitos funcionales y no funcionales identificados para el sistema.

2.1 Análisis del problema

A continuación se pretende introducir al lector en el desarrollo de agentes virtuales para dispositivos móviles. Con este fin, se lleva a cabo un breve resumen de las características propias de este tipo de agentes virtuales, se explican las plataformas de desarrollo utilizadas en trabajos anteriores y se acaba por describir la plataforma elegida para la realización del proyecto.

2.1.1 Agentes virtuales sobre dispositivos móviles

En las siguientes líneas se explican brevemente las características más representativas de los agentes virtuales desarrollados para dispositivos móviles y se presentan varios ejemplos ya existentes de este tipo de agentes (explicado todo ello con mayor detalle en la sección A.1 del Anexo A).

a) Características de los agentes

A pesar de ser utilizados en ámbitos muy diversos, los agentes virtuales desarrollados para ser utilizados sobre dispositivos móviles poseen múltiples similitudes entre sí. En las siguientes líneas se procede a describir los distintos aspectos en los que se asemejan este tipo de agentes virtuales (detallados en mayor profundidad en la sección A.1.1 del anexo A), a saber: el personaje, las animaciones y la interacción con el usuario.

Personaje

Todos los agentes virtuales suelen estar representados gráficamente por un personaje. Este personaje, que es el elemento principal de la interfaz gráfica del sistema, es el resultado de un proceso de diseño y modelado en el que el desarrollador define el tipo de personaje que mejor se adecúa, tanto a nivel computacional como estético, al nuevo sistema.

Por un lado, el desarrollador del sistema puede optar por utilizar imágenes bidimensionales para conformar gráficamente al nuevo agente virtual. De esta manera, se reducen tanto el espacio de memoria como el tiempo de procesado necesarios para generar al agente, lo que permite al sistema ser ejecutado por un mayor rango de dispositivos móviles. Sin embargo, disminuye el grado de realismo del agente virtual. Por otra parte, se puede optar por el uso de un personaje tridimensional para la representación gráfica del nuevo agente virtual. En este caso, es necesario que los dispositivos móviles sobre los que se ejecute el sistema dispongan de una cierta capacidad de procesado, ya que se precisa renderizar[1] la imagen continuamente. Cabe destacar que, si bien ambas opciones se han utilizado con éxito en el pasado, cada vez son más los sistemas que incorporan un personaje tridimensional debido al continuo desarrollo de los dispositivos móviles. Es por ello que para el desarrollo del nuevo sistema se opta por hacer uso de un personaje tridimensional para representar gráficamente al agente virtual, asumiendo las peculiaridades de diseño y modelado que conlleva el desarrollo de este tipo de agentes para dispositivos móviles (véase apartado Personaje de la sección A.1.1 del anexo A).

[1] *Renderizar: (Del inglés rendering) proceso de generar una imagen o animación en 3D a partir de un modelo, usando para ello una aplicación de computador.*

Animación

Con el objetivo de dotar de un mayor realismo a los agentes virtuales, todos ellos suelen incorporar una serie de animaciones que les permite actuar y reaccionar de forma similar al comportamiento humano. Dichas animaciones, ya sean faciales, corporales o mixtas, deben ser definidas por el desarrollador en el proceso de animación del agente virtual. Las dos técnicas de animación más utilizadas en el campo de la informática gráfica son la captura de movimiento (*motion capture*) y la animación por planos clave (*key-frame animation*).

La técnica de captura de movimiento (*motion capture*) se basa en la grabación de los movimientos y gestos realizados por un actor, generalmente una persona o animal, y el traslado de éstos a un modelo digital, en este caso el agente virtual. Con esta técnica se logran movimientos dotados de un gran realismo y expresividad en períodos de tiempo no muy grandes; sin embargo, la exageración de ciertos movimientos o expresiones no puede ser lograda mediante esta técnica ya que el actor no es capaz de realizarlos. Además, otro de los inconvenientes de esta técnica es que si la estructura esqueletal del agente es demasiado simple, una gran cantidad de datos se pierden. Por todo ello, se descarta esta técnica de animación para el desarrollo de un agente virtual para dispositivos móviles.

En contraposición a la técnica anterior, en la animación por planos clave (*key-frame animation*) no existe actor alguno. Esta técnica consiste en que el diseñador del agente virtual fija una posición inicial y una final para cada movimiento de un elemento, generándose de forma automática las posiciones intermedias que permiten mover el elemento seleccionado de forma suave y continua desde la posición inicial hasta la posición final. Utilizando esta segunda técnica, los movimientos y expresiones pueden ser tan exagerados y particulares como sea posible en la imaginación del diseñador. Además, otra gran ventaja que ofrece esta técnica es que otorga un mayor control sobre las posiciones de la animación.

Por otro lado, existen sistemas dedicados a la animación de agentes virtuales que incorporan herramientas que guían al usuario a través de cada una de las fases del proceso de modelado de animaciones, como es el caso del sistema MAge-AniM [Chittaro et al, 2006] y su herramienta H-Animator [Nadalutti et al, 2006] (explicados en el apartado Animación de la sección A.1.1 del Anexo A).

Interacción

Una de las características más importantes de los agentes virtuales es que deben ser capaces de interactuar con el usuario. Esta interacción se puede llevar a cabo a través de diversos canales, a saber: escrito, oral y visual. A continuación, se detallan los elementos y módulos necesarios para permitir los distintos tipos de interacción con el usuario mencionados.

La interacción con el usuario a través del canal visual se limita, en la inmensa mayoría de los sistemas, a la reproducción de las animaciones que incorpora el agente virtual. Estas animaciones son las encargadas de dar a conocer las reacciones y los distintos estados emocionales del agente. Con el fin de gestionar la interacción visual con el usuario, el sistema debe disponer de un módulo que almacene el estado del agente virtual y lo actualice en función de la información proveniente del usuario, disparando la animación correspondiente a cada cambio de dicho estado.

Por otro lado, para la interacción con el usuario a través del canal oral son necesarios dos elementos: un reconocedor de discurso (*Automatic Speech Recognition*), encargado de captar los mensajes pronunciados oralmente por el usuario, y un sintetizador de voz a partir de texto (*Text To Speech*) que permita reproducir oralmente las respuestas del agente virtual.

Por último, la interacción con el usuario a través del canal escrito se fundamenta en la existencia de un área de texto y un panel de texto. En el caso del área de texto, es utilizada para que el usuario escriba la información que desea transmitir al agente virtual, por lo que suele ser lo suficientemente amplia para que quepan varias líneas de texto. Por su parte, sobre el panel se muestran las respuestas del agente virtual, impidiendo que el usuario pueda modificar en modo alguno el texto mostrado. Ambos elementos son parte de la interfaz gráfica del sistema, y su implementación depende de la plataforma de desarrollo y del lenguaje de programación utilizados.

b) Ejemplos

Debido al continuo aumento tanto de la capacidad de procesado como de la memoria de los dispositivos móviles, empiezan a aparecer sistemas basados en agentes virtuales desarrollados para ser ejecutados sobre estos dispositivos. A continuación, se citan algunos ejemplos de aplicación de estos sistemas (descritos más ampliamente la sección A.1.2 del anexo A).

- Intérprete del lenguaje de signos: a través de un sistema basado en un agente virtual se ha creado un foro específico para personas con problemas auditivos [Buttussi et al, 2007]. En este foro, el agente virtual actúa como intérprete, transmitiendo mediante el lenguaje de signos los mensajes escritos por los usuarios (véase Figura 2.1.1).

- Guía de entrenamiento: se ha desarrollado una aplicación en la que un agente virtual desempeña el papel de guía durante el entrenamiento del usuario, ayudando a este último a realizar correctamente los ejercicios presentes en una determinada pista de fitness [Buttussi et al, 2006] (veáse Figura 2.1.1).

- Presentador virtual: otra aplicación ya disponible es un presentador virtual encargado de narrar los titulares de las noticias más importantes del momento al usuario [Santi et al 2003].

- Entrenador personal: un nuevo ejemplo de sistema basado en agente virtual es el entrenador personal desarrollado en el proyecto Smarcos [Smarcos web]. Este entrenador personal virtual insta a los usuarios a lograr y mantener un estilo de vida saludable, siendo sensible al contexto de las actividades diarias que realizan los usuarios a través de una gama de dispositivos interconectados.

- Controlador del sistema domótico del hogar: se ha desarrollado un sistema que permite controlar el sistema domótico de un domicilio mediante la interacción con un agente virtual [Santos-Pérez et al, 2013]. En dicho sistema, el usuario puede controlar el cierre/apertura de puertas, el encendido/apagado de luces, la subida/bajada de persianas y la temperatura de la casa a través de la interfaz gráfica o de una conversación con el agente virtual.

Figura 2.1.1: *Ejemplos de aplicación de sistemas basados en agentes virtuales para dispositivos móviles. En la imagen de la izquierda aparece el agente virtual encargado de expresar en el lenguaje de los signos las oraciones que los usuarios escriben en el foro. En la imagen de la derecha se observa al agente virtual encargado de llevar a cabo las demostraciones de los ejercicios a realizar en cada una de las estaciones de ejercicios.*

2.1.2 Plataformas de desarrollo de agentes virtuales para dispositivos móviles

En este sub-apartado de la memoria principal se explican brevemente dos plataformas de desarrollo de agentes virtuales para dispositivos móviles ya existentes (descritas en detalle en la sección A.2 del Anexo A) y que han servido de base a este trabajo.

a) Elckerlyc

Elckerlyc [Klaasen et al, 2012] es una plataforma, basada en modelos, que sirve para la especificación y animación de agentes virtuales que soporten una interacción multimodal en tiempo real. Su uso está centrado en dispositivos móviles como *tablets* y *smartphones*.

Esta plataforma permite desarrollar sistemas basados en agentes virtuales capaces de interactuar con el usuario a través de diferentes canales, a saber: oral, textual y visual.

Comunicación Oral: con el fin de permitir a los agentes comunicarse oralmente con el usuario, la plataforma Elckerlyc incorpora el módulo SpeechEngine, módulo encargado de expresar de forma oral el discurso seleccionado para el agente virtual a través del altavoz del dispositivo. En las versiones anteriores, dirigidas al uso sobre un ordenador, el motor de conversión de texto a voz (*Text To Speech – TTS*) utilizado presenta múltiples dependencias del sistema operativo sobre el que trabaja, no siendo posible su reutilización sobre Android sin realizar cambios significativos. Por ello, en esta nueva versión se opta por utilizar el sistema TTS propio de Android, adaptando para ello el módulo SpeechEngine con el fin de cargar e inicializar dicho sistema.

Comunicación Escrita: el modo de interacción escrita se lleva a cabo a través del módulo TextSpeechEngine, módulo encargado de recibir el discurso proveniente del SpeechEngine y reproducirlo en el interior del cuadro de texto con formato PNG, compatible con el entorno gráfico de Android, existente en la interfaz del sistema.

Comunicación Visual: en cuanto a la comunicación visual, Elckerlyc se sirve de Picture Engine, un nuevo realizador gráfico desarrollado específicamente para esta plataforma. Picture Engine es un realizador gráfico sin grandes requerimientos computacionales, ejecutable sobre la plataforma Android, que se sirve de una colección de imágenes bidimensionales para la conformación gráfica del agente virtual. Aunque el modelado 2D del agente virtual conlleva algunas limitaciones, como la disminución del realismo, presenta a su vez ventajas especialmente notorias en este tipo de dispositivos. En primer lugar, supone una gran reducción del tiempo de procesado y espacio de memoria requeridos por la aplicación desarrollada, así como disminuye el consumo de batería de la misma. Además, soporta una gran variedad de técnicas de diseño para los agentes virtuales, siendo posible diseñarlos a partir de una figura de dibujos animados, de imágenes 3d pre-renderizadas o, incluso, a partir de fotografías de una persona real.

b) Plataforma de desarrollo de interfaces basadas en el uso de agentes virtuales para Android de la Universidad de Málaga

En este artículo [Santos-Pérez et al, 2013], elaborado por investigadores de la Escuela de Ingeniería de Telecomunicaciones de la Universidad de Málaga, se presenta una plataforma que permite el desarrollo de interfaces basadas en agentes virtuales para dispositivos móviles Android.

Los sistemas basados en agentes virtuales desarrollados con esta plataforma son capaces de interactuar con el usuario a través de dos canales, el oral y el visual.

Comunicación Oral: en esta modalidad de interacción intervienen un gran número de módulos del sistema, los cuales se describen a continuación según el orden de intervención:

- Voice Activity Detector (VAD): que tiene como principal cometido el discriminar los fragmentos de audio que contengan voz del usuario frente a los solamente contengan ruido. La implementación del VAD se basa en la biblioteca SphinxBase

- Automatic Speech Recognition (ASR): se encarga de convertir el discurso del usuario a texto. Toma como entrada los fragmentos de audio con el discurso del usuario provenientes del VAD y, tras el proceso de conversión, envia el texto resultante al Conversational Engine. El módulo ASR está basado en la biblioteca de reconocimiento de discurso PocketSphinx.

- Conversational Engine: este módulo es el encargado de extraer el significado de las distintas palabras y/o expresiones reconocidas por el ASR, gestionar el flujo de diálogo y generar respuestas en base a dicho significado de las expresiones, el historial del diálogo y el estado actual de la conversación. Está basado en PyAIML, un *chatbot* de AIML (Artificial Intelligence Markup Language).

- Text To Speech: su cometido es generar el discurso oral del agente a partir de las respuestas en forma de cadena de texto provenientes del Conversational Engine. La implementación de este módulo está basada en la biblioteca eSpeak.

Comunicación Visual: la interacción visual del sistema con el usuario depende de los dos módulos que se describen a continuación:

- Control Interface: es el módulo encargado de dotar funcionalidad a todos los elementos presentes en la interfaz gráfica del sistema.

- Virtual Head Animation: su cometido es generar las expresiones faciales y los visemas del agente virtual. El motor de renderizado utilizado es Ogre 3D.

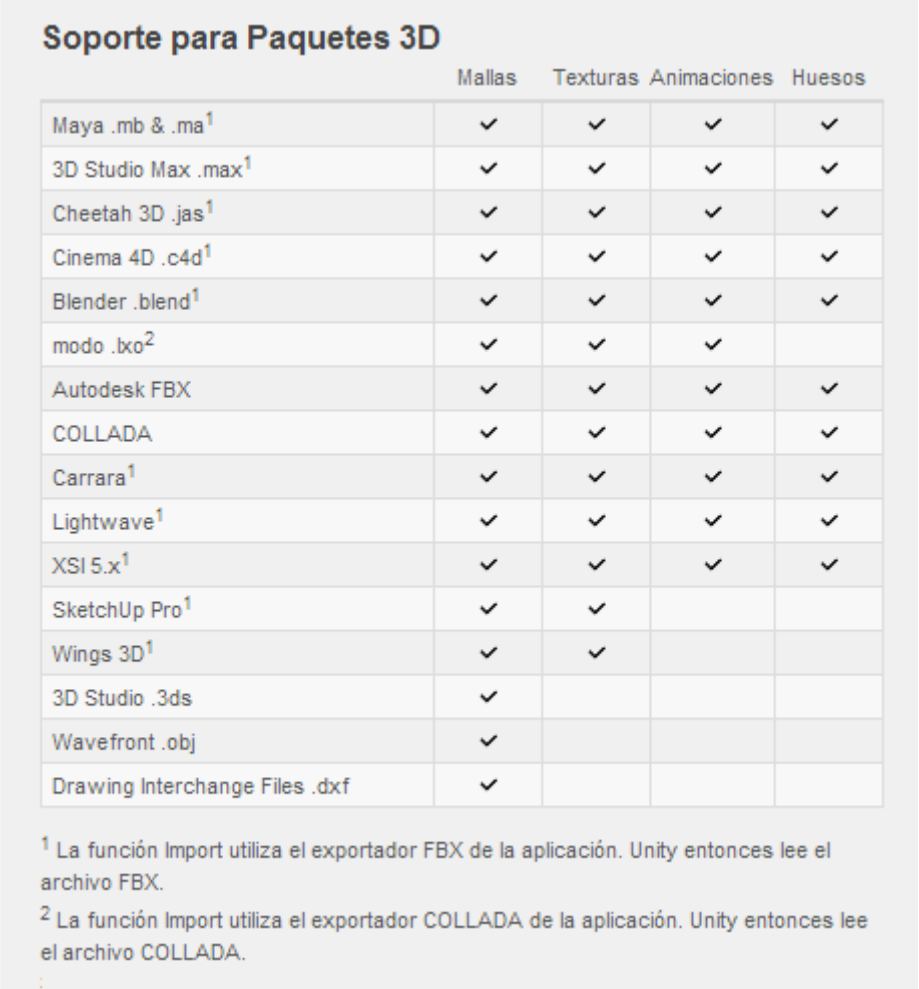
Cabe destacar que, a través del estudio realizado de estas plataformas, ha sido posible asentar las bases del proceso de desarrollo del nuevo sistema, desarrollo que viene descrito a lo largo de esta memoria principal.

2.1.3 Plataforma Unity 3D

Una vez identificadas las características que deben poseer los agentes virtuales destinados a ser ejecutados sobre dispositivos móviles y analizadas las distintas plataformas utilizadas en otros trabajos ya existentes, se procede a describir la plataforma Unity 3D [Unity3D web], plataforma seleccionada desde un principio por el grupo de trabajo para el desarrollo de este Proyecto Fin de Carrera. Esta elección se debe a que la plataforma Unity 3D permite hacer uso de los personajes tridimensionales de los que dispone el grupo de trabajo y está dotada de mecanismos con los que adicionar las distintas funcionalidades de interacción que precisan dichos personajes.

La plataforma Unity 3D es un plataforma de desarrollo, mayoritariamente de juegos, que presenta un potente motor de renderizado al que acompañan un juego completo de herramientas intuitivas y flujos rápidos de trabajo que permiten la creación de contenido 3D interactivo. Además, Unity 3D permite publicar contenido para múltiples plataformas como PC, Mac, Nintendo, Android e iPhone, siendo posible la adaptación a otras nuevas plataformas a través de *plugins*.

Una de las razones por la que se ha optado por utilizar Unity 3D es su canalización (pipeline) de activos, ya que presenta una amplia asistencia de herramientas industriales y es sencilla de manejar. En caso de que se desee importar en un determinado proyecto de Unity los modelos, texturas, audio, *scripts* y demás activos procedentes de otras aplicaciones 3D, únicamente se precisa guardar dichos activos en la carpeta del proyecto, realizando Unity la importación de forma automática para su uso inmediato. Además, es posible modificar los activos en cualquier momento, siendo capaces de observar las modificaciones realizadas sobre el proyecto de forma instantánea. En la Tabla 2.1.1 se muestra el listado de los programas y formatos 3D soportados por Unity 3D.



Soporte para Paquetes 3D

	Mallas	Texturas	Animaciones	Huesos
Maya .mb & .ma[1]	✓	✓	✓	✓
3D Studio Max .max[1]	✓	✓	✓	✓
Cheetah 3D .jas[1]	✓	✓	✓	✓
Cinema 4D .c4d[1]	✓	✓	✓	✓
Blender .blend[1]	✓	✓	✓	✓
modo .lxo[2]	✓	✓	✓	
Autodesk FBX	✓	✓	✓	✓
COLLADA	✓	✓	✓	✓
Carrara[1]	✓	✓	✓	✓
Lightwave[1]	✓	✓	✓	✓
XSI 5.x[1]	✓	✓	✓	✓
SketchUp Pro[1]	✓	✓		
Wings 3D[1]	✓	✓		
3D Studio .3ds	✓			
Wavefront .obj	✓			
Drawing Interchange Files .dxf	✓			

[1] La función Import utiliza el exportador FBX de la aplicación. Unity entonces lee el archivo FBX.

[2] La función Import utiliza el exportador COLLADA de la aplicación. Unity entonces lee el archivo COLLADA.

Tabla 2.1.1: *Soporte de paquetes 3D por parte de Unity3D. En la columna de la izquierda se listan los programas 3D junto a las extensiones de sus archivos generados que son soportados por Unity3D. Los ticks que aparecen en el resto de columnas indican si las mallas, texturas, animaciones o huesos generados en cada uno de estos programas pueden ser transportados a un proyecto Unity3D.*

Como se muestra en la Tabla 2.1.1, la plataforma Unity 3D soporta la importación de paquetes procedentes de 3D Studio. Este hecho permite la reutilización de alguno de los modelos tridimensionales de los que dispone el grupo de trabajo para la realización de este Proyecto Fin de Carrera, más concretamente todos aquellos modelos de personajes virtuales que no posean un alto número de polígonos.

Por otra parte, con el fin de desarrollar un sistema basado en un agente virtual para dispositivos móviles Android es necesario que la plataforma seleccionada permita publicar contenido para Android y sea capaz de comunicarse con los servicios que presta dicho sistema operativo. Por un lado, Unity 3D posee la capacidad de publicar contenido para múltiples plataformas, entre ellas Android, de forma transparente al usuario, lo que facilita la labor del desarrollador ya que no precisa realizar ninguna acción complementaria. En cambio, en el caso de la comunicación con los servicios propios de Android, la plataforma Unity 3D no dispone de la funcionalidad necesaria para llevarla a cabo, aunque permite la adhesión al proyecto Unity de *plugins* JAVA, implementados por el propio desarrollador, a través de los cuales comunicarse con los distintos servicios Android. Esta solución, si bien supone una mayor carga de trabajo para el desarrollador, permite la utilización por parte del nuevo sistema del *Text To Speech* o el reconocedor de discurso que incorporan todos los dispositivos móviles Android.

Por último, otra de las características interesantes de Unity 3D para el desarrollo de este Proyecto Fin de Carrera es su gestión rápida e intuitiva de los elementos presentes en la interfaz gráfica del sistema a través de *scripts*. Para la programación de estos *scripts* se utiliza uno de los tres lenguajes soportados:

JavaScript, C# y Boo. Los tres lenguajes son sencillos de utilizar y se ejecutan en la plataforma Open Source .NET Mono, con rápidos tiempos de compilación.

2.2 Análisis de requisitos

Tras haber conocido las principales características que presentan los agentes virtuales desarrollados para dispositivos móviles, estudiado los trabajos existentes y analizado las distintas plataformas utilizadas en éstos últimos, en esta sección se procede a identificar los requisitos del sistema a desarrollar en este Proyecto Fin de Carrera. Cabe destacar que para el desarrollo de este sistema se ha optado por seguir un modelo de proceso incremental (decisión que se explica detalladamente en el Anexo B), el cual consiste en ir realizando secuencialmente modificaciones o extensiones sucesivas a la aplicación hasta obtener el sistema final

Como ya se ha mencionado anteriormente, el objetivo que persigue este Proyecto Fin de Carrera es el desarrollo de un sistema basado en un agente virtual, para dispositivos móviles Android, que permita una interacción multimodal con el usuario lo más natural posible.

A continuación se muestra la lista de requisitos, tanto funcionales como no funcionales, que debe cumplir el sistema a desarrollar. Estos requisitos se obtienen en el proceso de análisis del problema, el cual se trata en mayor detalle en el apartado C.1 Metodología de Análisis del Anexo C.

Requisitos funcionales:

RF1 – El sistema debe permitir al usuario comunicarse oralmente con el agente virtual. Para ello, debe ser capaz tanto de reconocer el discurso del usuario, captado por el micrófono del dispositivo, como de expresarse de forma oral a través del altavoz del mismo.

RF2 – Se debe otorgar al usuario la posibilidad de comunicarse de forma escrita con el agente virtual, permitiendo así la interacción cuando la comunicación oral no sea posible. Para ello, se debe desarrollar un mecanismo de comunicación escrita entre ambos a través de campos de texto.

RF3 – El sistema debe permitir al usuario cambiar la modalidad de la interacción entre él y el agente virtual a través de su interfaz gráfica.

RF4 – El sistema debe ser capaz de gestionar de forma adecuada las animaciones, tanto faciales como corporales, que incorpora el agente virtual.

RF5 – El sistema debe permitir al agente virtual expresar su estado emocional. Dicho estado emocional debe ser apreciable a través de los gestos del agente, así como a través del tono de voz y velocidad de discurso utilizados.

RF6 – El sistema debe permitir al agente virtual variar su estado emocional en función de la conversación que mantenga con el usuario.

RF7 – El sistema debe ser capaz de gestionar el diálogo entre el agente virtual y el usuario. Para ello, debe analizar la información proveniente del usuario, vía texto o vía discurso, gestionar dicha información para buscar una respuesta adecuada y expresarla.

Requisitos no funcionales:

RNF1 – Usar la plataforma de programación Eclipse Juno para el desarrollo de *plugins* que comuniquen al sistema con los distintos servicios Android que se precisen.

RNF2 – Utilizar el motor de renderizado Unity 3D para el desarrollo gráfico tanto del agente virtual como de la interfaz, así como para la integración de los distintos módulos que conforman el sistema.

RNF3 – Usar el lenguaje de programación JAVA para la implementación de los *plugins* a desarrollar para el sistema.

RNF4 – Servirse del lenguaje de programación C# para generar los *scripts* que gestionen la funcionalidad tanto del agente virtual como de su interfaz gráfica.

RNF5 – Trabajar sobre el Smartphone Samsung Galaxy S3 del que dispone el grupo de trabajo**.**

3. Diseño de la interacción con el agente virtual

En este tercer capítulo de la memoria se detalla el diseño modular llevado a cabo en el sistema para dotar al agente virtual de la capacidad de interactuar con el usuario. En las siguientes líneas se muestra la propuesta de diseño del nuevo sistema, se explican los distintos tipos de comunicación con los que trabaja el agente virtual, se detalla la gestión de diálogo realizada durante la interacción con el usuario y se describe la interfaz gráfica del sistema. Exceptuando la propuesta, en el resto de apartados se explican las decisiones de diseño tomadas por el desarrollador, exponiendo las distintas posibilidades encontradas para cada problema y dando a conocer, de forma razonada, la solución adoptada.

3.1 Propuesta de diseño del nuevo sistema

Una vez llevado a cabo el análisis del sistema e identificados los requisitos, tanto funcionales como no funcionales, del mismo, se procede a presentar una primera propuesta de diseño del nuevo sistema a desarrollar (para mayor información acerca del proceso de diseño del sistema, véase la sección C.2 del Anexo C). Esta propuesta de diseño, mostrada en la Figura 3.1.1, está compuesta por distintos módulos, los cuales vienen descritos brevemente a continuación:

- **Comunicación Oral:** la función de este módulo es gestionar la interacción con el usuario a través del canal oral. Por un lado, incorpora un **Reconocedor de Discurso**, basado en la aplicación **Google Voice Search**, que se encarga de captar el mensaje hablado del usuario a través del **micrófono** del dispositivo móvil y transmitirlo al módulo **Gestor de Diálogo**. Por otra parte, incorpora un **Sintetizador de Voz**, basado en el sistema **TTS Android**, encargado de la conversión a voz de las respuestas en formato texto procedentes del **Gestor de Diálogo**. Este sintetizador dispone, a su vez, de una serie de **voces emocionales** con las que pronunciar el mensaje, lo que permite transmitir el estado emocional del agente virtual a través del discurso. La reproducción del discurso se lleva a cabo a través del **altavoz** del dispositivo.

- **Comunicación Escrita:** este módulo, paralelo al anterior, es el encargado de gestionar la interacción con el usuario a través del canal escrito. Dispone de un primer **Módulo de Lectura** cuya función es trasladar los mensajes del **Área de Texto**, escritos por el usuario a través del **teclado** del dispositivo, al módulo **Gestor de Diálogo**. Además, dispone de un **Módulo de Escritura** encargado de mostrar al usuario las respuestas procedentes del **Gestor de Diálogo** a través de un **Panel Deslizable** que se encuentra en la **Interfaz Gráfica**. Este segundo módulo se sirve de las connotaciones emocionales existentes en los colores para dar a conocer al usuario el estado emocional del agente virtual, expresando dichos estados a través de los distintos **colores emocionales** utilizados en la escritura por pantalla de los mensajes de respuesta.

- **Gestor de Diálogo:** este módulo es el encargado de recibir los mensajes provenientes del usuario, interpretar dichos mensajes y generar una respuesta acorde a los mismos. Para ello, el módulo debe estar conectado a los dos módulos anteriores, recibiendo la información obtenida tanto del **Reconocedor de Discurso** como del **Módulo de Lectura** y enviando las consiguientes respuestas al **Sintetizador de Voz** y al **Módulo de Escritura**. Este módulo está basado en el Programa AB [ProgramAB web].

- **Módulo Motor:** la función de este módulo es gestionar las animaciones que incorpora cada uno de los **personajes 3D** utilizados en el sistema. Este módulo recibe información proveniente del **Gestor de Diálogo**, información que se utiliza para establecer el **estado emocional** en el que se encuentra el agente virtual y transmitírselo al usuario a través de las distintas animaciones reproducidas sobre el personaje. Para la implementación de este módulo se utiliza un ***Plugin* de Unity 3D**.

- **Interfaz Gráfica:** la interfaz gráfica permite al usuario interactuar con el sistema a través de la **pantalla táctil** del dispositivo. Esta interfaz gráfica es la encargada de iniciar tanto el proceso de escucha del discurso del usuario como la lectura de los mensajes de texto del mismo, gestionando a su vez el nivel de volumen y los distintos modos de interacción del sistema.

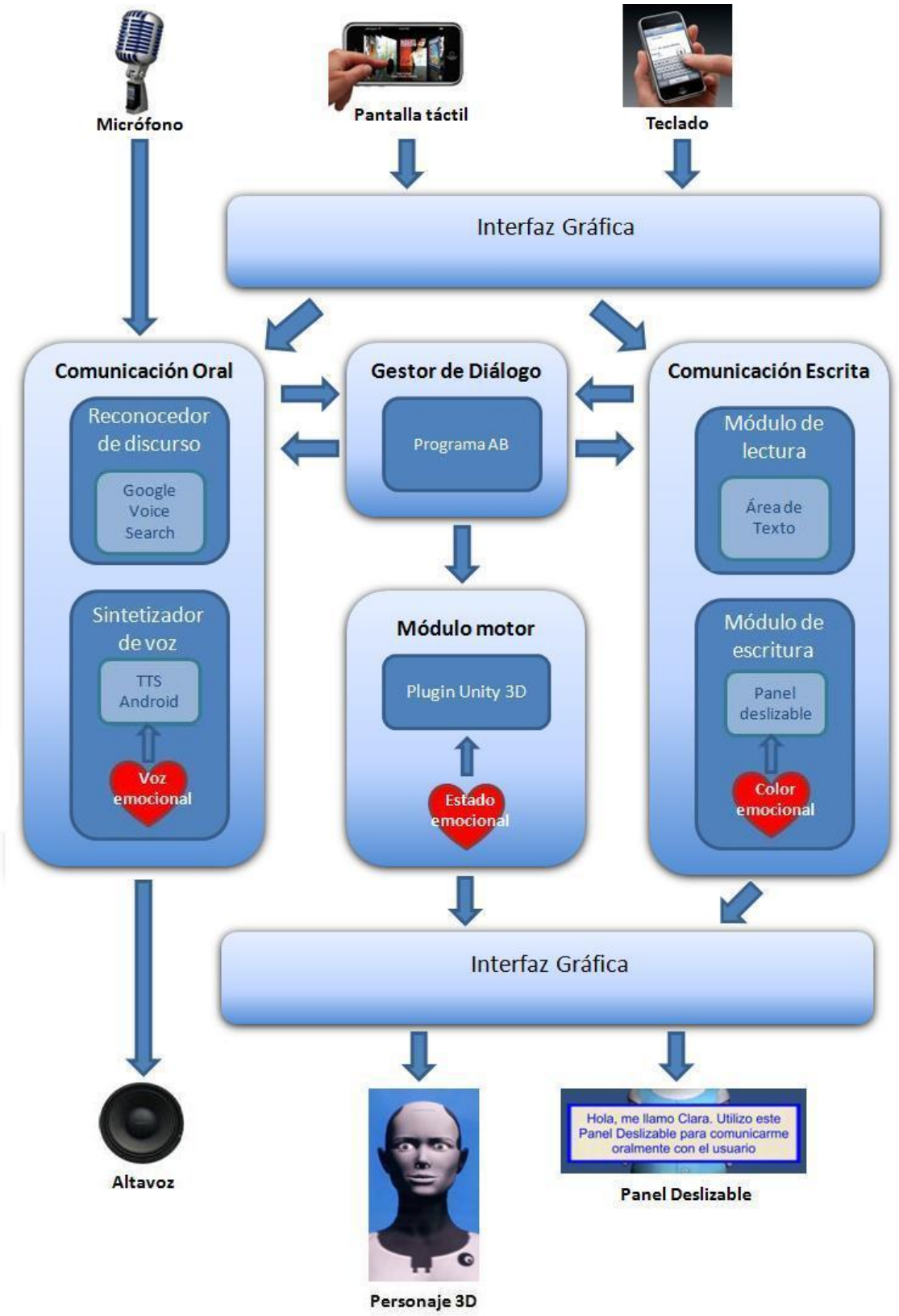


Figura 3.1.1: *Gráfico de la propuesta de diseño del nuevo sistema a desarrollar*

3.2 Comunicación oral

Con el objetivo de que la interacción con el usuario sea lo más natural posible, el sistema debe permitir que el usuario y el agente virtual sean capaces de comunicarse entre sí a través del canal oral.

La comunicación oral entre el usuario y el agente virtual es posible dividirla en dos procesos bien diferenciados, como son el proceso de escucha del discurso del usuario y el proceso de reproducción del discurso del agente virtual. En las siguientes líneas se explica en detalle ambos procesos así como el método seleccionado para expresar el estado emocional del agente virtual a través de sus mensajes orales.

3.2.1 Reconocedor de discurso

En primer lugar, es necesario desarrollar un módulo que permita al sistema escuchar al usuario. Este módulo va a ser el encargado de captar el discurso del usuario, llevar a cabo un reconocimiento que convierta el discurso en una cadena de texto y trasladar la información obtenida al sistema de gestión de diálogo para comenzar la búsqueda de respuestas adecuadas.

Decisiones de diseño

Tras investigar las distintas posibilidades para captar y reconocer el discurso del usuario en los dispositivos móviles Android, se observa que existen dos opciones que destacan sobre el resto:

- La primera de ellas se basa en utilizar una versión reducida del sistema de reconocimiento de voz Sphinx, PocketSphinx [PocketSphinx web], específica para sistemas embebidos y dispositivos móviles. Esta versión permite llevar a cabo el reconocimiento del discurso sobre el propio dispositivo donde se ejecuta el sistema, sin necesidad de conexión a internet. Sin embargo, esta opción presenta dos inconvenientes principales. Por un lado, al tratarse de una versión reducida, la precisión del reconocimiento de voz es inferior al de otras versiones de Sphinx. Por otra parte, realizar el reconocimiento del discurso sobre el propio dispositivo móvil aumenta de forma considerable las necesidades del procesado del sistema, lo que puede ralentizar la ejecución del mismo.

- En contraposición a esta primera opción, existe la posibilidad de que el sistema realice peticiones de reconocimiento de voz a una aplicación especializada instalada en el dispositivo móvil. Este tipo de aplicaciones vienen integradas en una arquitectura cliente-servidor, realizando una petición de reconocimiento de la captura de sonido a un servidor externo, el cual lleva a cabo el reconocimiento y le devuelve los resultados a la aplicación. El principal inconveniente de esta opción es la necesidad de disponer de una conexión a internet para realizar el reconocimiento. No obstante, los resultados obtenidos poseen una mayor precisión que en el caso de PocketSphinx utilizando una menor cantidad de recursos computacionales del dispositivo móvil.

Una vez analizadas en profundidad ambas opciones, se opta por que el sistema realice peticiones de reconocimiento de voz a una aplicación especializada, ya que se considera fundamental reducir los requerimientos de procesado del sistema. Esta opción, además de precisar de una menor cantidad de recursos computacionales, ofrece una mayor calidad en el reconocimiento del discurso, lo que permite dotar de un mayor realismo a la interacción oral con el usuario, a pesar de que sea necesaria una conexión a internet.

Para que un dispositivo móvil sea capaz de responder a una petición de reconocimiento de voz, dicho dispositivo debe disponer de al menos una aplicación que pueda procesar dicha petición, lo que no supone un gran problema debido a que la inmensa mayoría de los dispositivos móviles Android actuales poseen funciones de reconocimiento de voz activadas por defecto. De cualquier modo, con el fin de homogeneizar la forma de trabajar con el sistema a desarrollar, se considera necesario seleccionar una única aplicación de reconocimiento de voz con la que hacer uso del sistema. Para ello, se estudian las diversas aplicaciones de reconocimiento de voz disponibles para Android:

- Voice Action Plus [Voice Action Plus web], aplicación que permite la redacción de mensajes y correos electrónicos, la configuración de alarmas o la obtención de información de internet. Se descarta puesto que únicamente es compatible con el inglés.

- Cyberon Voice Commander [Cyberon Voice Commander web] y VoicePOD [VoicePOD web], aplicaciones especializadas en la escritura de mensajes de texto y búsquedas en internet. Se descartan ya que se tratan de aplicaciones de pago que no incorporan ninguna funcionalidad extra de utilidad para el sistema a desarrollar con respecto a otras aplicaciones gratuitas.

- Konele [Konele web], aplicación de reconocimiento de voz gratuita completamente compatible con el inglés y el estonio. Esta opción es descartada debido a que solamente dispone de una versión de prueba en castellano.

De este modo, la decisión se focaliza en dos de las aplicaciones de reconocimiento de voz más utilizadas por los dispositivos de hoy en día, a saber: Vlingo y Google Voice Search. Ambas son aplicaciones de distribución gratuita y están especializadas en la redacción de todo tipo de mensajes, búsquedas en internet y ejecución de acciones sobre el dispositivo móvil a través de comandos de voz. En cualquier caso, a pesar de que ambas aplicaciones ofrecen servicios similares y están muy bien valoradas por la comunidad de usuarios de Android, existen ligeras diferencias entre ellas:

- Google Voice Search [Google Voice Search web] presenta una mayor rapidez a la hora de reconocer el discurso del usuario, disponiendo además de un mayor rango de servicios a invocar a través de comandos de voz.

- Vlingo [Vlingo web] reconoce un mayor número de dialectos y acentos dentro de un mismo idioma.

Finalmente, para la realización de este Proyecto Fin de Carrera se ha optado por utilizar la aplicación Google Voice Search, aplicación que puede ser instalada sobre cualquier dispositivo móvil de forma gratuita y que viene ya integrada en los dispositivos móviles Android más modernos. Google Voice Search es uno de los mejores reconocedores del habla disponibles, siendo compatible con varios idiomas. Además, esta aplicación posee una actividad muy sencilla, ya que informa a los usuarios del momento en el que pueden comenzar su discurso y, una vez el usuario termina de hablar, cierra el diálogo y envía al sistema invocante (*Intent caller*) una gama de cadenas con el reconocimiento del discurso. No obstante, como ya se ha comentado anteriormente, este servicio precisa una conexión a Internet debido a que el reconocimiento de voz se lleva a cabo en los servidores de Google.

En cuanto a la información correspondiente a la implementación de este módulo reconocedor del discurso del usuario, viene mostrada en la sección D.1 del Anexo D.

3.2.2 Sintetizador de voz

Por otro lado, es necesario desarrollar un segundo módulo que permita al sistema reproducir de forma oral el discurso perteneciente al agente virtual. Este módulo es el encargado de recibir las cadenas de texto procedentes del gestor de diálogo que contienen las respuestas para el usuario y reproducir las mismas a través del altavoz del dispositivo móvil en el que se ejecute el sistema.

Decisiones de diseño

Para dotar de voz al agente virtual, existen dos posibles alternativas, a saber: desarrollar un motor de síntesis de voz propio que sea compatible con Android o hacer uso del motor de síntesis de discurso (también llamado "*Text To Speech*" o "*TTS*") que incorpora la plataforma Android desde su versión 1.6. A continuación se presentan las ventajas e inconvenientes de ambas opciones:

- Desarrollar un motor de síntesis de voz propio: el desarrollador debe implementar un nuevo motor de síntesis de discurso utilizando para ello alguna de las bibliotecas especializadas para el desarrollo de sistemas TTS compatibles con Android. Esta opción permite desarrollar motores de síntesis de voz más específicos para los sistemas en los que van a ser integrados, pero supone un

aumento considerable tanto en la capacidad de procesado como en el espacio de memoria requeridos por dichos sistemas.

- Hacer uso del motor de síntesis de discruso de Android: el sistema TTS que incorpora Android es más genérico, permitiendo tan sólo al desarrollador modificar unos pocos parámetros del discurso generado. Sin embargo, se encuentra bastante optimizado y requiere una menor cantidad de recursos computacionales durante su ejecución.

Igual que sucediera en el proceso de reconocimiento de voz (explicado en el apartado anterior), se elige la opción que minimiza los requerimientos de procesado del sistema, en este caso, hacer uso del motor de síntesis de voz que incorpora Android desde su versión 1.6. El sistema TTS de Android realiza la síntesis de discurso a partir de cadenas de texto, soportando múltiples idiomas y permitiendo modificar alguno de los parámetros del discurso como el volumen, el tono y la velocidad. Además, hace uso de la aplicación de síntesis de voz seleccionada por defecto en cada dispositivo móvil, de forma que el desarrollador puede seleccionar aquella que se adecúe más a su sistema. Por otro lado, el principal inconveniente que presenta esta opción es que no aporta ninguna clase de información temporal, esto es, el desarrollador no puede saber con exactitud el momento en el que se va a reproducir cada una de las palabras del discurso. En este mismo sentido, el sistema TTS de Android tampoco ofrece información para la reproducción de los visemas (representación visual de los fonemas), por lo que la sincronización labial (*lipsync*) se complica de forma considerable.

Si se desea consultar la implementación llevada a cabo para este módulo sintetizador de voz, véase la sección D.2 del Anexo D.

3.2.3 Expresión de emociones a través de la voz

Con el objetivo de dotar de mayor realismo y naturalidad al proceso de interacción oral con el agente virtual, se considera necesario otorgar de cierta emoción a la voz sintetizada durante el discurso del agente. De esta forma, se pretende que el usuario sea capaz de reconocer, en cierta medida, el estado emocional en el que se encuentra el agente virtual a partir del discurso reproducido por éste.

Decisiones de diseño

En primer lugar, se debe determinar los distintos estados emocionales en los que se puede encontrar el agente virtual. Debido a que el sistema a desarrollar está pensado para ser ejecutado sobre dispositivos móviles, tanto los personajes tridimensionales utilizados para representar gráficamente al agente virtual como el sistema TTS usado no poseen la misma potencia expresiva que aquellos pertenecientes a sistemas destinados a ser ejecutados sobre ordenadores. Por esta razón, se opta por dotar únicamente de cinco estados emocionales al agente virtual, a saber: alegre, sorprendido, neutro, triste y enfadado.

Una vez seleccionados los estados emocionales en los que se puede encontrar el agente virtual, se pretende que el usuario sea capaz de reconocer el estado emocional del agente a través de su discurso. Para ello, se procede a modificar los distintos parámetros que posee dicho discurso para cada uno de los estados emocionales del agente.

Como se ha argumentado en el apartado anterior de este capítulo, para la reproducción del discurso del agente se opta por utilizar el sistema TTS de Android, sistema que permite modificar únicamente tres parámetros a la hora de sintetizar dicho discurso: volumen, velocidad y tono.

- Volumen: determina la intensidad con la que se reproduce el discurso. Experimentalmente se comprueba que para valores de volumen inferiores a 4.0 el discurso se vuelve inapreciable para el usuario, mientras que los valores por encima de 16.0 provocan una distorsión a la hora de reproducir dicho discurso.

- Velocidad: indica la rapidez con la que el sistema debe articular las palabras a lo largo del discurso del agente virtual. El valor por defecto que se otorga a este parámetro es 1.0, valor con el que se reproduce el discurso a una velocidad estándar. Otros valores representativos de este

parámetro son 0.5 y 2.0, los cuales reproducen el discurso en el doble y la mitad de tiempo respectivamente.

- Tono: establece el grado de elevación de la voz sintetizada para reproducir el discurso. El valor por defecto que se otorga a este parámetro es 1.0, valor que implica un tono de voz estándar. Para valores superiores a 1.0 el tono del discurso pasa a ser más agudo, mientras que para valores inferiores el tono del discurso se vuelve más grave.

A través de la manipulación de estos tres parámetros del discurso se pretende sintetizar un tipo de voz distinta para cada uno de los estados emocionales del agente virtual. En este sentido, el objetivo principal es establecer una terna de valores, correspondientes a los parámetros del discurso mencionados con anterioridad, que se asocie a cada uno de los estados emocionales del agente. La elección de estos valores no es inmediata, siendo necesario llevar a cabo varias fases de pruebas con usuarios para su elección (capítulo 4), pero sí es posible delimitar, en cierto modo, el rango de valores que puede adoptar cada parámetro en función del estado emocional que se desee expresar a través de la voz sintetizada. A continuación se explican los distintos rangos seleccionados para cada estado emocional del agente.

Rango de valores en función de las emociones

Con el objetivo de conocer los rangos de valores en los que se mueven los parámetros del discurso para cada uno de los estados emocionales seleccionados, el desarrollador se sirve de varios trabajos anteriores relacionados con la expresión de emociones a través de la voz [Francisco et al, 2005] [Anaya, 2006]. Cabe destacar que dichos trabajos se utilizan a modo de guía orientativa, ya que las escalas usadas para los parámetros del discurso en estos trabajos difieren en gran medida de las escalas disponibles para dichos parámetros en el sistema TTS de Android. En las siguientes líneas se presentan, agrupados por estados emocionales, los rangos de valores seleccionados a partir de la información extraída de las diversas fuentes consultadas y su consiguiente extrapolación al sistema TTS de Android.

- Alegre: el volumen del discurso debe ser alto, la velocidad con la que se articula las palabras debe ser superior a la estándar y el tono debe ser elevado. Por tanto, se ha de considerar el rango [11.0, 16.0] para el volumen, el rango [1.2, 1.8] para la velocidad y el rango [1.8, 2.4] para el tono del discurso.

- Sorprendido: el volumen con el que se reproduce el discurso debe ser medio-alto, con una velocidad similar o ligeramente superior a la estándar y con un tono de voz alto. De esta forma, se han de considerar los rangos [10.0, 14.0], [1.0, 1.5] y [1.8, 2.4] para el volumen, velocidad y tono del discurso respectivamente.

- Neutro: tanto el volumen como la velocidad y el tono del discurso deben estar en torno a sus respectivos valores estándar. En este sentido, los rangos a tener en cuenta son [9.0, 12.0] para el volumen, [0.8, 1.2] para la velocidad y [0.8, 1.2] para el tono del discurso a reproducir.

- Triste: el volumen del discurso debe ser similar o inferior al estándar, con una velocidad de articulación de palabra baja y un tono de voz similar o ligeramente más grave que el estándar. Según esta información, los rangos de interés son [7.0, 11.0] para el volumen, [0.6, 1.0] para la velocidad y [0.7, 1.1] para el tono.

- Enfadado: el volumen con el que se reproduce el discurso debe ser elevado, con una velocidad de articulación de palabra alta y un tono de voz similar o ligeramente inferior al estándar. Por consiguiente, se estudia el rango [11.0, 16.0] para el volumen, el rango [1.2, 1.8] para la velocidad y el rango [0.7, 1.1] para el tono del discurso.

Una vez delimitados, en cierta medida, los rangos de valores de cada uno de los parámetros del discurso en función del estado emocional del agente que se desea expresar, se procede a realizar las distintas pruebas con usuarios. Estas pruebas, explicadas en detalle en el capítulo 4, tienen como fin determinar la terna de valores que mejor se asocie a cada uno de los estados emocionales del agente.

Por su parte, la información correspondiente a la manipulación mediante código de los tres parámetros del discurso que permite modificar el sistema TTS de Android se muestra en la sección D.3 del Anexo D.

3.3 Comunicación escrita

Debido a que el sistema a desarrollar en este Proyecto Fin de Carrera está pensado para su utilización sobre dispositivos móviles, es necesaria la existencia de un canal de interacción que no se vea afectado por condiciones externas al sistema como el ruido ambiental o la conectividad del dispositivo. Por este motivo, y con el fin de mejorar la accesibilidad del sistema, se otorga al usuario la posibilidad de interactuar con el agente virtual a través del canal escrito.

El proceso de comunicación escrita entre el usuario y el agente virtual se realiza, principalmente, a través de dos elementos existentes en la interfaz gráfica del sistema: un área de texto y un panel deslizable. Estos elementos, situados como se muestra en la Figura 3.3.1, únicamente serán visibles en dicha interfaz en el caso de que el canal textual esté activado como canal de entrada o de salida del sistema respectivamente. A continuación, se procede a explicar en detalle cada uno de estos elementos de la interfaz gráfica así como el método seleccionado para expresar el estado emocional del agente virtual a través de sus mensajes escritos.

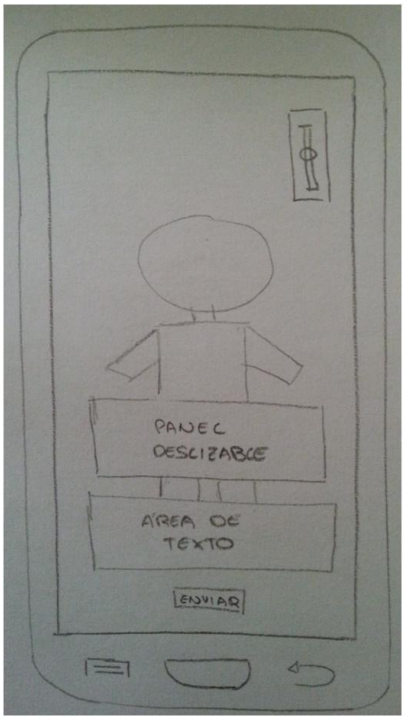


Figura 3.3.1: *Fotografía del boceto a papel correspondiente al diseño de la interfaz gráfica durante un proceso de comunicación escrita.*

3.3.1 Área de texto

El área de texto es la zona de la interfaz gráfica utilizada por el usuario para redactar el mensaje que desea transmitir al agente virtual.

Decisiones de diseño

Las dimensiones de esta área de texto vienen definidas en base a la anchura y altura de la pantalla del dispositivo en el que se ejecuta el sistema, evitando de esta forma los problemas característicos de utilizar elementos de un tamaño fijo en una interfaz gráfica pensada para ser utilizada sobre dispositivos móviles con pantallas de muy diversa índole. Por un lado, se opta por que el área de texto sea considerablemente ancha, ocupando un 90% del ancho de la pantalla, permitiendo de esta forma un mayor número de palabras por línea de texto, lo que facilita la revisión del mensaje escrito por parte del usuario. Por otro lado, no se considera necesario que el área muestre un gran número de líneas de texto ya que en una conversación los mensajes no suelen ser excesivamente largos, por lo que no se

precisa dotar al área de texto de una gran altura, ocupando tan sólo el 15% de la altura de la pantalla del dispositivo en el que se ejecute el sistema. De todas formas, con este 15% de altura se asegura mostrar un mínimo de dos líneas en el área de texto, ya que el tamaño de la letra queda establecido en función de la resolución del dispositivo en el que se está ejecutando el sistema.

Además, es necesario dotar al sistema de un mecanismo que permita al usuario enviar el mensaje una vez esté redactado. En este sentido, se opta por que el área de texto venga acompañada de un botón 'Enviar' situado justo debajo de la misma. Este botón es el encargado de transmitir el mensaje escrito al agente virtual, previa revisión por parte del usuario.

Para obtener la información correspondiente a la implementación de esta área de texto, consúltese la sección D.4 del Anexo D.

3.3.2 Panel deslizable

Por su parte, el panel deslizable es el elemento a través del que se transmite, por escrito, la información procedente del agente virtual al usuario.

Decisiones de diseño

La elección de hacer uso de un panel deslizable en lugar de utilizar cualquier otro elemento estático que pudiera mostrar mensajes de texto en su interior tiene que ver con el tamaño destinado para este elemento. Siguiendo un razonamiento análogo al realizado en el caso del área de texto anterior, se considera acertado que el panel posea las mismas dimensiones que dicha área de texto. Además, de esta forma se dota de cierta simetría al modo de interacción escrita del sistema, disponiendo de espacios idénticos tanto para la entrada como para la salida de texto. Sin embargo, existe la posibilidad de que el mensaje proveniente del agente virtual sea lo suficientemente extenso como para que sólo se visualice parcialmente dentro del espacio destinado para ello, lo que hace necesario que el elemento utilizado para mostrar el mensaje provea al usuario de un mecanismo que le permita desplazarse a través del interior del mismo con el fin de permitir la lectura completa del mensaje. El elemento que reúne estás características es el panel deslizable.

Por otro lado, es necesario que la información proveniente del agente virtual se muestre de forma sencilla y clara dentro del panel deslizable, procurando que la lectura del mensaje sea lo más amena posible para el usuario. Para ello, se opta por que el mensaje venga mostrado de forma centrada, comenzando en la parte central superior del panel deslizable y rellenando de un modo descendente el mismo. Además, se considera acertado incluir márgenes en el interior del panel deslizable, evitando así que exista cualquier tipo de problema durante la lectura del mensaje causado por la mala visualización de alguna de las letras del texto.

Finalmente, con el fin de adaptar el tamaño de letra utilizado para mostrar el mensaje a los distintos tipos de pantallas que poseen los dispositivos móviles de hoy en día, se opta por que dicho tamaño de letra sea seleccionado por el sistema en función de la resolución de la pantalla del dispositivo en el que se está ejecutando.

Con respecto a la información correspondiente a la fase de implementación de este panel deslizable, viene recogida en la sección D.5 del Anexo D.

3.3.3 Expresión de emociones a través del texto

Con el objetivo de que el uso de la comunicación escrita del sistema no implique una pérdida de información con respecto a la utilización de la comunicación oral del mismo, es necesario que el sistema permita al agente virtual expresar en todo momento su estado emocional a través de los mensajes de texto que se muestran al usuario.

Decisiones de diseño

Como se ha explicado en el apartado anterior, el elemento encargado de mostrar los mensajes provenientes del agente virtual al usuario es un panel deslizable. Es por ello que se opta por utilizar el color tanto de la fuente como de los bordes de dicho panel deslizable para dar a conocer al usuario el estado emocional del agente, de forma que un cambio en el color de ambos elementos implique una variación en el ánimo del agente virtual. Cabe destacar que también se sopesa la utilización de emoticonos e imágenes dentro del panel deslizable con el fin de que el usuario reconozca más fácilmente el estado emocional en el que se encuentra el agente virtual, pero se acaba descartando porque se considera suficiente refuerzo emocional para la comunicación escrita del sistema la reproducción de las animaciones que incorpora el agente virtual.

Con el objetivo de dar a conocer al usuario el estado emocional del agente virtual a través del color de la fuente y los bordes del panel deslizable, es necesario seleccionar una serie de colores que representen unívocamente cada uno de los estados emocionales del agente virtual y generar un conjunto de texturas para el panel deslizable acordes a los colores elegidos. A continuación se detallan estas dos fases del proceso de expresión de las emociones del agente a través de los mensajes de texto.

Selección de los colores

En primer lugar, se precisa asociar a cada uno de los estados emocionales del agente virtual un color determinado. Para ello, se estudian las connotaciones emocionales existentes en los colores, connotaciones obtenidas de distintos trabajos [Boyatzis et al, 1994] [Kaya et al, 2004] [Hemphill, 1996] y sitios web [COLORES EMOCIONES web] [SINESTESIA web] encontrados. En base a la información procedente de estas fuentes, se selecciona un color distinto para cada estado emocional. En la Tabla 3.3.1 se muestra un listado de los estados emocionales de los que dispone el agente virtual junto a los colores elegidos para representarlos.

Neutro → Azul	**Triste → Gris**
Contento → Verde	**Sorprendido → Amarillo**
Enfadado → Rojo	

Tabla 3.3.1: *Listado de los estados emocionales del agente virtual y los colores usados para representarlos*

En el caso del estado neutro, se opta por representarlo con el color azul oscuro. El azul oscuro, al pertenecer a la gama de colores fríos, evoca una sensación de frialdad y tranquilidad que se corresponde en gran medida con el estado emocional neutro del agente virtual. Además, los estudios citados anteriormente reflejan que la mayoría de las personas encuestadas consideran que el azul inspira una sensación de calma, sensación que se asemeja al estado emocional neutro del agente.

A su vez, el estado emocional de alegría del agente virtual viene representado por el color verde. Este color, de carácter más cálido que el azul, está asociado a la vida y a la naturaleza, evocando emociones muy positivas como la felicidad o el entusiasmo. En este sentido, en todos los estudios utilizados en este proceso de selección, el verde es uno de los colores con connotaciones más positivas para los encuestados, independientemente del sexo o edad de los mismos, asociándolo a sensaciones de felicidad, euforia o excitación. Es por ello que pasa por ser el color ideal para representar la emoción de alegría.

Por otro lado, el estado emocional de enfado se encuentra representado por el color rojo. El rojo es el color más cálido que existe y suele venir asociado a emociones fuertes, como la pasión, el enfado, la rabia o la ira. Además, los trabajos utilizados como fuente de información muestran como la mayoría de los encuestados asocian el rojo a sentimientos de excitación o enojo, lo que convierte a este color en el idóneo para representar el enfado del agente virtual.

Por su parte, el gris es el color seleccionado para representar el estado emocional de tristeza. Este color, oscuro y frío, evoca sensaciones negativas como la soledad, el miedo, la decepción o la tristeza. En este mismo sentido se manifiestan los encuestados de los distintos trabajos mencionados anteriormente,

considerando al gris como el color con connotaciones más negativas de toda la gama de colores y asociándolo a sentimientos de aburrimiento, depresión, y tristeza.

Finalmente, otro de los colores utilizados para representar emociones del agente virtual es el amarillo, el cual se utiliza para dar a conocer al usuario el estado de sorpresa en el que se encuentra el agente. El amarillo es un color cálido que evoca sensaciones de sorpresa, incredulidad o preocupación.

Generación de texturas

Tras haber determinado los colores encargados de expresar el estado emocional del agente virtual, se precisa crear las imágenes a partir de las que generar las texturas que van a servir de fondo del panel deslizable. Para ello, se utiliza como base la imagen de un cuadrado opaco de color hueso, color elegido por su contraste con los anteriores, y se colorea su contorno con cada uno de los colores seleccionados para representar un estado emocional, generando de esta forma tantas imágenes como estados emocionales presenta el agente virtual. A continuación, se muestra las distintas imágenes generadas para servir de fondo del panel deslizable.

Figura 3.3.2: *Imágenes utilizadas como texturas en el panel deslizable*

Como se observa en la Figura 3.3.2, las imágenes creadas difieren únicamente en el color de los bordes que delimitan el cuadrado que las conforma. De esta manera, en lo que respecta al fondo del panel deslizable, una variación en el estado emocional del agente virtual supone exclusivamente un cambio en el color de los bordes de dicho panel.

Si se desea consultar la gestión de los cambios tanto de la fuente como de los bordes de dicho panel deslizable a través de código, véase la sección D.6 del Anexo D.

3.4 Gestor de diálogo

Una vez se ha dotado al sistema de mecanismos para comunicarse de forma oral y escrita con el usuario, es necesario desarrollar un módulo gestor de diálogo que permita al agente virtual mantener una conversación fluida y coherente con dicho usuario. En las siguientes líneas se explica en detalle el proceso de desarrollo de este módulo gestor de diálogo así como la gestión realizada por el mismo en el sistema.

3.4.1 Desarrollo del gestor de diálogo

El módulo gestor de diálogo del sistema debe gestionar la información contenida en el discurso y los mensajes escritos del usuario, procesando dicha información y generando una respuesta acorde a la misma. De este modo se pretende dotar al agente virtual de la capacidad de conversar con el usuario de forma coherente.

Decisiones de diseño

Con el fin de desarrollar un módulo gestor de diálogo para el sistema, se estudian las distintas alternativas existentes para interpretar la información procedente del usuario y generar una respuesta

acorde a la misma. A continuación se presentan las diversas opciones contempladas, describiendo cada una de ellas y analizando sus ventajas e inconvenientes:

- Episteme Engine: se trata de una biblioteca de pago para la plataforma Unity 3D [Episteme Engine web], biblioteca que permite la incorporación de diálogos a videojuegos y la generación de respuestas de los *chatbots* desarrollados sobre dicha plataforma. La principal ventaja que posee este paquete de actualización de Unity 3D es que utiliza el lenguaje AIML (Artificial Intelligence Markup Language) [Bush, 2001], siendo capaz el desarrollador de modificar los patrones de reconocimiento a su voluntad. Además, incorpora una serie de métodos públicos y ejemplos de uso que facilitan mucho su integración en el sistema. Sin embargo, esta opción tiene como inconvenientes que es una actualización de pago y que ciertas secciones de código no son abiertas.

- Diccionario: esta opción consiste en implementar un diccionario que se encargue de generar las respuestas del agente virtual. El funcionamiento de este diccionario es bastante sencillo, ya que se basa en asociar a cada frase del usuario considerada una respuesta, de forma que si la entrada procedente del sistema coincide con alguna de las frases existentes en el diccionario, éste emite la respuesta asociada. Este sistema de gestión de diálogo tiene como principal ventaja su rápida y sencilla implementación, puesto que el lenguaje C# utilizado en los *scripts* del sistema incorpora una clase Dictionary [Dictionary web] con esta funcionalidad. No obstante, la gestión de diálogo que es posible llevar a cabo con un diccionario es muy básica e ineficiente, ya que se precisan una gran cantidad de entradas en el diccionario para poder mantener una conversación simple con el usuario y la búsqueda de coincidencias se realiza, para cada frase transmitida por el usuario, sobre todas las entradas existentes en dicho diccionario.

- Listas, tablas hash, árboles: otra posibilidad que se contempla es llevar a cabo la gestión de diálogo a través estructuras de datos ya existentes en Unity 3D como listas, tablas hash o árboles de decisión. A partir de la información proveniente del usuario, se realizaría una búsqueda dentro de la estructura de datos seleccionada para dar con la respuesta más adecuada, utilizando para ello determinadas palabras clave que guiarían la búsqueda. El uso de estas estructuras de datos permite llevar a cabo la búsqueda de respuestas de forma eficiente, lo que se erige en su principal ventaja frente al uso del diccionario. Sin embargo, el grado de complejidad de la implementación e inicialización de cualquiera de estas estructuras de datos es mayor que en el caso del diccionario y la gestión de diálogo que se puede llevar a cabo con dichas estructuras de datos sigue siendo bastante básica.

- Expresiones regulares: esta alternativa se basa en utilizar expresiones regulares para llevar a cabo el reconocimiento del mensaje transmitido por el usuario y seleccionar la respuesta más adecuada. En este sentido, cada una de las expresiones regulares iría asociada a un conjunto de respuestas similares, de forma que si una determinada expresión regular reconociera el mensaje del usuario, una de sus frases asociadas sería utilizada como respuesta. Con el fin de hacer uso de expresiones regulares para el reconocimiento del mensaje, el desarrollador se serviría de la clase pública Regex [Regex web] que incorpora el lenguaje C#. Sus principales ventajas son la precisión que es posible alcanzar en el proceso de reconocimiento y su sencilla implementación. Por otro lado, presenta la tremenda desventaja de tener que definir una gran cantidad de expresiones regulares para poder llevar a cabo un reconocimiento básico de los mensajes del usuario, creciendo exponencialmente conforme se va complicando el tipo de reconocimiento a realizar.

- Programa AB: la fundación A.L.I.C.E [A.L.I.C.E web], dedicada a la Inteligencia Artificial, dispone de una serie programas de libre distribución que incorporan intérpretes AIML para el desarrollo de *chatbots* sobre distintas plataformas. Debido a las características del sistema que se está desarrollando en este Proyecto Fin de Carrera, el programa que más se adecúa al mismo es el Progama AB [ProgramAB web], el cual incorpora un intérprete AIML que puede ser incluido como biblioteca en aplicaciones JAVA. La principal ventaja de utilizar el intérprete AIML que incorpora el Programa AB es que el desarrollador es capaz de modificar los patrones de reconocimiento a su voluntad, pudiendo llevar a cabo un gestor de diálogo todo lo complejo que se desee sin una gran carga de programación. Además, incorpora una serie de métodos públicos y ejemplos de uso que facilitan mucho su integración en el sistema. No obstante, este programa precisa de la generación de una serie de directorios y ficheros AIML en el dispositivo sobre el

que se va a ejecutar el sistema, por lo que precisa de una mayor capacidad de memoria que otras opciones contempladas. Estos ficheros sirven para almacenar toda la información correspondiente al *chatbot*.

Una vez analizadas en profundidad las distintas alternativas para desarrollar un módulo gestor de diálogo en el sistema, se opta por hacer uso del Programa AB desarrollado por la fundación A.L.I.C.E., ya que es de libre distribución y permite una gestión de diálogo más compleja y realista que en el caso de utilizar expresiones regulares o estructuras de datos como tablas hash y árboles de decisión. Además, como se ha comentado anteriormente, la gestión de diálogo se lleva a cabo mediante ficheros AIML, ficheros que contienen la información correspondiente al *chatbot* de forma estructurada, lo que permite al desarrollador comprender y modificar dicha información a voluntad sin gran dificultad. De este modo, a través de los ficheros AIML, el desarrollar posee la capacidad de focalizar las conversaciones que debe mantener el agente virtual, siendo posible especializar al agente en una determinada tarea o tema. Sin embargo, no todo son ventajas, ya que el uso del Programa AB implica la generación de todos los directorios y ficheros necesarios para la ejecución del mismo en el interior de los dispositivos sobre los que se vaya a ejecutar el sistema a desarrollar.

Para obtener información acerca de la integración del Programa AB en el sistema y del desarrollo de las funcionalidades del módulo gestor de diálogo, consúltese la sección D.7 del Anexo D.

3.4.2 Programación del *chatbot*: generación de respuestas

Las preguntas que es capaz de contestar el agente virtual, así como las respuestas que devuelve para cada una de dichas preguntas han de ser especificadas de alguna forma. El formato que utiliza el Programa AB, y por consiguiente el módulo Gestor de Diálogo del sistema, para especificar el conocimiento del agente virtual es el AIML (Artificial Intelligence Markup Language) [Bush, 2001]. La potencia del AIML se puede simplificar en dos aspectos:

- La sintaxis del AIML permite extraer fácilmente el contenido semántico de una pregunta, permitiendo devolver una respuesta adecuada de forma rápida.

- La utilización de etiquetas para combinar las respuestas, aumentando la variedad de éstas y el número de preguntas a las que se puede dar una contestación adecuada.

En este apartado se da una visión genérica y sin mucho detalle de la sintaxis AIML presente en los ficheros utilizados para la programación del *chatbot* "clara".

El formato AIML

AIML es un lenguaje de programación derivado de XML (Extensible Markup Language) [XML web], el cual fue diseñado específicamente para ayudar en la creación de la primera entidad *chatbot* informática de lenguaje artificial online (A.L.I.C.E.). Descrito muy ampliamente, el lenguaje AIML está especializado en la creación de agentes software con lenguaje natural.

Un objeto AIML está compuesto de cero o más tópicos y una o más categorías, pudiendo ser ordenados a gusto del usuario. Para comprender qué son estas dos entidades es necesario definir previamente qué es un patrón.

- Un patrón AIML simple está formado por palabras normales (compuestas por letras siempre en mayúsculas y/o dígitos) y *wildcards* (símbolos: '_' o '*'). Estos símbolos representan una o más palabras normales. La diferencia entre ambos es que el '_' tiene mayor prioridad que el '*' a la hora de buscar respuestas. Un patrón compuesto puede llevar además otros elementos.

Una vez explicado el concepto de patrón, se presentan las definiciones tanto de tópico como de categoría AIML:

- Un tópico es un elemento AIML de primer nivel compuesto de una o más categorías. Un tópico debe llevar siempre un atributo nombre que debe contener un patrón simple. Un tópico hace referencia al tema de la conversación sobre el que se está hablando en el momento actual.

- Una categoría es un elemento AIML de primer nivel (o segundo si está incluida en un tópico) que contiene un patrón y una plantilla o respuesta. Para aquellas categorías que no están incluidas específicamente dentro de un tópico, el intérprete AIML asumirá que están incluidas dentro de un tópico implícito cuyo atributo nombre tiene el valor '_'. Las categorías pueden llevar adicionalmente otro elemento llamado patrón *that* (o simplemente *that*) que contiene un patrón simple. Este patrón hace referencia a la última frase pronunciada por el agente virtual. Si la categoría no contiene un patrón *that*, el intérprete AIML asume un *that* implícito conteniendo la expresión '_'.

Con estos elementos fundamentales se generan los distintos ficheros AIML de programación del *chatbot* "clara", ficheros encargados de reconocer las preguntas y contestaciones provenientes del usuario así como de la definición y selección de las respuestas del agente virtual a las mismas.

Ejemplo de uso

Ahora que se conocen los principales elementos del lenguaje AIML, se procede a explicar su utilidad mediante el ejemplo de código que se muestra en el Cuadro 3.4.1:

```
< topicname =SALUDO>
    < category > <! − −categoría 1−− >
        < pattern >Bien< /pattern >
        < that >QUE TAL ESTAS< /that >
        < template >Me alegro de que te estés bien.< /template >
    < /category >
< /topic >

< category > <! − −categoría 2−− >
        < pattern >_ TU NOMBRE< /pattern >
        < template >Me llamo Maxine.< /template >
< /category >
```

Cuadro 3.4.1: *Código AIML utilizado como ejemplo para mostrar el funcionamiento del chatbot "clara"*

Por un lado, la categoría 1 se activa siempre que el usuario haga preguntas como "Cuál es tu nombre" o "Dime tu nombre", pero no se activa en caso de que el usuario pronuncie únicamente "Tu nombre" porque el símbolo '*' representa una o más palabras.

Por otra parte, la categoría 2 se activa siempre que el usuario transmita un "Bien", el tópico de la conversación sea "SALUDO" y la última frase dicha por el avatar se "Qué tal estás". Si alguna de estas tres condiciones no se cumple, la categoría no se activa.

3.5 Módulo motor

Con el fin de reforzar la expresividad del agente durante la interacción con el usuario así como aumentar el realismo de la misma, se desarrolla un módulo motor encargado de gestionar las animaciones del personaje tridimensional que representa al agente virtual en el sistema. Este módulo debe conocer el estado emocional en el que se encuentra el agente, reproducir las animaciones acordes a dicho estado emocional en el momento oportuno y gestionar la sincronización labial en el caso de que sea necesario. En las siguientes líneas se explica en detalle el proceso de desarrollo de este módulo motor así como las funciones que realiza el mismo en el sistema.

Decisiones de diseño

Debido a que el sistema se desarrolla sobre la plataforma Unity 3D y a que los personajes utilizados para representar al agente virtual, diseñados sobre 3D Studio, son exportables y reproducibles en dicha plataforma, se opta por desarrollar el módulo motor del sistema en el propio proyecto Unity 3D

sobre el que se trabaja. Para ello, se hace uso de un *Script* en código C# cuya función es gestionar la información procedente de la ejecución del sistema y reproducir las animaciones del agente que sean acordes a la misma.

Puesto que los distintos personajes tridimensionales utilizados a lo largo del desarrollo del sistema poseen animaciones diferentes, se estudian dos alternativas a la hora de desarrollar el módulo motor. Ambas opciones vienen descritas a continuación, junto al análisis de las ventajas e inconvenientes que presenta cada una de ellas:

- *Plugin* C# conjunto: esta alternativa se basaría en desarrollar un único *plugin* C# que sirviese para gestionar las animaciones de todos los personajes utilizados para representar al agente virtual. La principal ventaja de esta opción es la reutilización de código dentro del sistema, permitiendo que todos los personajes sean controlados por el mismo *plugin* independientemente de sus características particulares. Además, cabría la posibilidad de permitir elegir al usuario el personaje con el que desea interactuar en tiempo de ejecución sin que se tuviera que duplicar código o realizar operaciones complejas para ello. Sin embargo, dada la heterogeneidad de las animaciones que incorporan los tres personajes de índole similar (niño, niña y perro) y el tipo de animación completamente distinta que presenta Maxine, el desarrollo de un *plugin* único para todos ellos supondría una gran pérdida en términos de expresividad y realismo para el agente virtual, ya que solamente se podría hacer uso de aquellas animaciones de carácter similar que posean todos los personajes, teniendo que ignorar las características específicas de dichas animaciones en cada uno de los personajes.

- *Plugin* C# específico: esta opción consiste en implementar un *plugin* C# distinto para cada uno de los cuatro personajes tridimensionales utilizados para representar el agente virtual. La principal ventaja de esta alternativa es que permite hacer uso del conjunto completo de animaciones que incorpora cada personaje, extrayendo de esta forma toda la capacidad expresiva de los personajes y dotando de un mayor realismo a la interacción con el usuario. No obstante, esta opción dificulta en gran medida la posibilidad de que el usuario sea capaz de elegir el personaje con el que desea interactuar durante la ejecución del sistema, a la vez que supone un aumento en el tiempo de desarrollo del sistema debido a que se deben implementar cuatro *plugins* en vez de uno.

Una vez analizadas en profundidad ambas alternativas, se opta por implementar un *plugin* C# específico para cada personaje tridimensional utilizado, ya que se considera mucho más relevante dotar de un mayor realismo a la interacción con el usuario que permitir a este último seleccionar el personaje con el que desea interactuar.

Con respecto a la información correspondiente a la implementación del módulo motor a través de este *plugin* C#, viene mostrada en la sección D.8 del Anexo D.

3.6 Interfaz gráfica

Debido a que la interfaz gráfica es un elemento fundamental en la ejecución del sistema, se opta por diseñar e implementar dos versiones de la misma a lo largo de este proyecto:

- Prototipo funcional: este primer prototipo de la interfaz gráfica (cuya fase de desarrollo se explica en detalle en el Anexo Q) debe permitir la comprobación, de forma rápida y sencilla, de las distintas funcionalidades implementadas en el sistema.

- Versión definitiva: esta segunda versión de la interfaz tiene como objetivo dotar al sistema de una interfaz gráfica sencilla que permita a personas de distinta edad y condición hacer uso de la aplicación sin dificultades.

A continuación se describe de forma detallada la fase de desarrollo de la versión definitiva de la interfaz, fase que se inicia una vez implementadas y validadas las distintas funcionalidades del sistema a través del prototipo funcional.

Requisitos de la interfaz gráfica definitiva

En primer lugar, se definen los servicios que esta versión de la interfaz gráfica debe prestar al usuario, servicios que se asemejan a los prestados por el prototipo funcional (véase sección Q1 del anexo Q) pero que incorporan ciertas modificaciones en la forma de uso.

Por un lado, el usuario debe ser capaz de seleccionar los modos de interacción con el agente virtual que desee a través de la interfaz gráfica del sistema, siendo posible cualquier combinación de los mismos, siempre y cuando esté habilitado, al menos, un modo de interacción de entrada y otro de salida.

Por otra parte, esta versión definitiva de la interfaz debe dotar al usuario de los mecanismos necesarios para iniciar la comunicación con el agente virtual a través de cualquier de los canales de interacción seleccionados.

A su vez, en caso de estar habilitada la comunicación oral para el agente virtual, se debe permitir al usuario modificar el volumen del discurso del agente.

Finalmente, se debe dotar a la interfaz de algún mecanismo que dé por concluida la interacción con el usuario y cierre el sistema.

Diseño del prototipo final

A pesar de que el sistema a desarrollar se lleva a cabo sobre la plataforma Unity 3D, se realiza un estudio en detalle la guía de estilo Android para aplicaciones [ANDROID web]. Este estudio se debe a que el sistema está dirigido a los dispositivos móviles Android, hecho por el que se considera necesario seguir los consejos y prácticas habituales, en lo que a interfaces gráficas se refiere, que aparecen en dicha guía de estilo. De este modo se pretende que la interfaz gráfica del sistema le resulte lo más sencilla e intuitiva al usuario, habituado en mayor medida a interactuar con interfaces propias de la plataforma Android.

Una de las recomendaciones presentes en la guía de estilo de Android es la existencia de un menú de opciones. Toda modificación de los valores por defecto del sistema se debe abordar a través de este menú de opciones, el cual debe aparecer en pantalla siempre que el botón físico “Menú” sea pulsado por el usuario. Además, se insta a incorporar en dicho menú una opción de ayuda que explique al usuario las distintas tareas que puede llevar a cabo sobre el sistema.

A su vez, la guía de estilo de Android insta a todas las aplicaciones que trabajan con archivos de sonido a que permitan al usuario modificar el volumen con el que se reproducen los mismos a través de los botones físicos destinados para ello. No obstante, estas aplicaciones también pueden contar con reguladores del volumen en su propia interfaz, como presenta la aplicación de radio que incorporan todos los dispositivos móviles Android.

Finalmente, tal y como se explica en la guía de estilo de Android, las aplicaciones desarrolladas para esta plataforma no deben incorporar un botón de cierre de forma explícita. Esto se debe, principalmente, a que la mayoría de estas aplicaciones están diseñadas para mantenerse en ejecución en un segundo plano en el momento que el usuario deja de utilizarlas, permitiendo al usuario retomar cualquier tarea en el punto donde la dejó al salir de la aplicación. La decisión de finalizar la ejecución de este tipo de aplicaciones se delega en el sistema operativo, el cual cierra toda aquella aplicación abierta que ha superado un determinado período de tiempo sin haber sido utilizada de nuevo por el usuario. No obstante, existen aplicaciones para Android que, debido a sus requerimientos de procesado o a que ofrecen un servicio de respuesta inmediata, no tiene sentido mantener en ejecución en un segundo plano. Para este tipo de aplicaciones, Android recomienda el uso del botón físico “Atrás” para dar por concluida la labor de la aplicación y llevar a cabo el cierre de la misma.

Con estas premisas, se procede a diseñar la interfaz gráfica del sistema del prototipo final. El resultado de este proceso de diseño se observa en la Figura 3.6.1, tras la que se detallan las características más importantes de la interfaz gráfica de este prototipo.

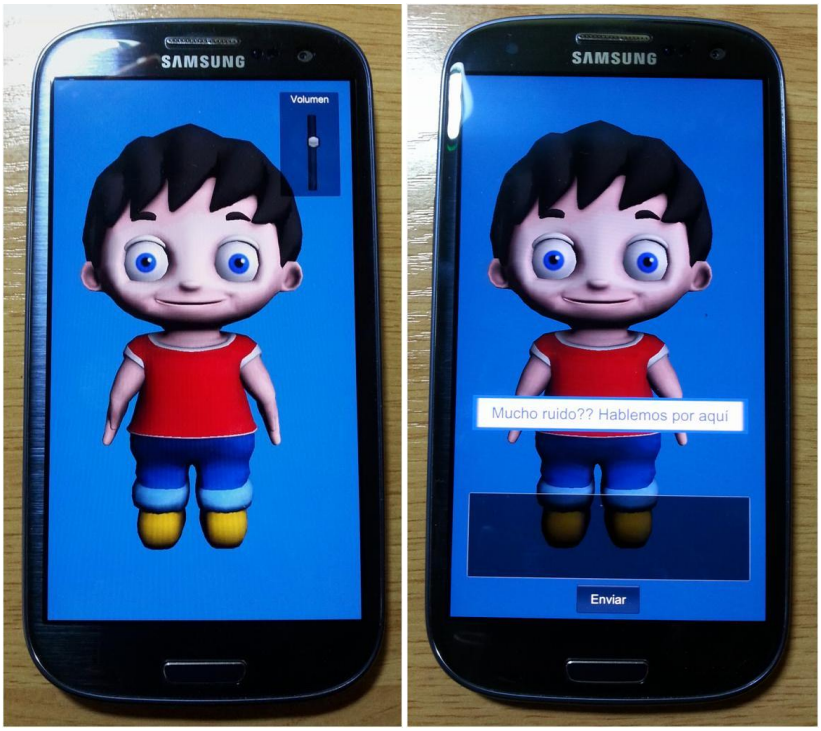


Figura 3.6.1: *Imágenes de la versión definitiva de la interfaz gráfica del sistema. (Izquierda) Interfaz gráfica utilizada durante un proceso de interacción oral. (Derecha) Interfaz gráfica utilizada durante un proceso de interacción textual*

Agente virtual

Se trata del elemento principal de la interfaz gráfica, situándose en el centro de la misma, mirando de frente al usuario. La representación gráfica del agente virtual se lleva a cabo a través de distintos personajes tridimensionales de los que disponía el grupo, seleccionados todos ellos por su reducido número de polígonos. Estos personajes tridimensionales son un niño, una niña, un perro y el torso de la mujer de Maxine.

El agente virtual desempeña dos funciones principales dentro de la interfaz gráfica del sistema. En primer lugar, es el encargado de dotar de naturalidad y realismo a la interacción con el usuario a través de las múltiples animaciones, tanto faciales como corporales, de las que dispone, dando a conocer en todo momento el estado emocional en el que se encuentra al usuario. Por otro lado, el agente virtual es un elemento activo dentro de la interfaz, ya que incorpora un "*box collider*" que detecta si el usuario pulsa sobre el agente. De esta forma, el sistema se sirve del agente virtual para dar comienzo a la escucha del discurso del usuario, puesto que una vez se aprieta sobre el agente, se inicia el proceso de reconocimiento de voz.

Menú de opciones

La interfaz gráfica del sistema también incorpora un menú de opciones, menú que, según aconseja la guía de estilo Android, es mostrado por pantalla siempre que el usuario pulsa el botón físico "MENU" del dispositivo en el que se ejecuta el sistema.

Debido a que el sistema está desarrollado sobre la plataforma Unity 3D, no es posible desplegar el menú de opciones propio de Android, por lo que se opta por desarrollar un menú de opciones que simule el menú de Android original. De esta forma, al usuario, más habituado a interactuar con el entorno de Android, le resulta más sencillo manejarse dentro del menú de opciones desarrollado. El modo de despliegue seleccionado para el sistema es el modo expandido (*expanded mode*), en donde los elementos del menú quedan representados en forma de lista flotante.

Para la generación del menú de opciones se utiliza un rectángulo, de color gris oscuro y bordes blanquecinos, sobre el que se representan las pestañas correspondientes a cada una de las opciones de dicho menú. Estas pestañas poseen también un color gris oscuro de fondo, color que cambia a azul turquesa una vez que el usuario las selecciona, y se encuentran rotuladas con texto de color blanco. El resultado, mostrado en la Figura 3.6.2, se asemeja al menú de opciones que utiliza el tema "Holo" de Android.

Figura 3.6.2: *Imagen del menú de opciones desarrollado para el sistema*

En cuanto al contenido del menú, es necesario decidir las opciones que se van a incorporar a dicho menú y desarrollar sus respectivas ventanas. Cabe señalar que, del mismo modo que ocurre con el menú de opciones, el hecho de que el sistema esté desarrollado sobre la plataforma Unity 3D impide lanzar las ventanas desplegables propias de Android, por lo que se opta por diseñar e implementar en Unity 3D las distintas ventanas desplegables que se requieren, tratando de simular en todo momento las ventanas desplegables originales del tema "Holo" de Android.

En primer lugar, debido a que la guía de estilo de Android insta a realizar toda modificación de los valores por defecto del sistema a través del menú de opciones, se opta por crear una primera pestaña en el menú que gestione los distintos modos de interacción con el agente virtual. La selección de dicha pestaña, titulada "Modos de interacción", despliega una nueva ventana que permite al usuario seleccionar los canales de comunicación con el agente virtual que desee utilizar, siendo posible cualquier combinación de los mismos, siempre y cuando esté habilitado, al menos, un canal de entrada y otro de salida.

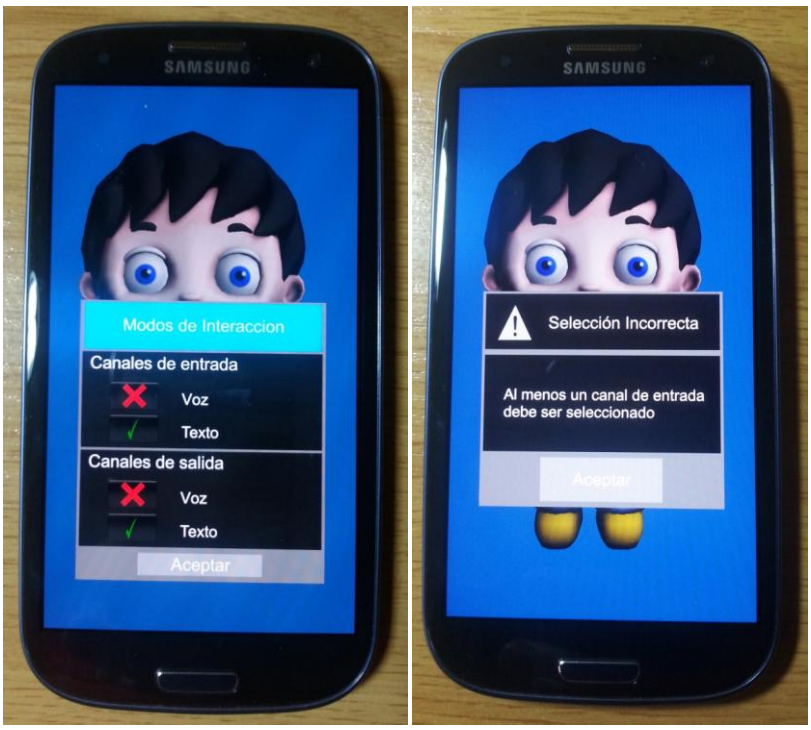


Figura 3.6.3: *(Izquierda) Imagen de la ventana desplegable para la gestión de los modos de interacción con el agente virtual. (Derecha) Imagen del cuadro de diálogo de alerta desarrollado para el sistema*

En la imagen izquierda de la Figura 3.6.3 se puede apreciar la ventana desplegada al pulsar sobre la pestaña “Modos de interacción” del menú de opciones. Como se muestra en la imagen, la ventana presenta dos secciones bien diferenciadas: una primera sección para la gestión de los canales de interacción de entrada y una segunda sección para la gestión de los canales de interacción de salida. En ambas secciones se muestra al usuario los dos posibles modos de interacción existentes, oral y textual, junto a su respectivos botones de estado. Estos botones de estado cumplen una doble función en el sistema. Por un lado, son los encargados de informar al usuario acerca de los canales de interacción que se encuentran habilitados en el sistema, presentando un *tick* verde todos aquellos botones de estado cuyos canales de interacción estén activados y una cruz roja los restantes. Por otro lado, el usuario se sirve de los botones de estado para seleccionar los canales de interacción, tanto de entrada como de salida, que desea que estén activos en el sistema, habilitando o deshabilitando dichos canales a través del pulsado de sus respectivos botones de estado.

Por su parte, la imagen derecha de la Figura 3.6.3 muestra la ventana de error disparada por el sistema en caso de que el usuario no habilite ningún canal de interacción de entrada, ventana que es análoga a la utilizada en el caso de no habilitarse ningún canal de interacción de salida. Esta ventana, desarrollada enteramente en Unity 3D, pretende imitar el estilo y forma de los “*AlertDialog*” propios de Android, también conocidos como cuadros de diálogo de alerta. Este tipo de cuadro de diálogo se limita a mostrar un mensaje sencillo de alerta al usuario, presentando un único botón “Aceptar” que debe ser pulsado por el usuario para confirmar la lectura del mensaje.

Cabe destacar que, una vez el usuario ha seleccionado los canales de interacción que desea tener habilitados con el agente virtual, el sistema activa todos los elementos extras necesarios para llevar a cabo el tipo de comunicación elegida por el usuario, desactivando los elementos restantes. En la Figura 3.6.1, mostrada con anterioridad, se puede observar las interfaces gráficas resultantes de elegir el canal oral y el canal escrito, tanto de entrada como de salida, respectivamente.

Por otro lado, siguiendo una de las recomendaciones de la guía de estilo Android que se han comentado anteriormente, se opta por incorporar al menú de opciones una pestaña de Ayuda. Esta opción, mostrada en la última pestaña del menú, sirve para informar al usuario acerca de las distintas acciones que puede realizar sobre el sistema. En este sentido, si el usuario selecciona la pestaña de Ayuda, se despliega una nueva ventana que explica al usuario cómo cambiar el modo interacción con el agente virtual, cómo modificar el volumen del sistema o la manera de iniciar una comunicación, tanto oral como escrita, con el agente virtual. En la Figura 3.6.4 se muestra una imagen de la ventana desplegada al seleccionar la pestaña Ayuda del menú de opciones del sistema.

Figura 3.6.4: *Ventana desplegable de Ayuda desarrollada para el sistema*

Control de volumen

Con respecto al control del volumen del sistema, se opta por desarrollar las dos opciones permitidas por la guía de estilo Android, a saber: controlar el volumen del sistema a través de los botones físicos destinados para ello en el dispositivo o a través del regulador existente en la interfaz gráfica. De esta forma, el usuario puede hacer uso de aquella opción que le resulte más cómoda y sencilla.

En este sentido, con el fin de permitir la segunda opción se desarrolla un controlador de volumen táctil en la propia interfaz gráfica del sistema. Este controlador está formado por una caja con una barra de desplazamiento en su interior, situándose en la esquina superior derecha de la interfaz. Para ello, se sirve de las funciones de gestión del volumen implementadas para el sintetizador de voz (como por ejemplo la función "cambiaVolumen", función que viene mostrada en el Cuadro D.2.3 del Anexo D).

Cierre del sistema

Debido a que el sistema a desarrollar requiere una gran capacidad de procesado y que carece de sentido mantenerlo en ejecución en un segundo plano, se opta por utilizar el botón físico "Atrás" del dispositivo para dar por concluida la interacción con el agente virtual y proceder al cierre del sistema.

Tanto el reconocimiento como la gestión del pulsado de los botones físicos destinados al control del volumen, el botón "Salir" y el botón "Atrás" (utilizados a la hora de modificar el volumen del sistema, mostrar el menú y cerrar la aplicación respectivamente) se explica detalladamente en la sección D.9 del anexo D.

4. Pruebas con usuarios: generación y evaluación de voces emocionales

En este cuarto capítulo de la memoria se presentan las distintas pruebas realizadas con usuarios finales durante el desarrollo del sistema. El capítulo se encuentra dividido en tres apartados, un primer apartado en el que se dan a conocer las distintos métodos de evaluación utilizados en las pruebas realizadas, un segundo apartado en el que se explican brevemente las pruebas preliminares llevadas a cabo para la generación de voces emocionales realistas y un último apartado donde se presentan cada una de las distintas pruebas finales realizadas para la evaluación de la calidad y relevancia de dichas voces emocionales con respecto a otros factores que denotan emociones en el proceso de interacción con el usuario.

4.1 Métodos de evaluación

En las siguientes líneas se describen brevemente los distintos métodos de evaluación utilizados en las pruebas realizadas. El objetivo es que el lector sepa a qué tipo de prueba se hace referencia en cada momento, ya que los métodos que se definen a continuación son citados en varias ocasiones a lo largo de este capítulo:

- Elección libre: este método de evaluación consiste en no restringir la respuesta del usuario a un conjunto cerrado de términos. Este tipo de evaluación está especialmente indicada para encontrar fenómenos no esperados durante la prueba.

- Elección forzada: al contrario que el método de evaluación anterior, este método se fundamenta en facilitar a los usuarios que van a llevar a cabo la prueba un conjunto finito de posibles respuestas que engloban todas las emociones que han sido modeladas.

- Elección libre modificada: es un método de evaluación intermedio entre la elección libre y la elección forzada. Se basa en introducir ciertas modificaciones en el paradigma de elección forzada como, por ejemplo, dotar al usuario de un gran conjunto de emociones entre las que seleccionar su respuesta, emociones entre las cuales existan varias categorías de distracción y la categoría "otros". Otra modificación muy extendida consiste en combinar las voces emocionales modeladas con textos cuyo contenido denota cierto estado emocional.

- Self-Assessment Manikin (SAM) [Morris, 1995]: este método de evaluación se sirve de escalas pictográficas de fácil entendimiento para medir la percepción emocional del usuario hasta en tres dimensiones: activación, valencia y dominancia. En las pruebas con usuarios realizadas únicamente se han utilizado las dos primeras dimensiones que son las habituales.

Tras haber definido los métodos de evaluación utilizados, se procede a presentar las dos técnicas de evaluación de las que se ha hecho uso:

- Encuesta: esta técnica de evaluación se basa en obtener información a partir de las respuestas de los usuarios a una serie de preguntas escritas y organizadas en un cuestionario impreso. Al usuario encuestado se le permite leer previamente el cuestionario, respondiendo el mismo por escrito y sin la intervención directa de ninguna persona que colabore en el estudio.

- Entrevista: esta técnica de evaluación consiste en que la persona que está llevando a cabo la investigación realice una serie de preguntas acerca de diversos aspectos del sistema a los usuarios, pretendiendo conocer de primera mano sus opiniones y apreciaciones al respecto.

Una vez definidos los métodos y técnicas de evaluación utilizados, se procede a explicar cada una de las pruebas realizadas con usuarios finales en este Proyecto Fin de Carrera.

4.2 Pruebas preliminares: generación de voces emocionales

En este apartado de la memoria se presentan el conjunto de pruebas llevado a cabo para la generación de voces emocionales lo más realistas posibles (la metodología seguida para la realización de estas pruebas se describe en la sección F.1 del Anexo F). Este conjunto de pruebas se encuentra dividido en dos fases bien diferenciadas, como son la fase de calibración y la fase de selección de las voces emocionales. A continuación se describen brevemente ambas fases:

- Calibración: esta primera fase (explicada de forma más detallada en la sección F.2 del Anexo F) tiene como objetivo calibrar las voces emocionales generadas por el desarrollador. Esta fase consta de dos módulos de pruebas en los que los usuarios deben evaluar los distintos bloques de voces emocionales generados, módulos a los que siguen sendas etapas de modificación de las voces emocionales en función de los resultados obtenidos en los mismos. Durante esta fase, el desarrollador se sirve de encuestas de elección libre y elección libre modificada para realizar la evaluación de los bloques con los usuarios.

- Selección: esta segunda fase (explicado de forma más detallada en la sección F.3 del Anexo F) tiene como fin seleccionar, para cada uno de los estados emocionales que se desea representar, la voz emocional que mejor transmita la emoción correspondiente. Esta fase de selección consta de dos módulos de pruebas en los que los usuarios deben evaluar los distintos bloques de voces emocionales generados, basándose el desarrollador en los resultados de cada uno de estos módulos a la hora de llevar a cabo sendas cribas respectivas de las voces emocionales generadas. El resultado final de estas dos cribas es la definición de una única voz emocional para cada uno de los estados emocionales del agente virtual. Durante esta fase, el desarrollador se sirve de encuestas de elección forzada y elección libre modificada para realizar la evaluación de los bloques con los usuarios.

El objetivo de estas pruebas es la determinación de los valores más adecuados de volumen, velocidad y tono (los tres únicos que el sistema TTS de Android permite manipular) para cada una de la emociones consideradas en el sistema (neutra, tristeza, alegría, sorpresa y enfado). Los valores seleccionados para cada voz emocional se presentan en la tabla 5.1.1 perteneciente al capítulo 5.

4.3 Pruebas finales: evaluación de las voces emocionales

Tras haber seleccionado las voces emocionales correspondientes a cada uno de los estados emocionales del agente virtual, se lleva a cabo una última fase de evaluación, mucho más exhaustiva, que pretende determinar la calidad de dichas voces emocionales y su relevancia con respecto a otros aspectos del sistema que denotan emociones en el agente durante el proceso de interacción con el usuario. En este sentido, esta última fase de pruebas tiene tres objetivos principales:

- Conocer la capacidad del usuario para reconocer los estados emocionales del agente a través de las voces emocionales

- Observar el impacto del contenido semántico de las frases reproducidas en esta percepción emocional

- Determinar la relevancia de la voz escuchada con respecto a la imagen observada durante la interacción con el agente virtual.

En los siguientes apartados se explica en detalle la metodología seguida a la hora de realizar todas las pruebas y el diseño de cada una de estas pruebas para evaluar los distintos ámbitos de la interacción con el agente virtual descritos anteriormente.

4.3.1 Metodología utilizada durante las pruebas con usuarios

Antes de definir las diferentes pruebas a realizar en esta última fase de evaluación, es necesario establecer los usuarios que participarán en las pruebas y la metodología para llevarlas a cabo.

Participantes

Los usuarios con los que se va a realizar esta última fase de evaluación pertenecen a cuatro conjuntos de personas distintos:

- Amigos y familiares del evaluador: es el conjunto más heterogéneo de todos ya que lo conforman usuarios de rangos de edad muy dispares y con formación enormemente diversa. Las pruebas con este conjunto de usuarios se llevan a cabo en el domicilio del evaluador o en el laboratorio de trabajo del mismo.

- Alumnos de Interacción Persona-Ordenador: este grupo de usuarios está formado por alumnos del Grado en Ingeniería Informática con edades comprendidas entre los 19 y los 22 años. La evaluación se realiza en el horario de prácticas establecido para esta asignatura, en el laboratorio donde se llevan a cabo dichas prácticas.

- Alumnos de Diseño Centrado en el Usuario: es un nuevo conjunto de usuarios formado por alumnos del Grado en Ingeniería Informática, cuyas edades están comprendidas entre los 21 y los 25. Del mismo modo que sucediera en el conjunto anterior, las pruebas se realizan en el horario de prácticas establecido para esta asignatura, en el laboratorio donde se llevan a cabo dichas prácticas.

- Alumnos de Entornos 3D Interactivos: este grupo de usuarios está formado por alumnos del Grado en Ingeniería en Diseño Industrial y Desarrollo de Producto con edades comprendidas entre los 20 y los 22 años. La evaluación se realiza en el horario de clase teórica establecido para esta asignatura, en el aula donde se llevan a cabo dichas clases.

Con estos conjuntos de usuarios se pretende alcanzar un mínimo de 40 personas encuestadas, otorgando cierta representatividad a los resultados que se extraigan del estudio.

Estructura de las sesiones de evaluación

Cada sesión de evaluación consta de pre-test, test (o evaluación propiamente dicha) y post-test.

Pre-Test

El pre-test pretende definir el perfil de los participantes en la evaluación y consta únicamente de los siguientes apartados:

- Nombre
- Sexo
- Edad
- Ocupación

A estos datos, el evaluador incorpora información extra como, por ejemplo, el grupo y el horario en el que se encuentran un determinado conjunto de alumnos que realizan las prueba.

Test

En cada sesión de evaluación se llevan a cabo tres pruebas:

- Prueba Nº1: Reconocimiento de las voces emocionales
- Prueba Nº2: Influencia del contenido semántico de las frases en la percepción emocional
- Prueba Nº3: Relevancia de la imagen respecto a la voz en la percepción emocional

Como método de evaluación se opta por utilizar la técnica de test o encuesta.

Por un lado, el usuario debe determinar el estado emocional que percibe en cada una de las reproducciones emocionales de las distintas pruebas, labor para la que se ha seleccionado el método de evaluación de la elección libre modificada. Con este método de evaluación se pretende evitar que el usuario se vea obligado a seleccionar un determinado estado emocional en caso de no estar seguro, añadiendo al conjunto de términos NEUTRO, ALEGRE, SORPRENDIDO, TRISTE y ENFADADO una nueva categoría, NO LO RECONOZCO, categoría que permite al usuario no decantarse por ningún estado emocional si no lo ha percibido claramente.

Por otra parte, se utiliza la escala SAM para que los usuarios determinen el grado de activación y valencia percibidas en las distintas reproducciones emocionales.

Para facilitar la labor de evaluación a los usuarios, se les hace entrega de tres hojas de encuesta, una por cada prueba a realizar, con preguntas numeradas acerca del estado emocional percibido en cada una de las veinte reproducciones existentes en cada una de las tres pruebas. En la figura 4.3.1 se muestra una de estas preguntas numeradas.

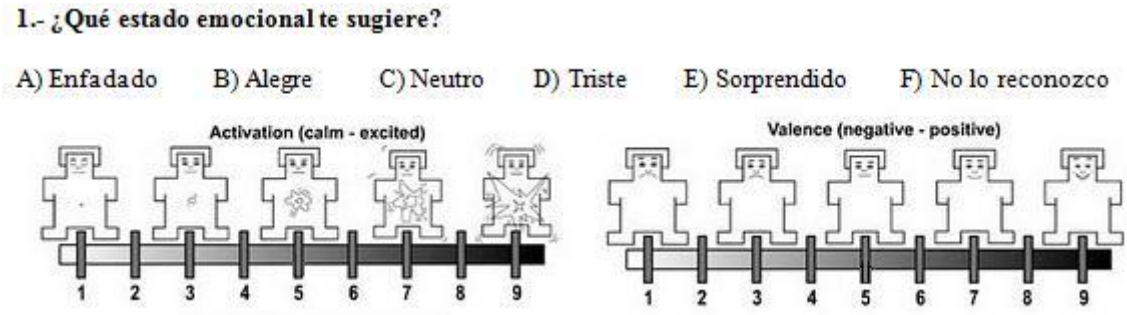


Figura 4.3.1: *Primera pregunta de las veinte existentes en cada una de las hojas de encuesta facilitadas a los usuarios*

Una vez que se le ha entregado un bloque de hojas de respuestas, el evaluador procede a explicar al usuario el procedimiento a seguir en las distintas pruebas. A continuación se le hace entrega del dispositivo móvil sobre el que se va a realizar las distintas pruebas, se le otorga unos auriculares para favorecer la escucha del discurso del agente y se lanza la aplicación de evaluación. La duración total del test para cada usuario es de unos 10 minutos.

Post-test

Finalmente, se lleva a cabo una entrevista de carácter informal con los usuarios, donde el evaluador pregunta sobre las impresiones y opiniones de los usuarios acerca de las distintas pruebas realizadas. Con esta información se pretende mejorar tanto la experiencia de los futuros usuarios encuestados como el diseño de las propias pruebas.

Una vez definida la metodología a seguir durante la realización de las pruebas finales se procede a explicar en detalle cada una de dichas pruebas.

4.3.2 Reconocimiento de las voces emocionales

En primer lugar, se desea conocer la capacidad de los usuarios para reconocer el estado emocional en el que se encuentra el agente virtual únicamente a través de la voz escuchada. Para ello, se desarrolla una primera prueba en la que se reproducen distintas voces emocionales y se insta a los usuarios a determinar el estado emocional percibido en cada una de las voces reproducidas, evitando en todo momento que cualquier otro aspecto emocional influya en la percepción de los usuarios encuestados. Con este fin, se adoptan una serie de medidas a la hora de diseñar la prueba, medidas que se explican en detalle a continuación:

- Frase neutra: con el objetivo de que el usuario no se vea influenciado por el contenido de la frase reproducida con cada voz emocional, se hace uso de una misma frase neutra para todas las voces emocionales utilizadas. La frase neutra elegida es "Los viernes la fruta está mucho más barata", frase que ha sido usada en los módulos de pruebas previos y que, según los propios usuarios encuestados, no les ha aportado emotividad alguna.

- Sin imagen: para evitar que la percepción emocional del usuario pueda verse influenciada por el aspecto del agente virtual, se opta por no reproducir al personaje del agente virtual en la pantalla del dispositivo durante la prueba.

- Varias repeticiones: se considera necesario reproducir un mínimo de cuatro veces cada una de las voces emocionales seleccionadas, ya que la frase utilizada no es lo suficientemente larga para distinguir las distintas voces emocionales generadas con una sola reproducción. Además, los módulos de pruebas llevados a cabo anteriormente sugieren un cierto aprendizaje por parte de los usuarios conforme avanza la prueba realizada.

- Aleatoriedad: otra de las conclusiones a las que se llega tras realizar los módulos de pruebas anteriores es que la percepción de una voz emocional viene influida, en cierto modo, por la voz emocional previa. En este sentido, una voz neutra es percibida de forma más negativa si la precede una voz alegre, mientras que es percibida con un carácter más positivo si es reproducida detrás de una voz de enfado. Es por ello que se considera necesario que el conjunto de las veinte voces emocionales a reproducir durante la prueba siga un orden totalmente aleatorio, de forma que se mitigue la influencia de este factor en el cómputo global de los resultados obtenidos.

- Espacio entre reproducciones: con el fin de asegurar que el usuario disponga del tiempo necesario para clasificar cada una de sus percepciones emocionales, se opta por separar cada una de las reproducciones por diez segundos. De esta forma, se reduce también la influencia de la reproducción previa sobre la percepción de la reproducción actual.

4.3.3 Influencia del contenido semántico de las frases reproducidas

Además de la calidad de las voces emocionales seleccionadas para transmitir los estados emocionales del agente virtual, se considera interesante estudiar la influencia del contenido semántico de las frases reproducidas en la percepción emocional del usuario durante el discurso del agente. El principal objetivo de este estudio es conocer qué factor tiene una mayor relevancia para los usuarios a la hora de determinar el estado emocional en el que se encuentra en agente virtual, si el tipo de voz con el que se reproduce el discurso o el contenido semántico del mismo.

En primera instancia, es necesario seleccionar un conjunto de cinco frases con distintas connotaciones emocionales, las cuales se deben corresponder con cada uno de los estados emocionales del agente virtual. A continuación se muestran las frases seleccionadas para cada estado emocional considerado:

- Triste: "He perdido mi peluche favorito"

- Enfadado: "La última vez que me faltas al respeto"

- Neutro: "La lámpara de mi cuarto es verde"

- Alegre: "He sacado un diez en matemáticas"

- Sorprendido: "Ese perro me está hablando"

Una vez seleccionadas las frases emocionales, se procede a diseñar una segunda prueba que evalúe la influencia del contenido semántico de las frases reproducidas en la percepción emocional del usuario durante el discurso del agente.

Inicialmente se mezclan las distintas voces emocionales modeladas con las frases con connotación emocional seleccionadas anteriormente. Para ello, se ordenan de forma aleatoria tanto el conjunto de las veinte voces emocionales a utilizar como el conjunto de las veinte frases a reproducir, procediendo posteriormente a la combinación de las voces y frases que se encuentran en la misma posición dentro de sus respectivos conjuntos. Cabe destacar que el evaluador se encarga de que, al menos una vez, la voz y frase correspondientes a un mismo estado emocional se reproduzcan simultáneamente, puesto que se desea estudiar los resultados de este tipo de combinaciones.

Tras haber llevado a cabo la mezcla, se reproduce la combinación resultante a los usuarios e instándoles a determinar qué estado emocional perciben en cada caso. De esta forma, se pretende conocer cómo afecta el contenido semántico del discurso en la percepción del estado emocional del agente por parte del usuario, pudiendo comparar los resultados obtenidos en esta prueba con los obtenidos en la prueba anterior, en la cual únicamente influía la voz emocional reproducida. Al igual que en la prueba anterior se toman las siguientes medidas: no se hace uso de imagen, se llevan a cabo cuatro repeticiones de cada reproducción, el orden es aleatorio y se separan las reproducciones con 10 segundos de espera.

4.3.4 Relevancia de la imagen con respecto a la voz

Por último, otro de los aspectos que se desea estudiar es la influencia de la imagen del agente virtual en la percepción emocional del usuario durante del proceso de interacción.

Una de las labores del agente virtual es dotar de mayor realismo y naturalidad a la interacción con el usuario, sirviéndose para ello de distintas animaciones tanto faciales como corporales que permiten enfatizar las emociones y actitudes del agente frente a la información proveniente del usuario. El estudio que se desea realizar pretende estudiar la importancia relativa que poseen estas animaciones del agente virtual con respecto a las voces emocionales en la percepción de los estados emocionales del agente por parte del usuario.

Para realizar esta prueba, se ha de seleccionar uno de los personajes tridimensionales con los que se ha trabajado en este Proyecto Fin de Carrera. Puesto que se pretende valorar la influencia de las animaciones del personaje en la percepción emocional del usuario, se selecciona el torso de la mujer de Maxine como representación gráfica del agente virtual durante la prueba (ver Figura 3.5.1). Esta elección se debe a que las animaciones faciales y corporales que incorpora este personaje para representar estados emocionales son de una mayor complejidad y poseen un mayor realismo que las que incorporan los demás personajes disponibles.

Inicialmente se mezclan las distintas voces emocionales modeladas con las animaciones que incorpora Maxine para representar los diversos estados emocionales del agente virtual. Para ello, se ordenan de forma aleatoria tanto el conjunto de las veinte voces emocionales a utilizar como el conjunto de las veinte animaciones a reproducir, procediendo posteriormente a la combinación de las voces y animaciones que se encuentran en la misma posición dentro de sus respectivos conjuntos. Cabe destacar que el evaluador se encarga de que, al menos una vez, la voz y animación correspondientes a un mismo estado emocional se reproduzcan simultáneamente, puesto que se desea estudiar los resultados de este tipo de combinaciones.

Una vez realizada la mezcla, esta tercera prueba consiste en reproducir la combinación resultante a los usuarios e instarles a determinar qué estado emocional perciben en cada caso. De este modo, a través de la comparación de los resultados obtenidos con los resultados de la primera prueba, se pretende conocer la influencia de la imagen del agente virtual en la percepción emocional del usuario. Además de las medidas de repetición, aleatoriedad y espacio entre reproducciones se hace uso siempre de la frase neutra con el objetivo de que el usuario no se vea influenciado por el contenido de la frase reproducida.

5. Resultados de las pruebas con usuarios

En este capítulo se presentan los resultados obtenidos en las distintas pruebas realizadas con usuarios. En primer lugar, se dan a conocer los valores de volumen, velocidad y tono seleccionados para generar cada una de las voces emocionales del sistema. Posteriormente, se presentan los resultados más significativos de cada una de las tres pruebas llevadas a cabo para la evaluación de la calidad y relevancia de las voces emocionales con respecto a otros factores que denotan emociones en el proceso de interacción con el usuario.

5.1 Caracterización de voces emocionales

El objetivo las fases de calibración y selección, explicadas en el capítulo anterior, es determinar los valores más adecuados de volumen, velocidad y tono para cada una de la emociones consideradas en el sistema (neutra, tristeza, alegría, sorpresa y enfado). A continuación, en la Tabla 5.1.1, se muestra los valores seleccionados para cada una de las voces emocionales generadas.

	Voz Triste	Voz Enfadada	Voz Neutra	Voz Alegre	Voz Sorprendida
Volumen	9.28	12.8	10.7	12.6	13.6
Velocidad	0.8	1.6	1.0	1.3	1.2
Tono	1.3	1.2	1.7	2.1	2.3

Tabla 5.1.1: *Tabla con los valores seleccionados para el volumen, velocidad y tono de cada una de las voces emocionales generadas*

5.2 Reconocimiento de las voces emocionales

Uno de los objetivos perseguidos con la realización de este conjunto de pruebas finales era evaluar la capacidad de los usuarios para reconocer el estado emocional en el que se encuentra el agente virtual únicamente a través de la voz emocional escuchada. Para ello, se ha llevado a cabo la prueba descrita en el apartado 4.3.1 de esta memoria, prueba que han realizado un total de 48 usuarios finales. En las siguientes líneas se dan a conocer los resultados más relevantes obtenidos de la realización de esta primera prueba (explicados más en detalle en la sección G.1 del Anexo G).

En primer lugar, se procede a analizar los resultados obtenidos por cada una de las voces emocionales generadas en la encuesta de elección libre modificada llevada a cabo con los usuarios. Estos resultados pueden ser observados de forma gráfica en la Figura 5.2.1, donde se muestra el porcentaje de respuestas obtenido por las distintas voces emocionales para cada categoría existente en la encuesta.

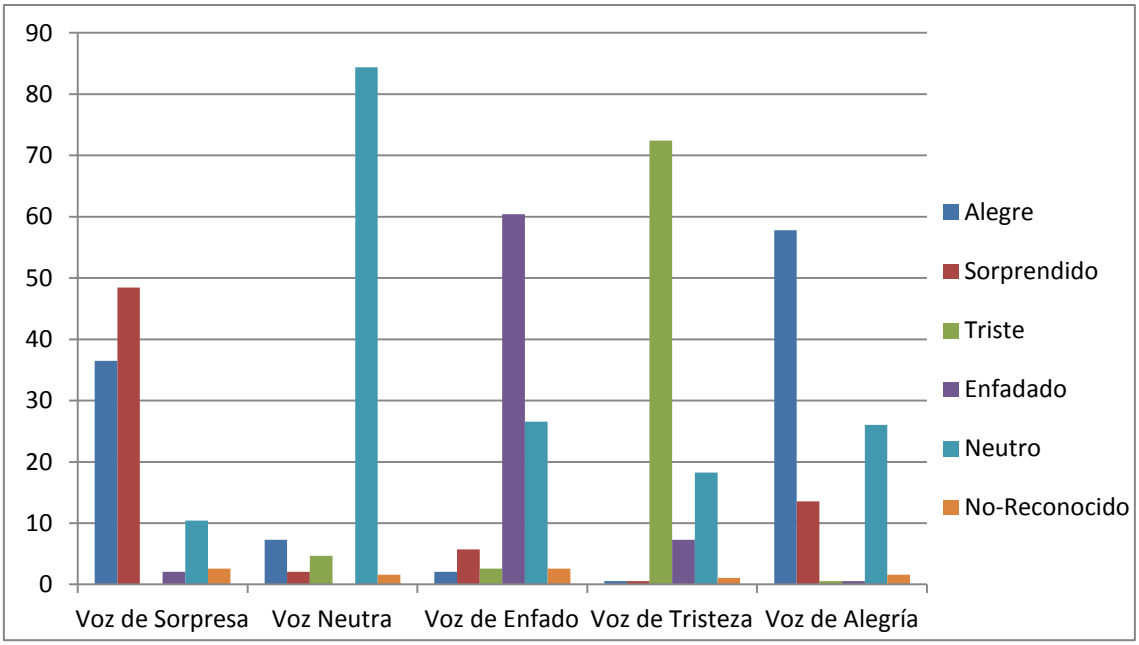


Figura 5.2.1: *Resultados obtenidos por cada una de las voces emocionales generadas en la encuesta de elección libre modificada realizada por los 48 usuarios encuestados. En el eje horizontal se clasifican las distintas voces emocionales generadas. En el eje vertical se muestra el porcentaje de respuestas obtenido por dichas voces emocionales para cada término existente en la encuesta de elección libre modificada.*

Como se puede apreciar en la figura anterior, las voces emocionales neutra y triste obtienen unos resultados notablemente satisfactorios, puesto que son reconocidas por más del 80% y 70% de los usuarios encuestados respectivamente. Estos datos las convierten en las dos voces emocionales más conseguidas de las cinco voces generadas para el sistema.

En cuanto a las voces emocionales de alegría y enfado, presentan unos resultados moderadamente aceptables, siendo reconocidas por el 60% de los usuarios encuestados aproximadamente. Es importante reseñar que, en el caso de la voz emocional de alegría, la segunda opción más votada es la de sorpresa, aunque con un porcentaje muy inferior al obtenido por la opción alegre en el caso de la voz emocional de sorpresa. Por su parte, sorprende que la segunda opción más votada para la voz emocional de enfado sea la neutra, ya que no se habían detectado confusiones notables entre ambas en las pruebas previas.

Finalmente, la voz emocional menos lograda a raíz de los resultados es la de sorpresa, siendo reconocida por menos del 50% de los usuarios encuestados. Este hecho, unido a que aproximadamente un tercio de los usuarios se decanten por el estado emocional de alegría al escuchar esta voz emocional, viene a confirmar las dificultades ya detectadas en pruebas anteriores (explicadas en la sección F.2 del Anexo F) para generar voces emocionales de alegría y sorpresa que no conlleven confusión entre sí.

Además, analizando en profundidad los resultados obtenidos en esta primera prueba, cabe destacar la mejor percepción emocional de las mujeres con respecto a los hombres. Este hecho viene reflejado en la Tabla 5.2.1, donde se muestran numéricamente los porcentajes medios de acierto tanto de los usuarios como de las usuarias encuestadas.

Porcentaje de Aciertos Total	Porcentaje de Aciertos en Chicos	Porcentaje de Aciertos en Chicas
64.58%	61.86%	71.92%

Tabla 5.2.1: *Porcentajes medios de aciertos obtenidos en la prueba de reconocimiento de las voces emocionales generadas.*

5.3 Influencia del contenido semántico de las frases reproducidas

Además de la calidad de las voces emocionales seleccionadas para transmitir los estados emocionales del agente virtual, se ha considerado interesante estudiar la influencia del contenido semántico de las frases reproducidas en la percepción emocional del usuario durante el discurso del agente. Con este fin, se ha llevado a cabo una segunda prueba, descrita en el apartado 4.3.2 de esta memoria, cuyos resultados más relevantes se explican a continuación (el conjunto total de estos resultados viene detallado de forma más amplia en la sección G.2 del Anexo G).

En las siguientes líneas, se analizan los resultados obtenidos por las reproducciones formadas por la voz y frase emocionales correspondientes a una misma emoción. Se desea comprobar si la conjunción de estos dos factores emocionales permite al usuario reconocer más fácilmente los estados emocionales que se pretenden transmitir.

Con un porcentaje medio de acierto del 88.89% en las reproducciones formadas por voces y frases emocionales acordes, se confirma que los usuarios encuestados perciben el estado emocional de forma más sencilla y acertada en el caso de que ambos factores emocionales correspondan a una misma emoción. De este modo, los resultados obtenidos avalan que el usuario pueda ser capaz de reconocer el estado emocional en el que se encuentra el agente virtual durante la ejecución del sistema, puesto que durante la misma tanto frase como voz emocional serán reproducidas de forma acorde.

A continuación, en la Tabla 5.3.1, se muestran los porcentajes medios de acierto asociados a cada una de las conjunciones de voces y frases emocionales correspondientes a una misma emoción.

Porcentaje Aciertos del par Sorprendido	Porcentaje Aciertos del par Neutro	Porcentaje Aciertos del par Enfadado	Porcentaje Aciertos del par Triste	Porcentaje Aciertos del par Alegre
75.00%	97.91%	97.92%	94.79%	72.92%

Tabla 5.3.1: *Porcentaje medio de acierto obtenido por la conjunción de las voces y frases emocionales correspondientes a una misma emoción en la prueba que estudia la influencia del contenido semántico de las frases reproducidas en la percepción del usuario*

Como se puede apreciar, el porcentaje de aciertos obtenido supera en todos los casos el 70%, alcanzando tasas cercanas al 98% de acierto en los pares emocionales de enfado y neutro. A pesar de haber cosechado resultados altamente satisfactorios en todos los casos, vuelve a ser significativo que los pares emocionales peor evaluados sean el sorprendido y el alegre, cuyas voces emocionales son las menos logradas a la luz del análisis llevado a cabo en el apartado 5.2 de este capítulo.

5.4 Relevancia de la imagen con respecto a la voz emocional

Finalmente, otro de los aspectos que se desea estudiar es la relevancia que adquiere la imagen del agente con respecto a las voces emocionales modeladas a la hora de que el usuario reconozca el estado emocional del agente virtual. Con este fin, se ha llevado a cabo una tercera prueba con usuarios, descrita en el apartado 4.3.3 de esta memoria, cuyos resultados más relevantes se explican a continuación (el conjunto total de estos resultados se describe en detalle en la sección G.3 del Anexo G).

En las siguientes líneas se analizan los resultados obtenidos por las reproducciones formadas por la animación y la voz emocional correspondientes a una misma emoción. Se desea comprobar si la conjunción de estos dos factores emocionales permite al usuario reconocer más fácilmente los estados emocionales que se pretenden transmitir.

Con un porcentaje medio de acierto del 92.5% en las reproducciones formadas por animaciones y voces emocionales acordes, es posible afirmar que los usuarios encuestados perciben el estado emocional de forma más sencilla y acertada en el caso de que ambos factores emocionales correspondan a una misma emoción. De esta manera, los resultados obtenidos invitan a pensar que el usuario será capaz de reconocer el estado emocional en el que se encuentra el agente virtual durante la ejecución del sistema

sin dificultad, puesto que durante dicha ejecución tanto las animaciones como las voces emocionales serán reproducidas de forma acorde.

A continuación, en la Tabla 5.4.1, se muestran los porcentajes medios de acierto asociados a cada una de las conjunciones de emociones y voces emocionales correspondientes a una misma emoción.

Porcentaje Aciertos del par Sorprendido	Porcentaje Aciertos del par Neutro	Porcentaje Aciertos del par Enfadado	Porcentaje Aciertos del par Triste	Porcentaje Aciertos del par Alegre
89.58%	93.75%	95.83%	100%	83.33%

Tabla 5.4.1: *Porcentaje medio de acierto obtenido por la conjunción de las animaciones y voces emocionales correspondientes a una misma emoción en la prueba que estudia la relevancia de la imagen con respecto a la voz en la percepción del usuario*

Como se puede apreciar, el porcentaje de aciertos obtenido supera en todos los casos el 80%, alcanzando el 100% en el par emocional triste y valores superiores al 90% en los pares neutro y enfadado. A pesar de haber cosechado excelentes resultados en todos los casos, se vuelve a repetir la circunstancia de que los pares emocionales peor evaluados sean el sorprendido y el alegre, cuyas voces emocionales son las menos logradas.

6. Conclusiones y trabajo futuro

En este último capítulo se analiza el cumplimiento de los objetivos, se proponen posibles ideas para un trabajo futuro sobre el sistema desarrollado y se muestra la valoración personal del autor sobre el trabajo realizado a lo largo del Proyecto Fin de Carrera.

6.1 Conclusiones

Una vez finalizado el presente proyecto, es posible afirmar que se ha conseguido desarrollar un sistema basado en un agente virtual para dispositivos móviles Android, el cual permite una interacción multimodal y relativamente natural con el usuario, teniendo en consideración las limitaciones de procesado intrínsecas a estos dispositivos. Para ello, se han logrado llevar a cabo los objetivos planteados en el primer capítulo de esta memoria, como se muestra a continuación:

- Se ha llevado a cabo la elección de las herramientas con las que implementar el agente virtual, específico para dispositivos móviles, tanto desde el punto de vista gráfico como para desarrollar cada uno de sus modos de interacción a incorporar al propio agente. Con este fin se ha realizado un exhaustivo análisis de las distintas opciones existentes así como de las utilizadas en otros proyectos de índole similar.

- Se han desarrollado los módulos necesarios para hacer posible la comunicación oral entre el agente virtual y el usuario. En este sentido, se han implementado un sintetizador de voz o TTS, que permite reproducir oralmente el discurso del agente virtual, y un reconocedor de discurso o ASR, que permite al usuario dirigirse al agente de forma oral.

- Se han desarrollado dos módulos que permiten la comunicación escrita entre el agente virtual y el usuario. En primer lugar, se ha implementado un panel deslizable, el cual reproduce textualmente los mensajes provenientes del agente virtual en el interior de la interfaz gráfica del sistema. En segunda instancia, se ha implementado un área de texto, situada dentro de la interfaz, la cual se encarga de gestionar el proceso de escritura del mensaje por parte del usuario y de enviar dicho mensaje al agente virtual.

- Se ha dotado al sistema de una interfaz gráfica en la que el agente virtual es el elemento central de la misma, permitiendo al agente expresarse y comunicarse de forma visual con el usuario. Además, se ha desarrollado un módulo motor encargado de gestionar las animaciones, tanto corporales como faciales, que incorpora el agente virtual, permitiendo la mezcla o reproducción simultánea de varias de estas animaciones.

- Se ha dotado de ciertos aspectos emocionales a todas las modalidades de interacción entre el agente virtual y el usuario. En este sentido, se han generado distintas voces emocionales para cada uno de los estados emocionales en los que se puede encontrar el agente, se han seleccionado distintos colores de fuente y texturas para denotar diversas emociones a través del panel deslizable y se han utilizado personajes tridimensionales que incorporan animaciones que se corresponden con cada uno de los estados emocionales del agente virtual a representar.

- Se ha implementado un prototipo que permite, en un escenario sencillo de aplicación, valorar con usuarios los modos de interacción y las prestaciones del sistema.

Además de estos objetivos iniciales, se ha desarrollado también un módulo Gestor de Diálogo que permite programar al agente virtual de forma que sea capaz de mantener una conversación fluida y coherente con el usuario. Este módulo Gestor de Diálogo, basado en el Programa AB, utiliza ficheros AIML para reconocer los mensajes del usuario y llevar a cabo la búsqueda de las respuestas más adecuadas a cada uno de estos mensajes.

A su vez, se han llevado a cabo pruebas exhaustivas y sistemáticas con usuarios finales para evaluar las voces emocionales que incorpora el sistema, así como la influencia del contenido semántico de las frases reproducidas y la imagen en la percepción emocional del usuario.

6.2 Trabajo futuro

En esta sección se describen los posibles trabajos futuros que podrían complementar o ampliar el proyecto realizado.

- Aumentar el realismo y naturalidad de las voces emocionales generadas para el sistema. En este sentido, se podría optar por ampliar la funcionalidad del módulo *Text To Speech* del sistema, desarrollando mecanismos que permitieran modificar los valores de los parámetros ya existentes a lo largo de la reproducción del discurso, dando lugar a distintas entonaciones dentro de una misma frase o pudiendo enfatizar determinadas partes o palabras del discurso del agente. Para ello, sería necesario implementar algún mecanismo (pre-procesado del discurso, estimación estadística, etc.) que otorgara al desarrollador información temporal de la reproducción del discurso, funcionalidad de la que no dispone el sistema TTS Android utilizado.

- Hacer uso de sintetizadores de voz TTS distintos para cada personaje tridimensional utilizado. Para realizar esta mejora en el sistema, se debería llevar a cabo un estudio de los sintetizadores de voz en castellano existentes para Android, seleccionando aquellos cuya voz sea más acorde a las características de cada uno de los personajes utilizados. Posteriormente, sería necesario llevar a cabo un proceso de generación de voces emocionales con todos los sintetizadores del discurso seleccionados, puesto que es muy probable que los rangos de valores correspondientes a cada uno de los parámetros del discurso sean completamente diferentes de un sintetizador de voz a otro. Finalmente, sería necesario incorporar las modificaciones necesarias para asociar cada uno de estos sintetizadores al uso de un determinado agente virtual en el sistema.

- Mejorar el Gestor de Diálogo del sistema. Por un lado, se propone dotar de una mayor capacidad conversacional al agente virtual a través de la incorporación de nuevos ficheros AIML cuya complejidad sea superior a la de los actuales. En este sentido, se recomienda hacer un mayor uso de la recursividad soportada por AIML, generar nuevas variables que permitan al agente virtual almacenar una mayor cantidad de información y utilizar los distintos mecanismos existentes para enlazar respuestas. Por otra parte, se propone desarrollar varios *chatbots* para el sistema, cada uno de los cuales estuviera focalizado a un determinado ámbito como, por ejemplo, la enseñanza de matemáticas. De esta manera, al inicio del sistema, el usuario podría determinar el ámbito del que desea hablar con el agente virtual, cargando el sistema el *chatbot* específico para dicho ámbito y reduciendo el tiempo de respuesta del agente.

6.3 Valoración personal

A título personal, estoy realmente satisfecho con el trabajo realizado. He sido capaz de cumplir los objetivos inicialmente fijados y he podido trabajar en un campo de la informática que siempre me ha gustado. Además, la realización de este proyecto ha sido una experiencia muy enriquecedora desde diversos puntos de vista:

Desde el punto de vista técnico, he tenido que investigar cómo llevar a cabo cada una de las operaciones que debía realizar el sistema, estudiando las distintas alternativas posibles y seleccionado las alternativas que mejor se adaptaban a lo que precisaba. Además, he trabajado con múltiples tecnologías y lenguajes de programación a lo largo del desarrollo del sistema, muchos de ellos nuevos para mí, lo que me ha permitido aumentar mi formación y ampliar mis conocimientos en Informática Gráfica e Inteligencia Artificial.

En cuanto a la gestión del proyecto se refiere, he aplicado los conocimientos teóricos y prácticos adquiridos a lo largo de la carrera. He podido experimentar lo importante que resulta hacer una buena planificación y el ir modificándola conforme los problemas aparecen.

Desde el punto de vista del resultado, no solo me siento satisfecho por haber conseguido los objetivos fijados, sino también por haber podido disfrutar del sistema desarrollado con mis familiares y amigos, viviendo la cara más amena y divertida del desarrollo de este tipo de sistemas.

Finalmente, este proyecto me ha brindado la oportunidad de conocer y trabajar con un gran grupo de personas y profesionales como son la gente del Grupo de Informática Gráfica Avanzada, con los que la relación ha sido siempre muy agradable.

Anexo A. Estudios previos relacionados

Este anexo complementa al capítulo 2 de la memoria principal, más concretamente al apartado 2.1 Análisis del problema. A lo largo de este documento se pretende introducir al lector en el desarrollo de agentes virtuales para dispositivos móviles, dando a conocer las características propias más relevantes de este tipo de agentes, describiendo varios de los agentes virtuales para dispositivos móviles ya existentes y explicando en detalle las plataformas de desarrollo utilizadas en trabajos anteriores.

A.1 Agentes virtuales sobre dispositivos móviles

En esta sección se presentan las características más relevantes que suelen poseer los agentes virtuales desarrollados para ser ejecutados sobre dispositivos móviles y se explican varios de los agentes de este tipo ya existentes.

A.1.1 Características de los agentes

A lo largo de los últimos veinte años se han venido desarrollando distintos sistemas que permiten la generación de entornos y agentes virtuales tridimensionales. Un objetivo fundamental al desarrollarlos es que dichos agentes soportasen una interacción multimodal y en tiempo real con el usuario. Ello ha dado lugar a sistemas compuestos por módulos muy pesados que precisan una gran capacidad de procesamiento. Es por esto que la gran mayoría son ejecutados sobre uno o varios ordenadores.

No obstante, la gran proliferación de dispositivos móviles en la última década, unido al continuo incremento de sus capacidades y recursos computacionales, ha hecho posible el desarrollo de agentes virtuales interactivos sobre estos dispositivos. Bien es cierto que las prestaciones de estos agentes virtuales son inferiores a los originales debido a las limitaciones intrínsecas de los dispositivos móviles.

Los agentes virtuales desarrollados para ser utilizados sobre dispositivos móviles, a pesar de ser usados en ámbitos muy diversos, poseen múltiples similitudes entre sí. En las siguientes líneas se detallan los distintos aspectos en los que se asemejan este tipo de agentes virtuales, a saber: el personaje, las animaciones y la interacción con el usuario.

Personaje

Todos los agentes virtuales suelen estar representados gráficamente por un personaje. Este personaje, que es el elemento principal de la interfaz gráfica del sistema, es el resultado de un proceso de diseño y modelado en el que el desarrollador define el tipo de personaje que mejor se adecúa, tanto a nivel computacional como estético, al nuevo sistema.

Por un lado, el desarrollador del sistema puede optar por utilizar imágenes bidimensionales para conformar gráficamente al nuevo agente virtual. De esta manera, se reducen tanto el espacio de memoria como el tiempo de procesado necesarios para generar al agente, lo que permite al sistema ser ejecutado por un mayor rango de dispositivos móviles. Sin embargo, disminuye el grado de realismo del agente virtual. Por otra parte, se puede optar por el uso de un personaje tridimensional para la representación gráfica del nuevo agente virtual. En este caso, es necesario que los dispositivos móviles sobre los que se ejecute el sistema dispongan de una cierta capacidad de procesado, ya que se precisa renderizar[2] la imagen continuamente. Cabe destacar que, si bien ambas opciones se han utilizado con éxito en el pasado, cada vez son más los sistemas que incorporan un personaje tridimensional debido al continuo desarrollo

[2] *Renderizar: (Del inglés rendering) proceso de generar una imagen o animación en 3D a partir de un modelo, usando para ello una aplicación de computador.*

de los dispositivos móviles. Es por ello que esta sección se va a centrar en el diseño y modelado de un agente virtual tridimensional.

A la hora de diseñar un agente virtual tridimensional para una aplicación interactiva, la primera decisión que se debe tomar es el nivel de detalle del propio agente, influyendo de forma determinante en dicha decisión la plataforma hacia la que va dirigida la aplicación. El nivel de detalle está directamente relacionado con el número de triángulos o polígonos de la malla que lo conforma. Los agentes virtuales con un número elevado de polígonos son utilizados para el pre-renderizado de películas o dibujos animados, mientras que para las animaciones en tiempo real de las aplicaciones interactivas es conveniente el uso de agentes con un número de polígonos reducido, de forma que el dispositivo sea capaz de procesar estos polígonos con fluidez. A su vez, a la hora de abordar el nivel de detalle de un agente virtual se debe considerar la configuración esqueletal del mismo, directamente relacionada con su animación, siendo conscientes de que un mayor número de huesos requiere un mayor tiempo de cálculo. En la Tabla A.1.1, obtenida de la Tesis Fin de Máster de Lars Zilmer [Zilmer, 2012], se muestran los rangos de polígonos y huesos recomendados para los agentes virtuales según la plataforma a la que estén dirigidos.

Plataforma	**Nº de Polígonos**	**Límite de Huesos**
Sistemas operativos de ordenadores (Windows, OS X, etc.)	1500 – 4000	Sin límite
Videoconsolas (Play Station3, Xbox 360, etc.)	5000 – 7000	Sin límite
Dispositivos móviles (iPhone, Android, etc.)	300 – 1500	30

Tabla A.1.1: *Número de polígonos y huesos recomendados para cada plataforma*

Los dos factores que marcan la diferencia entre las distintas plataformas que se muestran en la tabla son la capacidad de procesado y el motor gráfico. De cualquier modo, como es imposible dar una cifra exacta de polígonos o huesos que dichas plataformas puedan soportar, los valores que aparecen arriba se deben considerar como una aproximación orientativa. Además, es importante resaltar que la Tabla A.1.1 procede de una Tesis Fin de Máster del año 2012, por lo que los datos que se muestran en ella deben ser tenidos en cuenta desde la perspectiva de que los dispositivos móviles han evolucionado de forma muy notable en los últimos dos años.

Como se muestra en la Tabla A.1.1, la estructura ósea del modelo tiene que ser muy simple, puesto que no debe superar el límite de 30 huesos que se sugiere para las plataformas de los dispositivos móviles. En general, las zonas que precisan de un mayor número de huesos en los agentes virtuales son las manos y la cara, sin embargo, este número puede verse reducido drásticamente en función del tipo de animaciones utilizadas por el agente. En el caso de la cara, el descarte de las animaciones faciales en virtud de las animaciones corporales permite que un único hueso sea capaz de dotar de una rotación básica a la cabeza del agente. Por otro lado, con el fin de reducir el número de huesos necesarios en ambas manos, es posible configurar el modelo de forma que cada mano esté dotada de dos huesos, uno encargado del movimiento del dedo pulgar y otro del resto de dedos. Esta opción impide el control individual de los dedos, pero permite un control de la mano suficiente como para abrir y cerrar la misma, o llevar a cabo gestos de aprobación, saludo, etc.

Por otra parte, la Tabla A.1.1 sugiere que el modelo del agente virtual a desarrollar no debe superar los 1500 polígonos. Con este fin, elementos redondos o cilíndricos, como son la pierna o el brazo, son representados mediante formas poligonales más simples, reduciendo de forma considerable el número de polígonos necesarios para modelar dichos elementos. Además, en el caso de que las manos del agente virtual posean únicamente dos huesos cada una, carece de sentido el modelar todos los dedos de la mano puesto que no van a ser capaces de moverse individualmente. Por ello, se opta por modelar únicamente el dedo pulgar, modelando el resto como un solo dedo, lo que supone un ahorro considerable en el número de polígonos a utilizar. En este mismo sentido, la no utilización de animaciones faciales permite al desarrollador ahorrarse una gran cantidad de detalles en la cabeza del agente, pudiendo servirse de una cabeza con forma básica a la que las texturas añaden los detalles necesarios.

Además de decidir el nivel de detalle, otra de las tareas que debe abordar el desarrollador es la de determinar las características físicas del personaje que va a representar al nuevo agente virtual. Con este fin, el desarrollador debe definir aspectos como la condición, el género o la edad del personaje de forma que se adecúen al contexto de aplicación del sistema. A su vez, se debe seleccionar la representación del agente virtual que mejor se adapte a la funcionalidad del sistema. En este sentido, el desarrollador debe decidir entre utilizar una representación completa del cuerpo del agente virtual, usar tan sólo una representación de la cabeza del mismo, o servirse de representaciones intermedias en las que se muestran la parte superior del torso y la cabeza del agente. Esta elección, lejos de afectar únicamente al ámbito estético de la interfaz gráfica del sistema, influye también en el nivel de detalle que se puede aplicar a cada una de las partes del cuerpo del agente virtual.

Animación

Con el objetivo de dotar de un mayor realismo a los agentes virtuales, todos ellos suelen incorporar una serie de animaciones que les permite actuar y reaccionar de forma similar al comportamiento humano. Dichas animaciones, ya sean faciales, corporales o mixtas, deben ser definidas por el desarrollador en el proceso de animación del agente virtual. Las dos técnicas de animación más utilizadas en el campo de la informática gráfica son la captura de movimiento (*motion capture*) y la animación por planos clave (*key-frame animation*).

La técnica de captura de movimiento (*motion capture*) se basa en la grabación de los movimientos y gestos realizados por un actor, generalmente una persona o animal, y el traslado de éstos a un modelo digital, en este caso el agente virtual. Con esta técnica se logran movimientos dotados de un gran realismo y expresividad en períodos de tiempo no muy grandes; sin embargo, la exageración de ciertos movimientos o expresiones no puede ser lograda mediante esta técnica ya que el actor no es capaz de realizarlos. Además, otro de los inconvenientes de esta técnica es que si la estructura esqueletal del agente es demasiado simple, una gran cantidad de datos se pierden. Por todo ello, se descarta esta técnica de animación para el desarrollo de un agente virtual para dispositivos móviles.

En contraposición a la técnica anterior, en la animación por planos clave (*key-frame animation*) no existe actor alguno. Esta técnica consiste en que el diseñador del agente virtual fija una posición inicial y una final para cada movimiento de un elemento, generándose de forma automática las posiciones intermedias que permiten mover el elemento seleccionado de forma suave y continua desde la posición inicial hasta la posición final. Utilizando esta segunda técnica, los movimientos y expresiones pueden ser tan exagerados y particulares como sea posible en la imaginación del diseñador. Además, otra gran ventaja que ofrece esta técnica es que otorga un mayor control sobre las posiciones de la animación.

Por otro lado, existen sistemas dedicados a la animación de agentes virtuales que incorporan herramientas que guían al usuario a través de cada una de las fases del proceso de modelado de animaciones, como es el caso del sistema MAge-AniM [Chittaro et al, 2006] y su H-Animator [Nadalutti et al, 2006], el cual se procede a comentar brevemente a continuación.

En el sistema MAge-AniM, las animaciones se organizan en animaciones simples y animaciones compuestas. Las primeras son animaciones cortas, como realizar una acción (señalar, saltar…) o hacer un simple gesto (mover las manos, lenguaje de signos…). Por su parte, las animaciones compuestas están formadas por una secuencia de animaciones simples.

Para las animaciones simples, H-Animator utiliza una aproximación basada en la cinemática directa para cada fotograma. Esta aproximación divide la animación en una secuencia de fotogramas en los que el agente virtual adopta las posiciones principales de dicha animación. Para cada una de estas posiciones, el usuario debe establecer los valores de rotación que han de ser aplicados a cada una de las articulaciones del agente virtual y especificar el momento en el que la posición debe ser adoptada. Es por ello que el proceso de modelado por cinemática directa para cada fotograma suele dividirse en dos subtareas: la especificación de los valores del fotograma (*posing*) y el tiempo de reproducción del mismo (*timing*). Una vez terminadas, las animaciones simples son almacenadas en una base de datos.

Por otro lado, el usuario se sirve de las animaciones simples procedentes de las fases anteriores para crear las nuevas animaciones complejas (*joining*). En esta fase, las animaciones simples seleccionadas son obtenidas de la base de datos, mientras que las transiciones entre una animación y su siguiente son generadas directamente por el sistema. Para generar estas transiciones entre animaciones

simples, el sistema se sirve de la técnica de la interpolación lineal, que es fácilmente entendible por un usuario inexperto. Aunque es una solución simple, las transiciones generadas son en su gran mayoría realistas, presentando problemas únicamente en aquellos casos donde las animaciones a concatenar poseen alguna de las extremidades del agente en posiciones opuestas del cuerpo, lo que provoca que la interpolación lineal haga que dichas extremidades atraviesen el torso del agente durante la transición. De cualquier forma, es relativamente fácil para el usuario predecir si dos posiciones van a dar lugar a una transición problemática, ya que el resultado de la técnica de interpolación lineal es siempre el camino más corto entre la primera posición y la segunda. En caso de darse cualquier problema en una transición, basta con incluir una posición intermedia que evite el problema localizado.

Interacción

Una de las características más importantes de los agentes virtuales es que deben ser capaces de interactuar con el usuario. Esta interacción se puede llevar a cabo a través de diversos canales, a saber: escrito, oral y visual. A continuación, se detallan los elementos y módulos necesarios para permitir los distintos tipos de interacción con el usuario mencionados.

La interacción con el usuario a través del canal visual se limita, en la inmensa mayoría de los sistemas, a la reproducción de las animaciones que incorpora el agente virtual. Estas animaciones son las encargadas de dar a conocer las reacciones y los distintos estados emocionales del agente. Con el fin de gestionar la interacción visual con el usuario, el sistema debe disponer de un módulo que almacene el estado del agente virtual y lo actualice en función de la información proveniente del usuario, disparando la animación correspondiente a cada cambio de dicho estado.

Por otro lado, para la interacción con el usuario a través del canal oral son necesarios dos elementos: un reconocedor de discurso (*Automatic Speech Recognition*) y un sintetizador de voz a partir de texto (*Text To Speech*).

En el caso del ASR, una de las opciones más utilizadas para su implementación en dispositivos móviles es el uso de las clases públicas JAVA ***SpeechRecognizer*** [SpeechRecognizer web] y ***RecognizerIntent*** [RecognizerIntent web]. Ambas clases incorporan métodos ya implementados que gestionan el reconocimiento del discurso del usuario, de forma que el desarrollador únicamente debe inicializar la clase seleccionada y hacer uso de sus métodos para obtener la información comunicada oralmente por el usuario. El principal inconveniente de esta opción reside en la necesidad de disponer de conexión a internet durante el reconocimiento del discurso. Por otra parte, también existe la posibilidad de desarrollar módulos propios que, sin necesidad de conexión a internet, se encarguen del reconocimiento del discurso del usuario. Esta opción permite al desarrollador implementar un reconocedor de discurso mucho más específico para su sistema, aunque aumenta considerablemente tanto el tiempo de desarrollo del sistema como la carga de trabajo sobre el dispositivo móvil. Algunas de las bibliotecas utilizadas para el desarrollo de este tipo de módulos son SphinxBase y PocketSphinx [CMU Sphinx web].

Con respecto al TTS, una de las opciones más usadas por los desarrolladores de sistemas para dispositivos móviles es la de utilizar la clase JAVA ***TextTospeech*** [TextToSpeech web] para comunicarse con el sintetizador de voz nativo del sistema operativo del dispositivo. Dicha clase dispone de múltiples métodos ya implementados que facilitan la gestión de la síntesis del discurso del agente virtual, permitiendo al desarrollador modificar aspectos como el idioma, el volumen, el tono o la velocidad del discurso. Sin embargo, esta opción presenta el inconveniente de encontrarse sujeta al sistema de generación de voz del que disponga cada dispositivo, dando lugar a resultados muy distintos según el sintetizador de voz utilizado. Como alternativa a esta primera opción, cabe la posibilidad de desarrollar módulos *Text To Speech* específicos para el nuevo sistema a través de bibliotecas especializadas en la generación de discurso como eSpeak [eSpeak web].

Por último, la interacción con el usuario a través del canal escrito se fundamenta en la existencia de un área de texto y un panel de texto. En el caso del área de texto, es utilizada para que el usuario escriba la información que desea transmitir al agente virtual, por lo que suele ser lo suficientemente amplia para que quepan varias líneas de texto. Por su parte, sobre el panel se muestran las respuestas del agente virtual, impidiendo que el usuario pueda modificar en modo alguno el texto mostrado. Ambos elementos son parte de la interfaz gráfica del sistema, y su implementación depende de la plataforma de desarrollo y del lenguaje de programación utilizados.

A.1.2 Ejemplos de agentes virtuales sobre dispositivos móviles

Debido al continuo aumento tanto de la capacidad de procesado como de la memoria de los dispositivos móviles, empiezan a aparecer sistemas basados en un agente virtual desarrollados para ser ejecutados sobre estos dispositivos. A continuación, se describen brevemente algunos ejemplos de aplicación de estos sistemas basados en agentes virtuales.

Intérprete del lenguaje de signos

Una de las aplicaciones más relevantes en las que está presente el sistema Mage-AniM es un foro específico para personas con problemas auditivos [Buttussi et al, 2007], foro donde el agente virtual actúa como intérprete, transmitiendo mediante el lenguaje de signos los mensajes escritos por los usuarios. Este foro permite a cada uno de los usuarios componer de forma sencilla animaciones que expresen oraciones en lenguaje de signos. Los usuarios únicamente deben escribir las oraciones utilizando la gramática propia de la lengua de signos, y el mismo sistema es el encargado de recuperar de la base de datos las animaciones correspondientes a cada una de las palabras de la oración. En caso de no encontrar una animación que corresponda con una de las palabras utilizadas por el usuario, el sistema permite que modele una nueva animación para dicha palabra a través del H-Animator. De todos modos, para evitar que un solo usuario deba diseñar una gran cantidad de animaciones, MAge-AniM otorga a las comunidades del foro la posibilidad de compartir una base de datos de animaciones simples, reduciendo en gran medida el número de animaciones no disponibles. Además, se incorpora una función de deletreado dactilar en la que se muestra una palabra letra por letra a través de movimientos de dedos. Esta función es de gran utilidad para expresar palabras no asociadas a signos como nombres propios. Por lo tanto, la aplicación del lenguaje de signos es capaz de recuperar o crear una animación para todas las palabras existentes en una oración, de forma que cualquier usuario pueda generar la animación que exprese su oración en el lenguaje de los signos. En la imagen izquierda de la Figura 2.1 se puede observar a uno de los agentes virtuales utilizados para la representación de oraciones en el lenguaje de los signos.

Guía de entrenamiento

Una segunda aplicación de MAge-AniM es el desarrollo de un agente virtual que ayuda a los usuarios a realizar correctamente los ejercicios presentes en una determinada pista de fitness [Buttussi et al, 2006]. Una pista de fitness es una pista donde el usuario debe alternar la carrera continua con ejercicios estáticos. El usuario debe recorrer un circuito en el que se van alcanzando distintas estaciones de ejercicios. En dichas estaciones el usuario encuentra un aparato o herramienta con el que realizar una serie de ejercicios específicos, los cuales se encuentran explicados en unas placas existentes en las propias estaciones. En general, estas placas son difíciles de entender para usuarios inexpertos, siendo conveniente en estos casos el uso de agentes virtuales que muestren de forma visual cómo debe ser realizado cada uno de los ejercicios de la estación. En la imagen derecha de la Figura A.1.1 se muestra a un agente virtual explicando un ejercicio de anillas al usuario.

Figura A.1.1: *Ejemplos de aplicación de MAge-AniM. En la imagen de la izquierda aparece el agente virtual encargado de expresar en el lenguaje de los signos las oraciones que los usuarios escriben en el foro. En la imagen de la derecha se observa al agente virtual encargado de llevar a cabo las demostraciones de los ejercicios a realizar en cada una de las estaciones de ejercicios.*

Presentador virtual

Otro ejemplo de sistema basado en agentes virtuales para dispositivos móviles es el de un presentador virtual encargado de narrar los titulares de las noticias más importantes del momento al usuario. En este caso, la gestión del comportamiento del agente se lleva a cabo a través de una versión del Multimodal Presentation Markup Language específicamente diseñada para dispositivos móviles [Santi et al 2003].

La aplicación consta de un analizador MPML, un controlador del agente virtual y un gestor de diálogo. El teléfono genera los datos MPML a partir de la información obtenida de un servidor remoto, el cual contiene las noticias más importantes del día y las va actualizando cada 30 minutos. Los títulos de estas noticas son reformateados en un documento de edición MPML, el cual contiene tanto el contenido como los controles del agente virtual. La Figura A.1.2 muestra la reproducción de uno de estos documentos MPML sobre un teléfono móvil NTT Docomo Serie 540i.

El analizador MPML es un analizador XML con una pequeña huella de 3 KB de tamaño. Este tamaño de perfil tan pequeño se logra suponiendo que la información proveniente del MPML es correcta y sigue un formato XML bien formado. De esta forma, se reduce en gran medida el código de gestión de errores necesario en el proceso de análisis.

El controlador del agente virtual es el módulo que se encarga de manipular el motor gráfico 3D en el interior del dispositivo móvil. Por el momento, tanto el agente virtual como sus animaciones se tienen que crear de antemano, debiendo ser dichas animaciones previamente definidas y empaquetadas con el modelo de datos. Además, el número de animaciones debe ser reducido ya que los datos correspondientes a cada una de las animaciones pueden llegar a ocupar mucho espacio en el paquete JAVA (fichero JAR), pudiéndose sobrepasar el límite de espacio del que dispone el dispositivo. En el ejemplo de aplicación que se está explicando, el agente virtual cuenta con 15 animaciones distintas, animaciones que permiten mostrar comportamientos afectivos y realizar los movimientos oportunos para presentar los titulares de las noticias.

Por último, el gestor de diálogo se encarga del visualizado del discurso del agente virtual. Para ellos, se utiliza el texto deslizante, que ocupa solamente una línea en la pequeña pantalla del dispositivo móvil, con el fin de que el agente virtual disponga de más espacio.

Figura A.1.2: *A la derecha, dos capturas de pantalla donde se observa a la presentadora virtual narrando una determinada noticia al usuario desde sus distintos planos de enfoque. A la izquierda, representación de la ejecución del sistema sobre un teléfono móvil NTT Docomo Serie 540i*

Entrenador personal

Un nuevo ejemplo de sistema basado en agente virtual es el entrenador personal desarrollado en el proyecto Smarcos [Smarcos web], integrado dentro del proyecto europeo Artemis, el cual se ha llevado a cabo enteramente sobre la plataforma Elckerlyc, plataforma que se analiza en detalle en el siguiente sub-apartado de esta memoria. Este entrenador personal virtual insta a los usuarios a lograr y mantener un estilo de vida saludable, siendo sensible al contexto de las actividades diarias que realizan los usuarios a través de una gama de dispositivos interconectados.

Los dos grupos de interés en los que se centra el sistema son: trabajadores de oficina y personas que padecen diabetes de segundo tipo. En el caso de los primeros, el sistema informa a los usuarios acerca de su nivel de actividad física, nivel medido a través de un acelerómetro 3D. Por otro lado, a los usuarios con diabetes se les informa acerca de la toma de su medicación, la cual es controlada a través de un dispensador de pastillas inteligente.

Independientemente del grupo de interés al que vaya dirigido, el entrenador personal virtual mantiene un seguimiento continuo del usuario a través de los datos que va recibiendo de los dispositivos específicos de control. En el momento que recibe un aviso por parte de alguno de estos dispositivos, el sistema comienza a evaluar las normas de entrenamiento de las que dispone, transmitiendo al usuario la información oportuna en caso de que alguna de dichas normas de entrenamiento se cumpla. El sistema permite trasladar la información al usuario a través de los distintos modos de interacción que posee, siendo posible comunicar el mensaje a través de un mensaje de texto, un gráfico o el propio discurso del entrenador virtual.

Controlador del sistema domótico del hogar

Como último ejemplo de aplicación se presenta a un agente virtual que permite controlar el sistema domótico de un domicilio [Santos-Pérez et al, 2013]. Este sistema basado en un agente virtual se ha llevado a cabo sobre una plataforma desarrollada por investigadores de la Universidad de Málaga, plataforma que se explica de forma detallada en el siguiente sub-apartado de esta memoria. En dicho sistema, el usuario puede controlar el cierre/apertura de puertas, el encendido/apagado de luces, la subida/bajada de persianas y la temperatura de la casa a través de la interfaz gráfica o de una conversación con el agente virtual.

A.2 Plataformas de desarrollo de agentes virtuales para dispositivos móviles

En esta sección del anexo se explica de forma detallada dos plataformas de desarrollo de agentes virtuales para dispositivos móviles ya existentes y que han servido de base a este trabajo. En las siguientes líneas se dan a conocer las principales características de ambas plataformas y se describen los principales módulos que incorporan.

A.2.1 Elckerlyc

Elckerlyc [Klaasen et al, 2012] es una plataforma, basada en modelos, que sirve para la especificación y animación de agentes virtuales que soporten una interacción multimodal en tiempo real. Su uso está centrado en dispositivos móviles como *tablets* y *smartphones*.

Esta plataforma permite desarrollar sistemas basados en agentes virtuales capaces de interactuar con el usuario a través de diferentes canales, a saber: oral, textual y visual.

Comunicación Oral

Con el fin de permitir a los agentes comunicarse oralmente con el usuario, la plataforma Elckerlyc incorpora el módulo SpeechEngine, módulo que se encarga de expresar de forma oral el

discurso seleccionado para el agente virtual a través del altavoz del dispositivo. En las versiones anteriores, dirigidas al uso sobre un ordenador, el motor de conversión de texto a voz (*Text To Speech – TTS*) existente en dicho módulo presenta múltiples dependencias del sistema operativo sobre el que trabaja, no siendo posible su reutilización sobre Android sin realizar cambios significativos. Por ello, en esta nueva versión se opta por utilizar el sistema TTS propio de Android, adaptando para ello el módulo SpeechEngine con el fin de cargar e inicializar dicho sistema. A pesar de que el uso del sistema TTS nativo de Android implica el ahorro de los costes derivados de portar a dicho sistema operativo el motor TTS de versiones anteriores de Elckerlyc, hay que tener en cuenta los distintos inconvenientes que la utilización de este TTS conlleva. El principal inconveniente del sistema TTS de Android es que no aporta ninguna clase de información temporal, esto es, el desarrollador no puede saber con exactitud el momento en el que cada una de las palabras del discurso va a ser reproducida. Este hecho impide que el planificador que incorpora Elckerlyc utilice puntos de sincronización en las expresiones del discurso, dificultando la sincronización entre la pronunciación de determinadas palabras y alguna actitud o comportamiento específicos del agente ante las mismas. Además, el sistema TTS de Android tampoco ofrece información para la reproducción de los visemas (representación visual de los fonemas), por lo que la sincronización labial (*lipsync*) se hace imposible en estos dispositivos.

Comunicación Escrita

Debido a que la plataforma Elckerlyc está concebida para su utilización sobre dispositivos móviles, la probabilidad de que el usuario encuentre problemas a la hora de escuchar el discurso del agente virtual es considerablemente alta. Estas dificultades en la percepción del discurso del agente tienen como causas más comunes el ruido ambiental, un volumen máximo no muy elevado del dispositivo o un mal funcionamientos de los altavoces del mismo. Por ello, es necesaria la existencia de una vía alternativa de comunicación que permita al usuario interactuar con el agente virtual en dichas situaciones Esta segunda vía de comunicación es la escrita.

El modo de interacción escrita se lleva a cabo a través del módulo TextSpeechEngine, módulo encargado de recibir el discurso proveniente del SpeechEngine y reproducirlo en el interior del cuadro de texto con formato PNG, compatible con el entorno gráfico de Android, existente en la interfaz del sistema. Además, este módulo permite combinar ambos modos de interacción, oral y escrita, a través de la sincronización por expresiones del cuadro de texto con el TTS.

Comunicación Visual

El canal visual es el modo de interacción más característico de esta plataforma. En primer lugar, en Elckerlyc se considera que el desarrollo de un agente virtual tridimensional, dotado de una cinemática completa y con un aspecto humano realista no es viable por varias razones. La principal razón es la elevada capacidad de procesado que este tipo de agente y entorno virtuales precisan, la cual reduciría de forma notable el número potencial de dispositivos móviles a los que iría dirigida la aplicación desarrollada, y consumiría rápidamente las baterías de aquellos dispositivos que sí soportasen la demanda de procesado de la misma. Además, se considera poco práctico la reproducción de una escena en la que el agente virtual se represente a cuerpo completo, ya que el reducido tamaño de la pantalla de los dispositivos a los que va dirigido la aplicación haría casi inapreciables las expresiones del agente virtual. Por ello, con el fin de evitar estos problemas, Elckerlyc presenta un nuevo motor gráfico, el Picture Engine.

Picture Engine es un realizador gráfico sin grandes requerimientos computacionales, ejecutable sobre la plataforma Android, que se sirve de una colección de imágenes bidimensionales para la conformación gráfica del agente virtual. Aunque el modelado 2D del agente virtual conlleva algunas limitaciones, como la disminución del realismo, presenta a su vez ventajas especialmente notorias en este tipo de dispositivos. En primer lugar, supone una gran reducción del tiempo de procesado y espacio de memoria requeridos por la aplicación desarrollada, así como disminuye el consumo de batería de la misma. Además, soporta una gran variedad de técnicas de diseño para los agentes virtuales, siendo posible diseñarlos a partir de una figura de dibujos animados, de imágenes 3d pre-renderizadas o, incluso, a partir de fotografías de una persona real.

Con el fin de dotar de cierto dinamismo al agente virtual, el cual es generado a partir de una colección de imágenes bidimensionales estáticas, Picture Engine utiliza una aproximación basada en capas. Esta aproximación posibilita que distintas partes del cuerpo del agente virtual se localicen en capas

diferentes, permitiendo así que cada una de dichas partes pueda encontrarse en un estado distinto. Generalmente, la primera capa contiene al agente virtual en estado base, mostrándose al usuario dicha capa siempre que el agente virtual se encuentre en estado neutro o pasivo. Esto implica que, a pesar de que diversas partes del agente puedan localizarse en capas distintas, todas ellas están presentes en la primera capa. La principal ventaja de esta aproximación es que cada una de las partes del agente virtual se puede manipular de forma independiente, combinando posteriormente todas para generar diferentes expresiones. Además, permite al agente realizar varias tareas a la vez, como pestañear y hablar al mismo tiempo. No obstante, el sistema de capas también presenta ciertas limitaciones. El mayor inconveniente del uso de esta aproximación procede del carácter estático del agente virtual en la primera capa, necesario para reproducir las distintas partes que se encuentran en capas superiores, pero que dificulta en gran medida cualquier movimiento del agente virtual entero como el balanceo del peso. De todas formas, como Picture Engine está diseñado para ser utilizado en las pequeñas pantallas de los dispositivos móviles, los agentes virtuales suelen constar únicamente de una cabeza que ocupa prácticamente la totalidad de la pantalla, por lo que los movimientos de todo el cuerpo del agente carecen de sentido.

Aunque en la mayor parte de los casos las imágenes estáticas pueden bastar a la hora de representar expresiones, existen otros casos en los que el agente virtual debe realizar algún movimiento para dotar de mayor realismo a la expresión. Con el fin de permitir la especificación de animaciones, Picture Engine se sirve de ficheros con formato XML. En dichos ficheros se lista un número determinado de imágenes junto al tiempo de reproducción de cada una de ellas, quedando así definida la animación. Además, estos ficheros XML permiten incluir información de sincronización en la especificación de las animaciones, posibilitando la inclusión de puntos de sincronización entre dos frames de la animación.

Finalmente, como ya se ha comentado en el apartado de comunicación oral, la falta de información proveniente del sistema *Text To Speech* de Android hace imposible la sincronización labial (*lipsync*). Es por ello que Picture Engine otorga una opción rudimentaria de lypsinc al desarrollador, en donde éste último especifica una única animación que es reproducida siempre que el agente virtual se encuentre hablando, repitiéndose dicha animación el número de veces que se adecúe más a la duración del discurso del agente.

A.2.2 Plataforma de desarrollo de interfaces basadas en el uso de agentes virtuales para Android de la Universidad de Málaga

En el artículo elaborado por investigadores de la Escuela de Ingeniería de Telecomunicaciones de la Universidad de Málaga [Santos-Pérez et al, 2013], se presenta una plataforma que permite el desarrollo de interfaces basadas en agentes virtuales para dispositivos móviles Android.

La arquitectura de esta plataforma sigue un diseño modular, donde cada componente se puede modificar sin afectar al resto. Los módulos que componen esta plataforma son: Voice Activity Detector, Automatic Speech Recognition, Conversational Engine, Control Interface, Text To speech y Virtual Head Animation. Todos estos módulos están basados en bibliotecas de libre de distribución, exceptuando el Conversational Engine que está implementado como un módulo Python.

Los sistemas basados en agentes virtuales desarrollados con esta plataforma son capaces de interactuar con el usuario a través de dos canales, el oral y el visual.

Comunicación Oral

En esta modalidad de interacción intervienen un gran número de módulos del sistema, a saber: Voice Activity Detector, Automatic Speech Recognition, Conversational Engine y Text To Speech. A continuación, se describe tanto el cometido como la implementación de estos módulos, apareciendo en el orden en el que cada uno de los módulos es utilizado por el sistema.

El primer módulo que interviene es el Voice Activity Detector (VAD), que tiene como principal cometido el discriminar los fragmentos de audio que contengan voz del usuario frente a los solamente contengan ruido. De este modo, el VAD permite la segmentación del discurso del usuario en expresiones, filtrando además el audio grabado por el micrófono del dispositivo. Una vez leído y filtrado el audio captado por el micrófono de dispositivo, este módulo envía los fragmentos de audio que contienen el

discurso del usuario al Automatic Speech Recognition. La implementación del VAD se basa en la biblioteca SphinxBase, aunque ha sido modificada ligeramente para que sea compatible con las bibliotecas de audio OpenSL ES presentes en Android.

El Automatic Speech Recognition (ASR) es el modulo encargado de convertir el discurso del usuario a texto. Su cometido es tomar como entrada los fragmentos de audio con el discurso del usuario provenientes del Voice Activity Detector y, tras el proceso de conversión, enviar el texto resultante al Conversational Engine. En la plataforma propuesta, el módulo ASR está basado en la biblioteca de reconocimiento de discurso PocketSphinx, aunque se han realizado ligeros cambios con el fin de disminuir el tiempo de respuesta del módulo.

El tercer módulo que interviene en el proceso de gestión de la comunicación oral con el agente virtual es el Conversational Engine. Este módulo es el encargado de extraer el significado de las distintas palabras y/o expresiones reconocidas por el ASR, gestionar el flujo de diálogo y generar respuestas en base a dicho significado de las expresiones, el historial del diálogo y el estado actual de la conversación. Este módulo está basado en PyAIML, un *chatbot* de AIML. AIML (Artificial Intelligence Markup Language) es una extensión de XML que provee de reducción simbólica, recursividad, conocimiento del contexto y gestión del historial del diálogo, lo que permite al sistema entender en un mayor grado las expresiones del usuario y generar respuestas más concretas y adecuadas.

Finalmente interviene el Text To Speech, subsistema que se encarga de la generación de voz sintetizada a partir de las respuestas en forma de cadena de texto provenientes del Conversational Engine. La implementación de este módulo está basada en la biblioteca eSpeak.

Comunicación Visual

En la plataforma propuesta, la interacción visual del sistema con el usuario depende de dos módulos, el Control Interface y el Virtual Head Animation. A continuación, se describe tanto el cometido como la implementación de ambos módulos.

El Control Interface es el módulo encargado de dotar funcionalidad a todos los elementos presentes en la interfaz gráfica del sistema. Adicionalmente, puede ser el módulo que gestione la conversión de los comandos pronunciados por el usuario a un formato que sea entendible para la aplicación. En todo caso, depende del dominio específico del sistema, y por tanto, debe ser implementado o adaptado para cada nueva aplicación que se desarrolle.

Por otro lado, el Virtual Head Animation es el módulo encargado de generar las expresiones faciales y los visemas del agente virtual. Este módulo recibe como entradas tanto la información acerca del estado de ánimo del agente, procedente del Conversational Engine, como la lista de los fonemas y sus respectivas duraciones provenientes del Text To Speech. A través del procesado de esta información, el Virtual Head Animation es capaz de reproducir los visemas y las expresiones faciales de forma sincronizada con el discurso del agente virtual. El motor de renderizado utilizado es Ogre 3D.

Anexo B. Proceso de ingeniería del software

En este anexo se explica el modelo de proceso seguido a lo largo de este Proyecto Fin de Carrera junto con sus ventajas e inconvenientes.

B.1 Modelo de proceso

Se selecciona el modelo de proceso incremental, que viene descrito en [Sommerville, 2005], el cual combina las ventajas del modelo en cascada y del modelo evolutivo.

Este modelo fue propuesto por Harlan Mills en el año 1980, surgiendo como forma de reducir la repetición de trabajo en el proceso de desarrollo y permitiendo retrasar en cierta medida la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema.

Al inicio de un proceso de desarrollo incremental, los clientes o los usuarios identifican, a grandes rasgos, las funcionalidades y servicios que debe proporcionar el sistema. Para ello, se confecciona un primer listado de requisitos funcionales, encargándose el cliente de definir las prioridades de las distintas funcionalidades a desarrollar en el sistema. En este sentido, deben poseer una mayor prioridad las funcionalidades que precise con mayor urgencia el cliente, los requerimientos más básicos y aquellos requerimientos más complejos que impliquen un mayor grado de riesgo. Tras haber definido estas prioridades, se procede a confeccionar un plan de incrementos, donde cada incremento proporciona un subconjunto de las funcionalidades del sistema. La asignación de las funcionalidades a implementar en cada incremento depende de las prioridades establecidas anteriormente. Una vez que los incrementos se han identificado, se definen en detalle los requerimientos para los servicios que deben ser entregados en el primer incremento y se procede a desarrollar dicho incremento (Figura B.1.1). De esta forma, se desarrolla una aplicación ejecutable con parte de la funcionalidad del sistema (primera versión), la cual se entrega al usuario para que trabaje y experimente con ella, ayudando al cliente a clarificar sus requerimientos para los incrementos posteriores y permitiendo al desarrollador conocer las recomendaciones del cliente para mejorar el producto. Estas mejoras deberán esperar a ser integradas en la siguiente versión junto con los demás requerimientos que no fueron tomados en cuenta en la versión anterior. Tan pronto como se completan los nuevos incrementos, se integran en los ya existentes, de tal forma que cada vez se entrega un producto con mayor funcionalidad que el previo, repitiendo este proceso hasta alcanzar el completo desarrollo del sistema. Cabe destacar que con el modelo incremental se entrega un producto parcial pero completamente operacional en cada incremento.

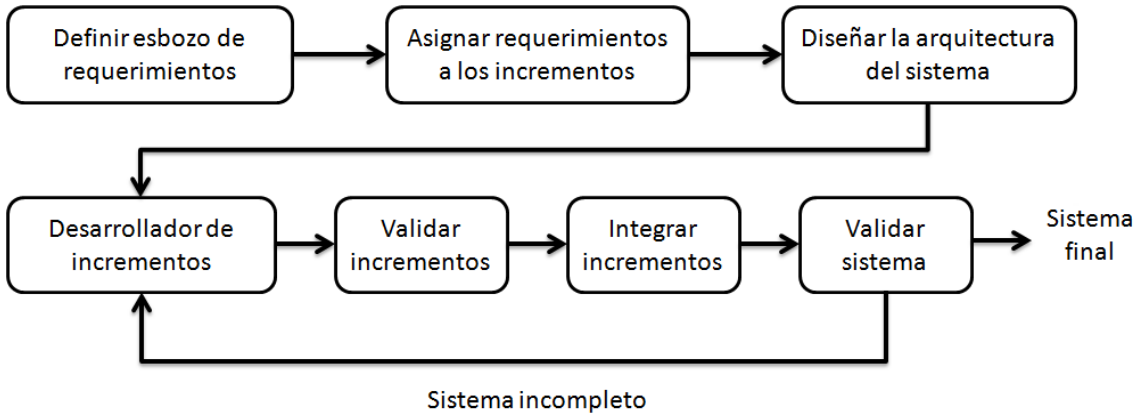


Figura B.1.1: *Esquema de entrega en el modelo incremental [Sommerville, 2005]*

El incremental es un modelo de tipo evolutivo, de filosofía iterativa, basado en varios ciclos Cascada Realimentados aplicados repetidamente. En la Figura B.1.2 se muestra el esquema del modelo de ciclo de vida Iterativo Incremental con sus actividades genéricas asociadas. En dicha figura se observa que cada ciclo cascada es aplicado para la obtención de un incremento, integrándose en última instancia estos incrementos para obtener el producto final completo. Cada incremento es un ciclo Cascada Realimentado, aunque, por simplicidad, en la Figura B.1.2 se muestra como secuencial puro.

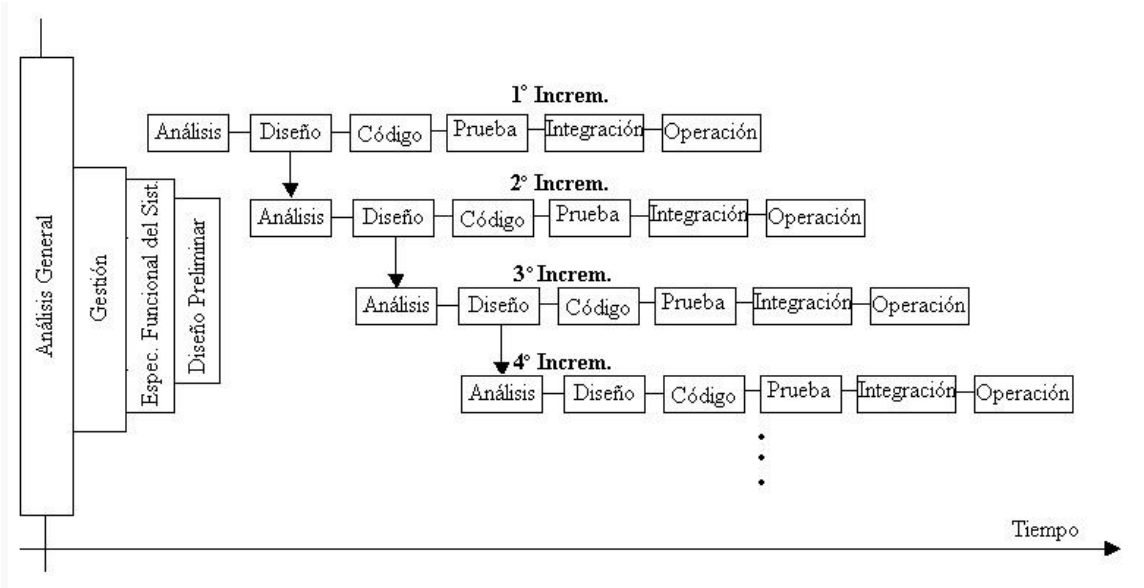


Figura B.1.2: *Modelo iterativo incremental para el ciclo de vida del software*

También se observa que existen actividades de desarrollo, pertenecientes a distintos incrementos, que son realizadas en paralelo o concurrentemente. Así por ejemplo, en la Figura B.1.2, mientras se realiza el diseño del primer incremento ya se está realizando el análisis del segundo. Cabe destacar que esta figura es solamente esquemática, por lo que no es necesario iniciar un nuevo incremento durante la fase de diseño del anterior, si no que es posible comenzar con el análisis de un nuevo incremento en cualquier momento de la etapa previa. El momento de inicio de cada incremento depende de varios factores: tipo de sistema; independencia o dependencia entre incrementos (dos de ellos totalmente independientes pueden ser fácilmente iniciados al mismo tiempo si se dispone de personal suficiente); capacidad y cantidad de profesionales involucrados en el desarrollo; etc. Cada incremento concluye con la actividad de «operación y mantenimiento» (indicada como «Operación» en la Figura B.1.2), que es donde se produce la entrega del producto parcial al cliente. En general cada incremento se construye sobre aquel que ya fue entregado.

B.1.1 Ventajas del modelo

El modelo de proceso incremental reduce el tiempo de desarrollo inicial, ya que se implementa parcialmente la funcionalidad del sistema. De esta forma, se evitan fases de desarrollo de larga duración a lo largo del proyecto, entregando al usuario versiones simplificadas pero operativas del sistema con cierta frecuencia. En primer lugar se llevan a cabo las partes más prioritarias del sistema, integrando sobre estas los incrementos posteriores. De esta manera se realizan un mayor número de pruebas sobre los servicios con una alta prioridad para el usuario, aumentando la posibilidad de encontrar y solucionar fallos de funcionamiento en las partes más importantes del sistema.

Además, el modelo de proceso incremental involucra en mayor medida al usuario en el proceso de desarrollo. Por un lado, la entrega temprana de partes operativas del sistema evita que el usuario tenga que esperar al final del proyecto para sacar provecho del sistema, mientras que, por otra parte, el usuario puede utilizar los incrementos iniciales como prototipos, obteniendo experiencia para definir los requerimientos de los incrementos posteriores.

Finalmente, a través del modelo de proceso incremental resulta más sencillo llevar a cabo cambios en el sistema debido a la acotación en el tamaño de los incrementos. A su vez, este modelo de proceso reduce el riesgo a un fallo total del proyecto.

B.1.2 Desventajas del modelo

Sin embargo, el modelo de proceso incremental presenta algunas dificultades técnicas que se proceden a comentar.

En primer lugar, este modelo precisa gestores experimentados puesto que se requiere mucha planificación, tanto administrativa como técnica. En este sentido, los incrementos deben planificarse de forma que sean relativamente pequeños (no más de 20000 líneas de código) y que cada uno incorpore una nueva funcionalidad al sistema. Además, se deben definir una serie de metas claras que permitan conocer el estado del proyecto.

Por otra parte, debido a que los requerimientos no se definen en detalle hasta que un incremento se implementa, resulta difícil identificar los recursos comunes que precisan todos los incrementos, y en consecuencia, determinar el coste total del proyecto.

Finalmente, resulta complicado aplicar este modelo de proceso a los sistemas transaccionales, ya que tienden a ser integrados y operar como un todo

B.1.3 Conclusión

A pesar de las desventajas que pueda presentar el modelo de proceso incremental, éstas no afectan de forma relevante al desarrollo de este Proyecto Fin de Carrera, convirtiéndolo sus ventajas en la mejor opción de modelo de proceso software. Esto se debe a que a lo largo del desarrollo del sistema se van a implementar diferentes incrementos que van aumentar de forma gradual la funcionalidad del mismo. Cada uno de estos incrementos deberá ser sometido a pruebas con el cliente (en este caso el propio desarrollador y las tutoras del proyecto) para seguir definiendo el resto de partes y realizar las modificaciones pertinentes en cada fase.

B.2 Aplicación del modelo de proceso al sistema

Debido a que el nuevo sistema pretende interactuar con el usuario a través de diversos canales, se opta por desarrollar varios incrementos en la funcionalidad del mismo, de manera que el cometido de cada uno de estos incrementos sea dotar al sistema de un modo de interacción distinto a los anteriores. En este sentido, con la ayuda de un primer prototipo funcional de la interfaz gráfica del sistema (descrito en el Anexo E), se llevan a cabo tres incrementos iniciales en el sistema:

- Primer incremento: con el fin de que la interacción con el usuario sea lo más natural posible, se considera prioritario desarrollar inicialmente los módulos necesarios para permitir al sistema comunicarse oralmente con el usuario (explicados en detalle en el apartado 3.1 de la memoria principal). Por tanto, en este primer incremento se incorporan al sistema los módulos encargados de escuchar el discurso del usuario y de reproducir a través del altavoz los mensajes del agente virtual.

- Segundo incremento: en esta fase se desarrollan los módulos necesarios para dotar al sistema de la capacidad de comunicarse de forma escrita con el usuario (explicados en detalle en el apartado 3.2 de la memoria principal). En este sentido, se incorporan al sistema los módulos encargados de leer los mensajes escritos por el usuario y de escribir por pantalla las respuestas del agente virtual.

- Tercer incremento: tras haber dotado al sistema de la capacidad de interactuar tanto oral como textualmente con el usuario, se considera necesario desarrollar un módulo motor (explicado en

detalle en el apartado 3.5 de la memoria principal) que se encargue de gestionar las animaciones que incorpora el agente virtual, permitiéndole de este modo interactuar visualmente.

Una vez el sistema ya es capaz de interactuar con el usuario por los canales oral, escrito y visual, se plantean dos nuevos incrementos con el fin de incluir aspectos emocionales en la comunicación agente-usuario y dotar de cierta capacidad conversacional al agente virtual.

- Cuarto incremento: en esta fase se opta por dotar de cierto carácter emocional a los distintos modos de interacción del agente virtual. En este sentido, se generan una serie de voces emocionales (cuyos procesos de generación y evaluación se describen en los capítulos 4 y 5 de la memoria principal) y se seleccionan una serie de colores para representar los distintos estados emocionales del agente a través de los mensajes de texto (explicados en el apartado 3.3.3 de la memoria principal).

- Quinto incremento: finalmente, se considera acertado otorgar al agente virtual cierta capacidad conversacional, de forma que sea capaz de mantener un diálogo sencillo con el usuario. Para ello se desarrolla un módulo gestor de diálogo (descrito en detalle en el apartado 3.5 de la memoria principal).

Como se ha comentado anteriormente, las pruebas de cada uno de los incrementos desarrollados las llevan a cabo el propio desarrollador junto a las directoras del proyecto y consisten (exceptuando el caso del cuarto incremento, cuyas pruebas fueron mucho más exhaustivas) en corroborar el correcto funcionamiento de dichos incrementos.

Anexo C. Documentación del desarrollo del software

Este anexo complementa a los capítulos 2 y 3 de la memoria principal. En primer lugar, se explica la metodología de análisis, en la que se definen los requisitos y los diagramas utilizados, así como los pasos seguidos para desarrollar cada modelo de descripción del sistema.

C.1 Metodología de análisis

En esta sección se explica la metodología de análisis del problema, tras la cual se definen los requisitos y los diagramas utilizados, así como los pasos seguidos para desarrollar cada modelo de descripción del sistema. La metodología de análisis seleccionada es OMT [Rumbaugh, 1996]. Según esta metodología, la fase de análisis se realiza en tres pasos:

- Modelo de objetos: Describe la estructura estática de los objetos del sistema (relaciones, atributos y operaciones), el cual se representa mediante diagramas de objetos.

- Modelo dinámico: Describe los aspectos de un sistema estudiando la organización de estados y la secuencia de operaciones mediante diagramas de estado.

- Modelo funcional: Describe las transformaciones que pueden sufrir los datos dentro del sistema, representado gráficamente mediante diagramas de flujo de datos.

Se ha creído oportuno añadir a estas tres etapas la fase de análisis de requisitos, propia de la metodología UML, pero que permite estudiar más fondo las necesidades del sistema a desarrollar en este Proyecto Fin de Carrera.

C.1.1 Análisis de requisitos

Como se ha mencionado en el primer capítulo de la memoria principal, el objetivo que persigue este Proyecto Fin de Carrera es el desarrollo de un sistema basado en un agente virtual, para dispositivos móviles Android, que permita una interacción multimodal con el usuario lo más natural posible.

En esta sección se muestra la lista de requisitos, tanto funcionales como no funcionales, que debe cumplir el sistema a desarrollar.

Requisitos funcionales:

RF1 – El sistema debe permitir al usuario comunicarse oralmente con el agente virtual. Para ello, debe ser capaz tanto de reconocer el discurso del usuario, captado por el micrófono del dispositivo, como de expresarse de forma oral a través del altavoz del mismo.

RF2 – Se debe otorgar al usuario la posibilidad de comunicarse de forma escrita con el agente virtual, permitiendo así la interacción cuando la comunicación oral no sea posible. Para ello, se debe desarrollar un mecanismo de comunicación escrita entre ambos a través de campos de texto.

RF3 – El sistema debe permitir al usuario cambiar la modalidad de la interacción entre él y el agente virtual a través de su interfaz gráfica.

RF4 – El sistema debe ser capaz de gestionar de forma adecuada las animaciones, tanto faciales como corporales, que incorpora el agente virtual.

RF5 – El sistema debe permitir al agente virtual expresar su estado emocional. Dicho estado emocional debe ser apreciable a través de los gestos del agente, así como a través del tono de voz y velocidad de discurso utilizados.

RF6 – El sistema debe permitir al agente virtual variar su estado emocional en función de la conversación que mantenga con el usuario.

RF7 – El sistema debe ser capaz de gestionar el diálogo entre el agente virtual y el usuario. Para ello, debe analizar la información proveniente del usuario, vía texto o vía discurso, gestionar dicha información para buscar una respuesta adecuada y expresarla.

Requisitos no funcionales:

RNF1 – Usar la plataforma de programación Eclipse Juno para el desarrollo de plugins que comuniquen al sistema con los distintos servicios Android que se precisen.

RNF2 – Utilizar el motor de renderizado Unity 3D para el desarrollo gráfico tanto del agente virtual como de la interfaz, así como para la integración de los distintos módulos que conforman el sistema.

RNF3 – Usar el lenguaje de programación JAVA para la implementación de los plugins a desarrollar para el sistema.

RNF4 – Servirse del lenguaje de programación C# para generar los *scripts* que gestionen la funcionalidad tanto del agente virtual como de su interfaz gráfica.

RNF5 – Trabajar sobre el Smartphone Samsung Galaxy S3 del que dispone el grupo de trabajo**.**

C.1.2 Modelo de objetos

El modelo de objetos define la estructura estática de los objetos del sistema y proporciona el entorno en el que situar los modelos dinámico y funcional.

En esta sección se presenta el diagrama de clases, un diccionario de datos y una descripción de los métodos contenidos en el diagrama de clases. En este sentido, inicialmente se hace un estudio de las principales clases del sistema, mostrando las relaciones que existen entre todas ellas, y una vez construido el diagrama de clases, se elabora el diccionario de datos y se definen los métodos existentes en dicho diagrama.

a) Diagrama de Clases

A continuación, en la Figura C.1.1, se muestra el diagrama de clases donde las flechas indican el sentido en que se hacen las llamadas a los distintos métodos.

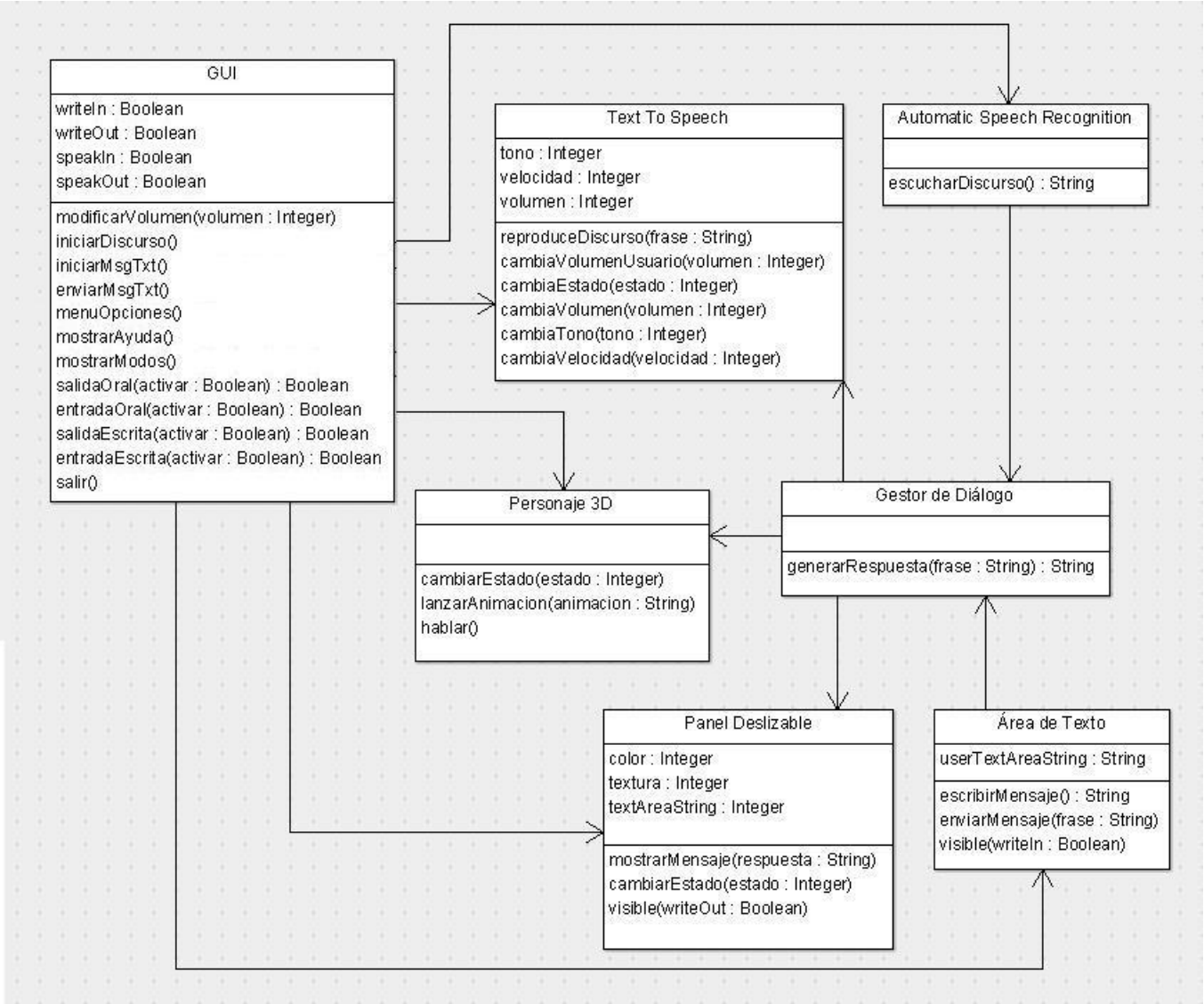


Figura C.1.1: *Diagrama de clases del sistema a desarrollar*

b) Diccionario de Datos

El diccionario de datos contiene la descripción de las clases presentadas en el diagrama anterior, haciendo referencia al alcance de cada una de ellas dentro del problema y cualquier suposición o restricción relativa a su uso.

GUI	Gestiona la interacción del usuario con el sistema. Es la encargada de activar y desactivar los distintos modos de interacción, de modificar el volumen y de cerrar el sistema.
Personaje 3D	Gestiona la comunicación visual del agente virtual. Su función es gestionar la reproducción de las distintas animaciones que incorpora el personaje tridimensional utilizado
Text To Speech	Gestiona todos los aspectos relacionados con la reproducción oral de las respuestas del agente virtual. Su función principal es convertir las respuestas, en forma de cadenas de texto, provenientes del Gestor de Diálogo en discurso.

Automatic Speech Recognition	Gestiona el reconocimiento del discurso del usuario y su posterior envío al Gestor de Diálogo en forma de cadena de texto.
Panel Deslizable	Gestiona todos los aspectos relacionados con la transmisión por escrito de las respuestas del agente virtual al usuario
Área de Texto	Gestiona la escritura de los mensajes del usuario y su posterior envío al Gestor de Diálogo
Gestor de Diálogo	Gestiona la interpretación y búsqueda de una respuesta adecuada a cada uno de los mensajes provenientes del usuario

c) Definición de Métodos

En este apartado se explican los principales métodos de las clases que han sido creadas en este proyecto. Dichas clases, junto a sus respectivos métodos, han sido mostradas en el diagrama de clases anterior.

GUI

```
modificarVolumen (Integer volumen);
// establece el volumen del sistema al valor indicado por el
// usuario a través de los botones físico o la barra deslizante
// situada en la esquina superior derecha de la interfaz gráfica
// Este valor viene indicado por el parámetro "volumen"

iniciarDiscurso();
// inicia el proceso de escucha del discurso del usuario por
// parte del ASR

iniciarMsgTxt();
// inicia el proceso de escritura en el Área de Texto

enviarMsgTxt();
// envía al Gestor de Diálogo el mensaje escrito por el usuario
// en el Área de Texto

menuOpciones();
// muestra por pantalla el menú de opciones del sistema

mostrarAyuda();
// muestra por pantalla el manual de ayuda del sistema

mostrarModos();
// muestra por pantalla el menú de modos de interacción. En este
// menú, el usuario puede seleccionar los canales a través de
// los cuales desea interactuar con el agente virtual

salidaOral(activar: Boolean):Boolean;
// activa/desactiva la reproducción oral de las respuestas del
// agente virtual en función del valor del parámetro "activar".
// Este método devuelve el valor que debe adoptar el atributo
// "speakOut"
```

```
entradaOral(activar: Boolean):Boolean;
// activa/desactiva el reconocedor del discurso del sistema en
// función del valor del parámetro "activar".
// Este método devuelve el valor que debe adoptar el atributo
// "speakIn"

salidaEscrita(activar: Boolean):Boolean;
// muestra/oculta el panel deslizable donde se escriben las
// respuestas del agente virtual en función del valor del
// parámetro "activar".
// Este método devuelve el valor que debe adoptar el atributo
// "writeOut"


entradaEscrita(activar: Boolean):Boolean;
// muestra/oculta el área de texto utilizado por el usuario para
// comunicarse de forma escrita con el agente virtual en función
// del valor del parámetro "activar"
// Este método devuelve el valor que debe adoptar el atributo
// "writeIn"

salir();
// cierra el sistema
```

Personaje 3D

```
cambiarEstado(estado: Integer);
// el personaje tridimensional que representa al agente virtual
// pasa a estar en el estado emocional indicado por el parámetro
// "estado"

lanzarAnimacion(animación:String);
// inicia la reproducción de la animación indicada por el
// parámetro "animación"

hablar();
// reproduce la animación correspondiente al habla de forma
// combinada con cualquier animación en reproducción
```

Text To Speech

```
reproduceDiscurso(frase: String);
// reproduce oralmente el mensaje de texto contenido en el
// parámetro "frase", el cual proviene del Gestor de Diálogo

cambiaVolumenUsuario(volumen: Integer);
// establece el volumen del sistema al valor indicado por el
// usuario a través de los botones físico o la barra deslizante
// situada en la esquina superior derecha de la interfaz gráfica
// Este valor viene indicado por el parámetro "volumen"

cambiaEstado(estado:Integer);
// establece la voz emocional correspondiente al estado
// emocional indicado por el parámetro "estado" como voz para
// reproducir los distintos mensajes del agente virtual

cambiaVolumen(volumen:Integer);
// establece el volumen del discurso al valor indicado por el
// parámetro "volumen"
```

```
cambiaTono(tono:Integer);
// establece el tono del discurso al valor indicado por el
// parámetro "tono"

cambiaVelocidad(velocidad:Integer);
// establece la velocidad del discurso al valor indicado por el
// parámetro "velocidad"
```

Automatic Speech Recognition

```
escucharDiscurso():String;
// inicia el proceso de reconocimiento del discurso del usuario
// por parte del ASR. Devuelve en un String el resultado del
// reconocimiento de voz llevado a cabo
```

Panel Deslizable

```
mostrarMensaje(respuesta:String);
// muestra sobre el panel deslizable el mensaje de texto que
// contiene el parámetro "respuesta"

cambiarEstado(estado:Integer);
// establece el color de la fuente y la textura del panel
// deslizable de forma acorde al estado indicado en el parámetro
// "estado"

visible(writeOut:Boolean);
// muestra/Oculta el panel deslizable donde se escriben las
// respuestas del agente virtual en función del valor del
// parámetro "writeOut".
```

Área de Texto

```
escribirMensaje():String;
// gestiona la escritura del mensaje por parte del usuario.
// Devuelve el resultado de este proceso en forma de String

enviarMensaje(frase:String);
// transmite el mensaje escrito por el usuario, almacenado en el
// parámetro "frase", al agente virtual

visible(writeIn:Boolean);
// muestra/oculta el área de texto utilizado por el usuario para
// comunicarse de forma escrita con el agente virtual en función
// del valor del parámetro "writeIn"
```

Gestor de Diálogo

```
generaRespuesta (String frase):String;
// Interpreta y busca la respuesta adecuada al mensaje contenido
// por el parámetro "frase". Devuelve la respuesta en forma de
// String
```

C.1.3 Modelo dinámico

El modelo dinámico describe los aspectos de control de un sistema, determinando las operaciones que intervienen en dicho sistema y el orden en que se llevan a cabo las mismas.

En esta sección se muestran los aspectos de control y los procesos que intervienen en el sistema a desarrollar, así como los eventos que se pueden producir a lo largo de la ejecución del mismo. Para ello se utilizan tanto diagramas de estado como diagramas de traza de eventos.

a) Diagramas de Estados

Los conceptos más importantes del modelado dinámico son los eventos, que representan estímulos externos, y los estados, abstracciones de los valores de los atributos y enlaces de los objetos.

Para una determinada clase, es posible abstraer la trama de eventos, estados y transiciones entre estados, representando dicha trama en forma de un diagrama de estados.

El modelo dinámico para un sistema consta de múltiples diagramas de estados, uno para cada clase que posea un comportamiento dinámico relevante. A continuación se presentan los diagramas de estado que modelan el funcionamiento del sistema a desarrollar.

En la Figura C.1.2 se muestra el diagrama de estados de la clase Text To Speech del sistema. Esta clase, tras inicializar sus parámetros al comienzo de la ejecución del sistema, entra en un estado "WAIT" a la espera de nuevos eventos que procesar. Los eventos que es capaz de procesar esta clase son aquellos que implican la reproducción oral de un mensaje de texto, una modificación en el volumen del sistema o un cambio de la voz emocional a utilizar. Únicamente en el caso de recibir una petición de reproducción de discurso, la clase entra en un estado diferente, "Reproduciendo Discurso", estado en el que permanece durante toda la reproducción oral del mensaje pasado como argumento en la petición, volviendo al estado "WAIT" una vez que haya finalizado.

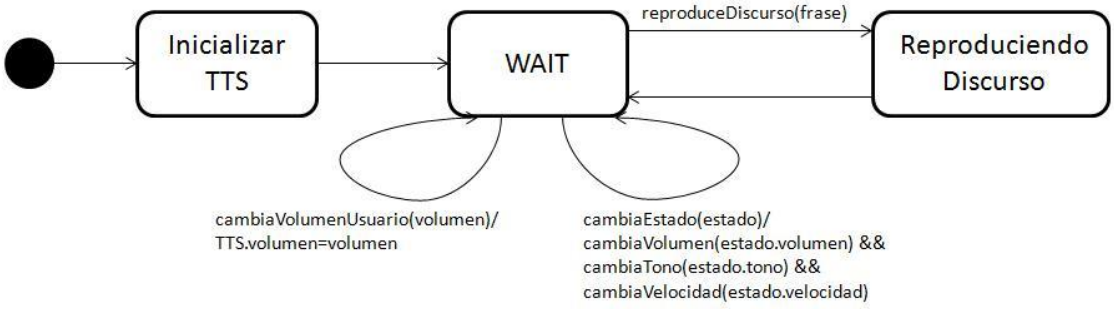


Figura C.1.2: *Diagrama de estados de la clase Text To Speech*

Por su parte, la Figura C.1.3 muestra el diagrama de estados de la clase Automatic Speech Recognition. Esta clase, tras inicializar sus parámetros al comienzo de la ejecución del sistema, entra en un estado WAIT a la espera de una petición de reconocimiento de discurso. En el momento que recibe una petición de este tipo, entra en el estado "Reconociendo Discurso", permaneciendo en dicho estado durante todo el proceso de escucha. Una vez finalizada la escucha, la clase vuelve al estado WAIT anterior, enviando el resultado del reconocimiento del discurso al Gestor de Diálogo.

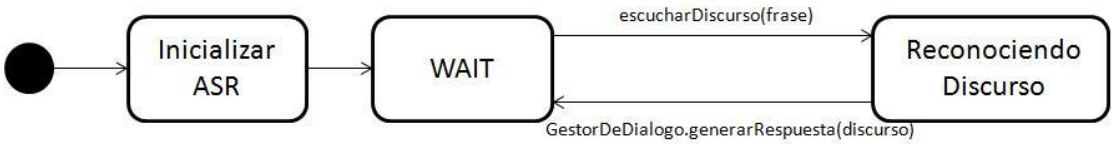


Figura C.1.3: *Diagrama de estados de la clase Automatic Speech Recognition*

El diagrama de estados de la clase Panel Deslizable se muestra en la Figura C.1.4. Este diagrama consta únicamente posee dos estados, cuya principal diferencia reside en la visibilidad del panel deslizable en la interfaz del sistema. De este modo, la clase Panel Deslizable es capaz de procesar peticiones de modificación tanto de la visibilidad del panel como del estado emocional a representar con la fuente y la textura del mismo en cualquiera de ambos estados. Sin embargo, solamente el estado "Mostrado WAIT" es capaz de mostrar por escrito los mensajes provenientes del Gestor de Diálogo.

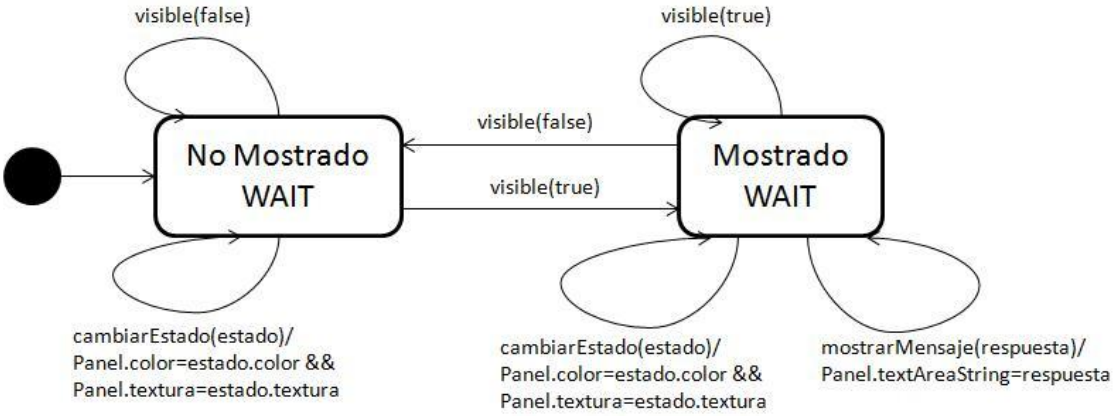


Figura C.1.4: *Diagrama de estados de la clase Panel Deslizable*

En la Figura C.1.5 se muestra el diagrama de estados de la clase Área de Texto. Inicialmente, esta clase se encuentra en el estado "No Mostrado WAIT", estado donde espera el evento que haga visible el área de texto en la interfaz gráfica del sistema. Una vez en el estado "Mostrado WAIT", la clase puede procesar peticiones de escritura sobre el Área de Texto, peticiones que trasladan la clase al estado "Escribiendo", donde permanece hasta que el usuario da por finalizado la escritura del mensaje, pasando al estado "Sin Enviar". En este estado, es posible enviar el mensaje al Gestor de Diálogo y volver al estado "Mostrado WAIT" o retornar al estado "Escribiendo" para seguir modificando el mensaje. Además, cualquiera de los estados existentes en este diagrama es capaz de procesar el evento "visible(false)", evento que devuelve a la clase al estado inicial "No Mostrado WAIT".

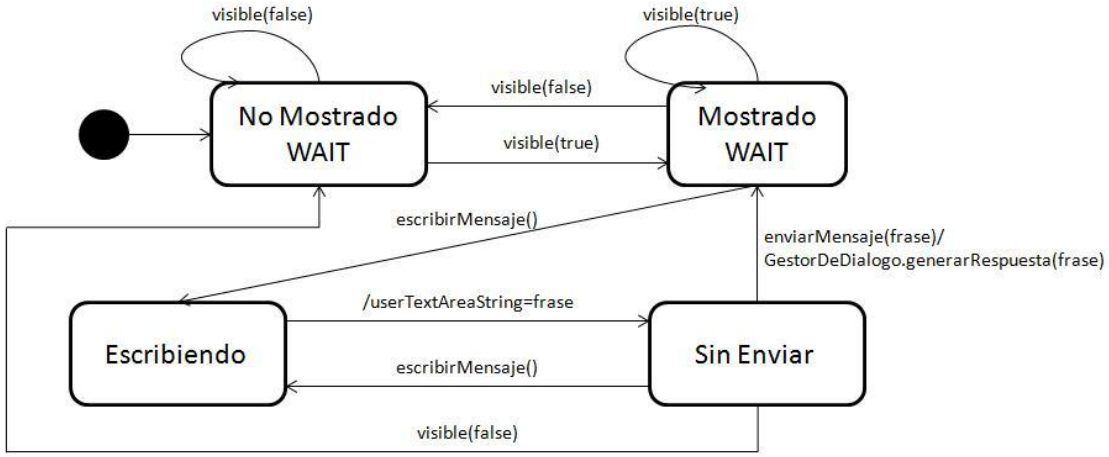


Figura C.1.5: *Diagrama de estados de la clase Área de Texto*

Por otro lado, el diagrama de estados correspondiente a la clase Gestor de Diálogo se muestra en la Figura C.1.6. Esta clase, tras inicializar sus parámetros al comienzo de la ejecución del sistema, entra en un estado "WAIT" a la espera de recibir una petición de generación de respuesta a cualquier mensaje proveniente del usuario. Una vez recibida esta petición, el Gestor de Diálogo entra en el estado "Generar Respuesta", estado conformado por dos sub-estados como son "Interpretar mensaje" y "Búsqueda de una respuesta". El primero de estos sub-estados se encarga de analizar el mensaje proveniente del usuario y gestionar, si es necesario, un cambio en el estado emocional del agente virtual. Por su parte, el segundo de

estos sub-estados es el encargado de buscar una respuesta acorde al mensaje del usuario, gestionando la transmisión de dicha respuesta al usuario antes de retornar al estado WAIT inicial.

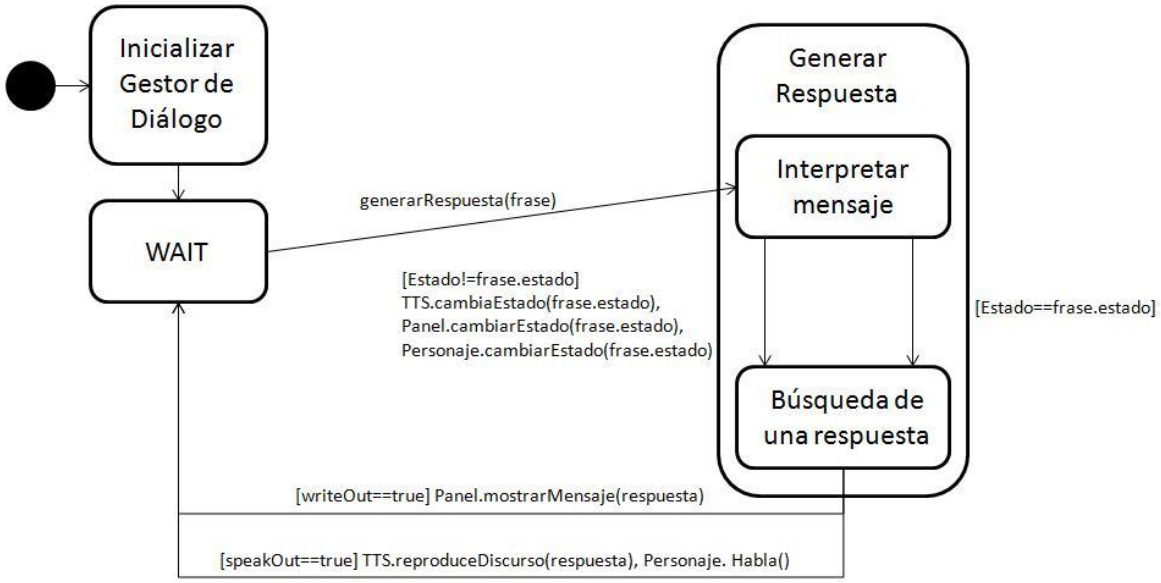


Figura C.1.6: *Diagrama de estados de la clase Gestor de Diálogo*

Por último, en la Figura C.1.7 se muestra el diagrama de estados de la clase GUI. Este diagrama presenta un estado "WAIT", estado donde la clase permanece en todo momento excepto cuando se muestra el menú de opciones del sistema por pantalla. Este estado "WAIT" es capaz de procesar peticiones de inicio de escucha del discurso del usuario, inicio de escritura en el área de texto, envío del mensaje escrito en el área de texto, modificación del volumen del sistema y mostrado del menú de opciones. Es esta última petición la que conlleva un cambio de estado en la clase, alcanzando el estado "Menú de Opciones". Este estado se encarga de mostrar el menú de opciones del sistema por pantalla, procesando a su vez las peticiones de mostrado de tanto el manual de ayuda como del menú de modos de interacción, peticiones que conllevan un cambio de estado distinto cada una. En el estado "Ayuda", el menú de ayuda se muestra por pantalla, pudiendo en cualquier momento retornar a estados anteriores con la petición de salida. Por su parte, en el estado "Modos de Interacción" se muestra el menú de los modos de interacción disponibles en el sistema, permitiendo la modificación de los modos seleccionados a través de las consiguientes peticiones de activación y desactivación de cada uno de los modos. Del mismo modo que en el caso del estado "Ayuda", es posible retornar a estados anteriores con la petición de salida.

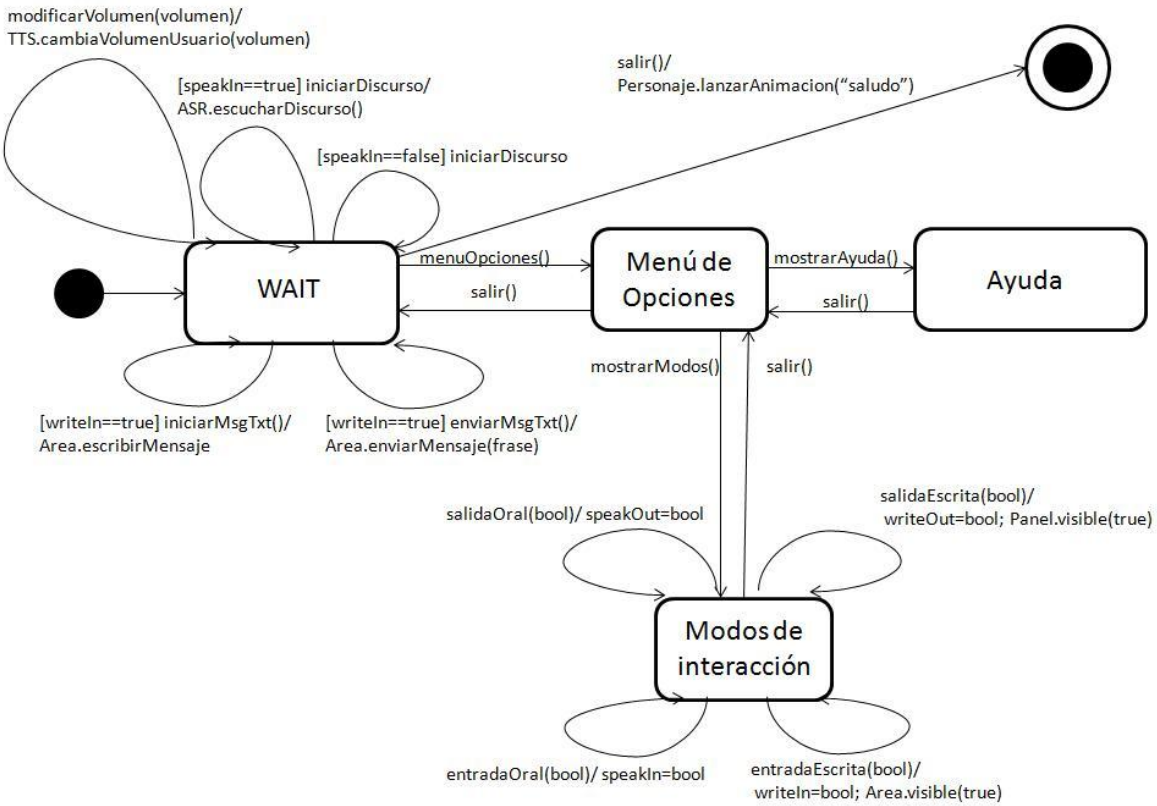


Figura C.1.7: *Diagrama de estados de la clase GUI*

b) Diagramas de Traza de Eventos

A través de las trazas de eventos se describen las interacciones principales entre los objetos que intervienen en el sistema y los eventos e intercambios de información que se producen entre los mismos.

En el caso del sistema que se está desarrollando, únicamente se han modelado dos trazas de eventos, correspondiéndose ambas con sendos ejemplos de uso básico del sistema:

- La primera de estas trazas, mostrada en la Figura C.1.8, se corresponde con un proceso de comunicación oral entre el usuario y el agente virtual.

- La segunda traza de eventos modelada, mostrada en el Figura C.1.9, se corresponde con una de modificación de los modos de interacción establecidos por defecto e inicio de un proceso de comunicación por escrito con el agente virtual.

De esta forma, a través de estas dos trazas de eventos se pretende modelar las principales actividades que pueden tener lugar durante la ejecución del sistema. El resto de procesos también se podrían modelar con una traza de eventos, donde se mostraría el intercambio de información que se lleva a cabo entre los distintos objetos del sistema en las múltiples actividades secundarias existentes. Sin embargo, aunque estas actividades son necesarias para el correcto funcionamiento del sistema, se considera que no requieren tanta atención como para especificarlas con una traza de eventos.

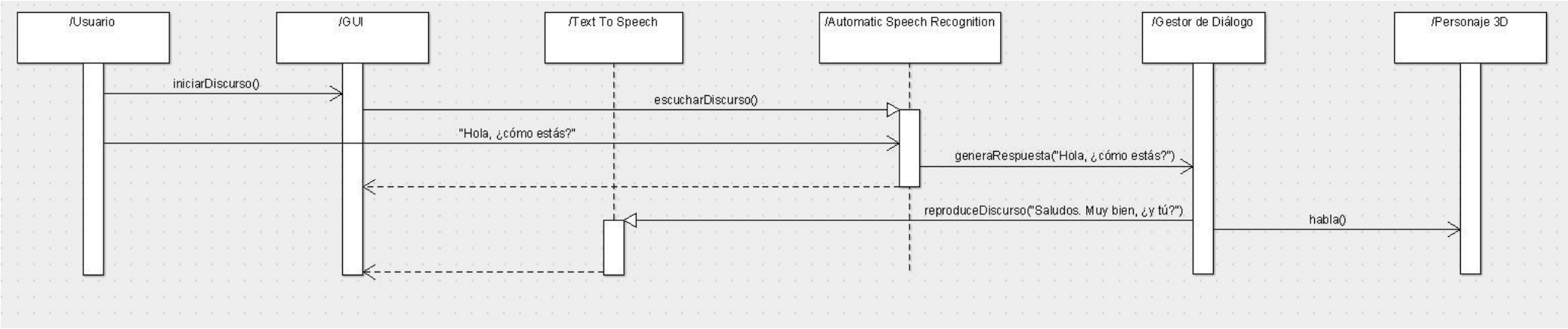


Figura C.1.8: *Traza de eventos del proceso de comunicación oral entre el usuario y el agente virtual*

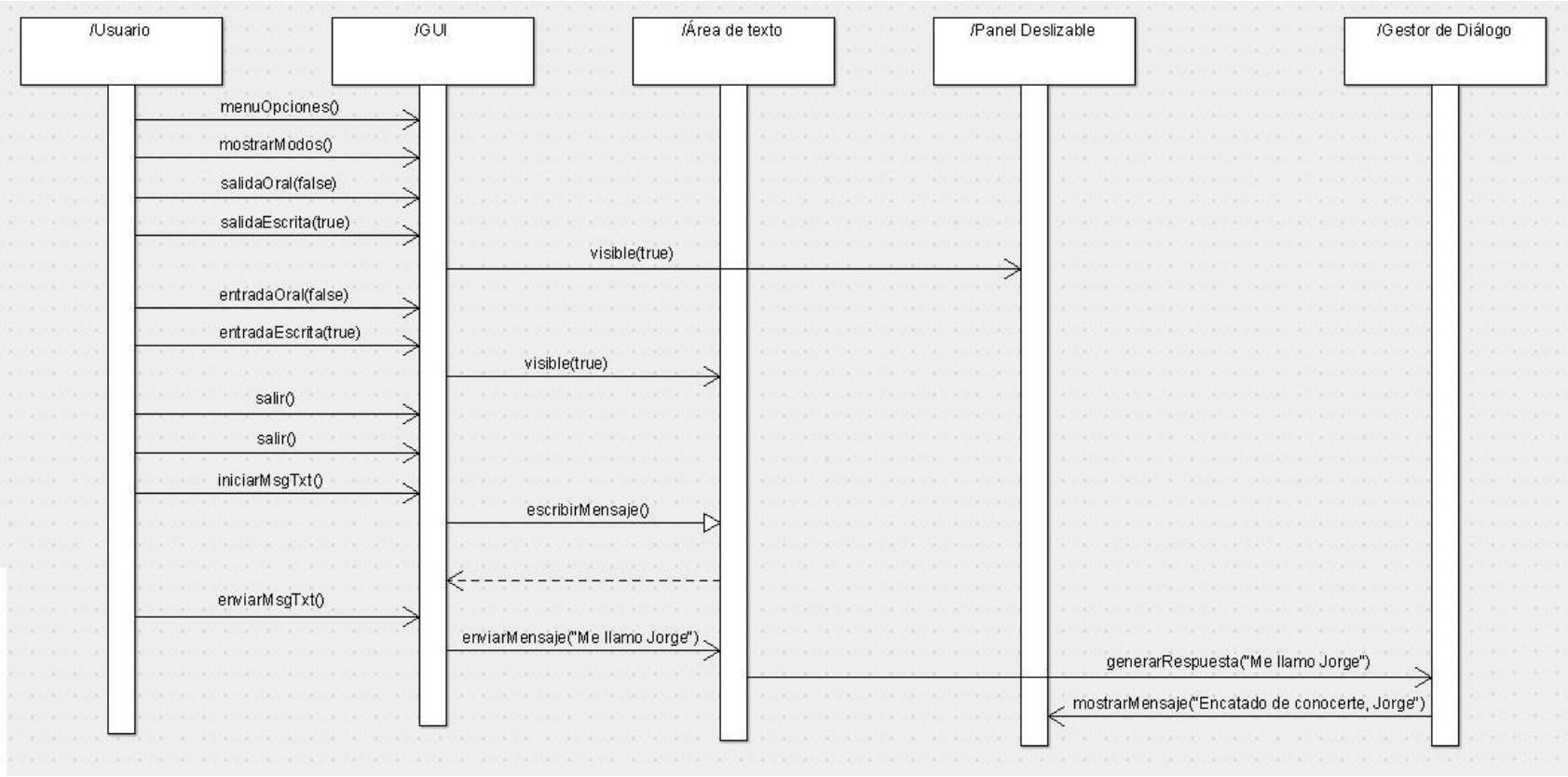


Figura C.1.9: *Traza de eventos del proceso de comunicación textual entre el usuario y el agente virtual*

C.1.4 Modelo funcional

El modelo funcional se emplea para especificar el significado tanto de las distintas operaciones que aparecen en el modelo de objetos como de las acciones y actividades del modelo dinámico. El modelo funcional describe la transformación de los datos de entrada y de salida del sistema mediante diagramas de flujo de datos (DFD).

Los DFD son grafos compuestos por nodos y arcos. Los nodos representan actores (producen/consumen datos), procesos (transforman datos) o almacenes de datos (elementos pasivos que únicamente guardan determinada información). Por su parte, los arcos representan valores de entrada/salida o flujos de datos (valores intermedios).

A través del modelo funcional se muestra el sistema subdividido en procesos que, aunque no se especifica cómo realizan sus funciones, es posible comprender qué labor desempeñan dentro del sistema, cuáles son los datos que precisan y qué datos devuelven. Además, se observa cuáles son los datos cuyo almacenamiento es necesario.

A continuación se muestran las figuras correspondientes al diagrama de flujo de datos del sistema a desarrollar.

En primer lugar, la figura C.1.10 muestra el DFD que se corresponde con el uso genérico del sistema, donde el usuario realiza una pregunta al agente virtual y éste le responde de forma acorde.

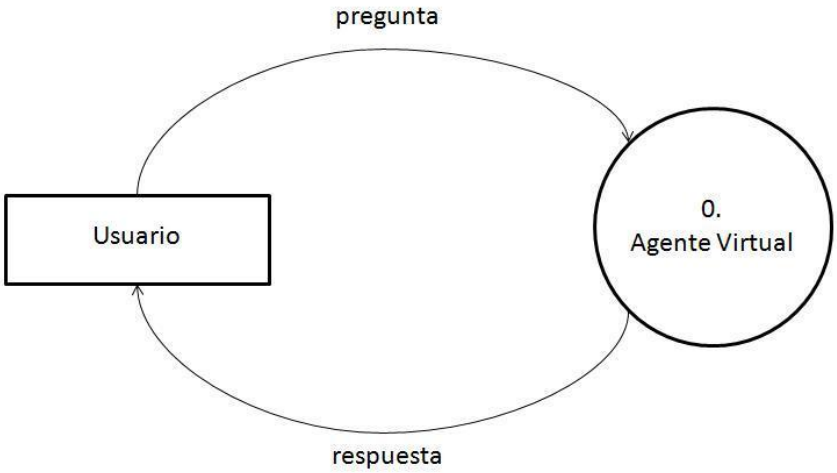


Figura C.1.10: *Diagrama de Flujo de Datos del contexto del problema*

Por su parte, la Figura C.1.11 muestra la división del sistema a desarrollar en los distintos subsistemas existentes.

La pregunta del usuario puede venir formulada a través del discurso oral del usuario o a través de un mensaje de texto, por lo que se generan dos procesos de entrada de datos, ”Preguntar oralmente al Agente Virtual” y “Preguntar por escrito al Agente Virtual”, los cuales consumen datos en formato audio o caracteres respectivamente. Estos procesos transforman las distintas entradas del sistema en cadenas de texto que contienen las preguntas formuladas por el usuario, pasando dichas cadenas de texto al proceso “Generar Respuesta”, proceso que se encarga de, a través de ficheros AIML, generar una respuesta acorde a cada una de las preguntas. Este proceso “Generar Respuesta” transmite tanto la respuesta que se debe devolver al usuario como el estado emocional en el que se debe encontrar el agente virtual a la hora de contestar a los dos procesos siguientes, “Reproducir respuesta oralmente” y “Reproducir respuesta por escrito”, los cuales se encargan de reproducir la respuesta oral o textualmente respectivamente.

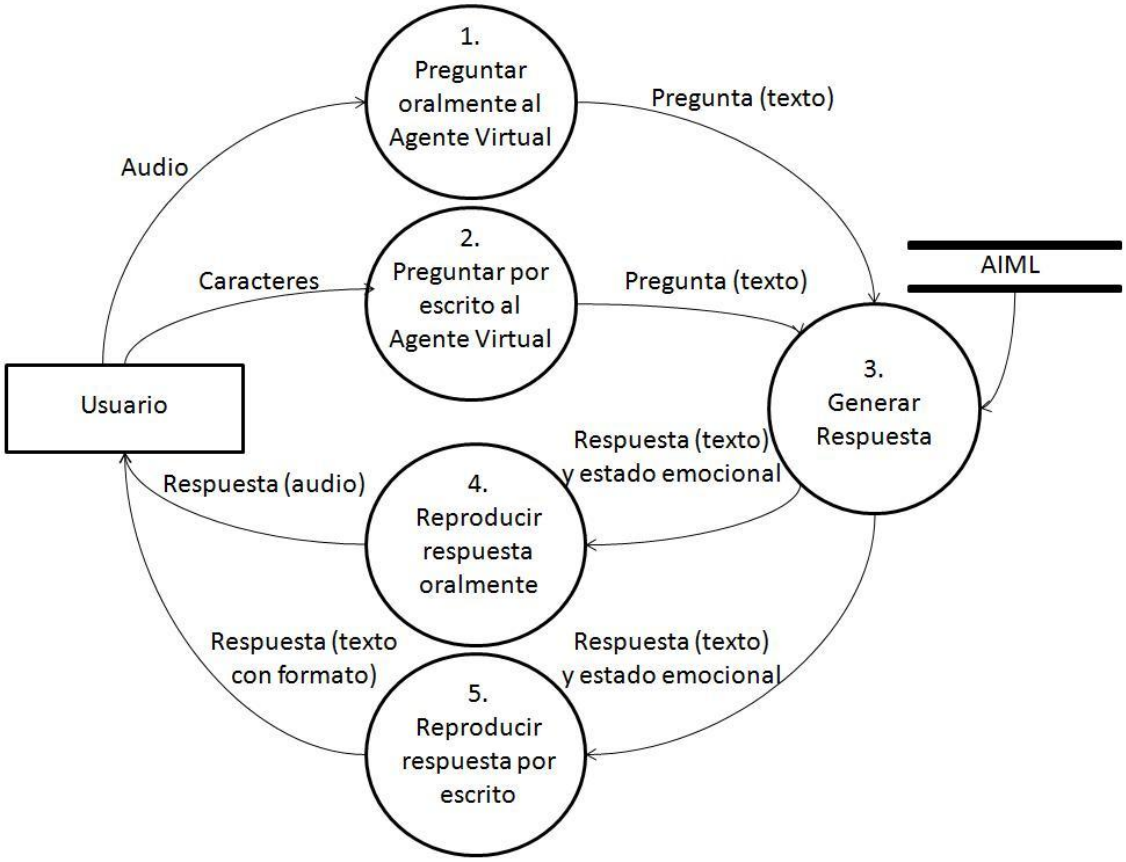


Figura C.1.11: *Diagrama de Flujo de Datos del sistema principal dividido en subsistemas*

La Figura C.1.12 descompone el proceso "Preguntar oralmente al Agente Virtual" en subprocesos. A través de este diagrama es posible apreciar cómo la pregunta oral del usuario es recogida por el proceso "Capturar audio", proceso que prepara el audio para su tratamiento posterior en el proceso "Extraer frase", el cual, sirviéndose de una gramática regular, transforma la pregunta del usuario en una cadena de texto.

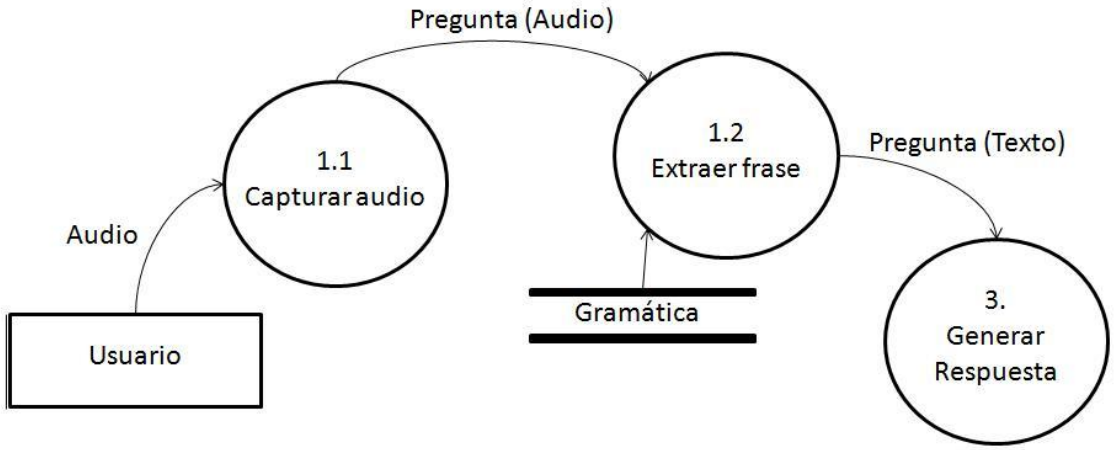


Figura C.1.12: *Diagrama de Flujo de Datos del proceso "Preguntar oralmente al Agente Virtual"*

En cuanto a la Figura C.1.13, descompone el proceso "Preguntar por escrito al Agente Virtual" en subprocesos. A través de este diagrama es posible apreciar cómo los caracteres tecleados por el usuario son recogidos por el proceso "Escribir pregunta", proceso que genera una cadena de texto que contiene la pregunta escrita por el usuario. Esta cadena es recibida por el proceso "Enviar pregunta", mediante el cual se opta por volver a reescribir la pregunta o por enviarla al agente virtual.

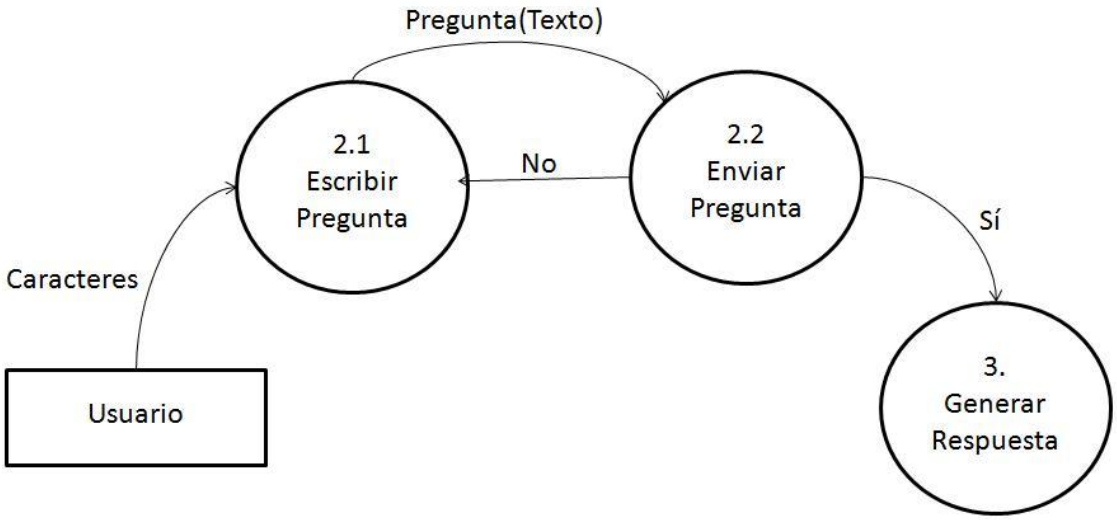


Figura C.1.13: *Diagrama de Flujo de Datos del proceso "Preguntar por escrito al Agente Virtual"*

Por otro lado, la Figura C.1.14 divide el proceso "Generar Respuesta" en distintos subprocesos. El proceso "Elaborar respuesta" se encarga de generar la respuesta que va a ser devuelta al usuario, indicando el estado emocional en el que se debe encontrar el agente virtual, el cual viene establecido por el proceso "Interpretar pregunta" tras el análisis AIML de la pregunta del usuario, y la contestación en formato texto, contestación basada en la respuesta AIML obtenida en el proceso "Buscar respuesta".

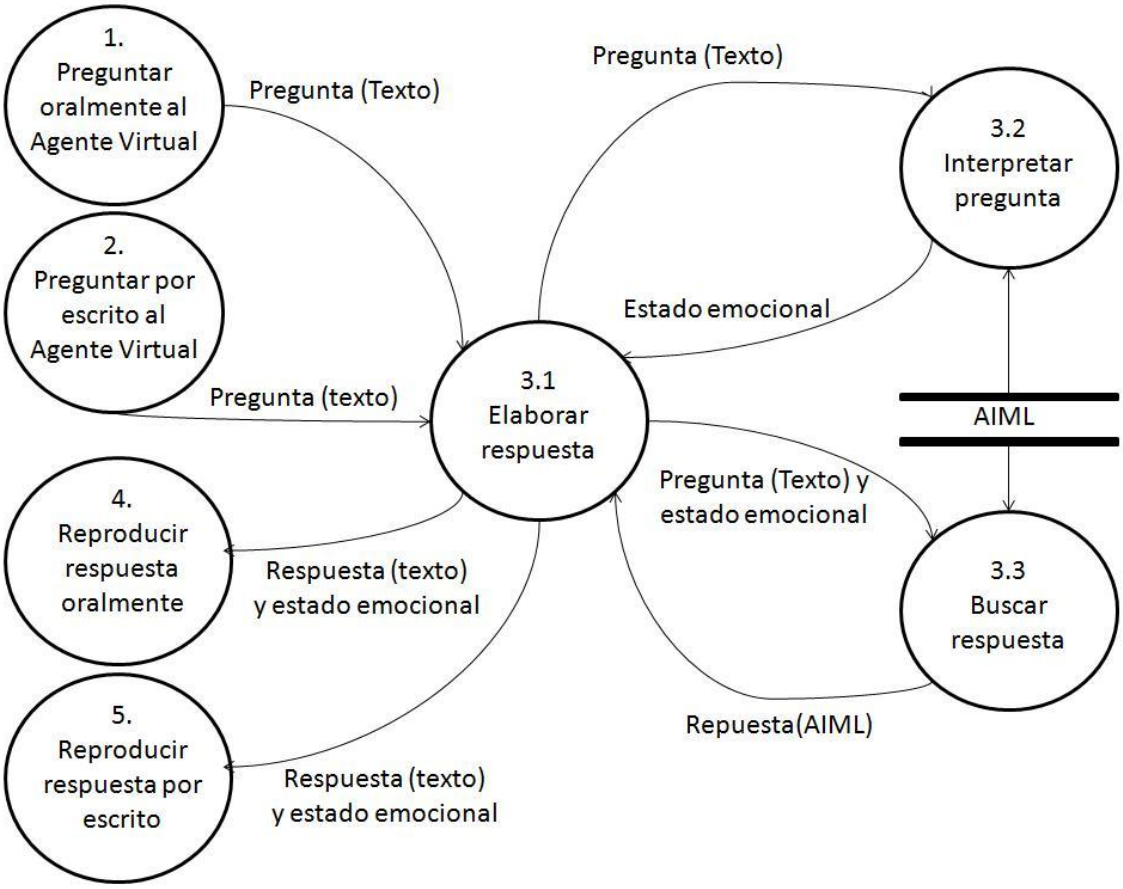


Figura C.1.14: *Diagrama de Flujo de Datos del proceso "Generar Respuesta"*

La Figura C.1.15 descompone el proceso "Reproducir respuesta oralmente" en subprocesos. En primer lugar, el proceso "Sintetizar respuesta" transmite el nuevo estado emocional del agente virtual al proceso "Modificar voz emocional", proceso encargado de determinar si es necesario modificar la voz

emocional utilizada para reproducir el discurso del agente. En caso de ser necesario un cambio de voz emocional, el proceso "Establecer parámetros del discurso" se encarga de transmitir los nuevos valores de tono, volumen y velocidad de discurso establecidos al estado "Sintetizar respuesta", que haciendo uso del conjunto de fonemas del que dispone, comienza la reproducción oral de la respuesta.

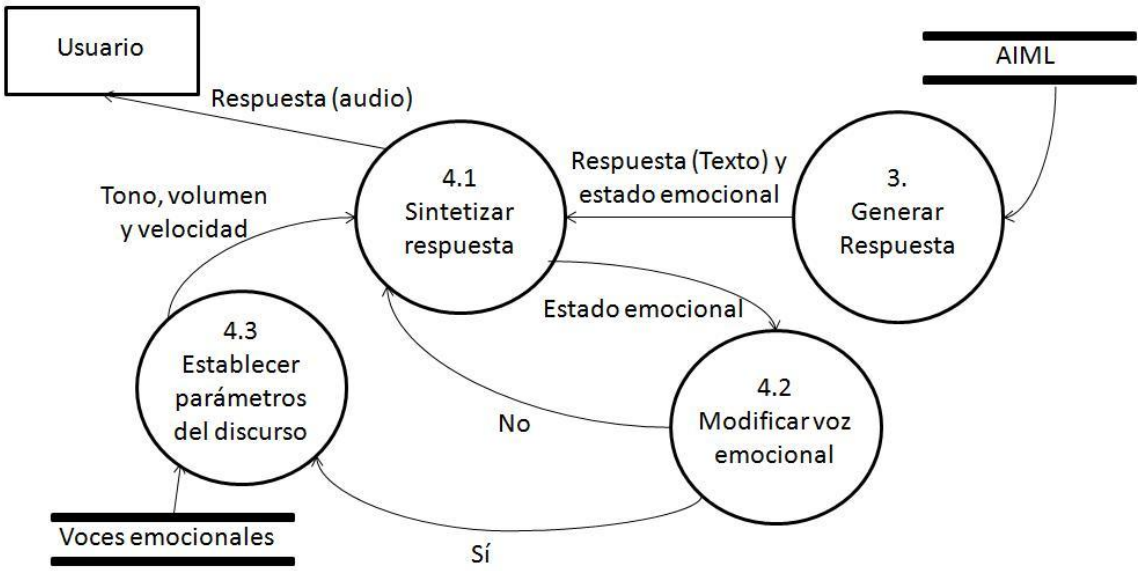


Figura C.1.15: *Diagrama de Flujo de Datos del proceso "Reproducir respuesta oralmente"*

Por último, la Figura C.1.16 divide el proceso "Reproducir respuesta por escrito" en distintos subprocesos. En primer lugar, el proceso "Mostrar respuesta" transmite el nuevo estado emocional del agente virtual al proceso "Modificar color y textura", proceso encargado de determinar si es necesario realizar un cambio tanto del color de la fuente como de la textura del panel deslizable. En caso de ser necesario un cambio, el proceso "Establecer color y textura" se encarga de transmitir el color de la fuente y la textura que han de ser utilizados al estado "Mostrar respuesta", que haciendo uso del conjunto de los colores de fuente y texturas almacenados en el sistema, reproduce por escrito la respuesta del agente virtual.

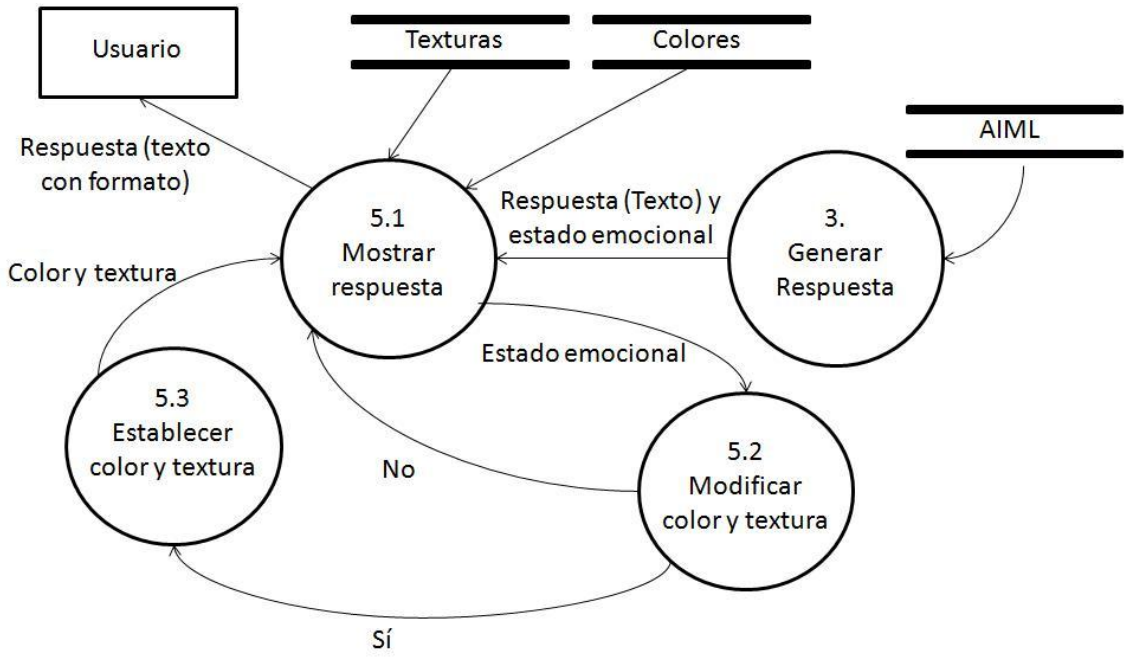


Figura C.1.16: *Diagrama de Flujo de Datos del proceso "Reproducir respuesta por escrito"*

C.2 Diseño

En esta sección del anexo se explica la metodología de diseño seguida para el desarrollo del sistema realizado en este proyecto y que complementa a la propuesta de diseño mostrada en el apartado 3.1 de la memoria principal.

El objetivo de la etapa de diseño es definir la estructura modular y los detalles del sistema partiendo del estudio realizado en la fase de análisis. En este sentido, se pretende hacer uso de un diseño que no sólo funcione, sino que también se pueda mantener con relativa facilidad, promueva la reutilización de código y se pueda entender y probar sin dificultad.

C.2.1 Patrón de diseño Modelo-Vista-Controlador

Debido a las características del sistema que se pretende desarrollar, se opta por hacer uso del patrón de diseño Modelo-Vista-Controlador (MVC).

El patrón MVC, mostrado en la Figura C.2.1, propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, de manera que el sistema quede dividido en tres capas donde la encapsulación de los datos, la interfaz o vista del sistema y la lógica interna o controlador estén completamente diferenciadas. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, ideas que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

De forma genérica, los componentes del patrón MVC se podrían definir como sigue:

- El modelo: es la representación de la información con la cual opera el sistema, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica interna). Envía a la "vista" aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al "modelo" a través del "controlador".

- El controlador: responde a eventos (generalmente acciones del usuario) e invoca peticiones al "modelo" cuando se hace alguna solicitud sobre la información como, por ejemplo, editar un documento o un registro en una base de datos. También puede enviar comandos a su "vista" asociada si se solicita un cambio en la forma en que se presenta el "modelo", por ejemplo, realizar un desplazamiento o scroll por un documento o por los diferentes registros de una base de datos. Es por ello que se podría considerar que el "controlador" hace de intermediario entre la "vista" y el "modelo".

- La vista: presenta el "modelo" en un formato adecuado para interactuar (normalmente a través de una interfaz de usuario), por tanto requiere de dicho "modelo" la información que debe representar como salida.

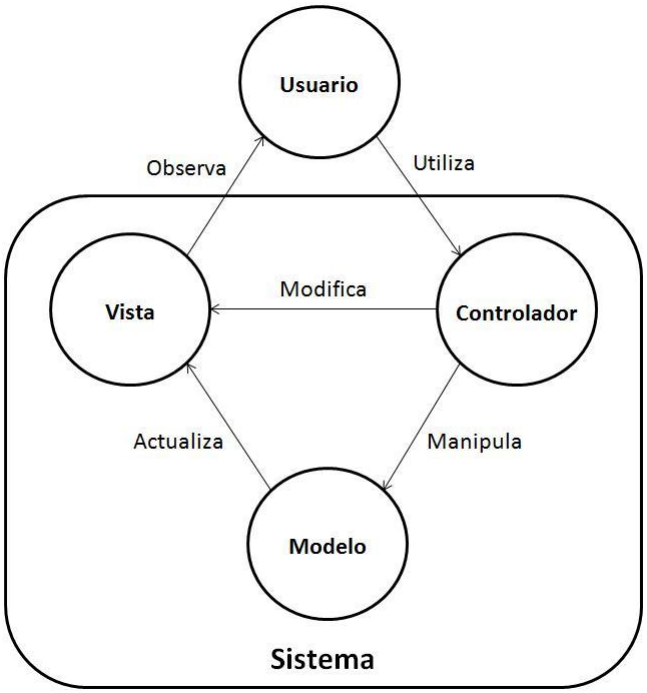


Figura C.2.1: *Esquema correspondiente al patrón de diseño Modelo-Vista-Controlador*

Una vez definido el patrón MVC con sus respectivos componentes, cabe destacar que en el proceso de diseño del sistema a realizar en este Proyecto Fin de Carrera se ha hecho uso de una de las múltiples versiones existentes de este patrón, más concretamente de la versión Modelo-Vista, también conocida como Documento-Vista. Esta versión del patrón MVC se caracteriza por unir las responsabilidades de la "Vista" y el "Controlador" del sistema en un único componente.

C.2.2 Diseño del sistema

Como se ha mencionado en el apartado anterior, se opta por utilizar el patrón Modelo-Vista para el diseño del sistema. Este patrón se suele utilizar en sistemas donde el pintado de la GUI y gestión de eventos están estrechamente unidas, como ocurre en los sistemas desarrollados sobre la plataforma Unity 3D, plataforma que se ha usado para la realización de este proyecto. A continuación, en la Figura C.2.2, se muestra una primera visión genérica del diseño del sistema.

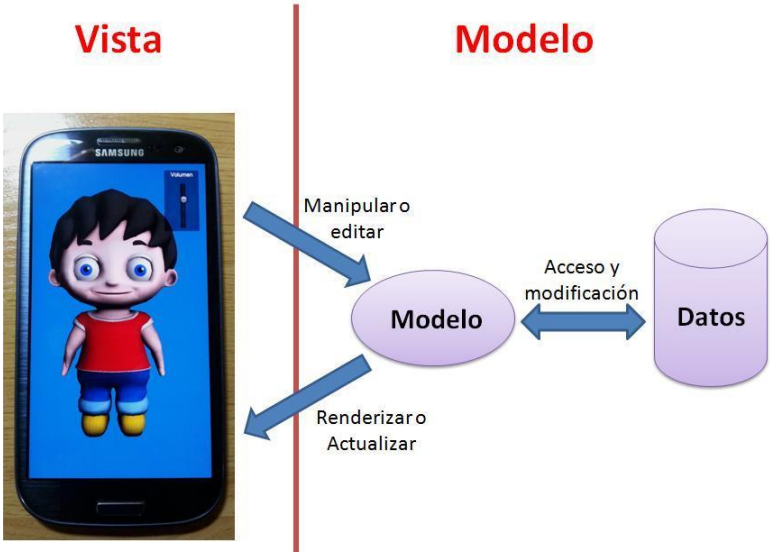


Figura C.2.2: *Esquema del patrón de diseño Modelo-Vista utilizado en el sistema*

Como se puede apreciar en la figura anterior, la comunicación entre el "Modelo" y la "Vista" es directa, existiendo dos tipos de mensajes entre ambos:

- Los mensajes que recibe la "Vista", procedentes del "Modelo", informan acerca de los cambios que se producen en los datos del sistema.

- Los mensajes que recibe el "Modelo", procedentes de la "Vista", informan acerca de la interacción llevada a cabo por el usuario con los elementos que se encuentran en reproducción en la interfaz gráfica del sistema.

De esta forma, se puede hacer una clasificación en dos capas (Vista y Modelo) de las clases identificadas en la sección C.1.2 de este anexo, tal y como se muestra en la Figura C.2.3.

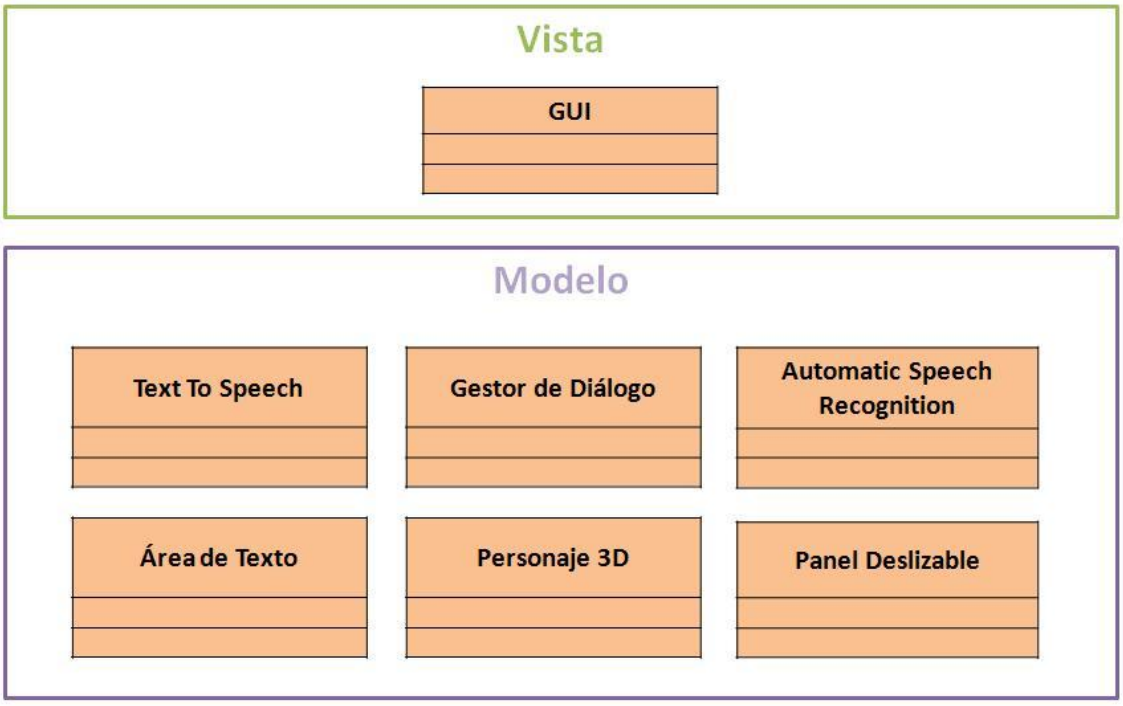


Figura C.2.3: *Clasificación de las clases identificadas en dos capas, Vista y Modelo.*

Anexo D. Implementación de la interacción con el agente virtual

Este anexo complementa al capítulo 3 de la memoria principal. A lo largo de este documento se explica la implementación llevada a cabo en cada uno de los módulos que intervienen en el proceso de interacción con el agente virtual y cuyo diseño se ha explicado anteriormente en el capítulo 3 de la memoria.

D.1 Reconocedor del discurso

Una vez decidido que el proceso de escucha del discurso proveniente usuario se va a llevar a cabo a través de peticiones de reconocimiento de voz a la aplicación Google Voice Search (decisión explicada en el apartado 3.2.1 de la memoria principal), se debe dotar al sistema, desarrollado sobre la plataforma Unity 3D, de la capacidad de realizar este tipo de peticiones a aplicaciones externas de Android. Para ello, se precisa implementar y añadir al proyecto Unity 3D un *plugin* JAVA que gestione las peticiones de reconocimiento del discurso. A continuación se muestran las líneas de código más relevantes del *plugin* JAVA desarrollado.

```
public void escuchar()
{
      Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

      intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
      RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

      intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Habla, te escucho");

      try
      {
            startActivityForResult(intent, RESULT_SPEECH);
      }

      …

}
```

Cuadro D.1.1: *Función en código JAVA que gestiona la escucha del discurso*

En el Cuadro D.1.1 se presenta el fragmento de código que gestiona la escucha del discurso proveniente del usuario. En primer lugar, se realiza una petición de reconocimiento de voz a la aplicación especializada instalada en el propio dispositivo, en este caso Google Voice Search. Este tipo de peticiones a aplicaciones externas al sistema en ejecución se llevan a cabo a través de los denominados *Intent*, que son ejecutados al invocarse sus respectivas acciones de lanzamiento. En el caso del reconocimiento del discurso, la acción utilizada es *RecognizerIntent.ACTION_RECOGNIZE_SPEECH*. Tras esto, se indica el modelo de lenguaje que se va a utilizar durante el reconocimiento llevado a cabo con dicho *Intent*, en este caso, el modelo específico para dictados *RecognizerIntent.LANGUAGE_MODEL_FREE_FORM*; y se define el mensaje encargado de avisar al usuario que puede comenzar a hablar. Por último, se lanza la actividad y se espera la obtención de resultados.

Por su parte, en el Cuadro D.1.2 se muestran las líneas de código que gestionan el reconocimiento del discurso del usuario, transmitiendo la información obtenida al sistema.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode== RESULT_SPEECH)
    {
        if (resultCode == RESULT_OK && null != data)
        {
            ArrayList<String> text = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

            Texto = (text.get(0));

            UnityPlayer.UnitySendMessage("niña Prefab", "interpretaEscuchado", Texto);
        }
        else
        {
            UnityPlayer.UnitySendMessage("niña Prefab", "interpretaEscuchado", "Lo siento, no te he entendido
            bien”);
        }
    }
}
```

Cuadro D.1.2: *Función en código JAVA que gestiona los resultados del reconocimiento del discurso*

Inicialmente, tras recibir los resultados procedentes de la actividad lanzada, se realizan una serie de comprobaciones para determinar si el proceso de reconocimiento de voz se ha llevado a cabo correctamente y se han obtenido resultados no vacíos.

En caso de superar estas comprobaciones, el *Intent* devuelve un *ArrayList* de cadenas de texto con los resultados del reconocimiento de voz. Este *ArrayList* está ordenado de forma descendente con respecto al grado de fidelidad del reconocimiento, por lo que se opta por utilizar el resultado almacenado en la primera posición. Finalmente, se envía la cadena de texto seleccionada al gestor de diálogo del sistema, sirviéndose para ello del método estático *UnityPlayer.UnitySendMessage("GameObjectName1", "MethodName1", "Message to send")*. Este método, perteneciente a la clase *unity3d.player.UnityPlayer*, permite invocar desde un *plugin* Android, desarrollado en Java, un método asociado a algún objeto presente en la escena de Unity3D, posibilitando de esta forma la comunicación bidireccional entre el *plugin* y el sistema desarrollado sobre la plataforma Unity3D.

En caso de no superar alguna de las comprobaciones anteriores, se envía un mensaje de error al gestor de diálogo, instando al usuario a repetir el mensaje que desea transmitir al agente virtual.

D.2 Sintetizador de voz

Una vez seleccionada la metodología para reproducir el discurso del agente virtual (elección que se detalla en el apartado 3.2.2 de la memoria principal), se debe dotar al sistema, desarrollado sobre la plataforma Unity 3D, de la capacidad de comunicarse con el motor de síntesis de voz nativo de Android. Para ello, se precisa implementar y añadir al proyecto Unity 3D un nuevo *plugin* JAVA que gestione esta comunicación. A continuación se muestran los fragmentos de código más relevantes del *plugin* JAVA desarrollado.

En primer lugar, el Cuadro D.2.1 muestra la creación e inicialización de la clase TTSActivity, clase que se encarga de gestionar la comunicación entre Unity 3D y el sistema de síntesis de voz de Android.

```
public class TTSActivity extends UnityPlayerActivity implements TextToSpeech.OnInitListener
{
    private TextToSpeech tts;
    private float tono;
    private float velocidad;
    private float volumen;

    ...

    @Override
    protected void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        tts = new TextToSpeech(this, this);
        tono = 1;
        velocidad = 1;
        volumen = 12.0f;
    }

    @Override
    public void onInit(int status)
    {
        if (status == TextToSpeech.SUCCESS)
        {
            Locale locSpanish = new Locale("spa", "ESP");

            int result = tts.setLanguage(locSpanish);

            if (result == TextToSpeech.LANG_MISSING_DATA || result == TextToSpeech.LANG_NOT_SUPPORT)
            {
                Log.e("TTS", "This Language is not supported");
            }
        }
        else
        {
            Log.e("TTS", "Initilization Failed!");
        }
    }
```

Cuadro D.2.1: *Código JAVA que crea e inicializa la clase TTSActivity, clase encargada de gestionar la comunicación entre Unity 3D y el sistema TTS de Android*

Inicialmente, con el fin de gestionar el motor de síntesis de discurso que incorpora Android, se lleva a cabo una instancia de la clase pública JAVA *TextToSpeech*, cuyos métodos permiten reproducir un texto dado de forma inmediata o crear un archivo de audio con el discurso generado.

Posteriormente, se establecen los valores por defecto de los parámetros del discurso. De esta forma, se determina que el tono y la velocidad de discurso sean los estándares, otorgándoles el valor de uno, valor intermedio dentro de su rango de valores aceptados. Por otro lado, al volumen se le otorga el valor 12, ligeramente por encima del valor medio de su rango de valores, con el fin de que el discurso sea percibido de forma clara por el usuario en lugares donde el ruido exterior sea moderado.

Una vez establecidos los valores por defecto de los parámetro del discurso, es necesario completar la inicialización de clase pública JAVA *TextToSpeech* a través del método *onInit(int status)*, método presente en la interfaz *TextToSpeech.OnInitListener*. Este método, además de comprobar que la inicialización se ha realizado correctamente, permite establecer el idioma del motor de síntesis de voz. En este caso, a través de la instrucción *new Locale("spa", "ESP")*, se ha seleccionado como idioma el español y como variante del dicho idioma la de España.

A continuación, el Cuadro D.2.2 muestra la función encargada de sintetizar el discurso del agente virtual. En primer lugar, se realizan las comprobaciones necesarias para determinar si la cadena de texto pasada como argumento de la función se trata de la cadena vacía. En caso afirmativo, se selecciona un mensaje de error como texto a reproducir; mientras que, en caso de no ser vacía la cadena pasada como argumento, se utiliza el texto contenido en la misma. A continuación, se hace uso del método *speak(String text, int queueMode, HashMap <String, String> params)* de la clase pública *TextToSpeech* para la reproducción oral del texto seleccionado. Como primer parámetro se pasa la variable “text”,

variable que almacena el texto a reproducir resultante de las comprobaciones. Como segundo parámetro se selecciona la constante "QUEUE_FLUSH", constante que obliga al sintetizador de voz a empezar un nuevo discurso en el momento que le llega una petición, interrumpiendo cualquier discurso anterior. Finalmente, se deja como "null" el último parámetro.

```
private void speakOut(String frase)
{
    String text = "No reconozco el input";
    if ((frase!=null)&&(!(frase.equals(""))))
    {
        text = frase;
    }
    tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}
```

Cuadro D.2.2: *Función en código JAVA que reproduce el discurso del agente*

Para concluir, en el Cuadro D.2.3 se muestran los diversos métodos implementados para modificar algunos de los parámetros del discurso sintetizado, como son el volumen, el tono o la velocidad del mismo. A través de la variación de estos parámetros es posible dotar de cierta emoción al discurso pronunciado por el agente virtual, cuestión que se explica en mayor detalle en el siguiente apartado.

```
private void setVolume(float volume)
{
    AudioManager am = (AudioManager)getSystemService(Context.AUDIO_SERVICE);
    int vol = (int) volume;
    am.setStreamVolume(am.STREAM_MUSIC,vol, 0);
}

private void cambiaVolumen (float vol)
{
    /*Valores de volumen entre 0.0 y 20.0*/
    float proporcion = this.volumen/12.0f;
    setVolume(proporcion*vol);
}

private void cambiaVelocidad (float velocidad)
{
    /*Valores de velocidad entre 0.0 y 2.0*/
    if (this.velocidad!=velocidad)
    {
        this.velocidad=velocidad;
        tts.setSpeechRate (velocidad);
    }
}

private void cambiaTono (float tono)
{
    /*Valores de tono entre 0.0 y 3.0*/
    if (this.tono!=tono)
    {
        this.tono=tono;
        tts.setPitch(tono);
    }
}
```

Cuadro D.2.3: *Funciones en código JAVA que modifican parámetros del discurso*

D.3 Expresión de emociones a través de la voz

Como se explica en el apartado 3.2.3 de la memoria principal, el sistema TTS de Android permite manipular únicamente tres parámetros a la hora de sintetizar la voz del agente virtual: volumen, velocidad y tono. A continuación, se explica la gestión, a través de código, de cada uno de estos parámetros por separado.

- Volumen: el valor de este parámetro se establece a través del método *setStreamVolume (int streamType, int index, int flags)* de la clase pública JAVA *AudioManager*. Este método, utilizado en la función "setVolume" mostrada en el Cuadro D.2.3, recibe como segundo parámetro un número entero comprendido entre 0 y 20, número encargado de determinar el nivel de volumen con el que se reproduce el discurso.

- Velocidad: el método utilizado para modificar la velocidad del discurso es *int setSpeechRate (float speechRate)*, método perteneciente a la clase pública JAVA *TextToSpeech*. Este método, usado en el interior de la función "cambiaVelocidad" del Cuadro D.2.3, recibe como parámetro un número decimal comprendido entre 0 y 2, decimal que establece la velocidad con la que se articulan las palabras a lo largo del discurso a reproducir.

- Tono: la gestión del tono del discurso se lleva a cabo a través del método *int setPitch (float pitch)* perteneciente a la clase pública JAVA *TextToSpeech*. Este método recibe como parámetro un número decimal comprendido entre 0 y 3, decimal encargado de establecer el tono del discurso a reproducir. En el Cuadro D.2.3 se muestra el uso de este método en el interior de la función "cambiaTono".

D.4 Área de texto

Con el fin de que el usuario pueda comunicarse a través de mensajes escritos con el agente virtual, se opta por implementar un área de texto con un botón de envío (cuyos diseños se explican en el apartado 3.3.1 de la memoria principal).

El área de texto se genera a partir de la función *static function* ***TextArea(position****: Rect* ***text****; string* ***style****: GUIStyle):string* que incorpora la clase GUI existente en Unity 3D. El parámetro ***position*** permite establecer las dimensiones del área de texto, el parámetro ***style*** su estilo y el parámetro ***text*** se corresponde con texto que escribe el usuario, texto que es devuelto a su vez como resultado de la función.

Por otro lado, para generar el botón de 'Enviar' se hace uso de la función *static function* ***Button(position****: Rect,* ***text****: string,* ***style****: GUIStyle): bool* que incorpora también la clase GUI existente en Unity 3D. Tanto el parámetro ***position*** como el parámetro ***style*** se utilizan de una forma análoga a como se usan en el caso del área de texto, mientras que el parámetro ***text*** se corresponde con el texto que se muestra sobre el propio botón.

D.5 Panel deslizable

Con el objetivo de transmitir por escrito al usuario la información procedente del agente virtual, se opta por implementar un panel deslizable (decisión que se explica en el apartado 3.3.2 de la memoria principal, junto al diseño del panel).

Dado que la plataforma Unity 3D no provee ninguna clase que implemente la creación de un panel deslizable, se opta por combinar varios de sus elementos ya disponibles para conseguirlo. En el Cuadro D.5.1 se muestra el fragmento de código utilizado para la generación del panel deslizable.

```
GUILayout.BeginArea (new Rect(ancho*0.05f, alto*0.60f, ancho*0.90f, alto*0.15f));
scrollPosition = GUILayout.BeginScrollView (scrollPosition, GUILayout.Width (ancho*0.90f), GUILayout.Height (alto*0.15f));
GUI.skin.box.wordWrap = true;
GUILayout.Box(textAreaString,myBox);
GUILayout.EndScrollView();
GUILayout.EndArea();
```

Cuadro D.5.1: *Código Script C# que implementa el panel desplegable*

En primer lugar, se crea un área rectangular que define las dimensiones del panel en la interfaz gráfica, utilizándose para ello la función *static function* ***BeginArea(screenRect****: Rect): void;* que incorpora la clase GUILayout existente en Unity 3D. A esta función se le pasa la instancia de un rectángulo con las mismas dimensiones que el área de texto anterior.

A continuación, haciendo uso de la función *static function* ***BeginScrollView(scrollPosition****: Vector2, params* ***options****: GUILayoutOption[]): Vector2* que incorpora la clase GUILayout, se añade un scroll al área creada. Este scroll permitirá al usuario desplazarse por el interior del panel en los casos en los que el mensaje proveniente del agente virtual sea lo suficientemente largo como para no poder ser mostrado en su totalidad en el espacio destinado al panel.

Finalmente, usando la función *static function* ***Box(text****: string,* ***style****: GUIStyle, params* ***options****: GUILayoutOption[]): void;* existente en la clase GUILayout, se añade al área creada una caja. Esta caja posee una gran relevancia dentro del conjunto de elementos que conforman el panel, puesto que sobre dicha caja se muestra el mensaje proveniente del agente virtual. La caja recibe como primer parámetro la variable global que contiene el mensaje actualizado del agente virtual, mientras que como segundo parámetro recibe una variable de tipo GUIStyle, tipo que permite personalizar el estilo de los elementos de la interfaz gráfica y que, en este caso, otorga la posibilidad de variar el aspecto del contenido de la caja. En este caso, la variable de tipo GUIStyle utilizada es 'myBox', variable encargada de que el mensaje proveniente del agente virtual se muestre de forma sencilla y clara al usuario. El Cuadro D.5.2 muestra el fragmento de código en el que se inicializa dicha variable.

```
myBox = new GUIStyle(GUI.skin.box);
myBox.fontSize = Textsize;
myBox.alignment = TextAnchor.UpperCenter;
myBox.padding = new RectOffset(15,15,20,20);
```

Cuadro D.5.2: *Código Script C# que inicializa la variable myBox*

A través de esta inicialización, el tamaño de letra utilizado para mostrar el mensaje del agente virtual pasa a ser idéntico al tamaño de letra establecido por el sistema para todos los elementos simples de la interfaz. Dicho tamaño es seleccionado por el sistema en función de la resolución de la pantalla del dispositivo en el que se esté ejecutando. A su vez, en la inicialización anterior se determina que el texto comience en la parte central superior del panel deslizable, quedando el mensaje en todo momento centrado en el interior de dicho panel y rellenando de forma descendente el mismo. Además, se establecen tanto los márgenes laterales como los márgenes superior e inferior del mensaje.

D.6 Expresión de emociones a través del texto

Una vez seleccionados los colores y generadas las imágenes que van a servir de textura para el panel deslizable (véase apartado 3.3.3 de la memoria principal), es necesario gestionar los cambios tanto de la fuente como de los bordes de dicho panel deslizable a través de código.

Gestión de los cambios

Con el fin de gestionar el color tanto de los bordes del panel deslizable como de la fuente utilizada en el interior del mismo, el sistema se sirve de las variables globales "textura" y "color". La primera de las variables es de tipo *Texture2D* y almacena, de forma actualizada, la textura que corresponde con el estado anímico del agente virtual en cada momento. Por su parte, "color" es una variable de tipo *Color* y se encarga de almacenar el color de fuente correspondiente al estado emocional del agente. En el Cuadro D.6.1 se ilustra el uso de estas variables

```
myBox.normal.background = textura;
myBox.normal.textColor = color;


color = Color.blue;
textura = Resources.Load("neutral") as Texture2D;
```

Cuadro D.6.1: *Código Script C# que gestiona el panel desplegable*

Como se puede observar en el cuadro, a los campos background y textColor de la variable global "myBox", anteriormente explicada (véase Cuadro D.5.2), se les asigna las variables "textura" y "color" respectivamente. De esta forma, el color y la textura almacenados en dichas variables se aplicarán al panel deslizable sin necesidad de actuar directamente sobre la variable de estilo "myBox".

Cabe destacar que las imágenes creadas anteriormente para servir de textura al panel deslizable se almacenan en el directorio "Resources" del proyecto Unity sobre el que se trabaja, directorio que permite el cargado de imágenes y su conversión a texturas bidimensionales a través de código *Script* C#.

D.7 Gestor de diálogo

Tras optar por el Programa-AB para el desarrollo del módulo gestor de diálogo del sistema a realizar (consúltese apartado 3.4.1 de la memoria principal), se deben llevar a cabo una serie de acciones que permitan la integración de dicho programa en el sistema.

En primer lugar, es necesario incorporar a los dispositivos móviles en los que se va a ejecutar el sistema una serie de ficheros que permitan la correcta ejecución del Programa-AB. Este conjunto de ficheros vendrá almacenado en el interior de un nuevo directorio, "/storage/sdcard0/BotPFC", creado previamente en cada uno de los dispositivos móviles utilizados.

Gran parte de los ficheros que se precisan provienen del repositorio de la fundación A.L.I.C.E. [AB ficheros web], más concretamente de la descarga y descompresión del fichero "ab.zip". Este archivo comprimido contiene la mayoría de los ficheros necesarios para la ejecución de un *chatbot* sobre el Programa-AB, aunque deben sufrir ciertas modificaciones antes de ser integrados en el sistema. A continuación, en la Tabla D.7.1, se listan los ficheros existentes en el directorio "/ab", directorio resultante de la descompresión del fichero "ab.zip", y se adjunta una breve descripción de los mismos:

ab/bots	Directorio en el que se encuentran los chatbots desarrollados
ab/lib	Directorio en el que se encuentran las bibliotecas Java necesarias para ejecutar el código AIML de los distintos chatbots
ab/out	Directorio de ficheros .class de Java
ab/run.bat	Fichero batch que permite ejecutar el Programa AB (no necesario)

Tabla D.7.1: *Tabla con los distintos ficheros y directorios que incluye el fichero "ab.zip"*

Previamente al traslado de estos ficheros al directorio "/storage/sdcard0/BotPFC" de cada uno de los dispositivos móviles utilizados, es necesario llevar a cabo varias modificaciones sobre los mismos.

En primera instancia, es necesario eliminar todos aquellos ficheros que no sean de utilidad al sistema, de forma que el espacio de memoria requerido sea el mínimo posible. En este sentido se elimina el fichero de ejecución "run.bat", ya que, en el sistema que se está desarrollando, el Programa-AB es llamado en todo momento mediante código, gestionando el diálogo del agente virtual cada vez que es requerido. A su vez, debido a que únicamente se desea incluir *chatbots* propios, se retira del directorio "/bots" el ejemplo de *chatbot* SUPER que viene por defecto, evitando en cualquier caso su eliminación completa del ordenador puesto que sirve de guía al desarrollador para generar un nuevo *chatbot*.

Seguidamente, se precisa generar los ficheros correspondientes a los *chatbots* que se desea desarrollar, ficheros que seguirán una estructura idéntica a la existente en el directorio SUPER que se ha retirado previamente. El primer paso a la hora de generar los ficheros de un *chatbot* es incorporar un nuevo directorio, cuyo nombre debe coincidir con el del *chatbot* a desarrollar, en el interior del directorio "/bots", directorio que contiene todos los *chatbots* AIML que pueden ser utilizados con el Programa-AB. En el caso del sistema que se está desarrollando, se opta por desarrollar un único *chatbot*, puesto que se pretende que los agentes virtuales sean capaces de mantener una conversación similar independiente del personaje tridimensional que los represente, por lo que carece de sentido desarrollar un *chatbot* específico para cada uno de los personajes de los que se dispone. De esta forma, en del directorio "/bots" se crea un nuevo directorio llamado "clara", el cual debe contener los ficheros que se listan y describen en la Tabla D.7.2:

clara/aiml	Directorio donde se almacenan los ficheros AIML que gestionan el diálogo del chatbot
clara/aimlif	Directorio en el que el Programa-AB almacena los ficheros AIMLIF
clara/config	Directorio en el que se encuentran los ficheros de configuración del chatbot
clara/sets	Directorio con los conjuntos AIML
clara/maps	Directorio que almacena los mapa AIML

Tabla D.7.2: *Tabla con los distintos directorios que deben ser creados para la correcta ejecución del chatbot "clara" en el sistema*

Estos directorios, presentes también en el ejemplo de *chatbot* "SUPER", son los encargados de almacenar los distintos ficheros con los que se programa el nuevo *chatbot*, aunque inicialmente todos ellos se encuentran vacíos. Tras la creación de estos directorios, se procede a generar los distintos ficheros que definirán la gestión de diálogo llevada a cabo por el nuevo *chatbot*, proceso de programación que se explica en el próximo apartado de la memoria principal debido a su complejidad y extensión.

Una vez realizadas las modificaciones comentadas anteriormente y finalizado el proceso de programación del nuevo *chatbot*, se procede a incorporar los ficheros resultantes en el directorio "/storage/sdcard0/BotPFC" de los dispositivos móviles en los que se va a ejecutar el sistema.

Tras haber generado los directorios necesarios para la correcta ejecución del nuevo *chatbot* en los distintos dispositivos móviles utilizados, es necesario desarrollar un mecanismo que permita al sistema contactar con dicho *chatbot* para que realice la gestión de diálogo del agente virtual. Con este fin, se procede a descargar la biblioteca AB.jar del repositorio oficial de la fundación A.L.I.C.E., haciendo uso de dicha biblioteca para implementar un nuevo *plugin* JAVA encargado de inicializar el *chatbot* y gestionar su comunicación con el sistema.

En el Cuadro D.7.1 se muestran las líneas de código más importantes de este *plugin*.

```
import org.alicebot.ab.Bot;
import org.alicebot.ab.Chat;

public class ClaraBot
{
    private Bot bot;
    private Chat chatSession;

    …

    @Override
    protected void onCreate()
    {
        String botname="clara";
        bot = new Bot(botname, "/storage/sdcard0/BotPFC");
        chatSession = new Chat(bot);
    }

    public String generaRespuesta (String frase)
    {
        return chatSession.multisentenceRespond(frase);
    }
}
```

Cuadro D.7.1: *Código JAVA que inicializa el chatbot y gestiona su comunicación con el sistema desarrollado sobre la plataforma Unity 3D*

Inicialmente, se importan las clases Bot y Chat pertenecientes a la biblioteca AB.jar, generando una variable privada de cada clase con los nombres de bot y chatSession respectivamente.

A continuación, se crea una instancia de la clase Bot a través del método constructor *Bot (String botname, String path)*, indicando el nombre del *chatbot* a utilizar y el directorio en el que se encuentran los ficheros asociados a dicho *chatbot*. Este método constructor se encarga de cargar todas las categorías, substituciones, ficheros de configuración y conjuntos definidos para el *chatbot* seleccionado. En este caso, como se ha explicado en los párrafos anteriores, el directorio donde se encuentra el *chatbot* a cargar es "/storage/sdcard0/BotPFC" y su nombre es "clara".

Una vez generado el *chatbot*, se crea una sesión de chat con el mismo a través del método constructor *Chat (Bot bot)*. Este método genera una instancia de la clase Chat sobre la variable privada chatSession, instancia que incorpora el método *String multisentenceRespond (String sentence)* que permite obtener respuesta del *chatbot* a una o más frases de entrada de forma simultánea.

Finalmente, se implementa la función pública *String generaRespuesta (String frase)*, función que permite al sistema trasladar la información proveniente del usuario al *chatbot* a través del parámetro "frase" y obtener la respuesta de éste. De esta forma, esta función gestiona la comunicación entre el módulo Gestor de Diálogo y el sistema.

D.8 Módulo motor

El desarrollo del módulo motor del sistema, tal y como se explica en el apartado 3.5 de la memoria principal, conlleva la implementación de un *plugin* C# específico para cada uno de los personajes tridimensionales que representan al agente virtual en el sistema. En las siguientes líneas se explican en detalle los aspectos más destacados de la gestión de las animaciones llevada a cabo para cada uno de estos personajes.

Niño, niña y perro

Debido a que los tres personajes más simples (niño, niña y perro) son de índole muy similar y poseen animaciones del mismo tipo, los *plugins* C# desarrollados para cada uno de ellos son muy similares entre sí, por lo que se procede a explicar las características más relevantes de la gestión de las

animaciones en uno de estos personajes, más concretamente el niño, haciendo extensible dicha gestión a los otros dos.

Uno de los aspectos más relevantes de la gestión de las animaciones en el personaje tridimensional del niño es que dichas animaciones se encuentran repartidas entre dos capas distintas:

- Capa estándar: también conocida como capa 0 ó capa por defecto, es la capa con menos prioridad de todas las capas existentes en el proyecto Unity 3D. En ella se sitúan tanto las animaciones que representan estados emocionales como las animaciones que representan acciones simples como saludar.

- Capa 1: es la capa inmediatamente superior a la estándar. En ella se localiza la animación "Hablar" del personaje.

Con esta distribución, todas las animaciones del personaje poseen la misma prioridad a la hora de ser reproducidas, exceptuando a la animación "Hablar", que se encuentra un nivel por encima. Esto se debe a que la animación "Hablar" debe poder ser utilizada en combinación con el resto de animaciones, siendo necesario situarla a distinto nivel que las demás animaciones para poder gestionar su reproducción de forma diferente.

En cuanto a la reproducción de las distintas animaciones que incorpora el personaje tridimensional del niño, se hace uso de dos modos de reproducción:

- Reproducción en serie: este tipo de reproducción es el que se utiliza con las animaciones que se encuentran en la capa estándar y consiste en reproducir una animación detrás de otra, siendo necesario que siempre exista una animación en reproducción. En este tipo de reproducción se utilizan los métodos para realizar un proceso de blending automático que incorpora la clase Animation de Unity 3D, a saber: *public AnimationState* ***CrossFadeQueued(animation**: string,* ***fadeLength**: float)* y *public void* ***CrossFade(animation**: string,* ***fadeLength**: float)*. El primer método sirve para comenzar a reproducir una animación concreta en el momento en el que acaba la animación que se encontraba en reproducción, evitando a su vez que hayan saltos o cortes durante el proceso de cambio entre ambas reproducciones. A través del parámetro ***animation*** se indica la nueva animación a reproducir, mientras que con el parámetro ***fadeLength*** se establece el tiempo durante el cual se debe llevar a cabo el cambio progresivo de una animación a otra. Gracias a este método, el sistema se asegura que siempre exista una animación en reproducción, puesto que, como se muestra en el Cuadro D.8.1, la función *void* ***Update()*** se encarga de reproducir la animación neutra en el momento que cualquier otra animación acabe de ser reproducida. Por su parte, el segundo método se utiliza para pasar de la reproducción de una determinada animación a la reproducción de otra animación distinta de forma progresiva y sin cortes, dotando de un mayor realismo y naturalidad al proceso de cambio entre animaciones. Este método es utilizado para comenzar a reproducir cualquiera de las animaciones de la capa estándar una vez que la animación neutra se esté reproduciendo. El uso de los parámetros es idéntico al método anterior.

- Reproducción combinada: este tipo de reproducción es la que se utiliza con la animación "Hablar". Como se ha comentado anteriormente, esta animación se sitúa en una capa superior a la capa estándar con el fin de poder reproducirla en combinación con cualquier otra animación que se esté reproduciendo al mismo tiempo. Para ello, como se muestra en el Cuadro D.8.1, es necesario generar una variable del tipo AnimationState, variable a la que se le asigna la animación "Hablar", la capa 1 y un peso de reproducción de 0.5, dejando el 0.5 restante para la reproducción de las distintas animaciones situadas en la capa estándar. De esta forma, la animación "Hablar", con mayor prioridad por estar situada en una capa superior, no utiliza todo el peso existente durante su reproducción, permitiendo a las animaciones de la capa inferior ser reproducidas simultáneamente.

A continuación se muestran las líneas de código más relevantes del *plugin* C# desarrollado para el personaje tridimensional del niño:

```
private AnimationState hablando;

void Start ()
{
    …

    hablando = animation["Hablar"];
    hablando.layer = 1;
    hablando.weight = 0.5f;

    …
}

void Update()
{
    if (animation.isPlaying)
    {
        animation.CrossFadeQueued("neutro", 1.0f);
    }
}

void gestorDialogo (string texto)
{
    if (texto.Contains("triste"))
    {
        animation.CrossFade("triste", 0.3f);
        animation.Play("Hablar");
    }

   …
}
```

Cuadro D.8.1: *Código perteneciente al plugin C# desarrollado para hacer las funciones de módulo motor con el personaje tridimensional del niño.*

Maxine

Por otro lado, debido a que las animaciones que incorpora el personaje tridimensional Maxine son mucho más complejas y realistas, se concentran en representar pequeños gestos como pestañear, sonreír o pronunciar una solo letra del abecedario, por lo que no es extraño que más de dos de estas animaciones sean reproducidas simultáneamente. Es por ello que la gestión de las animaciones de este personaje es diferente a la de los anteriores, debiendo desarrollar un módulo motor completamente distinto para Maxine. En las siguientes líneas se procede a explicar las características más relevantes de la gestión de las animaciones en este personaje.

Uno de los aspectos más relevantes de la gestión de las animaciones en el personaje tridimensional de Maxine es que dichas animaciones se encuentran repartidas entre cuatro capas distintas:

- Capa estándar: también conocida como capa 0 ó capa por defecto, es la capa con menos prioridad de todas las capas existentes en el proyecto Unity 3D. En ella se sitúa únicamente la animación neutra, animación que es utilizada como base para el resto de animaciones a lo largo de la ejecución del sistema.

- Capa 1: es la capa inmediatamente superior a la estándar. En ella se localiza las animaciones que representan los distintos estados emocionales del agente virtual.

- Capa 2: en esta capa se sitúa la animación “blink”, encargada de reproducir el pestañeo del agente a lo largo de la ejecución del sistema.

- Capa 3: es la capa más alta que se utiliza, situando en ella todas las animaciones correspondientes a la sincronización labial.

Con esta distribución, se pretende gestionar de forma independiente las animaciones de distinto tipo, siendo posible la reproducción de una animación de cada tipo simultáneamente.

En cuanto a la reproducción de las distintas animaciones que incorpora el personaje tridimensional Maxine, se hace uso de varios modos de reproducción:

- Reproducción base: este tipo de reproducción es la que se utiliza con la animación neutra que se encuentra en la capa estándar. La reproducción base consiste en asegurar en todo momento que al menos una animación se encuentra en reproducción, condición obligatoria para llevar a cabo otros modos de reproducción como la combinada o la aditiva que también se usan sobre el personaje tridimensional de Maxine. Debido a que la animación neutra mantiene estático al personaje, es posible reproducirla tantas veces como se desee sin que el usuario sea capaz de distinguir los cambios entre una reproducción y la siguiente, por lo que se opta por reproducir la animación neutra en el modo "Loop" (bucle). De esta manera, el modo "Loop" reproduce constantemente la animación neutra, volviendo a reproducirla nuevamente cada vez que termina la reproducción anterior, evitando de esta forma que no haya ninguna animación en reproducción.

- Reproducción en serie: este tipo de reproducción se utiliza con las animaciones que se encuentran en la capa 1, es decir, las animaciones que representan los distintos estados emocionales del agente virtual. Como se ha comentado anteriormente, la reproducción en serie únicamente puede ser utilizada entre animaciones pertenecientes a una misma capa y consiste en reproducir una animación detrás de otra, permaneciendo siempre una de estas animaciones en reproducción. Para este tipo de reproducción se sigue utilizando el método de clase Animation *public void* ***CrossFade(animation****: string,* ***fadeLength****: float)*, aunque se descarta el uso del método *public AnimationState* ***CrossFadeQueued(animation****: string,* ***fadeLength****: float)*, utilizado en los *scripts* C# del resto de personajes tridimensionales, ya que las animaciones correspondientes a los estados emocionales del agente son reproducidas en el modo "ClampForever", modo que reproduce la animación en su totalidad y mantiene en reproducción el último fotograma de la misma hasta que una nueva animación es reproducida. Este cambio se debe a que las animaciones de la capa 1 poseen una duración muy breve, por lo que la gestión con el método ***CrossFadeQueued*** hacía casi inapreciables dichas animaciones al usuario. Sin embargo, a través del método "ClampForever" la animación permanece en reproducción el tiempo que estime necesario el desarrollador, permitiendo que el usuario sea capaz de percibirla fácilmente.

- Reproducción combinada: este tipo de reproducción es el que se utiliza entre las animaciones pertenecientes a la capa 1 y la animación neutra de la capa estándar, aunque en ningún momento se pretende combinar las animaciones de ambas capas. El objetivo que se persigue es simular el funcionamiento de una reproducción en serie entre las animaciones de la capa 1 y la animación neutra, animación que se encuentra una capa por debajo y está reproduciéndose continuamente para servir de base a otros tipos de reproducción. Para ello, se opta por utilizar la reproducción combinada de una forma muy concreta, obligando en todo momento a que las animaciones de la capa 1 acaparen completamente el peso de reproducción, evitando de esta forma mezclarse con la animación neutra pero manteniendo a dicha animación en reproducción. De esta manera, las animaciones de la capa 1 pueden ser reproducidas de forma independiente a la animación neutra de la capa estándar, la cual únicamente actúa como base para otros tipos de reproducción.

- Reproducción aditiva: este tipo de reproducción se utiliza con las animaciones pertenecientes a las dos últimas capas, esto es, con las animaciones encargadas del pestañeo y el movimiento labial. La reproducción aditiva se basa en añadir los efectos de una determinada animación a una o varias animaciones se estén reproduciendo en ese mismo momento, siendo necesario disponer de una animación base que se reproduzca constantemente para asegurar en correcto funcionamiento de este tipo de reproducción. En este sentido, en el caso de que ninguna animación de la capa 1 esté siendo reproducida, la reproducción aditiva se llevaría a cabo sobre la animación neutra que actúa como base; mientras que si se estuviera reproduciendo cualquiera de las animaciones de la capa 1, al adquirir todo el peso de reproducción, actuarían como base para la reproducción aditiva en lugar de la animación neutra anterior. Además, dado que la animación de pestañeo se localiza en una capa distinta a las animaciones labiales, es posible realizar una reproducción aditiva simultánea de ambas, de forma que el personaje es capaz de mostrar el estado emocional en el que se encuentra mientras habla y pestañea.

A continuación, en el Cuadro D.8.2, se muestran algunas líneas de código del *plugin* C# desarrollado. En estas líneas de código se puede observar un ejemplo de cada uno de los tipos de reproducción llevados a cabo en el módulo motor del personaje tridimensional de Maxine.:

```
/*Variables de estado*/
private AnimationState hablando;
private AnimationState estadoBase;
private AnimationState estadoAnimico;
private AnimationState pestaneo;

void Start ()
{
    …

    /*Estado base*/
    estadoBase = animation["neutroNormal"];
    estadoBase.wrapMode = WrapMode.Loop;
    animation.Play ("neutroNormal");

    /*Pestañear*/
    pestaneo = animation["blink"];
    pestaneo.speed = 0.3f;
    pestaneo.layer = 2;
    pestaneo.blendMode = AnimationBlendMode.Additive;
    pestaneo.wrapMode = WrapMode.Once ;
    StartCoroutine(doBlinkAnimation());

    /*Estado anímico*/
    estadoAnimico = animation["sonreir"];
    estadoAnimico.layer = 1;
    estadoAnimico.wrapMode = WrapMode.ClampForever;
    estadoAnimico.weight = 1.0f;

    /*Hablar*/
    hablando = animation["hablar"];
    hablando.layer = 3;
    hablando.blendMode = AnimationBlendMode.Additive;
    hablando.wrapMode = WrapMode.Once ;

    …

    /*Saludo inicial*/
    animation.CrossFade("sonreir");
    animation.CrossFade("hablar");

    …
}

…

IEnumerator doBlinkAnimation()
{
    while (true)
    {
        // Wait for 5 to 8 seconds
        yield return new WaitForSeconds(Random.Range(5, 8));

        // Play the animation
        animation.CrossFade("blink", 0.03f);

        // Wait for the animation to complete
        yield return new WaitForSeconds(animation["blink"].length);
    }
}
```

Cuadro D.8.2: *Código perteneciente al plugin C# desarrollado para hacer las funciones de módulo motor con el personaje tridimensional de Maxine*

D.9 Interfaz Gráfica

En las siguientes líneas se describe el reconocimiento y la gestión realizada cada vez que el usuario pulsa sobre uno de los botones físicos. Cabe recordar que el usuario se sirve de estos botones para mostrar el menú de opciones, controlar el volumen de la aplicación o cerrar el sistema (véase apartado 3.6 de la memoria principal).

Con el fin de que el sistema sea conocedor de que el usuario ha pulsado alguno de los botones físicos del dispositivo, es necesario capturar el evento que provoca dicho botón al ser pulsado. Para ello, se añade una nueva función en el *plugin* JAVA desarrollado anteriormente, función que se encarga de gestionar todos los eventos producidos por los distintos botones físicos del dispositivo móvil. Esta función, llamada "*onKeyDown*" viene mostrada en el Cuadro D.9.1.

```
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN)
    {
        if (volumen>=5.0f)
        {
            this.volumen=this.volumen-1.0f;
            setVolume(this.volumen);
        }
        else
        {
            this.volumen=0.0f;
            setVolume(0.0f);
        }
        UnityPlayer.UnitySendMessage("Maxine Prefab", "cambiaVolumenPorHardware", "0");
        return true;
    }
    else if (keyCode == KeyEvent.KEYCODE_VOLUME_UP)
    {
        if (volumen<=15.0f)
        {
            this.volumen=this.volumen+1.0f;
            setVolume(this.volumen);
        }
        else
        {
            this.volumen=16.0f;
            setVolume(16.0f);
        }
        UnityPlayer.UnitySendMessage("Maxine Prefab", "cambiaVolumenPorHardware", "1");
        return true;
    }
    else if (keyCode == KeyEvent.KEYCODE_MENU)
    {
        UnityPlayer.UnitySendMessage("Maxine Prefab", "botonMenu", "");
        return true;
    }
    else if (keyCode == KeyEvent.KEYCODE_BACK)
    {
        UnityPlayer.UnitySendMessage("Maxine Prefab", "botonSalida", "");
        return true;
    }
  return true;
}
```

Cuadro D.9.1: *Función en código JAVA que captura el evento lanzado al pulsar cualquier tecla o botón del dispositivo*

La función "*onKeyDown*" es invocada en el momento que se produce un evento causado por el pulsado de una tecla, recibiendo como argumentos el identificador de la tecla pulsada (*keyCode*) y una descripción del evento capturado (*event*). Para poder reconocer si el usuario ha pulsado una determinada tecla, basta con comparar el identificador recibido como argumento con la constante que identifica dicha tecla, permitiendo gestionar de esta forma los eventos producidos por el pulsado de las distintas teclas y botones existentes en el dispositivo.

Menú de opciones

Para reconocer el pulsado del botón físico "Menú", la función compara el identificador recibido con la constante KeyEvent.KEYCODE_MENU. En caso de coincidir, se invoca a la función "botonMenu()" desarrollada en Unity 3D, función que se encarga de mostrar por pantalla el menú de opciones del sistema.

Control de volumen

Por su parte, con el fin de reconocer y gestionar el pulsado de los botones físicos de control del volumen, la función "*onKeyDown*" debe realizar un par de comprobaciones sobre el identificador de tecla recibido como argumento. En primera instancia, se compara dicho identificador con la constante correspondiente al botón de disminución del volumen, KeyEvent.KEYCODE_VOLUME_DOWN. En caso de coincidir, si el valor del volumen del sistema es superior a 4, se reduce en un punto dicho valor del volumen; mientras que si el valor del volumen del sistema es inferior o igual a 4, al tratarse del valor límite audible por el usuario, se procede a silenciar el dispositivo otorgando el valor 0 al volumen del sistema. Por otro lado, en caso de resultar negativa la primera comparación realizada sobre el identificador de tecla recibido como argumento, se lleva a cabo una segunda comparación, esta vez con la constante correspondiente al botón de aumento de volumen, KeyEvent.KEYCODE_VOLUME_UP. En caso de coincidir, si el valor del volumen del sistema es inferior o igual a 15, se aumenta un punto dicho valor del volumen; otorgando en cualquier otro caso el valor máximo de volumen seleccionado por el desarrollador, 16. De esta forma, el usuario es capaz de controlar el volumen del sistema a través de los botones físicos del dispositivo destinados para ello.

Cierre del sistema

Para el reconocimiento y gestión del pulsado del botón físico "Atrás", el sistema se sirve de la función "*onKeyDown*". Esta función, mostrada con anterioridad en el Cuadro D.9.1, lleva a cabo la comparación entre el identificador de tecla recibido como argumento y la constante correspondiente al botón físico "Atrás", KeyEvent.KEYCODE_BACK. En caso de resultar positiva la comparación, se procede a invocar a la función "botonSalida()" desarrollada en Unity 3D, la cual invita al agente virtual a despedirse cortésmente del usuario, para seguidamente cerrar la aplicación.

Anexo E. Prototipo funcional de la interfaz gráfica

Este anexo complementa al capítulo 3 de la memoria principal, más concretamente al apartado 3.6 Interfaz Gráfica. En este documento se presenta la fase de desarrollo del prototipo funcional de la interfaz gráfica, detallando los requisitos identificados y el diseño llevado a cabo para este prototipo.

E.1 Requisitos identificados

El objetivo principal de este primer prototipo de la interfaz gráfica es permitir la comprobación, de forma rápida y sencilla, de las distintas funcionalidades implementadas en el sistema. En las siguientes líneas se especifican los requisitos identificados para llevar a cabo este cometido.

En primer lugar, el prototipo funcional de la interfaz gráfica del sistema debe permitir al desarrollador cambiar el modo de interacción con el agente virtual de una forma simple. En este sentido, no es necesario poder combinar los distintos canales de interacción, únicamente se precisa que sea posible hacer uso de cada uno de dichos canales de interacción en un momento determinado, pudiéndose comprobar así su correcto funcionamiento.

Por otro lado, se considera interesante añadir a este primer prototipo de la interfaz gráfica algún elemento que permita la ejecución de acciones de forma asíncrona por parte del agente virtual. De esta manera, se permite al desarrollador realizar pruebas con las distintas animaciones que incorpora el agente virtual así como ejecutar directamente las nuevas funcionalidades en desarrollo para comprobar los cambios realizados sobre ellas.

Finalmente, el prototipo funcional de la interfaz debe otorgar al desarrollador la posibilidad de modificar el volumen del sistema y concluir la ejecución del mismo.

E.2 Diseño del prototipo funcional

Una vez definidas las funcionalidades mínimas que debe poseer este prototipo funcional de la interfaz gráfica del sistema, se inicia el proceso de diseño del mismo. En la Figura E.2.1 se muestra una imagen de la interfaz gráfica correspondiente a este prototipo funcional.

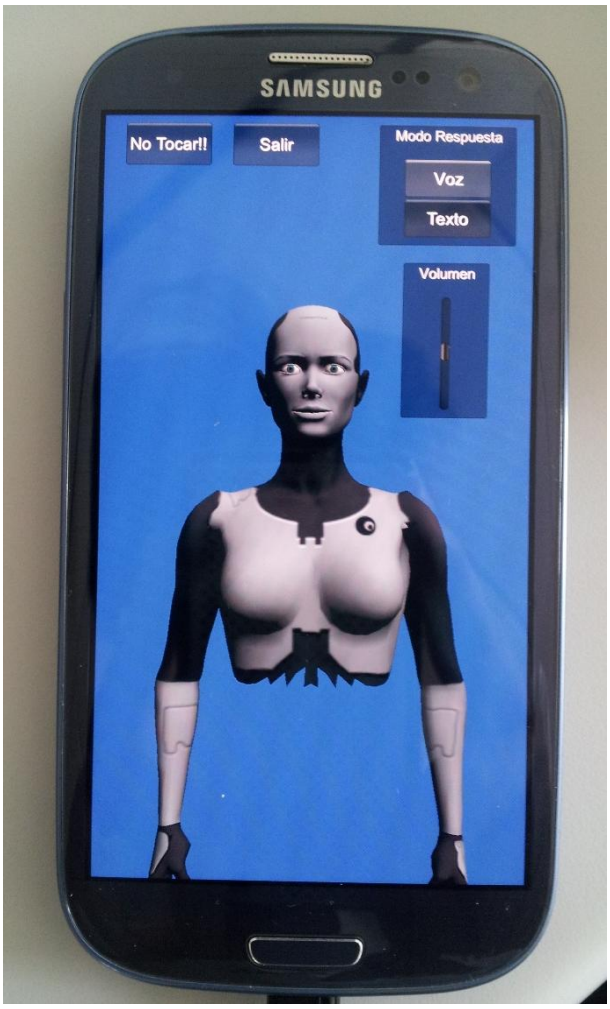


Figura E.2.1: *Imagen de la interfaz gráfica del prototipo funcional*

Como se puede observar, esta interfaz gráfica está conformada por la representación centrada del agente virtual, un conjunto de botones en la parte superior de interfaz y el control del volumen. A continuación se explican las características más importantes de este prototipo funcional.

Agente virtual

Al estar desarrollando un sistema basado en agente virtual, se opta por que dicho agente fuese el elemento central de la interfaz gráfica, situado en todo momento de frente al desarrollador. La representación gráfica del agente virtual se lleva a cabo a través del torso de uno de los personajes tridimensionales de los que disponía el grupo de trabajo, más concretamente, el torso de la mujer de Maxine. La elección de este personaje se debe a que las animaciones que incorpora son de una mayor complejidad que las incorporadas por el resto de personajes tridimensionales disponibles, permitiendo realizar un mayor rango de pruebas con las expresiones faciales del agente virtual y la sincronización labial.

Por otro lado, se decide hacer del agente virtual un elemento activo dentro de este prototipo inicial, añadiendo a dicho agente un "*box collider*" que detecta si el usuario pulsa sobre él. De esta forma, el desarrollador se sirve del agente virtual para dar comienzo al proceso de reconocimiento de voz, iniciando la escucha del discurso cada vez que el agente sea pulsado.

Modos de interacción

Con respecto a los modos de interacción, debido a que simplemente se precisa disponer de algún mecanismo que permita activar cada uno de los canales de interacción existentes para la comprobación de su correcto funcionamiento, se opta por permitir únicamente dos tipos de comunicación, la oral y la

escrita. En este sentido, si el desarrollador se decanta por hacer uso de la comunicación oral con el agente virtual, tanto la entrada como la salida de información se llevan a cabo a través de discursos orales; mientras que si se decide por utilizar la comunicación escrita, la entrada y salida de información se realizan en formato de texto. De esta forma, todos los canales de interacción pueden ser activados por el desarrollador en un momento determinado, aunque no es posible combinarlos entre sí.

Con el fin de permitir al desarrollador seleccionar cualquiera de los dos tipos de comunicación disponibles de forma rápida y sencilla, se opta por incorporar a la interfaz el panel "Modos Interacción". Dicho panel, situado en la esquina superior derecha, contiene una matriz de selección con dos botones, titulados "Oral" y "Escrito", que se encargan de gestionar la activación de los diversos servicios necesarios para establecer la comunicación con el agente virtual a través del canal oral o escrito respectivamente. Estos botones permanecen en todo momento en distinto estado, de manera que si el desarrollador selecciona el botón "Oral", el botón "Escrito" pasa a estar no seleccionado de forma instantánea y viceversa, permitiendo un único tipo de comunicación con el agente virtual.

Control de volumen

Por otra parte, se desarrolla un controlador de volumen táctil que permite gestionar el nivel de volumen en el sistema. Para ello, el controlador se sirve de las funciones de gestión del volumen implementadas en el *plugin* JAVA comentado en el apartado 3.2 de la memoria principal, como por ejemplo la función "cambiaVolumen", función que se presenta en el Cuadro 3.2.5 de dicha memoria principal. Este controlador de volumen, situado inmediatamente debajo de la matriz de selección anterior, está formado por una caja con una barra de desplazamiento en su interior.

Botón "No Tocar"

Al tratarse de un prototipo inicial de la interfaz gráfica, cuya finalidad es dar acceso a las funcionalidades básicas y facilitar el desarrollo del sistema, se considera adecuado incorporar un nuevo botón que permita comprobar, de forma inmediata, la correcta ejecución de acciones concretas. Este botón, situado en la esquina superior izquierda de la interfaz y titulado "No Tocar", permite al desarrollador realizar pruebas con las distintas animaciones que incorpora el agente virtual así como ejecutar, sin necesidad de implementar elementos más complejos sobre la interfaz, las nuevas funcionalidades en desarrollo para comprobar los cambios realizados sobre ellas.

Cierre del sistema

Finalmente, para llevar a cabo el cierre del sistema se opta por implementar un botón "Salir", el cual se sitúa justo encima del agente virtual, en el centro de la parte superior de la interfaz. Este botón, al ser pulsado, da por concluida la interacción con el agente virtual, invita al agente a despedirse cortésmente del desarrollador y procede cerrar la aplicación.

Anexo F. Generación de voces emocionales

En este anexo se explica el proceso de determinación de los parámetros que definen las voces correspondientes a las cinco emociones consideradas en el sistema a desarrollar: neutra, sorpresa, alegría, tristeza y enfado. En los siguientes apartados se explica la metodología seguida a lo largo de este proceso así como se detallan cada una de las pruebas llevadas a cabo.

F.1 Metodología

Antes de definir las diferentes pruebas a realizar en este proceso de generación de voces emocionales, es necesario determinar la estructura que va a seguir este proceso así como establecer los usuarios que participarán en las pruebas y la metodología para llevarlas a cabo. A continuación se explican los detalles.

F.1.1 Estructura del proceso de generación de las voces emocionales

En primer lugar, se procede a explicar la estructura que se va a seguir durante el proceso de generación de las voces emocionales. Esta estructura, mostrada en la Figura F.1.1, la conforman dos fases principales, como son las fases de calibración y selección, las cuales integran a su vez distintos módulos y operaciones. En las siguientes líneas se describe brevemente todos estas estapas de la estructura:

- **Fase de Calibración:** esta primera fase tiene como objetivo calibrar las voces emocionales generadas por el evaluador. Esta fase de calibración consta de dos módulos de pruebas, dos etapas de modificación de las voces emocionales y una última etapa de reagrupamiento de dichas voces en bloques emocionales. A continuación se explican todos estos elementos de la fase de calibración:
    - **Módulo de Pruebas:** los usuarios deben evaluar los distintos bloques de voces emocionales generados a través de dos encuestas, una de elección libre y otra de elección libre modificada.
    - **Modificación de las voces emocionales:** etapa en la que el evaluador varía los valores de los parámetros del discurso utilizados en cada una de las voces emocionales generadas en función de los resultados obtenidos en el módulo de pruebas previo.
    - **Reagrupamiento de las voces en bloques emocionales:** las voces generadas para representar una misma emoción del agente virtual se reagrupan dentro de un mismo bloque, generando así cinco bloques emocionales distintos. Estos bloques sirven de base a la fase de selección posterior.
- **Fase de Selección:** esta segunda fase tiene como fin seleccionar, para cada uno de los estados emocionales que se desea representar, la voz emocional que mejor transmita la emoción correspondiente. Esta fase de selección consta de dos módulos de pruebas, dos cribas de voces emocionales y sendas etapas donde se generan los bloques y voces emocionales definitivos respectivamente. A continuación se explican todos estos elementos de la fase de calibración:
    - **Módulo de Pruebas:** los usuarios deben evaluar los distintos bloques de voces emocionales generados a través de una encuesta. En el caso del tercer módulo de pruebas, la encuesta es de elección forzada, mientras que en el cuarto se utiliza la elección libre modificada.

- **Criba de voces emocionales:** etapa en la que el evaluador elimina las voces emocionales peor valoradas por los usuarios, sirviéndose para ello de los resultados obtenidos en el módulo de pruebas previo.

- **Generación de los bloques definitivos:** tras la primera criba de voces emocionales, se generan los bloques definitivos con los que se lleva a cabo el último módulo de pruebas de este proceso de generación de voces emocionales.

- **Definición de las voces definitivas:** una vez llevadas a cabo ambas cribas, se realiza un análisis de las voces emocionales restantes con el objetivo de definir una única voz emocional para cada uno de los estados emocionales del agente virtual.

Figura F.1.1: *Estructura del proceso de generación de las voces emocionales a utilizar en el sistema.*

F.1.2 Participantes y localización de la encuesta

Para la realización de los distintos módulos de pruebas presentes en este proceso de generación de voces emocionales, el evaluador se sirve de grupos de usuarios reducidos y heterogéneos que se encuentran a su alcance, más concretamente, sus familiares y amigos. Cabe destacar que, si bien el conjunto de usuarios encuestados en una misma fase del proceso es idéntico, varía de una fase a otra, siendo ligeramente mayor el número de usuarios encuestados en la fase de selección. En las siguientes líneas se explica el conjunto de usuarios utilizado en cada fase:

- Fase de Calibración: los usuarios encuestados proceden todos de la familia del evaluador. En total se encuestan a seis usuarios, tres varones y tres mujeres, de distinta edad y formación.

- Fase de Selección: los usuarios encuestados proceden tanto de la familia como del grupo de amigos del evaluador. En total se encuestan a nueve usuarios, cinco mujeres y cuatro varones, de distinta edad y formación. Cabe destacar que cinco de los nueve usuarios encuestados ya habían sido encuestados en la fase anterior.

Los distintos módulos de prueba se llevan a cabo en el domicilio del evaluador, más concretamente en el estudio, con el fin de evitar el ruido exterior que pudiera dificultar la escucha de las voces emocionales generadas. Además, todos los módulos de pruebas realizados en este proceso de generación de voces emocionales se llevan a cabo por la tarde, entre las 17:00 y las 20:00 horas.

Una vez definida la metodología, se procede a explicar en detalle el proceso de generación de voces emocionales, el cual se encuentra dividido en dos grandes bloques: calibración y selección.

F.2 Calibración de las voces emocionales

La primera fase de pruebas con usuarios tiene con objetivo principal calibrar las voces emocionales generadas por el desarrollador.

Como se ha explicado en capítulos anteriores de la memoria principal, se opta por hacer uso del sistema TTS de Android para reproducir el discurso del agente virtual. Este sistema TTS únicamente permite manipular tres parámetros del discurso, como son el volumen, la velocidad y el tono, lo que limita las posibilidades de modelar voces emocionales con un alto grado de realismo. De cualquier modo, en el apartado 3.1.3 de esta memoria se explica cómo, tras recabar información y haciendo uso de trabajos anteriores, se consigue delimitar, en cierta medida, el rango de valores entre los que debieran moverse cada uno de estos parámetros del discurso en función de la emoción que se desee modelar.

Una vez definidos los rangos de valores anteriores para cada uno de los parámetros del discurso, comienza el proceso de calibración de las voces emocionales.

F.2.1 Primer módulo de pruebas

El proceso de calibración de las voces emocionales comienza con un primer módulo de pruebas con usuarios cuyo objetivo es determinar qué conjuntos de valores hacen más reconocibles las emociones transmitidas a través de la voz sintetizada. A su vez, se pretende detectar cualquier anomalía o resultado inesperado para tenerlos en consideración en pruebas posteriores.

Este primer módulo de pruebas toma como punto de partida los rangos de valores seleccionados con anterioridad para los parámetros del discurso, utilizando dichos rangos como guía para generar distintos bloques de voces emocionales. La generación de estos bloques sigue un patrón similar para todas las emociones modeladas: se parte de un valor intermedio del rango seleccionado para cada uno de los parámetros del discurso y, a través de ligeras modificaciones en el valor de uno u otro parámetro, se generan bloques de tres o cuatro voces distintas para una misma emoción.

Tras haber generado del orden de dos o tres bloques por emoción, se procede a evaluar dichos bloques con usuarios finales. Al tratarse de un primer acercamiento a la generación de voces emocionales realistas, la prueba se lleva a cabo sobre un grupo reducido pero heterogéneo de usuarios, más concretamente, mi familia. Cabe destacar que, con el fin de no incorporar ningún componente emocional adicional que no fuera la voz, a lo largo de todo el módulo de pruebas se hace uso de una misma frase neutra (“Los viernes la fruta está mucho más barata”) y el agente virtual no aparece en ningún momento en la pantalla. Además, el orden en el que van reproduciéndose los bloques es aleatorio.

Elección libre

En primer lugar, se opta por utilizar la método de evaluación de la elección libre. A través de este método se pretende conocer, expresadas con sus propias palabras, las emociones que transmiten las voces modeladas a los usuarios encuestados.

Una de las conclusiones más importantes de esta primera prueba es la dificultad que tienen los usuarios para describir la emoción percibida a través de la voz. Esto se debe, en la gran mayoría de los casos, al poco léxico referido a estados emocionales que tiene en mente el usuario a la hora de realizar la prueba, lo que le lleva a utilizar ejemplos gráficos para dar a entender la emoción percibida (por ejemplo, "parece una secretaria con prisa por llegar a casa"). De cualquier modo, los resultados obtenidos permiten extraer múltiples conclusiones para pruebas posteriores:

- Triste: los bloques que intentan expresar un sentimiento de tristeza son reconocidos por los usuarios como bloques negativos, transmitiéndoles sensaciones de resignación, decepción, aburrimiento o tristeza. A su vez, se comprueba que para valores de velocidad inferiores a 0.7 el discurso se ralentiza en exceso, dificultando el reconocimiento de la emoción. Además, se observa que valores inferiores a 1.0 en el tono provocan que el discurso se reproduzca de forma monótona y robótica, lo que desvirtúa la emoción que se pretende transmitir.

- Enfadado: del mismo modo que en el caso anterior, los bloques con voces de emocionales de enfado son reconocidos por los usuarios como bloques negativos. Además, son varios los usuarios que reconocen la emoción de enfado, aunque términos como borde, iracunda o antipática también son utilizados para describir estos bloques. Se observa que valores de velocidad superiores a 1.8 no son recomendables puesto que dificulta el reconocimiento de cualquier emoción en el discurso. Además, el uso de tonos con valores por encima de 1.4 confunde al usuario con emociones contrarias como la alegría.

- Neutro: en el caso de los bloques que representan el estado anímico neutro, la mayoría de los usuarios reconocen la carencia de emoción en ellos, aunque algunos perciben cierta tristeza en el discurso. De cualquier modo, lo más preocupante es que los usuarios encuestados transmiten, de forma casi unánime, que la voz emocional neutra es la que más irreal les resulta puesto que les recuerda a la voz de un robot.

- Alegre: los resultados obtenidos para los bloques que contienen las voces emocionales alegres no son demasiado buenos. Si bien algunos usuarios reconocen cierto grado de alegría en estos bloques, la mayoría los asocian a estados de excitación, como la sorpresa, la histeria o incluso el miedo. Se comprueba que cuanto mayor es la velocidad y el tono del discurso utilizados para reproducir el estado anímico de alegría, peores resultados se obtienen.

- Sorprendido: en el caso de los bloques con voces emocionales de sorpresa, los resultados obtenidos son totalmente inesperados. Por un lado, tan sólo un usuario reconoce un estado de sorpresa en las voces modeladas, por lo que los resultados son innegablemente malos. Sin embargo, una inmensa mayoría de los usuarios encuestados relacionan las voces emocionales modeladas con estados de alegría, felicidad o entusiasmo, estados emocionales que se pretendían expresar, sin éxito, en el caso anterior. De esta forma, el desarrollador obtiene un punto de partida nuevo para generar voces emocionales alegres.

A pesar de que la información recabada es suficiente para proseguir con el modelado de voces emocionales más realistas, se procede a llevar a cabo una segunda prueba debido a la heterogeneidad de las respuestas obtenidas y a las múltiples interpretaciones que ha tenido que hacer el desarrollador de dichas respuestas.

Elección libre modificada

El método de evaluación seleccionado para esta segunda prueba es la elección forzada modificada. Con este método se pretende dotar al usuario de un conjunto bastante amplio de términos emocionales que evite los problemas de léxico detectados en la prueba anterior y, por consiguiente, las interpretaciones a posteriori llevadas a cabo por el desarrollador de las respuestas de índole gráfica de los usuarios.

Antes de comenzar la prueba, es necesario seleccionar el conjunto cerrado de términos entre los que el usuario debe elegir su respuesta. En primer lugar, como es obvio, se incluyen los cinco estados emocionales que se pretenden lograr expresar a través del discurso, como son ALEGRE, SORPRENDIDO, NEUTRO, TRISTE y ENFADADO. A estos cinco estados anímicos se les adicionó otros tres estados más, DECEPCIONADO, ASUSTADO y ENTUSIASMADO, y una última opción OTROS que evitara al usuario responder en caso de no haber reconocido ninguno de los estados anteriores.

Esta segunda prueba se lleva a cabo de forma idéntica a la prueba previa, utilizando los mismos bloques y en las mismas condiciones de aleatoriedad. En las siguientes líneas se analizan los resultados.

- Triste: de nuevo, los bloques que poseen voces emocionales tristes son reconocidos por los usuarios como bloques negativos. Sin embargo, la opción triste es seleccionada en la misma medida que las opciones decepcionado y asustado.

- Enfadado: los bloques con voces emocionales de enfado son los que mejores resultados obtienen en esta prueba, puesto que la gran mayoría de los usuarios encuestados seleccionan la opción enfadado al escucharlos. De cualquier modo, se observa que en los casos donde el tono se acerca a valores estándares los resultados empeoran, llegando los usuarios incluso a confundir las voces emocionales que denotan enfado con las voces emocionales neutras.

- Neutro: en el caso de los bloques con voces emocionales neutras, los resultados son aceptables, aunque se pueden mejorar en gran medida. Por un lado, más de la mitad de los usuarios encuestados reconocen dichos bloques como neutros, lo que lleva a pensar que los parámetros utilizados para modelar las voces emocionales son acertados. Sin embargo, muchos usuarios reiteran tener la sensación de estar escuchando a un robot, llegando algunos a seleccionar la opción triste a causa de la falta de expresividad de las voces emocionales escuchadas en estos bloques.

- Alegre: al igual que sucediese en la prueba anterior, los resultados obtenidos por los bloques con voces emocionales de alegría no son demasiado buenos. De hecho, la opción seleccionada por la mayoría de los usuarios encuestados para estos bloques es sorprendida, muy por delante de alegre. Se comprueba nuevamente que cuanto mayor son los valores de velocidad y tono del discurso, peor reconocen la emoción de alegría los usuarios.

- Sorprendido: los resultados obtenidos por los bloques con voces emocionales de sorpresa no hacen más que confirmar la tendencia que se intuía en la prueba anterior. La inmensa mayoría de los usuarios encuestados reconoce un estado emocional de alegría en las voces escuchadas en estos bloques, mientras que tan sólo dos usuarios reconocen un estado de sorpresa.

Tras analizar estos resultados y haber sacado las conclusiones pertinentes, se comienza un proceso de modificación de las voces emocionales modeladas para cada bloque con el fin de mejorar los resultados obtenidos en este primer módulo de pruebas con usuario.

F.2.2 Modificación de las voces emocionales

Este proceso tiene como objetivo modelar voces emocionales con un mayor grado de realismo que las del primer módulo de pruebas. Para ello, se hace uso de las conclusiones extraídas de los resultados obtenidos en el módulo de pruebas anterior, modificando los parámetros del discurso para cada uno de los estados anímicos que se desea modelar. A continuación se detallan las medidas adoptadas para mejorar los resultados de cada una de las emociones modeladas.

- Triste: en primer lugar, se decide que el rango de valores para la velocidad del discurso sea [0.75, 0.95], de forma que no se ralentice demasiado la reproducción del discurso ni se alcance los valores utilizados para representar el estado emocional neutro. A su vez, se opta por utilizar el rango de valores [1.1, 1.7] para el tono, puesto que valores inferiores a 0.9 robotizan demasiado el discurso y el uso del valor estándar 1.0 confunde mucho al usuario con la voz emocional neutra.

- Enfadado: a pesar de los buenos resultados obtenidos en el primer módulo de pruebas, se considera necesario llevar a cabo una serie de modificaciones en los bloques con voces emocionales de enfado. Por un lado, se opta por utilizar valores de velocidad por encima de lo estándar, pero sin superar el valor de 1.8 que dificulta el entendimiento del discurso por parte del usuario. A su vez, se delimita los valores del tono de forma considerable, considerando sólo valores comprendidos en el rango [1.1, 1.4]. Experimentalmente ha quedado demostrado que para valores de tono inferiores el usuario se confunde con la voz emocional neutra, mientras que para valores superiores a ese rango la voz emocional de enfado se empieza a confundir con la de alegría.

- Neutro: el principal problema a solucionar con la voz emocional neutra es reducir el carácter robótico que inspira a los usuarios. Para ello se procede a realizar variaciones en el tono del discurso utilizado, puesto que tanto volumen como la velocidad de discurso utilizados se consideran adecuados a tenor de los resultados obtenidos. De esta forma, se opta por utilizar el rango de valores [1.1, 1.7] para el tono de las voces emocionales neutras.

- Alegre: a raíz de los resultados obtenidos en el primer módulo de pruebas, se opta por invertir los rangos seleccionados para las voces emocionales de alegría y sorpresa. En este sentido, para representar un estado anímico alegre se opta por utilizar un rango inferior de velocidad de discurso [1.0, 1.5], el rango de volumen establecerlo en [10.0, 14.0] y el rango de valores posibles para el tono situarlo en [1.7, 2.2].

- Sorprendido: en el caso de las voces emocionales de sorpresa, el proceso es el inverso al de las voces emocionales de alegría. De este modo, se elevan ligeramente el rango de posibles valores para el volumen [11.0, 16.0] y la velocidad [1.2, 1.8] del discurso. A su vez, el rango de posibles valores para el tono del discurso pasa a ser [2.0, 2.5].

Una vez modificados los rangos de posibles valores de cada uno de los parámetros del discurso para cada estado emocional a representar, se procede a llevar a cabo un segundo módulo de pruebas con usuarios.

F.2.3 Segundo módulo de pruebas

Este segundo módulo de pruebas tiene como objetivo comprobar si los nuevos rangos de valores establecidos para cada uno de los parámetros del discurso hacen más reconocibles los estados anímicos que se pretenden expresar a través de las voces emocionales al usuario. En el caso de que los resultados obtenidos sean positivos, se pretende reorganizar los bloques en función de las respuestas de los usuarios encuestados para dar comienzo a la fase de selección de las voces emocionales.

En primer lugar, se toma como punto de partida los rangos de valores establecidos a partir de los resultados obtenidos en el primer módulo de pruebas con usuarios, sirviéndose el desarrollador de dichos rangos para generar los distintos bloques de voces emocionales. La generación de estos bloques sigue el mismo patrón que en el bloque de pruebas anterior: se parte de un valor intermedio del rango seleccionado para cada uno de los parámetros del discurso y, a través de ligeras modificaciones en el valor de uno u otro parámetro, se generan bloques de tres o cuatro voces distintas para una misma emoción.

Tras haber generado del orden de dos o tres bloques por emoción, se procede a evaluar dichos bloques con usuarios finales. De nuevo, la prueba se lleva a cabo sobre un grupo reducido pero heterogéneo de usuarios, más concretamente, los amigos y familiares cercanos del desarrollador. Cabe destacar que, al igual que sucediera en el primer módulo de pruebas, no se pretende incorporar ningún componente emocional adicional que no sea la voz. Es por ello que en todo momento se hace uso de la misma frase neutra (“Los viernes la fruta está mucho más barata”) y no se visualiza al agente virtual en la pantalla del dispositivo. Además, el orden en el que se van reproduciendo los bloques es aleatorio.

Elección libre

Con el objetivo de ser capaces de comparar los resultados de este segundo módulo de pruebas con los resultados obtenidos en el primero, se opta por utilizar los mismos métodos de evaluación y en el

mismo orden. De esta forma, en primera instancia, se hace uso del método de elección libre, con el que se pretende conocer, en palabras del usuario, las emociones que transmiten cada uno de los bloques con voces emocionales generados.

Antes de comenzar con la prueba, y con el fin de no sufrir los problemas de léxico que se detectaron en pruebas anteriores, se insta a los usuarios a recordar, durante un minuto y para sí mismos, distintos adjetivos que indiquen el estado emocional de una persona. Con este pequeño recordatorio de adjetivos emocionales, el número de veces que los usuarios recurren a ejemplos gráficos para expresar la emoción detectada en el discurso se reduce drásticamente, facilitando la labor del desarrollador al analizar los resultados. A continuación se presentan los resultados obtenidos para cada una de las emociones expresadas:

- Triste: todos los bloques con voces emocionales tristes son reconocidos por los usuarios como bloques negativos. El estado anímico más utilizado para describir estos bloques es triste, aunque siguen apareciendo con relativa frecuencia otros estados anímicos como aburrido y decepcionado en las respuestas de los usuarios encuestados.

- Enfadado: del mismo modo que sucede con la emoción anterior, los bloques que pretenden representar el estado anímico de enfado son reconocidos por los usuarios como bloques negativos. La mayoría de estos usuarios optan por el término enfadado para definir el estado emocional que transmiten estos bloques, pero términos como decepcionado o angustiado también aparecen entre las respuestas de los encuestados.

- Neutro: en el caso de los bloques que representan el estado anímico neutro se pueden extraer varias conclusiones de los resultados obtenidos. En primer lugar, la confusión con el estado de tristeza disminuye en gran medida con la elevación del rango de valores posibles para el tono del discurso. Además, se reducen las percepciones de carácter robótico del discurso por parte de los usuarios, definiendo las voces emocionales neutras como más naturales y realistas. Sin embargo, en los bloques donde se hace uso de los valores más bajos para el tono, los resultados empeoran, apareciendo estados anímicos como triste o aburrido entre las respuestas del usuario.

- Alegre: los resultados obtenidos por los bloques que contienen las voces emocionales de alegría son mejores que los conseguidos en el primer módulo de pruebas, siendo más de la mitad de los usuarios los que han optado por términos como alegre y contento para definir la emoción transmitida a través del discurso. No obstante, sigue resultando difícil distinguir esta voz emocional de la voz emocional de sorpresa, ya que aparecen múltiples respuestas de sorprendido entre las respuestas de los usuarios encuestados. También aparecen términos como entusiasta y excitado.

- Sorprendido: de la misma forma que sucede con la emoción de alegría, los bloques que pretenden representar el estado anímico de sorpresa obtienen mejores resultados que en el primer módulo de pruebas con usuarios. Esta vez, la mayoría de los usuarios encuestados se declinan por sorprendido para definir el estado emocional que transmite el discurso, aunque términos como excitado, alegre y contento también aparecen entre las respuestas de los usuarios.

Además las conclusiones expuestas para cada uno de los estados emocionales, es posible extraer una conclusión común para todos ellos. Analizando los resultados obtenidos se observa que, para cada uno de los estados anímicos que se pretenden expresar a través del discurso, existe un par de bloques que son más reconocibles que el resto por parte del usuario. Para confirmar esta tendencia se procede hacer una segunda prueba con los mismos bloques de voces emocionales.

Elección libre modificada

El método de evaluación utilizado en esta segunda prueba es la elección libre modificada. El conjunto cerrado de términos entre los que el usuario debe elegir su respuesta es el mismo que en el primer módulo de pruebas, esto es, las cinco emociones que se pretende representar a través del discurso (ALEGRE, SORPRENDIDO, NEUTRO, TRISTE y ENFADADO), tres categorías de distracción (DECEPCIONADO, ASUSTADO y ENTUSIASMADO) y una última opción OTROS que impida que el usuario se vea obligado a decantarse por un estado emocional determinado en caso de no haber reconocido la emoción transmitida.

Esta segunda prueba tiene dos objetivos principales. En primer lugar, se pretende confirmar la mejora en los resultados obtenidos con respecto al primer módulo de pruebas, de forma que sea posible determinar que los nuevos bloques de voces emocionales hacen más reconocibles al usuario las emociones que se desea transmitir. Por otro lado, se busca comprobar que, para cada una de las emociones modeladas, los resultados obtenidos por determinados bloques siguen siendo superiores al resto, de manera que el desarrollador pueda centrarse en los rangos de valores hacen más reconocibles las emociones al usuario.

La prueba se lleva a cabo de forma idéntica a la prueba anterior, utilizando los mismos bloques y en las mismas condiciones de aleatoriedad. En las siguientes líneas se analizan los resultados.

- Triste: los resultados obtenidos por los bloques con voces emocionales de tristeza son notablemente buenos. Una gran mayoría se decanta por la opción triste a la hora de definir la emoción transmitida a través del discurso, aunque también aparecen otros estados emocionales negativos como enfadado y decepcionado entre las respuestas de los usuarios. En este sentido, es posible observar que, para los valores más altos de velocidad considerados [0.85, 0.95], los resultados obtenidos empeoran puesto que los usuarios se decantan en una mayor medida por la opción enfadado. Por otro lado, las voces emocionales de tristeza con valores de tono del discurso más bajos [1.1, 1.2] son confundidas por el usuario con el estado emocional de decepción, mientras que para los valores más altos [1.6, 1.7] son confundidas con el enfado nuevamente.

- Enfadado: los bloques con voces emocionales de enfado son los que mejores resultados obtienen en esta segunda prueba, aunque repartidos de forma desigual entre los cuatro bloques generados. Si bien una inmensa mayoría de los usuarios encuestados reconocen el estado anímico enfadado en el discurso, los resultados empeoran para los bloques con voces emocionales con menor velocidad de discurso [1.2, 1.4], ya que los usuarios comienzan a confundirse con el estado anímico de decepción. A su vez, cabe destacar que los bloques que han cosechado mejores resultados son aquellos que han hecho uso de los valores intermedios del rango [1.1, 1.4] considerado para el tono de voz.

- Neutro: es uno de los estados emocionales más reconocido por los usuarios, obteniendo grandes resultados para todos los bloques neutros generados. Sin embargo, de los tres bloques neutros utilizados durante la prueba, el que más confusión genera a los usuarios es el bloque que posee las voces emocionales con un tono de voz más bajo [1.1, 1.3], decantándose varios de los usuarios encuestados por los estados de decepción y tristeza a la hora de evaluarlo.

- Alegre: como ya sucediera en la prueba anterior, los resultados obtenidos para los bloques con voces emocionales de alegría son mejores que los del primer módulo de pruebas, seleccionando la opción alegre más de la mitad de los usuarios encuestados. Sin embargo, en todos los bloques alegres generados, el grado de confusión con el estado emocional de sorpresa es elevado, llegando a ser seleccionada la opción sorprendido en casi la misma medida que la opción alegre en algún bloque. De cualquier modo, se puede apreciar que los bloques con valores más bajos de volumen, tono y velocidad de discurso obtienen mejores resultados.

- Sorprendido: en el caso de los bloques con voces emocionales de sorpresa, los resultados vuelven a mejorar los resultados obtenidos en el primer módulo de pruebas. Además, a pesar de que el grado de confusión con el estado emocional de alegría es elevado, los resultados obtenidos por los bloques con voces emocionales de sorpresa superan en aciertos a los bloques alegres anteriores. A su vez, se puede observar que los usuarios reconocen mejor la emoción de sorpresa para rangos [12.0, 14.0] de volumen, [1.2, 1.4] de velocidad y [2.2, 2.4] de tono del discurso.

Tras analizar estos resultados se pueden extraer un par de conclusiones fundamentales. En primer lugar, las voces emocionales modeladas para este segundo módulo de pruebas transmiten mejor al usuario los estados anímicos que se pretenden representar. Por otro lado, la variabilidad en los resultados obtenidos por los bloques de cada una de las emociones invita a centrarse en las voces emocionales correspondientes a los bloques con mejores resultados, reduciendo de esta forma el tamaño de los rangos de posibles valores para los parámetros del discurso correspondientes a cada una de las emociones modeladas.

Con esta información, se inicia un último proceso de modificación de las voces emocionales cuyo objetivo es generar las voces que conformarán los bloques emocionales a utilizar en la fase de selección posterior.

F.2.4 Modificación de las voces emocionales

Este proceso tiene como objetivo generar las voces que conformarán los bloques emocionales a utilizar en la fase de selección posterior. Con este fin, se procede a modelar voces emocionales con un mayor grado de realismo, utilizando para ello toda la información extraída de los resultados obtenidos en los módulos de prueba anteriores. En las siguientes líneas se explican el proceso de generación de nuevas voces emocionales para cada uno de los estados anímicos del agente virtual.

- Triste: para modelar nuevas voces emocionales que representen el estado anímico de tristeza de forma más fidedigna, se opta por hacer uso de los rangos de valores mejor valorados para cada uno de los parámetros del discurso en esta emoción. De este modo, para la velocidad del discurso se consideran los valores comprendidos en el rango [0.75, 0.85], mientras que el rango considerado para el tono del discurso es [1.3, 1.5]. A partir de dichos rangos se generan un total de nueve voces emocionales de tristeza, combinando entre sí los distintos valores seleccionados para la velocidad del discurso (0,75, 0.8 y 0.85) con los valores seleccionados para el tono (1.3, 1.4 y 1.5).

- Enfadado: el proceso de generación de nuevas voces emocionales para el estado anímico de enfado es idéntico al explicado para la emoción de tristeza. Se consideran, para cada uno de los parámetros del discurso, los rangos de valores mejor valorados por los usuarios en los módulos de pruebas anteriores. En este caso, para la velocidad de discurso se considera el rango [1.4, 1.7], mientras que para el tono, el rango a considerar es [1.2, 1.3]. De esta forma, se modelan ocho voces emocionales nuevas para el estado anímico de enfado, resultantes todas ellas de la combinación de los distintos valores seleccionados para la velocidad (1.4, 1.5, 1.6 y 1.7) y el tono (1.2 y 1.3) del discurso.

- Neutro: en el caso del estado anímico neutro, no se persigue generar nuevas voces emocionales, ya que los resultados obtenidos en todas las pruebas realizadas han sido notablemente buenos, si no que se busca determinar la voz emocional neutra que más realista y menos robotizada resulta a los usuarios. Para ello, y en base a los resultados obtenidos en los módulos de pruebas anteriores, se lleva a cabo una pequeña criba de las voces emocionales neutras ya modeladas, manteniendo sólo cuatro de ellas. Estas cuatro voces emocionales neutras seleccionadas poseen el mismo volumen y la misma velocidad de discurso, por lo que únicamente difieren en el valor utilizado en cada una de ellas para el tono. Los valores considerados para el tono del discurso son (1.4, 1.5, 1.6, 1.7).

- Alegre: el modelado de voces emocionales que representen fielmente el estado anímico de alegría es complicado. Por un lado, los resultados obtenidos en los módulos de pruebas anteriores no son excesivamente buenos, existiendo un alto porcentaje de usuarios que confunden el estado emocional de alegría con el de tristeza. Por otra parte, las conclusiones extraídas de estos resultados no son del todo concretas, si no que se tratan de tendencias que se intuyen de los mismos. De cualquier modo, se opta por generar nueve voces emocionales alegres nuevas en base a las tendencias observadas en los resultados y a la percepción del discurso llevada a cabo por el desarrollador. Los rangos de valores considerados son [1.0, 1.2] para la velocidad de discurso, [10.0, 12.0] para el volumen y [1.7, 1.9] para el tono.

- Sorprendido: el proceso de generación de nuevas voces emocionales para el estado anímico de sorpresa es similar al explicado para la emoción de tristeza. Se consideran, para cada uno de los parámetros del discurso, los rangos de valores mejor valorados por los usuarios en los módulos de pruebas anteriores. En el caso del estado emocional de sorpresa, el rango considerado para la velocidad de discurso es [1.2, 1.4], mientras que para el tono se considera el rango [2.2, 2.4]. Con respecto al volumen, el rango a considerar es [12.0, 14.0], pero el desarrollador opta por utilizar un valor constante que simplifique el proceso de generación de nuevas voces. El valor seleccionado es 13.6, valor de volumen correspondiente al bloque emocional sorprendido con

mejores resultados en el segundo módulo de pruebas realizado. De esta forma, se modelan nueve voces emocionales nuevas para el estado anímico de sorpresa, resultantes todas ellas de la combinación de los distintos valores seleccionados para la velocidad (1.2, 1.3 y 1.4) y el tono (2.2, 2.3 y 2.4) del discurso.

F.2.5 Reagrupamiento de las voces en bloques emocionales

Una vez generadas las nuevas voces emocionales, éstas se reagrupan en cinco bloques distintos, de manera que todas aquellas voces emocionales que pretendan transmitir un mismo estado anímico se encuentren en el mismo bloque.

Debido al distinto número de voces modeladas para cada una de las emociones a representar, se generan cinco bloques emocionales de distinto tamaño, bloques emocionales que van a servir como base a la fase de selección que se explica en detalle en el siguiente apartado de la memoria principal.

F.3 Selección de las voces emocionales

La segunda fase de pruebas con usuarios tiene con objetivo seleccionar, para cada uno de los estados anímicos que se desea representar, la voz emocional que mejor transmita las emociones correspondientes. Con este fin, se llevan a cabo dos nuevos módulos de pruebas con usuarios, módulos que sirven al desarrollador para hacer las cribas necesarias hasta hallar la voz emocional más adecuada para cada estado anímico. A continuación se detallan ambos módulos de pruebas.

F.3.1 Tercer módulo de pruebas

El proceso de selección de las voces emocionales se inicia con un tercer módulo de pruebas con usuarios que persigue dos objetivos principales: cerciorarse de que los usuarios reconocen el estado anímico que pretende transmitir cada bloque emocional generado y guiar una primera criba de voces emocionales en función de los resultados obtenidos.

Como se ha comentado anteriormente, los cinco bloques emocionales resultantes de la fase de calibración sirven como punto de partida a este tercer módulo de pruebas. Cada uno de estos cinco bloques es escuchado por los usuarios dos veces consecutivas. La primera vez, los usuarios deben decidir el estado anímico que les transmite cada uno de los bloques, mientras que la segunda debe servir a los usuarios para decidir las tres voces emocionales del bloque escuchado que mejor transmiten el estado anímico que han reconocido. De esta forma, se pretende conocer los conjuntos de voces emocionales mejor valorados por los usuarios para cada estado anímico a representar y proceder a la eliminación de las voces emocionales con peores resultados.

Para que los resultados obtenidos en este módulo de pruebas no se vean influenciados por ninguna componente emocional adicional, se sigue optando por usar la frase neutra utilizada en los dos módulos de pruebas anteriores (“Los viernes la fruta está mucho más barata”) y el agente virtual no aparece en ningún momento en la pantalla del dispositivo. Además, para no alterar los resultados, tanto el orden de las voces emocionales incluidas en cada uno de los bloques emocionales generados como el orden en el que los usuarios escuchan dichos bloques siguen un orden aleatorio.

Cabe destacar que este tercer módulo de pruebas también se lleva a cabo sobre un grupo reducido de personas. Este grupo reducido de usuarios está formado por nueve personas, cinco mujeres y cuatro varones, ninguno con la misma edad y con estudios o formación laboral de índole totalmente diversa.

Elección forzada

El método de evaluación utilizado en la primera reproducción de cada bloque es la elección forzada, ya que se considera necesario que, una vez terminado el proceso de calibración, el usuario se

decante por una de las cinco emociones a modelar como respuesta para cada uno de los bloques emocionales. De esta forma, el conjunto cerrado de términos entre los que el usuario debe elegir su respuesta es ALEGRE, SORPRENDIDO, NEUTRO, TRISTE y ENFADADO.

Por otro lado, con el objetivo de facilitar a los usuarios encuestados la selección de las tres voces que mejor transmiten la emoción reconocida en cada bloque emocional, se les otorga una hoja de respuestas con la información que se muestra en la Tabla F.3.1. Como se puede observar, aparecen los cinco bloques generados seguidos del número de voces emocionales que contienen cada uno, de forma que el usuario sólo debe rodear las tres voces emocionales que más le gusten de cada uno de los bloques.

BLOQUE 1	BLOQUE 2	BLOQUE 3	BLOQUE 4	BLOQUE 5
Voz emocional 1	Voz emocional 1	Voz emocional 1	Voz emocional 1	Voz emocional 1
Voz emocional 2	Voz emocional 2	Voz emocional 2	Voz emocional 2	Voz emocional 2
Voz emocional 3	Voz emocional 3	Voz emocional 3	Voz emocional 3	Voz emocional 3
Voz emocional 4	Voz emocional 4	Voz emocional 4	Voz emocional 4	Voz emocional 4
Voz emocional 5	Voz emocional 5	Voz emocional 5		Voz emocional 5
Voz emocional 6	Voz emocional 6	Voz emocional 6		Voz emocional 6
Voz emocional 7	Voz emocional 7	Voz emocional 7		Voz emocional 7
Voz emocional 8	Voz emocional 8	Voz emocional 8		Voz emocional 8
Voz emocional 9		Voz emocional 9		Voz emocional 9

Tabla F.3.1: *Tabla otorgada a los usuarios para que seleccionen las tres voces emocionales que mejor transmiten la emoción reconocida en cada bloque.*

A continuación se detallan los resultados obtenidos tanto en la fase de reconocimiento de la emoción que pretende transmitir cada bloque como en la fase de selección de las voces emocionales más acordes a la emoción reconocida.

- Triste: el bloque con las voces emocionales tristes es uno de los bloques mejor reconocidos por los usuarios, ya que siete de las nueve personas encuestadas optan por la opción triste al escucharlo. Los otros dos encuestados restantes optan por la opción neutra. En cuanto a la selección de las voces emocionales que mejor transmiten el estado anímico de tristeza, es reseñable que, de las nueve voces modeladas, tan sólo cinco de ellas son elegidas por los usuarios que han reconocido previamente la emoción.

- Enfadado: los resultados obtenidos por el bloque emocional de enfado son muy similares al caso anterior. Por un lado, el estado anímico de enfado que pretende transmitir este bloque es reconocido por siete de los nueve usuarios encuestados, optando los dos usuarios restantes por las opciones triste y neutro. Por otra parte, pese al haber modelado un total de ocho voces emocionales enfadadas, los votos de los usuarios encuestados se reparten entre cinco de ellas, quedando tres sin seleccionar.

- Neutro: el bloque emocional neutro es el que mejor resultados obtiene. En la fase de reconocimiento de la emoción que pretende transmitir este bloque, tan sólo un usuario no reconoce el estado emocional neutro, optando por la opción alegre. Por otro lado, si bien las cuatro voces emocionales neutras modeladas reciben votos de los usuarios que han reconocido la emoción, dos de ellas presentan resultados notablemente mejores que el resto.

- Alegre: como ya ocurriera en módulos de pruebas anteriores, el bloque que contiene las voces emocionales de alegría obtiene los peores resultados de los cinco bloques generados. En primer lugar, a pesar de que cinco de los nueve usuarios encuestados reconocen la emoción de alegría

que se pretende transmitir, sigue siendo elevado el número de usuarios que confunden esta emoción con la de sorpresa, en este caso tres de nueve, lo que viene a confirmar la dificultad que entraña modelar una voz emocional alegre fidedigna que se aleje de la voz emocional de sorpresa. Además, es destacable que un último usuario opta por el término enfadado para definir el estado emocional que percibe al escuchar este bloque. En cuanto a las voces emocionales mejor valoradas por los usuarios para representar el estado emocional de alegría, seis de las nueve voces modeladas reciben al menos un voto, quedando sin ninguno las tres restantes.

- Sorprendido: los resultados obtenidos por el bloque con voces emocionales de sorpresa se asemejan a los del bloque anterior, aunque los mejora ligeramente. A la hora de dar a conocer la emoción percibida al escuchar este bloque emocional, seis de los nueve usuarios optan por el término sorprendido, frente a los tres usuarios encuestados que optan por la opción alegre. Por otra parte, de las nueve voces emocionales modeladas, solamente cinco reciben algún voto por parte de los usuarios que han reconocido la emoción que se pretende transmitir.

Además de los resultados cosechados por cada uno de los bloques individualmente, cabe destacar un par de aspectos adicionales correspondientes a este tercer módulo de pruebas. En primer lugar, una inmensa mayoría de los usuarios expresa al desarrollador la gran complejidad que entraña la prueba, puesto que consideran que varias de las voces emocionales de algunos bloques son idénticas o excesivamente parecidas como para ser capaz de distinguirlas, aumentando de esta forma la dificultad de seleccionar las voces emocionales más conseguidas. Por otra parte, se observa que los bloques emocionales utilizados en este módulo de pruebas son, en algunos casos, demasiado extensos, resultando difícil al usuario recordar todas las voces emocionales escuchadas.

Con toda esta información, se procede a llevar a cabo una primera criba de voces emocionales, criba tras la que se procederá a generar los bloques de voces emocionales con los que realizar el siguiente módulo de pruebas con usuarios.

F.3.2 Primera criba de voces emocionales

Este proceso tiene como objetivo eliminar las voces emocionales peor valoradas por los usuarios encuestados para poder generar los distintos bloques de voces emocionales con los que realizar el último módulo de pruebas de esta fase de selección. Con este fin, se procede a realizar una criba de las voces emocionales utilizadas en el módulo de pruebas anterior en función de los resultados obtenidos en dicho módulo. En las siguientes líneas se explica el proceso de criba realizado.

- Triste: del conjunto de nueve voces emocionales de tristeza que se modelan para el módulo de pruebas anterior, se eliminan las cuatro voces emocionales que no han recibido ningún voto por los usuarios encuestados. Además, debido a que ningún usuario ha percibido la emoción triste en los otros bloques emocionales, no es necesario realizar ninguna acción extra.

- Enfadado: se descartan las tres voces emocionales de enfado que no han sido votadas por los usuarios encuestados, manteniendo las otras cinco. A su vez, se adiciona una nueva voz emocional proveniente del bloque emocional alegre, más concretamente, la voz emocional mejor valorada por el usuario que ha percibido la emoción de enfado en dicho bloque. De esta forma, se debe contar con seis voces emocionales de enfado para el siguiente módulo de pruebas.

- Neutro: a pesar de que las cuatro voces neutras modeladas han recibido votos de los usuarios encuestados, la notable diferencia entre los resultados obtenidos por dos de ellas y los conseguidos por el par restante hace que se opte por eliminar a estas dos últimas. Por otro lado, debido a que los dos usuarios que perciben un estado anímico neutro en el bloque emocional triste coinciden, a su vez, en la voz emocional que seleccionan como más realista, se añade esta voz emocional triste al conjunto de voces neutras a considerar en el próximo módulo de pruebas con usuarios.

- Alegre: de las nueve voces emocionales de alegría modeladas, se eliminan tres de ellas al no recibir ningún voto de los usuarios encuestados. Además, debido a la confusión observada en los usuarios a la hora de diferenciar el estado anímico alegre del sorprendido, se incorporan a las seis voces emocionales no eliminadas tres voces adicionales procedentes del bloque emocional de

sorpresa. Estas tres voces son las seleccionadas por los usuarios que han detectado la emoción de alegría en el bloque emocional sorprendido, y hacen que el número de voces alegres a considerar en el próximo módulo vuelva a ser nueve.

- Sorprendido: del conjunto de nueve voces emocionales de sorpresa modelado para el módulo de pruebas anterior, se opta por prescindir, en primera instancia, de las cuatro voces emocionales que no han recibido ningún voto. Además, debido a que los resultados obtenidos por dos de las voces restantes no superaban los dos votos, se procede también a su eliminación, quedando tan sólo tres de las voces emocionales iniciales. Posteriormente, a este grupo de tres voces se le adiciona tres nuevas voces emocionales procedentes del bloque emocional alegre, voces que han sido seleccionadas por la mayoría de los usuarios que han reconocido el estado anímico de sorpresa en dicho bloque. De esta forma, se ha de tener en cuenta un total de seis voces emocionales de sorpresa para el próximo módulo de pruebas.

F.3.3 Generación de los bloques definitivos

Una vez llevado a cabo el proceso de criba y tras haber determinado el conjunto de voces emocionales se van a utilizar para intentar transmitir cada uno de los estados anímicos que se desea representar, se procede a generar los distintos bloques con los que realizar el último módulo de pruebas con usuarios.

En primer lugar, se opta por que los bloques generados no contengan todas las voces emocionales correspondientes a un mismo estado anímico, si no que posean una mezcla heterogénea de las voces resultantes del proceso de criba anterior. El objetivo de esta medida es conocer la capacidad del usuario para reconocer el estado anímico que pretende transmitir cada voz emocional modelada sin que éstas se encuentren acompañadas de voces correspondientes al mismo estado emocional.

Por otro lado, se considera acertado no dotar de una gran cantidad de voces emocionales a cada uno de los bloques, ya que las pruebas con usuarios realizadas anteriormente demuestran que los resultados obtenidos mejoran con bloques no muy extensos. En este sentido, dado que el número de voces emocionales a distribuir entre los bloques es 28, se opta por generar cuatro bloques que posean siete voces emocionales distintas cada uno.

Finalmente, debido a que el número de voces emocionales para cada estado anímico modelado es distinto y dificulta la distribución compensada las voces emocionales, se lleva a cabo una distribución completamente aleatoria de todas las voces emocionales entre los cuatro bloques a generar, no teniendo en consideración el estado anímico representado por cada una de las voces emocionales distribuidas.

De esta forma, el resultado de este proceso de criba y reagrupación de las voces emocionales es un conjunto de cuatro bloques que incorporan siete voces emocionales, de procedencia diversa y mezcladas aleatoriamente, cada uno. A continuación se describe el último módulo de pruebas con usuarios de esta fase de selección, módulo que se sirve de los cuatro bloques generados para determinar qué voz emocional transmite mejora cada uno de los estados anímicos que se desea representar.

F.3.4 Cuarto módulo de pruebas

El proceso de selección de las voces emocionales continúa con un último módulo de pruebas con usuarios. El objetivo principal de este módulo es evaluar las voces emocionales de cada uno de los bloques generados, de forma que los resultados obtenidos en este módulo guíen la criba definitiva posterior.

Como se ha comentado anteriormente, los cuatro bloques emocionales resultantes de la fase de criba previa sirven como punto de partida a este cuarto módulo de pruebas. Cada uno de estos cuatro bloques es escuchado por los usuarios una única vez, no siendo posible la repetición de los mismos. Además, durante la reproducción de los bloques, el usuario debe responder a la encuesta que se le propone acerca de cada una de las voces emocionales escuchadas, encuesta que se explica en detalle en párrafos posteriores.

Para que los resultados obtenidos en este módulo de pruebas no se vean influenciados por ninguna componente emocional adicional, se sigue optando por usar la frase neutra utilizada en todos los módulos de pruebas anteriores (“Los viernes la fruta está mucho más barata”) y el agente virtual sigue sin aparecer en ningún momento en la pantalla del dispositivo. A su vez, para no alterar los resultados, el orden en el que cada uno de los usuarios encuestados escucha los distintos bloques generados es aleatorio.

Cabe destacar que este módulo de pruebas se lleva a cabo sobre el mismo grupo reducido de personas que se utilizó en el módulo de pruebas anterior. De esta forma, el grupo reducido de usuarios sigue formado por nueve personas, cinco mujeres y cuatro varones, ninguno con la misma edad y con estudios o formación laboral de índole totalmente diversa.

Elección libre modificada

Como se ha citado previamente, el usuario debe responder una encuesta acerca de cada una de las voces emocionales escuchadas durante la reproducción de los cuatro bloques. El objetivo de esta encuesta es conocer la capacidad del usuario para reconocer el estado anímico que se desea transmitir a través de cada una de las voces emocionales de los bloques.

El método de evaluación utilizado en este módulo de pruebas con usuarios es la elección libre modificada. Esta decisión se debe a que, a diferencia del módulo anterior, esta vez no se desea obligar al usuario a elegir entre los distintos estados anímicos que se desea representar, ya que se considera que la mezcla aleatoria de voces emocionales complica notablemente el reconocimiento de las mismas. En este sentido, al conjunto de términos ALEGRE, SORPRENDIDO, NEUTRO, TRISTE y ENFADADO se les adiciona una nueva categoría, NO LO RECONOZCO, que evita al usuario decantarse por un determinado estado anímico si no está seguro.

A continuación se detallan los resultados obtenidos en este último módulo de pruebas con usuarios de la fase de selección de las voces emocionales.

- Triste: de las cinco voces emocionales de tristeza incluidas en los bloques de este módulo de pruebas, tres de ellas reciben ocho, siete y seis votos de los nueve posibles respectivamente, resultados notablemente buenos que invitan a pensar que la voz emocional triste está bastante conseguida. No obstante, las dos voces restantes obtienen resultados bastante pobres, no superando ninguna el 50% de los votos.

- Enfadado: como opciones más votadas aparecen dos de las seis voces emocionales de enfado distribuidas entre los distintos bloques, recibiendo siete y seis votos cada una. Las demás voces emocionales de enfado no superan los cuatro votos, siendo una voz neutra la siguiente más votada con cinco votos.

- Neutro: el conjunto de voces emocionales que los usuarios consideran neutras es muy heterogéneo, apareciendo desde alguna voz emocional triste, hasta un par de voces emocionales de alegría, pasando por las tres voces emocionales neutras incluidas en los bloques en este módulo de pruebas. A pesar de esta diversidad de respuestas, los mejores resultados los obtienen una de las voces neutras modeladas y una voz emocional de alegría, con ocho y siete votos respectivamente, quedando notablemente por detrás la siguiente voz emocional más votada, una voz neutra con cinco votos.

- Alegre: del mismo modo que en los módulos de prueba anteriores, las voces emocionales de alegría siguen siendo difícilmente diferenciables de las voces emocionales de sorpresa para los usuarios, apareciendo voces correspondientes a ambas categorías en las respuestas de los mismos para la categoría ALEGRE. Sin embargo, destaca una de las voces emocionales de alegría con siete votos, seguida de otras dos voces emocionales, una de sorpresa y otra alegre, con seis. Las demás voces emocionales que aparecen no superan los cinco votos.

- Sorprendido: en el caso de las voces emocionales que transmiten sorpresa, el grado de confusión es menor ya que el número de voces alegres que son seleccionadas dentro de la categoría SORPRENDIDA es inferior al caso anterior. Además, las dos voces emocionales que obtienen mejores resultados pertenecen ambas al conjunto de voces emocionales de sorpresa incluidas en

los bloques de este módulo de pruebas, recibiendo 7 y 6 votos cada una. Las demás voces emocionales que aparecen en las respuestas de los usuarios no superar ninguna los cinco votos.

Con estos resultados, se procede a llevar a cabo una segunda criba de voces emocionales, criba tras la que se procederá a generar las voces emocionales definitivas que utilizará el sistema a desarrollar.

F.3.5 Segunda criba de voces emocionales

El objetivo principal de este proceso es llevar a cabo la criba definitiva de las voces emocionales modeladas, de forma que el número de voces emocionales seleccionadas para cada estado anímico no supere las dos unidades. Para ello, se procede a eliminar las voces emocionales que peores resultados hayan obtenido en el módulo de pruebas anterior. En las siguientes líneas se explica el proceso de criba realizado.

- Triste: de las cinco voces emocionales utilizadas a lo largo del módulo de pruebas anterior, dos de ellas son descartadas de forma inmediata al no superar el 50% de los votos. En cuanto a las tres voces restantes, se opta por mantener las dos más votadas, con ocho y siete votos cada una, desechando la tercera más votada.

- Enfadado: en el caso de las voces emocionales de enfado, la elección resulta bastante sencilla, ya que las dos voces más votadas por los usuarios, con siete y seis votos respectivamente, difieren en gran medida de las cuatro restantes en cuanto a número de votos.

- Neutro: el hecho de que los mejores resultados los obtengan una de las voces neutras modeladas y una voz emocional de alegría, con ocho y siete votos respectivamente, unido a la notable diferencia existente entre estas voces y la siguiente voz neutra más votada, hace que el evaluador seleccione las dos primeras voces, independientemente de su índole emocional, y se plantee estudiar los valores de los parámetros del discurso de ambas a la hora de definir la voz neutra definitiva.

- Alegre: a pesar de que las voces emocionales de alegría se siguen confundiendo en gran medida con las de sorpresa, se opta por seleccionar las dos voces alegres con mayor número de votos. De esta forma, se descartan todas las voces emocionales de alegría a excepción de las dos que han obtenido siete y seis votos respectivamente.

- Sorprendido: en el caso de las voces emocionales que transmiten sorpresa, se descartan todas aquellas que no superan los cinco votos. En este sentido, únicamente se salvan de la criba dos voces emocionales de sorpresa cuyos resultados son de siete y seis votos respectivamente.

Una vez finalizada la criba, se inicia un proceso de análisis de los parámetros de las voces emocionales mejor valoradas por los usuarios para cada estado anímico a representar, cuyo fin último es determinar la voz que transmitirá cada uno de dichos estados anímicos del agente virtual en el sistema.

F.3.6 Voces emocionales definitivas

Tras haber analizado los resultados obtenidos en el módulo de pruebas anterior y haber estudiado los parámetros de las voces emocionales seleccionadas por los usuarios para cada estado anímico, se procede a definir las voces emocionales finales cuyo cometido es dar a conocer al usuario las emociones del agente virtual a través del discurso.

En las siguientes líneas se presentan cada una de las voces emocionales definitivas, así como se detallan los parámetros de las mismas y se razona su elección.

- Triste: los valores de los parámetros de la voz emocional de tristeza seleccionada son: 9.28 para el volumen, 0.8 para la velocidad de discurso y 1.3 para el tono. Estos valores coinciden casi en su totalidad con los valores propios de la voz triste más votada por los usuarios en el módulo de pruebas anterior, a excepción del volumen, cuyo valor ha sido incrementado ligeramente debido

a que las otras dos voces consideradas por sus buenos resultados poseían un volumen superior. De esta forma, la voz emocional de tristeza seleccionada para el sistema se caracteriza por un discurso lento, con un volumen no muy elevado y un tono bajo pero no robótico.

- Enfadado: en el caso de la voz emocional de enfado, la decisión es más sencilla ya que las dos voces mejor valoradas por los usuarios en el módulo de pruebas anterior difieren únicamente en la velocidad del discurso, resultando casi imperceptible dicha diferencia al oír ambas voces emocionales de forma consecutiva. Es por ello que se opta por seleccionar como voz emocional de enfado definitiva la voz con mayor número de votos, no llevando ninguna modificación sobre los valores de los parámetros del discurso de la misma. De este modo, la voz emocional de enfado del sistema posee un valor de 12.8 para el volumen, de 1.6 para la velocidad de discurso y de 1.2 en el tono.

- Neutro: a la vista de los resultados obtenidos en el módulo de pruebas anterior se puede concluir que los usuarios perciben un estado emocional neutro en discursos con un volumen no muy elevado, una velocidad media y un tono ligeramente alto. De este modo, y tomando como referencia los parámetros de las dos voces emocionales mejor valoradas por los usuarios, una neutra y otra alegre respectivamente, se determina que la voz definitiva para transmitir el estado anímico neutro del agente posea los siguientes valores en los parámetro del discurso: 10.7 para el volumen, 1.0 para la velocidad y 1.7 para el tono. De esta forma, se mantiene la velocidad de discurso de la opción más votada, incrementando ligeramente tanto el volumen como el tono del discurso.

- Alegre: debido a la dificultad observada a lo largo de todos los módulos de pruebas realizados para distinguir las voces emocionales alegres de las sorprendidas por parte de los usuarios, los resultados notablemente aceptables obtenidos por la voz emocional alegre más votada en el módulo anterior la convierten en la mejor opción para transmitir la alegría del agente a través del discurso en el sistema. Es por ello que no se realiza ninguna modificación sobre los valores de los parámetros del discurso de dicha voz emocional alegre, la cual posee un volumen de 12.6, una velocidad de discurso de 1.3 y un tono elevado de 2.1.

- Sorprendido: los valores de los parámetros de la voz emocional de sorpresa seleccionada para el sistema son: 13.6 para el volumen, 1.2 para la velocidad de discurso y 2.3 para el tono. Estos valores son el resultado de combinar los valores de los parámetros de las voces emocionales de sorpresa mejor valoradas en el módulo de pruebas anterior, que pese a ser similares, diferían en el volumen y el tono de discurso utilizados.

Una vez definidas las voces emocionales encargadas de transmitir cada uno de los estados anímicos del agente virtual a través del discurso, se da por concluida la fase de selección de las voces emocionales, pudiéndose iniciar la última fase de evaluación programada. Esta última fase, que pretende estudiar la calidad de las voces emocionales seleccionadas y su influencia, junto a otros aspectos que denotan emoción, en la capacidad del sistema para expresar las emociones del agente virtual durante la interacción con el usuario, viene explicada en detalle en los siguientes párrafos.

Anexo G. Resultados adicionales

Este anexo complementa al capítulo 5 de la memoria principal, más concretamente a sus últimos tres apartados. En este documento se presentan los resultados adicionales obtenidos en las distintas pruebas finales realizadas para la evaluación de la calidad y relevancia de las voces emocionales con respecto a otros factores que denotan emociones en el proceso de interacción con el usuario. Este anexo está dividido en tres secciones, correspondientes a cada una de las tres pruebas llevadas a cabo con usuarios.

G.1 Reconocimiento de las voces emocionales

Uno de los objetivos perseguidos con la realización de este conjunto de pruebas finales era evaluar la capacidad de los usuarios para reconocer el estado anímico en el que se encuentra el agente virtual únicamente a través de la voz emocional escuchada. Para ello, se ha llevado a cabo una primera prueba con usuarios (descrita en detalle en la sección F.4.2 del Anexo F) en la que se reproducían distintas voces emocionales y se instaba a los usuarios a seleccionar el estado anímico percibido en cada una de las voces reproducidas, evitando en todo momento que cualquier otro aspecto emocional influyese en la percepción de los usuarios encuestados. En las siguientes líneas se dan a conocer los resultados adicionales obtenidos en esta primera prueba.

Tras haber analizado previamente los resultados desde el prisma de la calidad de las voces emocionales generadas (véase apartado 5.2 de la memoria principal), se procede a realizar un segundo análisis donde se estudian otros aspectos destacables de los resultados obtenidos en la prueba.

En la Figura G.1.1 se muestra el porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados en esta prueba.

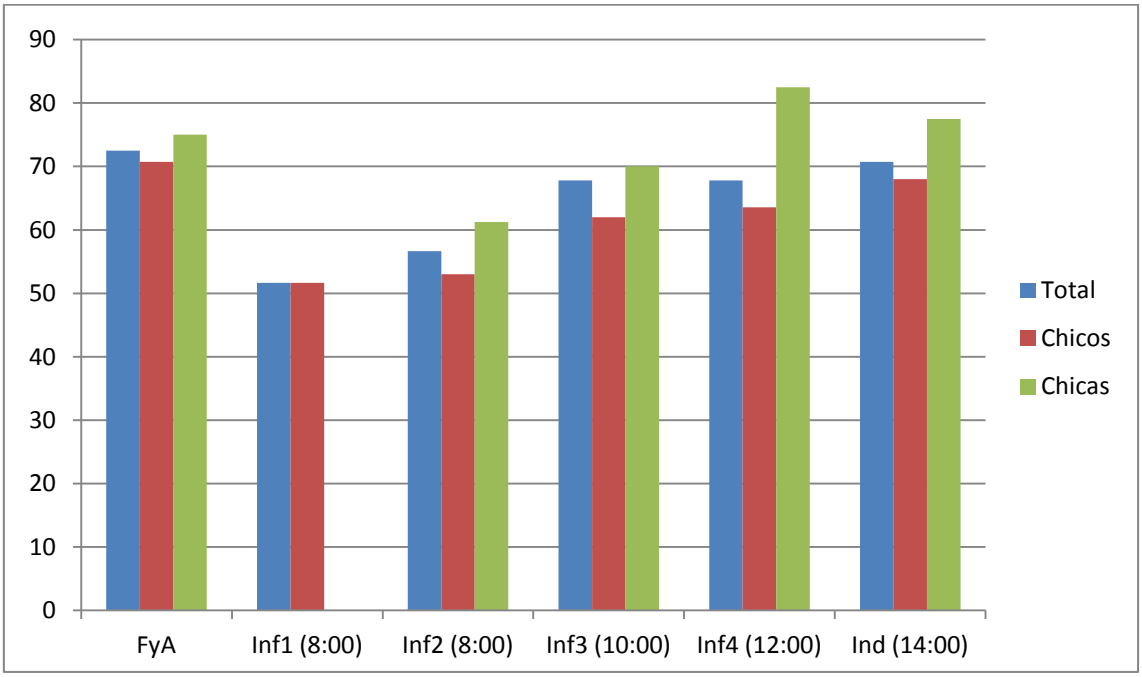


Figura G.1.1: *Porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios finales encuestados en la prueba de reconocimiento de las voces emocionales. En color azul se representa la media total de aciertos de cada grupo, mientras que en rojo y verde se muestran las medias de aciertos obtenidas por los chicos y chicas del grupo respectivamente.*

Como se puede observar en la figura anterior, se ha llevado a cabo una diferenciación por sexo de los resultados obtenidos por los distintos grupos de usuarios. Además, a excepción del primer grupo, referido a los familiares y amigos (FyA) del desarrollador, en el que sólo se ha hecho diferenciación por sexo debido a la heterogeneidad de los datos, los grupos de usuarios vienen clasificados en función de la formación de los usuarios y del horario de realización de la prueba.

Una de las primeras conclusiones que se pueden extraer de la figura anterior es que las usuarias poseen una mayor capacidad de percibir las emociones que se pretenden transmitir a través de las voces emocionales generadas. En este sentido, en todos los grupos de usuarios donde ha existido variedad de sexos, los resultados obtenidos por las féminas han sido mejores que los resultados de los varones, superando, como se muestra en la Tabla G.1.1, en más de un 10% el número de aciertos de éstos.

Porcentaje de Aciertos Total	Porcentaje de Aciertos en Chicos	Porcentaje de Aciertos en Chicas
64.58%	61.86%	71.92%

Tabla G.1.1: *Porcentajes medios de acierto obtenidos en la prueba de reconocimiento de las voces emocionales generadas.*

A su vez, cabe destacar que en grupos de usuarios de índole similar, como son los cuatro grupos de alumnos de ingeniería informática encuestados (inf1-4), los resultados obtenidos mejoran conforme va avanzando la mañana. Este hecho se puede apreciar en la Figura G.1.2, donde la media de aciertos total (representada en azul) de estos cuatro grupos sufre un ligero pero constante ascenso desde la primera hora de la mañana hasta el mediodía, lo que puede ser debido a un aumento de la motivación de los usuarios a la hora de realizar las pruebas.

Finalmente, llama la atención que los grupos de usuarios que obtienen peores resultados en esta prueba coinciden con los grupos conformados por alumnos de ingeniería informática. En este sentido, tanto el grupo de usuarios formado por los amigos y familiares del desarrollador como el único grupo de usuarios formado por alumnos de ingeniería industrial obtienen un porcentaje de aciertos superior a cualquiera de los cuatro grupos de informáticos.

G.2 Influencia del contenido semántico de las frases reproducidas

Además de la calidad de las voces emocionales seleccionadas para transmitir los estados anímicos del agente virtual, se ha considerado interesante estudiar la influencia del contenido semántico de las frases reproducidas en la percepción emocional del usuario durante el discurso del agente. Con este fin, se ha llevado a cabo una segunda prueba (descrita en detalle en la sección F.4.3 del Anexo F) que consiste en mezclar las distintas voces emocionales generadas con las frases con connotación emocional seleccionadas para esta prueba, reproduciendo la combinación resultante a los usuarios e instándoles a determinar qué estado anímico perciben en cada caso. En las siguientes líneas se dan a conocer los resultados, correspondientes a esta segunda prueba con usuarios, que complementan a los presentados en el apartado 5.3 de la memoria principal.

G.2.1 Análisis global de los resultados

En primera instancia, se analizan los resultados obtenidos en esta segunda prueba con el objetivo de conocer el grado de influencia del contenido semántico de las frases reproducidas en la percepción emocional del usuario. Uno de los datos más reveladores en este sentido es el porcentaje de aciertos conseguido por cada uno de los factores emocionales evaluados, en este caso, las voces y frases emocionales, en esta prueba. Este dato, mostrado en la Tabla G.2.1, define el factor emocional que adquiere un mayor peso en la percepción del usuario, pudiendo concentrar esfuerzos en el desarrollo de dicho factor en las futuras optimizaciones del sistema, permitiendo que la interacción con el usuario resulte más realista.

Porcentaje de Acierto en las Voces Emocionales (reproduciendo voz + frase)	Porcentaje de Acierto en las Frases Emocionales (reproduciendo voz + frase)
49.58%	57.19%

Tabla G.2.1: *Porcentajes medios de acierto obtenidos tanto por las voces como por las frases emocionales reproducidas en la prueba que estudia la influencia del contenido semántico de las frases en la percepción del usuario*

Como se puede observar, el porcentaje de acierto obtenido para las frases emocionales es superior al de las voces emocionales, aunque la diferencia no es muy amplia ya que no alcanza el 10% de los aciertos. Con estos datos, es posible determinar que el contenido semántico de las frases reproducidas tiene una gran influencia en la percepción emocional del usuario, teniendo una importancia relativa similar o, incluso, ligeramente superior al de la voz emocional con la que se reproduce el discurso.

Seguidamente, debido a que las reproducciones de esta segunda prueba resultan de la combinación de voces y frases emocionales de distinta índole, se opta por realizar un análisis de los resultados desde el prisma de cada uno de estos aspectos emocionales por separado.

Por un lado, se analizan los resultados considerando únicamente las distintas voces emocionales reproducidas a lo largo de la prueba. Cabe destacar que las voces emocionales vienen reproducidas junto a frases emocionales que no siempre se corresponden con la emoción de la voz utilizada. En la Figura G.2.1 se muestra el porcentaje de respuestas obtenido por estas voces emocionales para cada uno de los términos existentes en la encuesta.

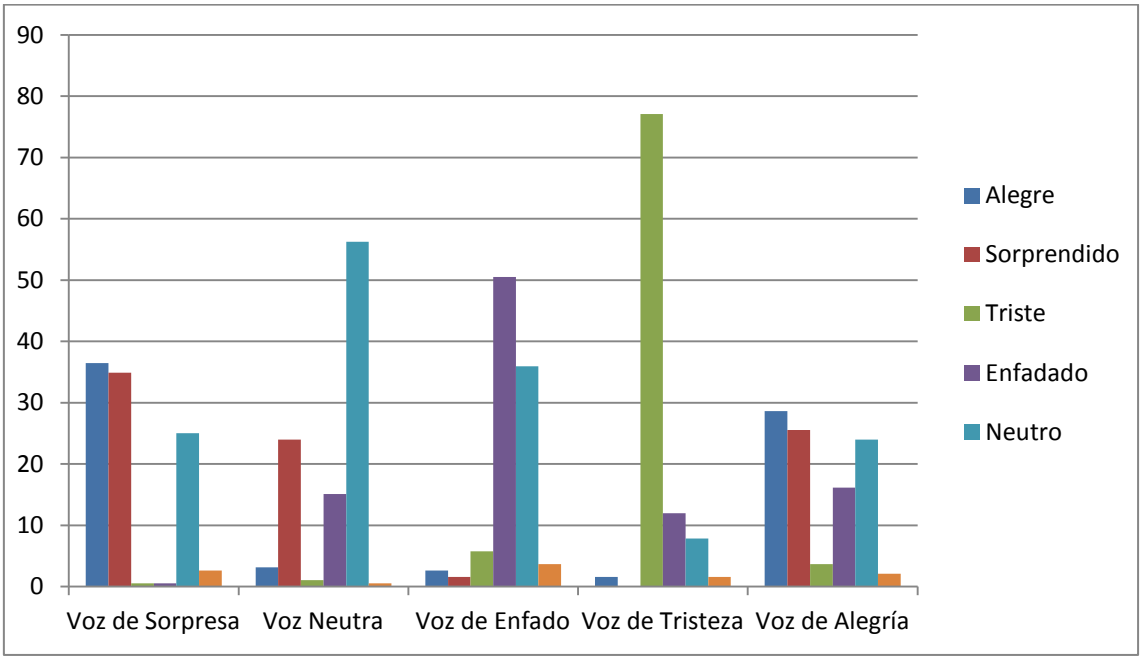


Figura G.2.1: *Resultados obtenidos por cada una de las voces emocionales en la encuesta de elección libre modificada realizada en la segunda prueba. Las voces emocionales son reproducidas junto a frases que corresponden a emociones diferentes. En el eje horizontal se clasifican las distintas voces emocionales reproducidas. En el eje vertical se muestra el porcentaje de respuestas obtenido por dichas voces emocionales para cada término existente en la encuesta de elección libre modificada.*

Como se puede apreciar en la figura anterior, la voz emocional triste es, con notable diferencia, la voz mejor reconocida por los usuarios encuestados, superando el 75% de aciertos. Este hecho, unido a los buenos resultados obtenidos por esta voz en la primera prueba, confirman a la voz emocional triste como la voz más lograda.

En contraposición, las voces emocionales de sorpresa y alegría obtienen resultados poco satisfactorios, rodando ambas voces el 30% de aciertos. Además, se vuelve a demostrar la confusión existente entre estas dos voces emocionales, llegando a superar en número de votos la opción alegre a la opción sorprendido en la evaluación de la voz emocional de sorpresa.

En cuanto a las voces emocionales de enfado y neutra, presentan unos resultados moderadamente aceptables, siendo reconocidas por más de la mitad de los usuarios encuestados. Es importante reseñar que, en el caso de la voz neutra, el porcentaje de aciertos se acerca al 60%, lo que junto a los excelentes resultados obtenidos por esta voz emocional en la prueba anterior hacen de la misma una de las voces emocionales más conseguidas. Por su parte, en la voz emocional de enfado aparece como segunda opción más votada la voz neutra, tal y como ocurre en la primera prueba.

Por otro lado, se realiza un nuevo análisis de los resultados obtenidos en el que se consideran únicamente las frases emocionales reproducidas a lo largo de la prueba. Cabe destacar que las frases emocionales vienen reproducidas junto a voces emocionales que no siempre se corresponden con la emoción de la frase utilizada. En la Figura G.2.2 se muestra el porcentaje de respuestas obtenido por estas frases emocionales para cada uno de los términos existentes en la encuesta.

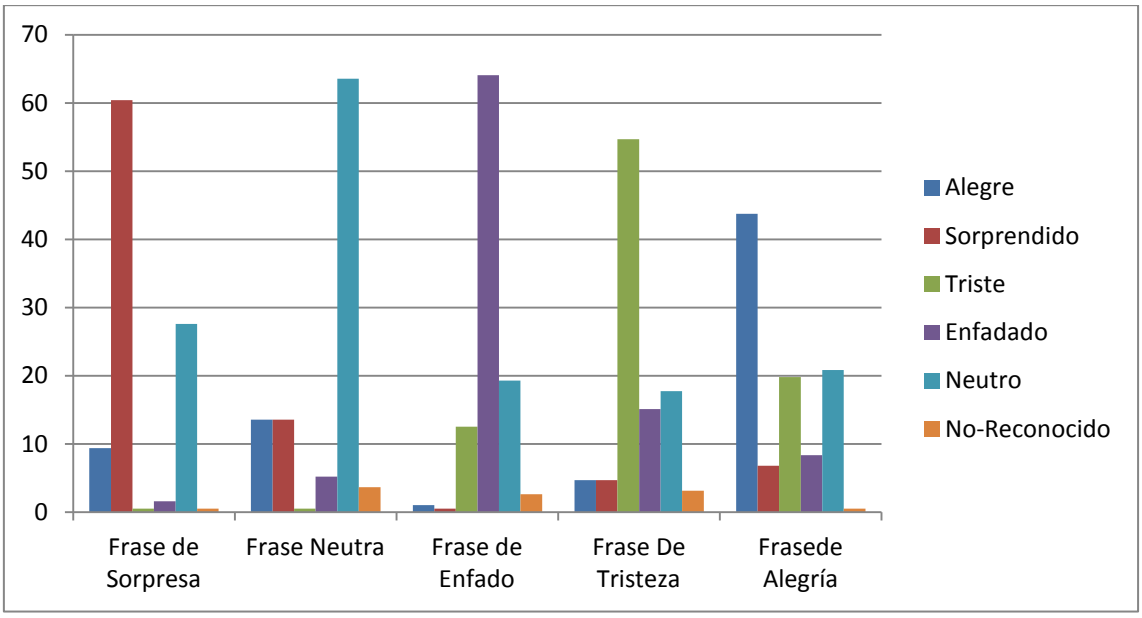


Figura G.2.2: *Resultados obtenidos por cada una de las frases emocionales en la encuesta de elección libre modificada realizada en la segunda pruebas. Las frases emocionales son reproducidas junto a voces que corresponden a emociones diferentes. En el eje horizontal se clasifican las distintas frases emocionales utilizadas. En el eje vertical se muestra el porcentaje de respuestas obtenido por dichas frases emocionales para cada término existente en la encuesta de elección libre modificada.*

Como se puede observar en la figura anterior, para cada una de las frases emocionales utilizadas a lo largo de esta segunda prueba existe una opción destacada en número de votos, siendo en todos los casos la opción correspondiente a la emoción que se pretende transmitir con las distintas frases. Este hecho no sólo confirma la influencia del contenido semántico de las frases reproducidas en la percepción emocional del usuario, sino que indica que el usuario reconoce mejor el estado emocional en el que se encuentra el agente a través del contenido de la frase que de la voz emocional, ya que, a la vista de los resultados, no existen grandes confusiones a la hora de seleccionar la emoción que se pretende transmitir con cada una. De cualquier modo, los resultados obtenidos difieren en función de la frase emocional.

Por una parte, cabe destacar los buenos resultados que presentan las frases emocionales de sorpresa y enfado junto con la frase neutra, obteniendo todas ellas unos resultados superiores al 60% de aciertos. Además, la única frase que posee una segunda opción destacada es la frase de sorpresa, pero los resultados obtenidos por esta segunda opción no alcanzan la mitad de los votos obtenidos por la opción sorprendido, por lo que se descarta la existencia de confusiones relevantes en estas tres frases.

En cuanto a las frases emocionales de tristeza y alegría, obtienen unos resultados del 55% y 45% de aciertos, siendo sus correspondientes emociones la opción más votada en la encuesta con amplia diferencia.

G.2.2 Análisis pormenorizado de los resultados

Una vez estudiado el grado de influencia del contenido semántico de las frases reproducidas en la percepción emocional de los usuarios, se procede a realizar un segundo análisis donde se estudian otros aspectos destacables de los resultados obtenidos en la prueba.

En la Figura G.2.3 se muestra el porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados en esta prueba con respecto a las voces emocionales reproducidas. Como se puede observar, se ha llevado a cabo una diferenciación por sexo de los resultados obtenidos por los distintos grupos de usuarios. Además, a excepción del primer grupo, referido a los familiares y amigos (FyA) del desarrollador, los grupos de usuarios vienen clasificados en función de la formación de los usuarios y del horario de realización de la prueba.

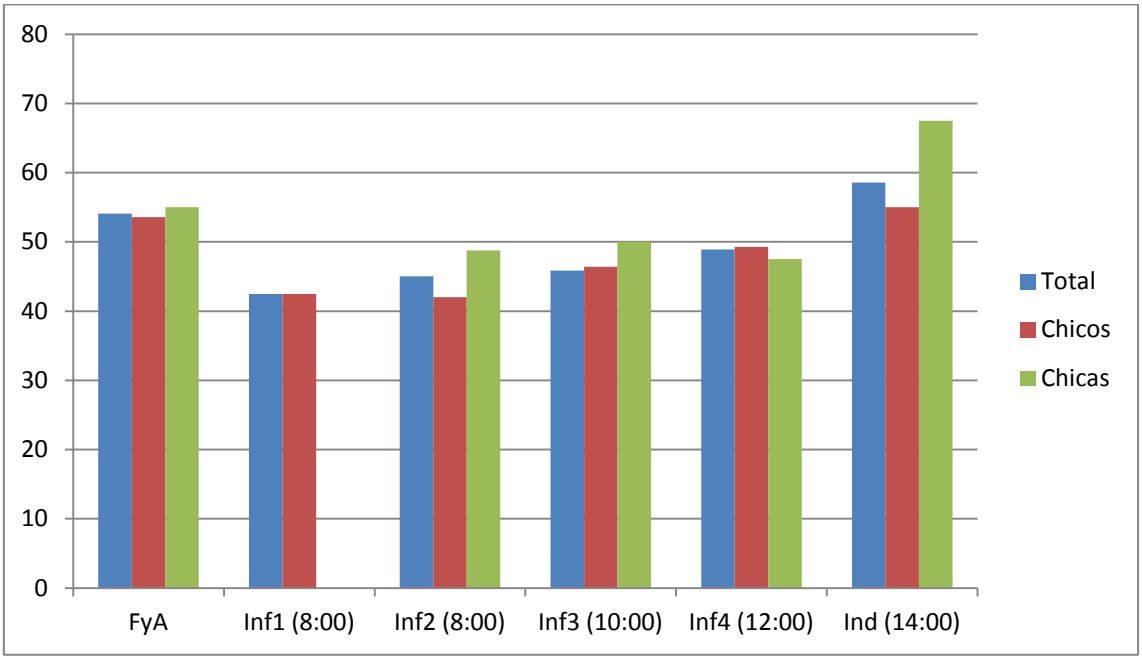


Figura G.2.3: *Porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados para las voces emocionales reproducidas en la prueba que determina la influencia del contenido semántico de las frases reproducidas. En color azul se representa la media total de aciertos de cada grupo, mientras que en rojo y verde se muestran las medias de aciertos obtenidas por los chicos y chicas del grupo respectivamente*

Esta figura, que presenta un cierto parecido a la Figura G.1.2, viene a confirmar las conclusiones extraídas en la primera prueba y que se han comentado en mayor detalle en el apartado G.1.2 de este capítulo. A continuación se explica brevemente cada una de estas conclusiones:

- Mejor percepción femenina: el porcentaje de acierto correspondiente a las usuarias encuestadas (representado en verde) es mayor que el de los usuarios (representado en rojo), superando los resultados obtenidos por estos últimos en la mayoría de los grupos.

- Influencia del horario de las pruebas: los cuatro grupos de alumnos de ingeniería informática presentan una ligera pero constante mejoría en los resultados obtenidos conforme se retrasa la realización de la prueba, siendo significantemente peores los resultados cosechados a primera hora de la mañana que al mediodía.

- Peores resultado en ingenieros informáticos: tanto el grupo conformado por los familiares y amigos del desarrollador como el de alumnos de ingeniería industrial vuelven a obtener mejores resultados que cada uno de los cuatro grupos de usuarios formados por alumnos de ingeniería informática.

Por otro lado, en la Figura G.2.4 se muestra el porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados en esta prueba con respecto a las frases emocionales reproducidas. Como se puede observar, se ha llevado a cabo la misma clasificación de los usuarios que en la figura anterior.

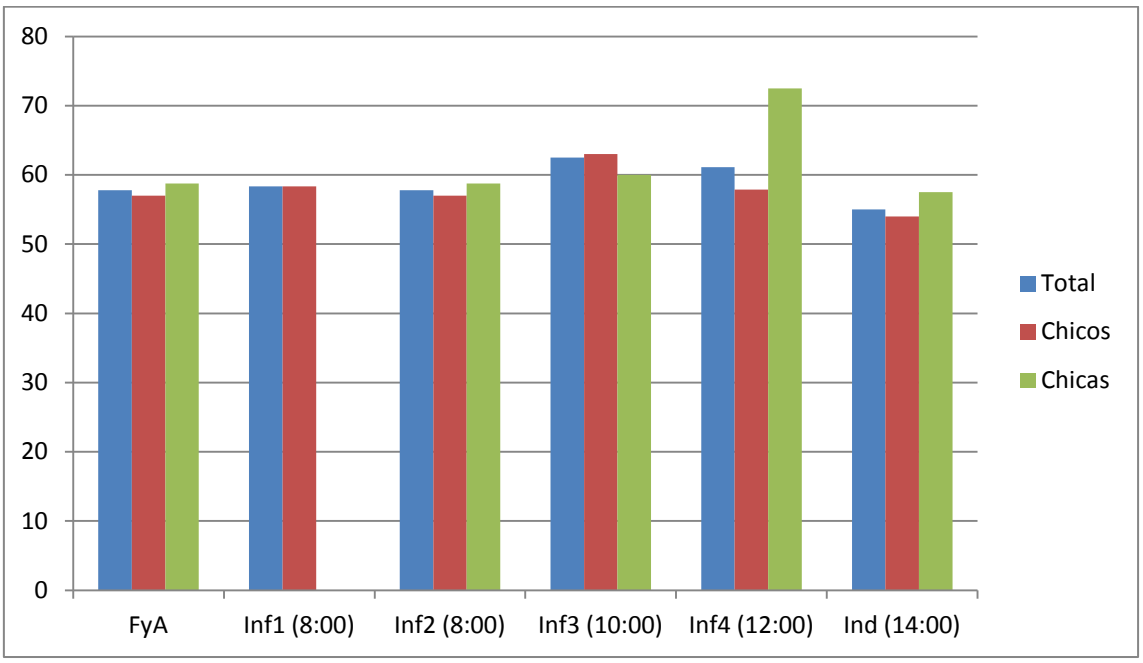


Figura G.2.4: *Porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados para las frases emocionales reproducidas en la prueba que determina la influencia del contenido semántico de las frases reproducidas. En color azul se representa la media total de aciertos de cada grupo, mientras que en rojo y verde se muestran las medias de aciertos obtenidas por los chicos y chicas del grupo respectivamente.*

Como se puede observar, los resultados obtenidos para las frases emocionales son notablemente más homogéneos que los obtenidos en el caso de las voces emocionales. En este sentido, la mayoría de los grupos encuestados se encuentran en torno al 60% de acierto, no habiendo grandes diferencias entre grupos. A su vez, la Tabla G.2.2 muestra el porcentaje de aciertos obtenido para las frases emocionales reproducidas es esta segunda prueba en función del sexo del encuestado, pudiéndose apreciar que la diferencia apenas alcanza el 1.3% de acierto favorable a las usuarias. Por todo ello, es posible concluir que la percepción del estado emocional que se pretende transmitir a través del contenido semántico de las frases reproducidas es similar en todos los usuarios finales encuestados.

Porcentaje de Aciertos Total	Porcentaje de Aciertos en Chicos	Porcentaje de Aciertos en Chicas
57.19%	56.86%	58.08%

Tabla G.2.2: *Porcentajes medios de acierto obtenidos para las frases emocionales reproducidas en esta segunda prueba*

G.3 Relevancia de la imagen con respecto a la voz

Por último, otro de los aspectos que se desea estudiar es la influencia de la imagen del agente virtual en la percepción emocional del usuario durante del proceso de interacción. En este sentido, se pretende conocer la relevancia que adquiere la imagen del agente con respecto a las voces emocionales modeladas a la hora de que el usuario reconozca el estado anímico del agente virtual. Con este fin, se ha llevado a cabo una tercera prueba con usuarios (descrita en detalle en la sección F.4.3 del Anexo F) que consiste en mezclar las distintas voces emocionales modeladas con las animaciones que incorpora el torso de la mujer de Maxine para representar los diversos estados anímicos del agente virtual, reproduciendo la combinación resultante a los usuarios e instándoles a determinar qué estado emocional perciben en cada caso. En las siguientes líneas se dan a conocer los resultados obtenidos de la realización de esta tercera prueba que complementan a los presentados en el apartado 5.4 de la memoria principal.

G.3.1 Análisis global de los resultados

En primer lugar, con el objetivo de conocer la relevancia que adquiere la imagen del agente con respecto a las voces emocionales generadas, se analiza el porcentaje de aciertos conseguido por cada uno de estos factores emocionales en la evaluación llevada a cabo durante la tercera prueba. Este dato, mostrado en la Tabla G.3.1, da una primera visión genérica de la influencia relativa tanto de la voz emocional como de la imagen del agente en la percepción emocional del usuario.

Porcentaje de Acierto en las Voces Emocionales (reproduciendo voz + imagen)	Porcentaje de Acierto en las Animaciones Emocionales (reproduciendo voz + imagen)
37.29%	75.31%

Tabla G.3.1: *Porcentajes medios de acierto obtenido tanto por las voces emocionales como por las animaciones reproducidas durante la prueba que estudia la relevancia de la imagen con respecto a la voz en la percepción emocional del usuario*

Como se puede observar, el porcentaje de aciertos obtenidos en el caso de las animaciones emocionales duplica al conseguido por las voces emocionales, lo que permite determinar a la imagen como el factor emocional más relevante en la percepción emocional del usuario ya que, como se ha mostrado en el apartado G.2.1 de esta memoria, el contenido semántico de las frases reproducidas apenas superaba en un 10% a las voces emocionales. A su vez, estos resultados tan contundentes invitan a centrarse en mayor medida en el desarrollo de animaciones emocionales más realistas en vez de en mejores voces emocionales en futuras optimizaciones del sistema.

Seguidamente, debido a que las reproducciones de esta segunda prueba resultan de la combinación de voces emocionales y animaciones de distinta índole, se opta por realizar un análisis de los resultados desde el prisma de cada uno de estos aspectos emocionales por separado.

Por un lado, se analizan los resultados considerando únicamente las distintas voces emocionales reproducidas a lo largo de la prueba. Cabe destacar que las voces emocionales vienen reproducidas junto a animaciones que no siempre se corresponden con la emoción de la voz utilizada. En la Figura G.3.1 se muestra el porcentaje de respuestas obtenido por estas voces emocionales para cada uno de los términos existentes en la encuesta.

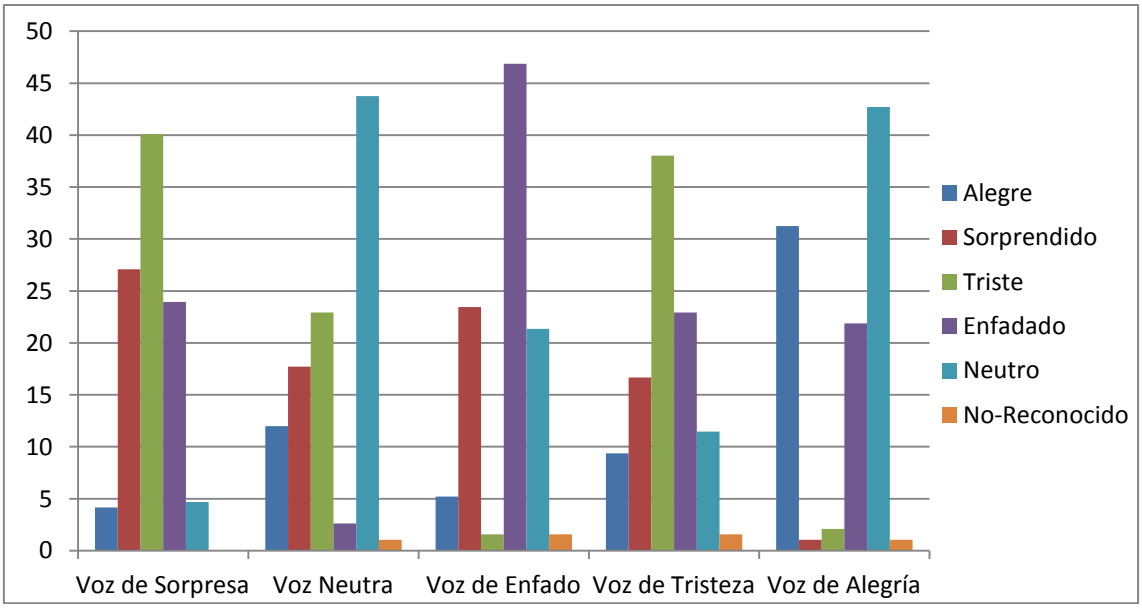


Figura G.3.1: *Resultados obtenidos por cada una de las voces emocionales en la encuesta de elección libre modificada realizada en la tercera prueba. Las voces emocionales son reproducidas junto a animaciones que corresponden a emociones diferentes. En el eje horizontal se clasifican las distintas voces emocionales reproducidas. En el eje vertical se muestra el porcentaje de respuestas obtenido por dichas voces emocionales para cada término existente en la encuesta de elección libre modificada.*

Como se aprecia en la figura anterior, ninguna de las voces emocionales reproducidas durante esta tercera prueba alcanza una tasa de acierto del 50%, quedando todas, a excepción de la voz de enfado, por debajo del 45% de acierto. Este hecho viene asociado a la enorme influencia de la imagen en la percepción emocional del usuario, influencia que provoca que las voces emocionales apenas tenga peso a en las respuestas de los usuarios encuetados.

Especialmente flojos son los resultados obtenidos por las voces emocionales de sorpresa y alegría, puesto que sus respectivas emociones no alcanzan ni siquiera a ser la opción más votada por los usuarios, demostrándose una vez más que son las dos voces emocionales menos logradas.

Por otro lado, se realiza un nuevo análisis de los resultados obtenidos en el que se consideran únicamente las animaciones emocionales reproducidas a lo largo de la prueba. Cabe destacar que las animaciones vienen reproducidas junto a voces emocionales que no siempre se corresponden con la emoción de la animación utilizada En la Figura G.3.2 se muestra el porcentaje de respuestas obtenido por estas frases emocionales para cada uno de los términos existentes en la encuesta.

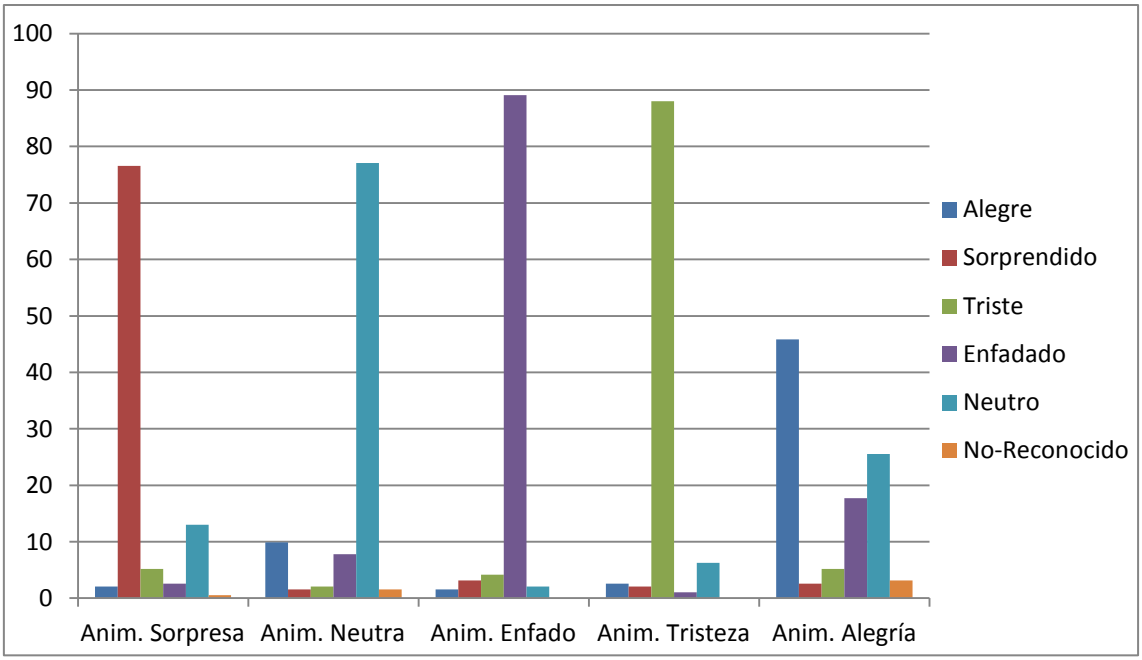


Figura G.3.2: *Resultados obtenidos por cada una de las animaciones emocionales reproducidas durante la encuesta de elección libre modificada realizada en la tercera prueba. Las animaciones son reproducidas junto a voces emocionales que corresponden a emociones diferentes. En el eje horizontal se clasifican las distintas animaciones emocionales utilizadas. En el eje vertical se muestra el porcentaje de respuestas obtenido por dichas animaciones emocionales para cada término existente en la encuesta de elección libre modificada.*

Como se observa en la figura anterior, para cada una de las animaciones emocionales reproducidas a lo largo de esta tercera prueba, a excepción de la animación de alegría, existe una opción extremadamente destacada en número de votos, siendo en todos los casos la opción correspondiente a la emoción que se pretende transmitir con las distintas animaciones. Este hecho confirma la gran influencia de la imagen en la percepción emocional del usuario, reconociendo en mucha mayor medida el estado emocional en el que se encuentra el agente a través de la animación reproducida que de la voz emocional utilizada.

Por una parte, cabe destacar los excelentes resultados obtenidos por las animaciones de tristeza y enfado, animaciones que alcanzan una tasa de acierto cercana al 90% y que, por tanto, transmiten de forma muy acertada las emociones que pretenden representar.

En contraposición, la animación de alegría no presenta unos resultados excesivamente satisfactorios, obteniendo una tasa de acierto en torno al 45%. A raíz de estos datos se considera necesario revisar el modelado de la animación de alegría para versiones posteriores del sistema, de manera que se genere una nueva animación que transmita el estado emocional alegre a los usuarios de forma más precisa.

En cuanto a las animaciones neutra y de sorpresa, consiguen una tasa de acierto que supera el 75%, resultados que también se puede considerar muy satisfactorios desde el prisma de transmitir ambos estados emocionales.

G.3.2 Análisis pormenorizado de los resultados

Una vez estudiado el grado de influencia de la imagen en la percepción emocional de los usuarios, se procede a realizar un segundo análisis donde se estudian otros aspectos destacables de los resultados obtenidos en la prueba.

En la Figura G.3.3 se muestra el porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados en esta tercera prueba con respecto a las voces emocionales reproducidas.

Como se puede observar, se ha llevado a cabo una diferenciación por sexo de los resultados obtenidos por los distintos grupos de usuarios. Además, a excepción del primer grupo, referido a los familiares y amigos (FyA) del desarrollador, los grupos de usuarios vienen clasificados en función de la formación de los usuarios y del horario de realización de la prueba.

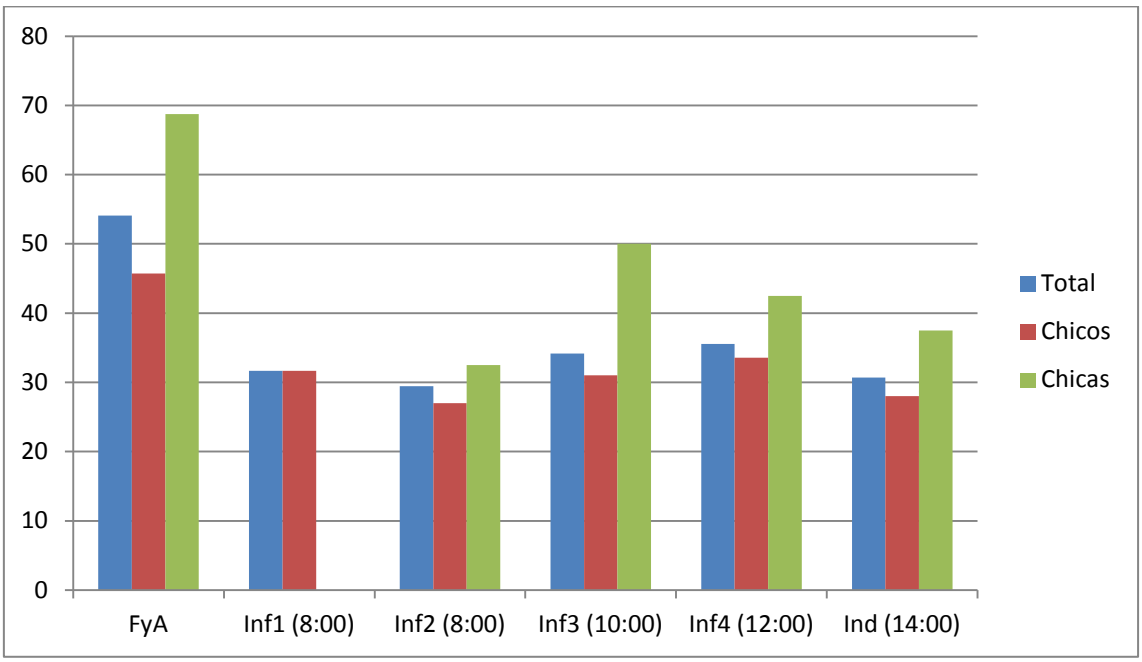


Figura G.3.3: *Porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados para las voces emocionales reproducidas en la prueba que determina la relevancia de la imagen respecto a la voz en la percepción emocional del usuario. En color azul se representa la media total de aciertos de cada grupo, mientras que en rojo y verde se muestran las medias de aciertos obtenidas por los chicos y chicas del grupo respectivamente.*

Una de las primeras conclusiones que se pueden extraer de la figura anterior es el bajo porcentaje de acierto alcanzado por todos los grupos de usuarios. Este hecho se debe a la fuerte influencia que ejerce la imagen mostrada del agente virtual en la precepción del usuario, la cual se ha detectado en el análisis realizado en el apartado anterior.

A su vez, cabe destacar que las usuarias poseen una mayor capacidad de percibir las emociones que se pretenden transmitir a través de las voces emocionales generadas. En este sentido, en todos los grupos de usuarios donde ha existido variedad de sexos, los resultados obtenidos por las féminas han sido mejores que los resultados de los varones, superando, como se muestra en la Tabla G.3.2, en más de un 13.5% el número de aciertos de éstos.

Porcentaje de Aciertos Total	Porcentaje de Aciertos en Chicos	Porcentaje de Aciertos en Chicas
37.29%	33.57%	47.31%

Tabla G.3.2: *Porcentajes medios de acierto obtenidos en el reconocimiento de las voces emocionales reproducidas en la tercera prueba.*

Por otro lado, en la Figura G.3.4 se muestra el porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados en esta prueba con respecto a las animaciones emocionales reproducidas. Como se puede observar, se ha llevado a cabo la misma clasificación de los usuarios que en la figura anterior.

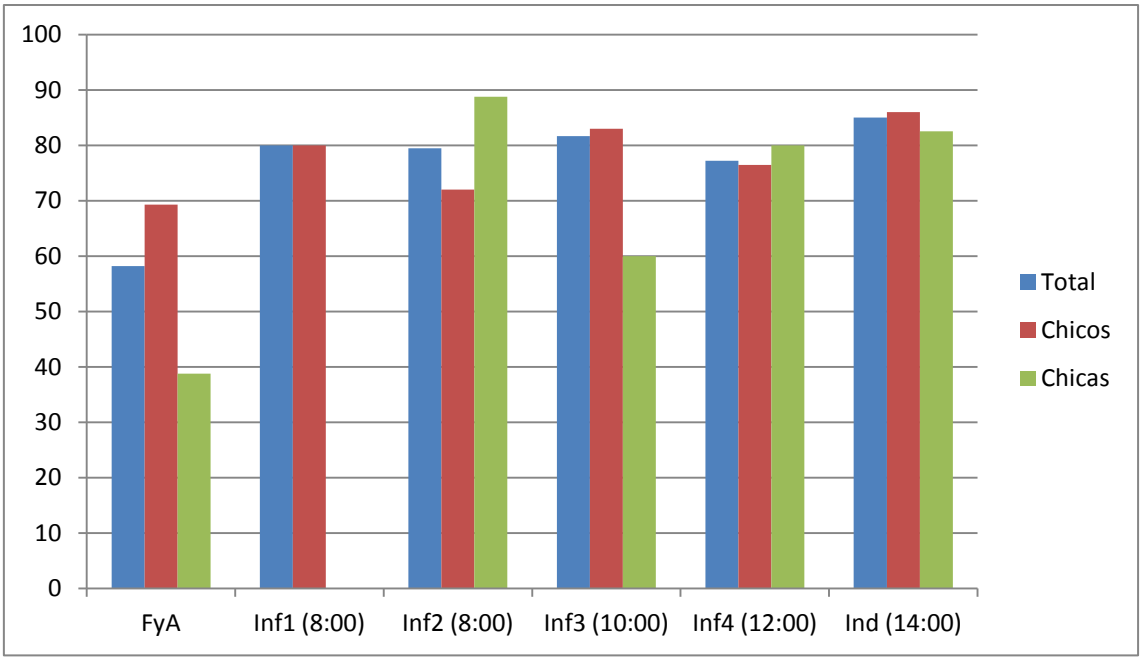


Figura G.3.4: *Porcentaje medio de aciertos obtenido por cada uno de los grupos de usuarios encuestados para las animaciones emocionales reproducidas en la prueba que determina la relevancia de la imagen con respecto a la voz en la percepción emocional del usuario. En color azul se representa la media total de aciertos de cada grupo, mientras que en rojo y verde se muestran las medias de aciertos obtenidas por los chicos y chicas del grupo respectivamente.*

Como se aprecia en la figura anterior, los resultados obtenidos para las animaciones reproducidas en esta tercera prueba son ostensiblemente mejores que los de las voces emocionales. En este sentido, a excepción del grupo de usuarios formado por los familiares y amigos del desarrollador (FyA), todos los grupos de usuarios restantes presentan un porcentaje medio de acierto superior al 70%, hecho que refuerza aún más la conclusión de que la imagen mostrada del agente influye de forma determinante en la percepción emocional del usuario.

Finalmente, la Tabla G.3.3 muestra el porcentaje medio de acierto obtenido para las animaciones reproducidas en esta tercera prueba en función del sexo del encuestado, pudiéndose apreciar que la diferencia ronda el 7.3% de acierto favorable a los usuarios.

Porcentaje de Aciertos Total	Porcentaje de Aciertos en Chicos	Porcentaje de Aciertos en Chicas
75.31%	77.29%	70.00%

Tabla G.3.3: *Porcentajes medios de acierto obtenidos para las animaciones reproducidas en esta tercera prueba.*

Referencias bibliográficas

[AB ficheros web] Link al repositorio oficial de la fundación dedicada a la Inteligencia Artificial A.L.I.C.E. donde se es posible descargarse los ficheros fuentes del programa AB *https://code.google.com/p/program- ab/downloads/list?can=2&q=AB&colspec=Filename+Summary +Uploaded+ReleaseDate+Size+DownloadCount* (último acceso 21-02-2013)

[A.L.I.C.E. web] Página web oficial de la fundación dedicada a la Inteligencia Artificial A.L.I.C.E. *http://www.alicebot.org/* (último acceso 31-07-2013)

[Anaya, 2006] Anaya, D. Desarrollo de un motor para la comunicación verbal usuario-avatar. Proyecto Fin de Carrera de la Universidad de Zaragoza (Zaragoza, España, Septiembre, 2006)

[Android web] Página web oficial de Google donde se dan a conocer guías, principios de diseño y sugerencias para el desarrollo de interfaces de usuarios para aplicaciones Android *http:/developer.android.com/design/index.html* (último acceso 07-02-2014)

[Baldassarri et al, 2007] Baldassarri S., Cerezo E., Serón F.J., Cuartero E., Montoro G., Haya P.A., Alamán X. Agentes virtuales 3D para el control de entornos inteligentes domóticos. Actas del VIII Congreso de Interacción Persona-Ordenador (Zaragoza, España, Septiembre, 11-14-2007)

[Baldassarri et al, 2008] Baldassarri S., Cerezo E., Serón F.J. Maxine: a platform for embodied Animated Agents. Computers & Graphics vol. 32(4), páginas 430-437, 2008

[Baldassarri et al, 2009] Baldassarri S., Cerezo E., Royo-Santas F. Automatic translation system to spanish sign language with a virtual interpreter. En las Actas del 12ª Conferencia IFIP TC13 sobre Human-Computer Interaction Lecture notes in Computer Science, VOL 5726/2009, ISBN978-3-642-03654-5, páginas 196-199 (Uppsala, Suecia, 2009)

[Boyatzis et al, 1994] Boyatzis C.J., Varghese R. Children's emotional associations with colors. En The Journal of Genetic Psychology: Research and Theory of Human Development, Volumen 155, páginas 77-85 (Estados Unidos, Marzo, 29-03-1994)

[Bush, 2001] Bush N. Artificial Intelligence Markup Languahe (AIML) Version 1.0.1. Documento propiedad de la A.L.I.C.E. AI Foundation Working Draft que se encuentra publicado en su web oficial (Octubre, 25-10-2001) *http://www.alicebot.org/TR/2001/WD-aiml/*

[Buttussi et al, 2006] Buttussi F., Chittaro L., Nadalutti D. Bringing mobile guides and fitness activities together: a solution based on an embodied virtual trainer. En las actas de la Octava Conferencia sobre Human-computer interaction with mobile devices and services, páginas 29-36, MobileHCI'06, (Helsinki, Finlandia, Septiembre, 12/15-09-2006)

[Buttussi et al, 2007] Buttussi F., Chittaro L., Coppo M. Using Web3D technologies for visualization and search of signs in an international sign language. En las actas de la duodécima conferencia internacional sobre 3D Web Technology, páginas 61-70, (Perugia, Italia, Abril, 15/18-04-2007)

[Bradley et al, 1994] Bradley M.M., Lang P.J. Measuring emotion: the self-assessment manikin and the semantic differential. En el Journal of Behavioral Therapy and Experimental Psychiatry, volumen 25, páginas 49-59 (Florida, USA, Marzo, 20-03-1994)

[Chittaro et al, 2006] Chittaro L., Buttussi F., Nadalutti D. MAge-AniM: a system for visual modeling of embodied agent animation and their replay on mobile devices. En las actas de la International Working Conference on Advanced Visual Interfaces, AVI'06, (Venecia, Italia, Mayo, 23/26-05-2006)

[CMU Sphinx web] Repositorio de los paquetes de CMU Sphinx donde se explica la funcionalidad que provee cada uno de estos paquetes *http://cmusphinx.sourceforge.net/wiki/download* (último acceso 23-12-2012)

[Colores Emociones web] Artículo sobre las connotaciones emocionales existentes en los colores. Este artículo se encuentra disponible en la página oficial de elsídromedelahojaenblanco *http://elsindromedelahojaenblanco.wordpress.com/2012/11/19/colores-y-emociones/* (último acceso 16-12-2013)

[CM Android web] Página web interactiva que muestra la cuota de mercado de los sistemas operativos para *smartphones* más difundidos en los países con mayor número de usuarios de estos dispositivos. *http://www.kantarworldpanel.com/smartphone-os-market-share/* (último acceso 25-04-2014)

[Cyberon Voice Commander web] Enlace web de la Google Play Store donde se describe la aplicación Cyberon Voice Commander *https://play.google.com/store/apps/details?id=com.cyberon.cvc.ESP&hl=es* (último acceso 11-04-2014)

[Dictionary web] Referencia a la clase pública Dictionary del paquete System.Collections.Generic, *http://msdn.microsoft.com/es-es/library/xfhwa508(v=vs.110).aspx* (último acceso 12-02-2014)

[Episteme Engine web] Enlace para comprar a través de la Asset Store de Unity 3D la biblioteca que contiene el Episteme Engine, *https://www.assetstore.unity3d.com/en/#!/content/4804* (último acceso 07-03-2014)

[eSpeak web] Página web donde se informa de las características del sintetizador de discurso eSpeak y se dispone de las últimas versiones de dicho software *http://espeak.sourceforge.net* (último acceso 02-02-2014)

[Francisco et al, 2005] Francisco, V., Gervás, P., Hervás, R. Expresión de emociones en la síntesis de voz en contextos narrativos. En las actas del Primer Simposio sobre Computación Ubicua e Inteligencia Ambiental (UCAmi'05), (Granada, España, Septiembre, 2005)

[Google Voice Search web] Enlace web de Softonic donde se describe la aplicación Google Voice Search y se permite al visitante descargarse la aplicación *http://voice-search.en.softonic.com/android* (último acceso 21-02-2014)

[Hemphill, 1996] Hemphill M. A note on adult's color-emotion associations. En The Journal of Genetic Psychology: Research and Theory of Human Development, Volumen 157, páginas 275-280 (Estados Unidos, Julio, 10-07-1995)

[Kaya et al, 2004] Kaya N., Epps H.H. Relationship between color and emotion: a study of college students. En la College Student Journal, Volumen 38, páginas 396-405, (Estados Unidos, Septiembre, 12-09-2004)

[Klaasen et al, 2012] Klaasen R., Hendrix J., Reidsma D., Akker op den H.J.A. Elckerlyc goes Mobile Enabling Technology for ECA's in Mobile Applications. En la Sixth International Conference on Mobile Ubiquitous Computing Systems, Services and Technologies, UBICOMM 2012, (Barcelona, España, Septiembre, 23/28-09-2012)

[Konele web] Enlace web donde se describe la aplicación de reconocimiento de voz Konele y se disponen los links para descargarse dicha aplicación *https://code.google.com/p/recognizer-intent/* (último acceso 15-08-2013)

[Nadalutti et al, 2006] Nadalutti D., Chittaro L., Buttussi F. Rendering of X3D content on mobile devices with OpenGL ES. En las actas de la undécima conferencia internacional sobre 3D Web Technology, Web3D'06, páginas 19-26, (Columbia, Estados Unidos, Abirl, 28/21-4-2006)

[PocketSphinx web] Tutorial de CMU Sphinx donde se explica, paso a paso, cómo hacer uso de PocketSphinx en un dispositivo Android. *http://cmusphinx.sourceforge.net/wiki/tutorialandroid* (Último acceso 08-04-2014)

[ProgramAB web] Enlace de la página oficial de la fundación A.L.I.C.E. donde se explica en detalle las funcionalidades del Programa AB y cómo incorporar dicho programa al proyecto sobre el que se trabaja. *https://code.google.com/p/program-ab/* (Último acceso 13-04-2013)

[RecognizerIntent web] Referencia a la clase pública RecognizerIntent del paquete android.speech, *http://developer.android.com/reference/android/speech/RecognizerIntent.html* (último acceso 08-02-2014)

[Regex web] Referencia a la clase pública Regex del paquete System.Text.RegularExpressions, *http://msdn.microsoft.com/en-us/library/system.text.regularexpressions.regex.aspx* (último acceso 21-05-2014)

[Rumbaugh, 1996] Rumbaugh J. Modelado y diseño orientado a objetos. Metodología OMT. Prentice Hall International Edition, España, 1996.

[Morris, 1995] Morris J.D. Observations: SAM: The Self-Assessment Manikin. Publicado en la Journal os Advertising Research, volumen 35, páginas 63-68, (Florida, Estados Unidos, Noviembre, 2-11-1995)

[Santi et al, 2003] Santi S., Koki U., Mitsuru I. Multimodal Presentation Markup Language on Mobile Phones. En las actas de la Fourth International Workshop, IVA 2003, páginas 226-230 (Kloster Irsee, Alemania, Septiembre, 15/17-09-2003)

[Santos-Pérez et al, 2013] Santos-Pérez M., González-Parada E., Cano-García J.M. ECA-based Control Interface on Android for Home Automation System. En la 2013 IEEE International Conference on Consumer Electronics (ICEE), páginas 70-71, (Las Vegas, Estados Unidos, Enero, 11/14-01-2013)

[Serón et al, 2008] Serón F.J., Baldassarri S., Cerezo E. Problem based learning and interactive embodied pedagogical agents. Anexo a las Actas de la Conferencia EUROGRAPHICS 2008 (Creta, Grecia, 2008)

[Sinestesia web] Trabajo existente en la página oficial del grupo de trabajo Shetshift de la Universidad de Málaga *http://www.ugr.es/~setchift/docs/cualia/sinestesia_colores_emociones.pdf* (último acceso 16-12-2013)

[Smarcos web] Página web oficial del proyecto Smarcos, perteneciente a la EU Artemis *http://www.smarcos-project.eu/* (último acceso 09-01-2013)

[Sommerville, 2005] Sommerville I. Ingeniería del Software (Software Engineering). Addison-Wesley, 7ª edition, 2005.

[SpeechRecognizer web] Referencia a la clase pública SpeechRecognizer del paquete android.speech, *http://developer.android.com/reference/android/speech/SpeechRecognizer.html* (último acceso 10-02-2014)

[TextToSpeech web] Referencia a la clase pública TextToSpeech del paquete android.speech.tts, *http://developer.android.com/reference/android/speech/tts/TextToSpeech.html* (último acceso 08-02-2014)

[Unity3D web] Página web oficial de la plataforma de desarrollo Unity3D, *http://unity3d.com/es* (último acceso 01-05-2014)

[Vlingo web] Página web oficial del asistente de voz Vlingo *http://www.vlingo.com/* (último acceso 02-10-2012)

[Voice Action Plus web] Enlace web donde se describe la aplicación Voice Action Plus y se muestran distintos links de descarga *http://android.downloadatoz.com/apps/com.pannous.voice.actions,70211.html* (último acceso 06-05-2014)

[VoicePOD web] Enlace web donde se describe la aplicación VoicePOD y se disponen varios links de descarga *http://android.downloadatoz.com/apps/it.mrqzzz.voicepod20,24758.html* (último acceso 16-03-2014)

[XML web] Enlace web a la página oficial del 3WC donde se presenta el lenguaje Extensible Markup Language (XML), *http*://www.w3.org/XML/ (último acceso 17-07-2014)

[Zilmer, 2012] Zilmer L. Animating emotions in ECA's for interactive applications. Tesis Final de Máster, Universidad de Aalborg (Copenhague, Dinamarca, Mayo, 23-05-2012)