

**Universidad
Rey Juan Carlos**

**ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS**

Curso Académico 2010/2011

Proyecto de Fin de Carrera

**Aplicación de ayuda a la comunicación,
para personas con discapacidad auditiva,
sin intérprete, u oyentes con desconocimiento
del idioma del receptor**

Autor: Jebel Gil Sánchez

Tutora: Estefanía Martín Barroso



AGRADECIMIENTOS

Recuerdo la fascinación que sentía de pequeño cuando veía a mi padre, sentado delante de su ordenador, escribiendo cosas inteligibles para mí. De aquella pantalla de fósforo verde ha pasado ya bastante tiempo, sin embargo la sensación de querer aprender qué significaban esas extrañas palabras sigue estando aún muy presente en mi interior. Por eso quiero agradecerle que plantara en mí esa semilla de curiosidad hacia su mundo, porque gracias a ella ahora también se ha convertido en el mío.

Pero no quiero dejar de dar también las gracias al resto de mi familia y amigos por haberme apoyado y ayudado siempre que lo he necesitado. También a mi tutora, Estefanía, por haber estado tan involucrada en este proyecto y por proporcionarme toda la ayuda que he requerido.

Y especialmente gracias a la persona que puso los cimientos de este proyecto y sobre todo de mi vida.

RESUMEN

La comunicación es uno de los pilares básicos en los que se sustenta la convivencia entre personas dentro de una sociedad. Necesitamos poder transmitir información, mensajes a nuestros semejantes ya sea para el ámbito personal como para el laboral. Los seres humanos disponemos de múltiples maneras de comunicarnos, pero se podrían catalogar en dos grupos bien diferenciados: lingüísticos y no lingüísticos. Es decir, los podemos clasificar en aquellos en los que se emplee el lenguaje (ya sea escrito u oral) y aquellos en los que no es necesario emplearlo para poder transmitir un mensaje (comunicación visual, gestual y acústica).

Después de este breve preámbulo vamos a centrarnos en un problema que se nos plantea, ¿cómo comunicarnos con personas que desconocen nuestro lenguaje o idioma? Este es un problema con el que tienen que lidiar a menudo las personas con discapacidad auditiva. Este colectivo dispone de un lenguaje propio, el lenguaje de signos, el cuál es prácticamente tan completo como cualquier lenguaje verbal. Pero presenta un problema, y es que pocas personas oyentes lo conocen. Por ello tienen que apoyarse en terceras personas, los intérpretes. Este mismo problema de comunicación puede extenderse a personas oyentes con desconocimiento de un idioma común.

Por todo ello podría considerarse de gran utilidad una herramienta que, en medida de lo posible, pueda solventar este gran obstáculo. El presente Proyecto Fin de Carrera, mediante una aplicación desarrollada para dispositivos móviles con sistema operativo *Android*, persigue este fin.

Con esta aplicación se pretende ayudar, al usuario de la misma, a comunicarse con otra persona que desconozca su idioma. La manera de conseguirlo es mediante un traductor, el cuál contiene palabras y frases útiles para unas determinadas situaciones cotidianas. Este traductor permitirá al usuario emplear tanto la comunicación lingüística (traduciendo las palabras o frases a diferentes idiomas) como la no lingüística (empleando en este caso símbolos que no requieren de un aprendizaje previo para comprenderlos). De esta manera no nos encontraremos ligados al número finito de idiomas que puede albergar un traductor si no que, a través de estos símbolos pictográficos, podremos comunicarnos con cualquier persona sin importarnos el idioma que use, logrando por tanto una comunicación básica universal.



TABLA DE CONTENIDOS

1. INTRODUCCIÓN	1
2. OBJETIVOS	7
A. Descripción del problema.....	7
B. Requisitos de usuario.....	7
C. Estudio de alternativas.....	8
D. Metodología empleada.....	16
3. DESCRIPCIÓN INFORMÁTICA	19
A. Casos de uso.....	19
a. Diagrama casos de uso.....	19
b. Descripción casos de uso.....	20
B. Diagrama de actividad.....	26
C. Interfaz de usuario de la aplicación.....	30
a. Menú aplicación “Dispositivo de Ayuda a la Comunicación”.....	31
b. Vocabulario.....	31
c. Uso de Vocabulario.....	33
D. Diseño de la base de datos.....	34
a. SQLite.....	34
b. Sqlite3.....	34
c. Diseño.....	35
E. Implementación.....	36
a. Estructura del proyecto.....	36
b. Componentes de la aplicación.....	38
4. EVALUACIÓN	51
A. Instalación de la aplicación.....	51
B. Tutorial y formularios.....	52
C. Estudio de los resultados obtenidos en la evaluación.....	52
5. CONCLUSIONES Y POSIBLES AMPLIACIONES	57
BIBLIOGRAFÍA	61
A. Referencias.....	61
B. Libros y sitios web.....	61
ANEXO A: INSTALACIÓN	63
A. Instalación del entorno de desarrollo.....	63
a. Entorno de desarrollo integrado.....	63
b. SDK (Software Development Kit).....	65
c. ADT (Android Development Tools).....	66
B. Instalación de la aplicación en un dispositivo móvil.....	68
a. Generación del fichero <i>.apk</i>	68
b. Instalación del fichero <i>.apk</i> en el dispositivo móvil.....	68
ANEXO B: TUTORIAL Y CUESTIONARIOS	71
A. Tutorial.....	71
a. Pantallas de la aplicación.....	72
B. Ejemplos de uso en posibles situaciones reales.....	81
a. En el restaurante.....	81
b. Buscando el teatro.....	86
C. Cuestionarios.....	92



a.	Cuestionario inicial personas sin discapacidad auditiva.....	93
b.	Cuestionario final personas sin discapacidad auditiva	95
c.	Cuestionario inicial personas con discapacidad auditiva	97
d.	Cuestionario final personas con discapacidad auditiva.....	99

ÍNDICE DE FIGURAS

Figura 1: Componentes de la arquitectura <i>Android</i>	15
Figura 2: Diagrama de patrón MVC.....	17
Figura 3: Casos de uso	20
Figura 4: Actividades principales de la aplicación	27
Figura 5: Actividades dentro del caso de uso “Vocabulario”	28
Figura 6: Actividades dentro del caso de uso “Uso de Vocabulario”	30
Figura 7: Boceto del menú principal de la aplicación	31
Figura 8: Boceto de la vista clásica del módulo Vocabulario	32
Figura 9: Boceto de la vista de grupos del módulo Vocabulario	32
Figura 10: Boceto de la pantalla de selección de situación del módulo Uso de Vocabulario	33
Figura 11: Boceto de la pantalla de selección de palabra para completar una frase incompleta	33
Figura 12: Estructura del proyecto implementado.....	38
Figura 13: Activity - Carga de la IU.....	39
Figura 14: Activity - Carga de la IU.....	40
Figura 15: Interfaz de usuario del menú principal	41
Figura 16: Interfaz de usuario - Botones.....	42
Figura 17: Interfaz de usuario - ListView.....	43
Figura 18: Interfaz de usuario - ListView.....	43
Figura 19: Interfaz de usuario - TabLayout.....	44
Figura 20: Interfaz de usuario - TabLayout.....	45
Figura 21: Interfaz de usuario – Menú	47
Figura 22: Intent – Emisión.....	49
Figura 23: Intent – Recepción	49
Figura 24: Intent – Declaración en AndroidManifest.....	49
Figura 25: Emulador Android	51
Figura 26: Perfil discapacitados auditivos.....	53
Figura 27: Perfil oyentes	53
Figura 28: Resultados sobre la utilización de la aplicación y facilidad de uso - Discapacitados auditivos	54
Figura 29: Resultados sobre la ayuda que proporciona a la comunicación - Discapacitados auditivos	54
Figura 30: Resultados sobre la facilidad de uso y la satisfacción - Oyentes.....	55
Figura 31. Pantalla inicial de la aplicación.....	72
Figura 32. Ayuda sobre la aplicación	73
Figura 33. Configuración de los idiomas y símbolos utilizados por la aplicación	73
Figura 34. Vista clásica del vocabulario.....	74
Figura 35. Vista por grupos dentro de la opción de vocabulario.....	75
Figura 36. Grupos de palabras del vocabulario de la situación Lugares	75
Figura 37. Traducción de una palabra determinada al idioma seleccionado e icono correspondiente.	76
Figura 38. Menú principal sobre el uso de vocabulario o frases de la aplicación.....	77
Figura 39. Frases asociadas a la temática alojamiento.....	78

Figura 40. Traducción de una frase determinada al idioma seleccionado y a los símbolos pictográficos.....	78
Figura 41. Frases asociadas a la temática transporte.....	79
Figura 42. Presentación de la frase incompleta seleccionada y de las palabras compatibles.	80
Figura 43. Traducción de una frase incompleta al idioma seleccionado y a los símbolos pictográficos.....	81
Figura 44. Situación de entrada en un restaurante.....	82
Figura 45. Pantalla inicial de la aplicación y selección del menú de configuración.....	82
Figura 46. Configuración de los idiomas de la aplicación incluyendo el inglés.....	83
Figura 47. Selección de la situación restaurante dentro del módulo uso del vocabulario.	83
Figura 48. Selección de la frase “Quiero una mesa para dos” y su correspondiente traducción.	83
Figura 49. Situación en la que se pide la cuenta al camarero.....	84
Figura 50. Selección de la frase “Quiero la cuenta” y posterior traducción.	84
Figura 51. Situación en la que se pregunta sobre el pago con tarjeta.....	85
Figura 52. Selección y traducción de la frase “¿Puedo pagar con tarjeta?”	85
Figura 53. Situación en la que los clientes intentan irse sin pagar y acude la policía.....	86
Figura 54. Traducción de la frase “¿Puedo hablar con un abogado?” dentro de la temática comisaria.	86
Figura 55. Medios de transporte - Teatro	87
Figura 56. Selección de la frase incompleta “Como llego a el/la” dentro del grupo “Lugares”.	87
Figura 57. Selección de una de las palabras compatibles junto con su posterior traducción.	88
Figura 58. Situación en la que se pregunta por el teatro a otra persona utilizando la aplicación.....	88
Figura 59. Selección de la temática transporte en respuesta a la pregunta realizada.....	89
Figura 60. Selección del autobús como medio de transporte para ir al teatro.....	89
Figura 61. De nuevo, selección de la temática transporte.	90
Figura 62. Pregunta sobre el número del autobús que hay que coger para ir al teatro....	90
Figura 63. Situación en la que una persona acompaña a otra a la parada de autobús apropiada.	90
Figura 64. Situación en la que se encuentran en la parada.	91
Figura 65. Indicación del color del autobús utilizando la aplicación.	91
Figura 66. Indicación del número de autobús.....	92
Figura 67. Fin del tutorial.....	92

1. INTRODUCCIÓN

“He aquí que todos forman un solo pueblo y todos hablan una misma lengua, siendo este el principio de sus empresas. Nada les impedirá que lleven a cabo todo lo que se propongan. Pues bien, descendamos y allí mismo confundamos su lenguaje de modo que no se entiendan los unos con los otros.”

El anterior párrafo presenta un fragmento del antiguo testamento de la Biblia, capítulo 11 del Génesis, en el que se relata la construcción de la torre de Babel. De este texto se quiere resaltar la importancia de la comunicación, ya que gracias a una buena comunicación todo es posible.

Por ello, para facilitar la comunicación dentro de un mismo país existe, al menos, un idioma común con el que sus habitantes puedan comunicarse. Pero ¿qué sucede si no podemos emplear este idioma común? Si ocurre esta situación, a diario nos encontraremos con dificultades para comunicarnos. Cualquier situación cotidiana puede convertirse en un problema si no se puede transmitir el mensaje deseado de una manera que el receptor lo entienda. Con este problema deben convivir muchas personas con discapacidad auditiva.

Estas personas, debido a la imposibilidad de comunicarse a través de un canal vocal-auditivo, emplean la lengua de signos, la cuál usa el canal gesto-viso-especial. La lengua de signos es una lengua natural como puede ser el castellano, con un léxico determinado y una estructura gramatical perfectamente definida. Sin embargo presenta el problema de que pocas personas lo comprenden, reduciéndose por tanto la comunicación de sus hablantes a un entorno reducido. Cuando estas personas necesitan comunicarse con un oyente con desconocimiento de la lengua de signos es cuando existen complicaciones, y aquí es donde entra el papel del intérprete. El intérprete se encarga de actuar de puente entre un sordo y un oyente. Debe estar presente en juzgados, comisarías, hospitales, oficinas de la administración pública, universidades, centros de formación, etc. Sin embargo su número es muy reducido en comparación con la demanda que requiere sus servicios, sin contar la imposibilidad que existe de disponer de un intérprete instantáneamente para solventar una situación determinada en un momento determinado. Es por ello que puede resultar de gran utilidad disponer de un

dispositivo que, de una manera rápida y sencilla, permita transmitir un mensaje de una manera que el receptor lo comprenda.

Hasta el momento se ha expuesto el problema que se pueden encontrar las personas con discapacidad auditiva en la comunicación con otra persona encontrándose cara a cara, pero no se ha comentado nada sobre la comunicación a distancia. Si se habla de comunicación a distancia una palabra viene a nuestras cabezas, el teléfono.

Desde que el teléfono fue inventado en el siglo XIX [1] se ha vuelto completamente imprescindible en la comunicación entre personas pues permite “acercarlas”, pudiendo hablar dos personas situadas en distintos lugares como si estuvieran una al lado de la otra. Pero nuevamente nos encontramos el problema de que lo que se transmite es sonido y si no somos capaces de recibir este sonido no podemos comunicarnos. Por ello surgieron los teléfonos de texto.

El teléfono de texto es un aparato que consta de un teclado y una pantalla, y que se conecta a la línea telefónica como un teléfono clásico, pudiendo establecer una comunicación “telefónica” intercambiando mensajes escritos. Sin embargo este aparato presenta inconvenientes, siendo el más importante de ellos que no haya una normalización estándar en el protocolo de comunicación que emplean. Actualmente se emplean varios de estos protocolos como pueden ser CF.PT, F.DT, DTMF, V21, BAUDOT [2]. Las personas que usen teléfonos de texto que empleen diferentes protocolos no podrán comunicarse. Al margen de este gran problema, también existe uno inherente al propio dispositivo y es que para comunicarse es necesario escribir en un teclado mecanográfico, con la lentitud que ello conlleva si no se tiene suficiente destreza. Por todo ello, muchas personas prefieren el uso del fax, aunque ello no les permita realizar una comunicación tipo diálogo.

Debido al problema de compatibilidad entre teléfonos de texto surge el centro de intermediación telefónica para sordos. Este centro es un servicio que facilita la comunicación telefónica entre personas sordas que disponen de aparatos telefónicos diferentes e incompatibles entre sí, a la vez que ofrece a la persona sorda la posibilidad de entrar en comunicación con las personas oyentes, usuarias del teléfono convencional. Todo esto es

posible gracias a la intervención de personas que funcionan en calidad de operadores, actuando como intercomunicadores entre mensajes escritos y/o hablados. El inconveniente de este centro es que se sigue manteniendo el problema de la lentitud en la conversación, puesto que sigue siendo necesario escribir en el teléfono de texto, y se le añade la falta de intimidad al existir un intermediario entre el emisor y el receptor.

Por tanto, el uso del teléfono de texto no solventa completamente los problemas de comunicación a distancia de personas con discapacidad auditiva. Sin embargo, actualmente está teniendo mucha proyección otro tipo de comunicación y que, al contrario que el teléfono, es perfectamente válido para personas con discapacidad auditiva. Hablamos de la videollamada.

La videollamada no sólo transmite sonido si no que transmite la imagen del interlocutor, por lo que puede emplear la lengua de signos para comunicarse con el receptor. El objetivo del teléfono, “acercar” a personas para que puedan comunicarse empleando su lengua natural, es de esta manera posible. Sin embargo al emplear la lengua de signos nos encontramos nuevamente con el problema del entendimiento con personas oyentes con desconocimiento de esta lengua. Por ello surge la video-interpretación en lengua de signos, *SVIsual* [3]. Este servicio es similar al que proporciona el centro de intermediación telefónica para sordos pero adaptado a la videollamada. En este caso, en lugar de que un operador se encargue de pasar de texto a voz y viceversa, nos encontramos con un intérprete que traduce la lengua de signos al castellano. El problema de falta de intimidad que se tenía en el centro de intermediación telefónica para sordos se sigue manteniendo también en esta modalidad.

Afortunadamente se están realizando grandes avances en reconocimiento de movimientos, curiosamente a través de sistemas de juego como puede ser *Kinect* [4] de *Microsoft*, el cuál permite controlar el sistema a través de los movimientos corporales del usuario. Por ello puede que no esté tan lejos el momento en el que exista un programa que pueda interpretar la lengua de signos sin necesidad de una tercera persona.

Hasta ahora hemos hablado de dispositivos que se encuentran en el hogar, pero centrémonos en otros dispositivos que actualmente están evolucionando a una gran velocidad y que permiten múltiples modos de comunicación. Estamos hablando de los *Smartphone* [5].

Un *Smartphone* permite centralizar en un sólo aparato distintos modos de comunicación ya que dispone de teléfono, envío-recepción de mensajes de texto, correo electrónico y videollamada. Parece que se quiere potenciar este modo de comunicación como podemos ver en el último anuncio de *Apple* y su *Iphone 4* con *FaceTime* [6] (aplicación que proporciona videollamada entre usuarios de dispositivos de esta marca). También la aplicación *Skype* [7], la cuál dispone de un gran número de usuarios, permite, entre otras cosas, realizar videollamada. Esto es una buena noticia para las personas con deficiencias auditivas puesto que, a priori el sistema que más ventajas le aporta, va a estar disponible en su palma de la mano.

Pero centrémonos de nuevo en el problema que habíamos aparcado para hablar de dispositivos de ayuda para personas con deficiencias auditivas, la comunicación cara a cara. Se ha hablado que un *Smartphone* puede ser de gran utilidad para comunicarse a distancia con otras personas, pero ¿por qué no usarlo también para comunicarse cara a cara? Parece una buena idea disponer en un único dispositivo, portátil y manejable, de los diferentes modos de comunicación a distancia y que también permita, de una manera rápida, una comunicación cara a cara transmitiendo al receptor un mensaje de un modo compresible.

Por este motivo, sería muy interesante disponer de aplicaciones en estos *Smartphones* que ayuden a comunicarse a las personas con discapacidad auditiva. Siguiendo esta línea podemos encontrar la aplicación *Slec* [8]. Esta aplicación para *Smartphone* ha sido desarrollada para ayudar a las personas con discapacidad auditiva en situaciones de emergencia. Para ello emplea frases predefinidas que le serán traducidas a lengua de signos utilizando videos. Esta aplicación ya ha sido utilizada por enfermeras y por personas con discapacidad auditiva. Sin embargo, y pese a que la idea es muy buena, presenta limitaciones y es que únicamente está pensada para situaciones de emergencia. No presenta frases que podamos considerar útiles para las actividades que realizamos en nuestra vida diaria. Por otro lado la aplicación está orientada a que la persona con discapacidad auditiva pueda entender el mensaje que se le

quiere transmitir, pero no se contempla el caso contrario. Es decir, no se contempla el caso de que una persona con discapacidad auditiva comunique un mensaje a otra que desconozca la lengua de signos. Además esta aplicación únicamente sería válida dentro de un mismo país, ya que la lengua de signos no es universal, presenta diferencias entre países.

Estos son los aspectos que se quieren cubrir en la aplicación de este Proyecto Fin de Carrera, proporcionar a las personas con deficiencias auditivas un modo rápido y eficaz de comunicación cara a cara que le ayude en las tareas cotidianas sin necesidad de recurrir a un intérprete. La aplicación constará de dos partes. Por un lado contendrá un vocabulario con palabras ordenadas alfabéticamente y por grupos de temática común y por otro lado frases prefijadas, también agrupadas por temáticas, que abarcan un gran número de situaciones cotidianas en las que el usuario puede encontrarse. Para permitir más flexibilidad, las frases podrán ser completas o incompletas. En el caso de frases incompletas, el emisor de la información tendrá que seleccionar aquellas palabras que encajan en el contexto de la frase. Cada una de las frases incompletas tiene asociado un conjunto de palabras. De este modo el usuario simplemente tendrá que seleccionar la palabra que desee o escoger su situación actual y, dentro de la misma, una de las frases que se adapte a lo que quiere transmitir.

Para que el receptor entienda el mensaje éste se mostrará, a elección del usuario, en uno o varios de los idiomas en los que está traducida la palabra o frase (comunicación lingüística ya que se emplea un lenguaje) y/o como secuencia de símbolos pictográficos (comunicación no lingüística, ya que no es necesario conocer ningún lenguaje para comprender el mensaje). De esta manera nos aseguramos que el receptor, independientemente del lenguaje que use o conozca va a entender el mensaje que le queramos comunicar, ampliando su uso no sólo a España, si no a cualquier parte del mundo. Por tanto esta aplicación no sólo va a resultar útil a personas con discapacidad auditiva si no a todas aquellas que quieran comunicarse con una persona que desconoce nuestro idioma, logrando de esta manera una comunicación básica universal.

A continuación, se detalla la estructura del presente documento:

- La sección 2 describe con detalle el problema a resolver, los requisitos de usuario (incluyendo los requisitos funcionales y no funcionales de la aplicación), el estudio de las alternativas y la metodología empleada.
- La sección 3 detalla la descripción informática. En concreto, la identificación y descripción de casos de uso, los diagramas de actividad, el diseño de la base de datos, el diseño de la interfaz gráfica a través de bocetos y la propia implementación de la aplicación, analizando alguno de sus componentes más característicos.
- La sección 4 indica los detalles de las pruebas realizadas sobre la aplicación. También presenta los resultados de las encuestas realizadas a personas con distintos grados de pérdida auditiva, logopedas especializados en niños con problemas auditivos y personas sin discapacidad auditiva, los cuáles han sido los encargados de realizar la evaluación de la aplicación.
- La sección 5 expone las conclusiones de este trabajo junto con las posibles mejoras y ampliaciones de la aplicación.

Para concluir, en este documento también se encontrarán dos anexos:

- El anexo A muestra el proceso de instalación y configuración del entorno de desarrollo.
- El anexo B consta de un tutorial de la aplicación, explicando su funcionamiento a través de las diferentes pantallas de la aplicación así como mostrando viñetas de comic de posibles situaciones de uso. También se encontrarán los formularios usados para la evaluación de la aplicación.

2. OBJETIVOS

A. Descripción del problema

La comunicación es de vital importancia para una convivencia satisfactoria dentro de una sociedad. Sin embargo las personas con deficiencias auditivas tienen, en ocasiones, problemas a la hora de desarrollarla con personas que se encuentran fuera de su entorno social. Este problema tiene su raíz en el idioma que emplean para comunicarse, la lengua de signos, el cual pocas personas conocen. Este mismo problema, no obstante, puede extenderse a cualquier persona que quiera comunicarse con otra y no exista un idioma común entre ellos.

Para paliar este problema en la medida de lo posible, se ha desarrollado la “Aplicación de ayuda a la comunicación, para personas con discapacidad auditiva, sin intérprete, u oyentes con desconocimiento del idioma del receptor.”

Esta aplicación, desarrollada para dispositivos móviles, se encarga de proporcionar al usuario una serie de situaciones de la vida cotidiana con frases útiles para cada una de ellas, así como de un vocabulario de palabras ordenadas alfabéticamente o agrupadas por temática. Cada una de estas frases o palabras podrán ser traducidas a uno o varios idiomas de los que dispone la aplicación y/o a una secuencia de símbolos pictográficos de carácter universal. De esta manera se logra que el receptor al que va dirigida la frase o la palabra pueda entender lo que le queramos transmitir, estableciendo de este modo la deseada comunicación.

B. Requisitos de usuario

La aplicación que aquí se presenta está enfocada a ayudar, en una comunicación cara a cara, a personas con deficiencias auditivas o personas oyentes que desconozcan el idioma del receptor del mensaje. Por tanto disponemos de un único rol de usuario, que es aquel que necesita comunicarse con otra persona sin conocer su idioma, no existiendo ninguna distinción entre si es oyente o no oyente. De esta manera, el usuario de la aplicación tendrá acceso total a la misma, pudiendo consultar todas las palabras o frases disponibles en ella, así como la posibilidad de configurarla para que la traducción se realice en los idiomas deseados.

Una vez comentados los requisitos funcionales veamos los requisitos no funcionales de la aplicación. La aplicación está desarrollada con vistas a ser usada en exteriores, es decir, no va a ser usada en el hogar si no fuera de él. Por este motivo el requisito más importante es que el dispositivo en el que esté alojada sea lo más portable posible, tanto en tamaño como en ligereza puesto que ha sido diseñada para un uso cotidiano y su transporte no debe ser un inconveniente. Pese a que el tamaño y peso deben ser comedidos, no debe sacrificarse un buen tamaño de pantalla ya que toda la interacción se va a producir en ella, por lo que también debe ser táctil. Debido a que el usuario va a usar la pantalla para manejarla, la interfaz de la misma debe ser simple y clara, manteniendo un orden lógico de las opciones que se le plantean al usuario en todo momento. Los datos almacenados en la aplicación (palabras/frases en diferentes idiomas y símbolos pictográficos) se encontrarán alojados dentro de una base de datos, por lo que el dispositivo debe tener la suficiente potencia para manejar texto e imágenes con cierta soltura. Esto es necesario, ya que otro de los objetivos de la aplicación es la fluidez en su uso, para que su utilización no conlleve un retardo que lo haga ineficiente en la comunicación.

Por todas las características requeridas, el dispositivo *hardware* que más se adapta a las mismas es un *Smartphone*, el cual proporcionará también otros métodos de comunicación alternativos, teniendo de esta manera, diversos modos de comunicación centralizados en un único dispositivo.

C. Estudio de alternativas

En el punto anterior se ha discutido sobre qué plataforma *hardware* se adaptaría mejor a las necesidades de la aplicación que aquí se presenta, llegando a la conclusión de que un *Smartphone* parece el dispositivo adecuado. Sin embargo dentro de estos dispositivos existen diversos sistemas operativos cada uno con sus puntos fuertes y débiles frente a la competencia. Veamos varios de los más importantes para tener una perspectiva general antes de indicar por qué *Android* ha sido la opción escogida.

- **iOS:** es el sistema operativo de los dispositivos móviles de la compañía *Apple*, es decir del *Iphone*, *iTouch* e *iPad*. Al ser la propia compañía la que diseña el *hardware* y el *software*, el sistema operativo está perfectamente optimizado para estos dispositivos, logrando un muy buen funcionamiento del mismo. Se trata, por tanto, de un sistema operativo estable y con gran rendimiento, gracias a su núcleo *Darwin* (el mismo sobre el que se sustenta su sistema operativo de escritorio *MacOS X*), y que proporciona al usuario una interfaz gráfica agradable e intuitiva.

Actualmente es la plataforma de dispositivos móviles en la que más desarrolladores se encuentran involucrados. El motivo de esto es, en parte, la *AppStore* [9]. La *AppStore* es la tienda de aplicaciones móviles de la plataforma *iOS*. Es uno de los pilares en los que se sustenta el éxito de este sistema operativo ya que proporciona al usuario una manera cómoda, rápida y sencilla de encontrar y adquirir aplicaciones. El desarrollador obtiene el 70% de las ganancias obtenidas por la aplicación mientras que el 30% restante va a parar a *Apple* a cambio del servicio de hospedaje, publicidad y gestión de la misma. También es *Apple* quien toma la decisión de aceptar o rechazar estas aplicaciones, a través de un listado de condiciones que deben cumplir todas ellas, siendo en ocasiones demasiado restrictivos.

Como puntos negativos indicar que el entorno de desarrollo (*X-Code*) no es gratuito y sólo es compatible con el sistema operativo de los equipos de *Apple*, *MacOS X*, siendo por tanto necesario disponer de uno de ellos para poder programar para esta plataforma. Con este entorno de desarrollo se puede programar y probar la aplicación en una máquina virtual pero si se quiere instalar en un dispositivo real o distribuirla en la *AppStore* es necesario comprar la llamada “Licencia de desarrollador de *Apple*”, la cuál tiene un coste anual de 100\$.

Por último indicar que dispone de licencia *EULA*¹, lo cual quiere decir que es *software* de código cerrado y que es *Apple* quien establece sus derechos de uso, distribución, copia, modificación, etc.

¹ EULA (End User License): Licencia de software propietario o privativo.

- **Android:** es un sistema operativo libre para dispositivos móviles, basado en *GNU/Linux* y con licencia *GPL*² (esto es si tenemos en cuenta el núcleo, sin él la licencia, libre igualmente aunque con más orientación comercial, es *ASL*³), creado por la *Open Handset Alliance*⁴, en la cual participa *Google* con otras empresas del calibre de *HTC*, *Motorola*, *Telefónica*, *Samsung* o *Intel*. Los dispositivos en los que podemos encontrar esta plataforma van desde lectores de tinta electrónica a teléfonos móviles. Según dice *Google* en su sitio *Web*, “*Android* ha sido diseñado con el objetivo explícito de constituirse en la primera plataforma abierta, integral y gratuita creada específicamente para dispositivos móviles”. Esto lo convierte en un proyecto complicado y ambicioso, pero que a su vez puede hacerle ser el dominador del mercado en el segmento de los teléfonos móviles.

Como ya se ha comentado, *Android* está basado en un *kernel* de *Linux* y su rendimiento es excelente, incorporando características nuevas cada poco tiempo. Esto es un punto positivo pero también negativo ya que está produciendo mucha fragmentación, encontrando dispositivos anclados en la versión 1.6 (por motivos que comentaremos más adelante) hasta dispositivos que ya soportan la última 2.3. Al ser un producto de *Google* éste se integra perfectamente con otros de sus servicios como *Gmail*, *Google Maps*, *Google Calendar* o *Gtalk*. Respecto a su interfaz gráfica, ésta ha evolucionado bastante y a mejor desde las primeras versiones a las actuales. Por tanto estamos ante un sistema operativo muy completo, con un gran respaldo y proyección que hacen que sea una plataforma muy recomendable y a tener en cuenta.

Al igual que *iOS*, también dispone de una tienda donde encontrar y descargar aplicaciones. En este caso se llama *Android Market*. Y pese a que dispone de menos aplicaciones que la anteriormente comentada *AppStore*, el número de ellas está creciendo a gran velocidad, superando ya las 80.000, siendo muchas de ellas gratuitas. *Google* también ha seguido el modelo de reparto de ganancias en la venta de aplicaciones impuesto por *Apple* (70%

² GPL (General Public License): Licencia orientada principalmente a proteger la libre distribución, modificación y uso de software.

³ ASL (Apache Software License): Licencia de software libre creada por la Apache Software Foundation (ASF).

⁴ Open Handset Alliance (OHA): Alianza comercial de 78 compañías para desarrollar estándares abiertos para dispositivos móviles.

desarrollador y 30% *Google*). Sin embargo el coste de la licencia para poder publicarlas es bastante menor, siendo en este caso 25\$ por una licencia vitalicia.

Como puntos negativos podemos indicar algo que se ha citado anteriormente, la fragmentación. Al ser *software* libre, cualquiera puede modificarlo. Eso es lo que hacen los fabricantes de teléfonos móviles, crear una “capa” por encima de éste que modifique su interfaz gráfica para distinguir sus productos de la competencia. Esto hace que, aunque *Google* libere una nueva versión del sistema operativo, hasta que el fabricante no decida si quiere incorporarla a sus productos, modificándola nuevamente, el usuario no tiene acceso a ella. Por ello hay una gran fragmentación, encontrando numerosos dispositivos cada uno con una versión y características distintas, creando confusión y descontento entre los usuarios. También indicar que su tienda de aplicaciones todavía no se encuentra muy madura y, debido a la total libertad que proporciona *Google* para publicar aplicaciones, podemos encontrar multitud de aplicaciones con dudosa utilidad.

- **Microsoft Windows Mobile / Windows Phone:** es un sistema operativo para dispositivos móviles desarrollado por *Microsoft*. Se basa en el núcleo de su sistema operativo embebido *Windows CE*. Como principal característica podemos indicar que estaba orientado al mercado empresarial mostrando una interfaz similar a la que podemos encontrar en las versiones de escritorio. Esto no es probablemente lo más adecuado considerando el tamaño de los dispositivos a los que va dirigido. La última versión, la 7, ha presentado un cambio tan radical tanto en rendimiento como en interfaz y orientación que incluso ha cambiado el nombre de *Windows Mobile* a *Windows Phone*. Se ha intentado con este cambio no sólo seguir manteniendo clientes para uso empresarial, disponiendo de las versiones móviles de las aplicaciones de su *suite* ofimática *Office* y de la plataforma *SharePoint Workspace Mobile* con la cual se puede compartir y colaborar en proyectos de manera remota, si no captar a un público más general.

Como se ha dicho anteriormente, la interfaz ha sufrido una completa renovación y no sólo no se parece en nada a las anteriores versiones si no que le permite distinguirse claramente de la competencia debido a su originalidad e impacto visual. Sin embargo, todos estos cambios han llevado consigo un aumento importante de los requisitos *hardware*, algo

lógico por otra parte. Por tanto, en esta nueva versión, *Microsoft* ha dado un paso al frente para poder ofrecer una buena resistencia frente a la competencia proporcionando un sistema operativo fluido, atractivo y sencillo de manejar.

Sin embargo, al tratarse de una plataforma muy joven, presenta carencias importantes como puede ser la multitarea o incluso funcionalidades tan básicas como copiar-pegar. También hay que observar como evoluciona su tienda *MarketPlace* (actualmente con 4000 aplicaciones disponibles) y la acogida que tiene por parte de los desarrolladores. Por el momento tiene el respaldo de grandes fabricantes de móviles y parece que también ha tenido una buena acogida por parte de los usuarios. Sin embargo, es pronto para saber si podrá hacerse un hueco debido al tiempo de retraso que tiene respecto a sus dos grandes competidores *iOS* y *Android*.

- **BlackBerry OS:** si anteriormente hemos comentado que *Windows Mobile* ha estado orientado al mercado empresarial ahora nos disponemos a analizar el sistema operativo de los dispositivos que han dominado y dominan este segmento, estamos hablando de los dispositivos móviles *BlackBerry*.

Los dispositivos móviles *BlackBerry* son *Smartphones* de la compañía *RIM* y siempre han sido el referente como dispositivo móvil empresarial, por su seguridad y su integración con las mayores plataformas de correo electrónico como son *Exchange*, *Domino* y *GroupWise*. Curiosamente, a partir de la versión 6 de su sistema operativo, se ha captado también al público joven. Los motivos de ello son que, además de mejorar la navegación web gracias a su motor *WebKit*⁵, se han añadido aplicaciones nativas para la gestión de redes sociales y fuentes *RSS*⁶. Esto da a los jóvenes la posibilidad de acceder a *BlackBerry Messenger*, *Facebook*, *Twitter*, *MySpace* y otras tantas aplicaciones de mensajería instantánea, las cuales les permiten estar conectados continuamente con sus amigos y chatear entre ellos.

⁵ WebKit: Motor de buscador Web de código abierto.

⁶ RSS (Really Simple Syndication): Formato XML usado para difundir información actualizada a través de la Web.

Estamos, por tanto, ante uno de los sistemas operativos más extendido del mundo. Su núcleo está basado en el lenguaje de programación *Java* y al desarrollador se le proporciona tanto las *APIs*⁷ de *J2ME* (*suite* reducida de *Java* orientada a dispositivos móviles) como a las propietarias de *BlackBerry*. Su tienda de aplicaciones se llama *BlackBerry App World*. Sin embargo, debido a la orientación que tienen estos dispositivos, el peso de la misma no es muy importante, no encontrándose ni siquiera instalada por defecto. Ello hace que los desarrolladores no se encuentren muy atraídos por esta plataforma, limitando por tanto mucho su potencial, lo que no quiere decir que no sea excelente para el ámbito para el que fue desarrollada.

- **Symbian:** es el sistema operativo del mayor fabricante del mundo de móviles, *Nokia* [10], el cuál siempre ha sido un referente en el sector. Sin embargo, y pese que sigue siendo el fabricante que más móviles vende, no ha sabido adaptarse al nuevo mercado que se abría con los *Smartphone* y se ha quedado retrasada. El causante de este retraso ha sido precisamente su sistema operativo, demasiado anclado en el pasado, y que no ofrece novedades frente a la competencia si no todo lo contrario, es tosco y lento y no hace justicia al buen *hardware* que fabrica *Nokia*.

Como alternativa a *Symbian*, *Nokia* se ha aliado con *Intel* y, trabajando junto con la fundación *Linux* [11], están desarrollando un sistema operativo multiplataforma y libre llamado *Meego* [12]. Se tienen puestas muchas esperanzas en este nuevo sistema operativo y se espera que con él puedan hacer frente a sus competidores.

Una vez vistos varios de los sistemas operativos de dispositivos móviles más reseñables del mercado podemos entrar a valorar, con más criterio, por qué la elección ha sido *Android*. Las causas principales son las siguientes:

- Se trata de un *software* libre con un gran rendimiento, posibilidades y proyección.
- Se dispone de un buen *SDK*⁸ multiplataforma y gratuito que contiene las herramientas, documentación, librerías, emulador y la *API* necesaria para desarrollar aplicaciones.

⁷ API (Application Programming Interface): Conjunto de rutinas y métodos que ofrece una biblioteca para facilitar el desarrollo de una aplicación.

⁸ SDK (software development kit): Conjunto de herramientas de desarrollo para la realización de aplicaciones para un determinado sistema.

- El coste de la distribución de la aplicación es reducido y a cambio el usuario dispone de una manera sencilla de buscar y descargar la aplicación.
- El gran respaldo de los fabricantes de móviles en este sistema operativo hace que encontremos multitud de modelos, de diferentes características y precios, lo cual permite que la aplicación pueda llegar a un mayor número de personas.

Como se ve, ha tenido mucho peso en esta elección que se trate de un *software* libre que dispone de respaldo tanto de fabricantes como de usuarios por su buen funcionamiento. Pero también lo ha sido el hecho de que esta plataforma sea atractiva tanto para el desarrollador como para el usuario. Por un lado, al primero le proporciona buenas herramientas multiplataforma gratuitas. Por otro lado, el segundo dispone de una tienda donde encontrar e instalar la aplicación fácilmente. Todo esto sin olvidarnos que el objetivo de la aplicación es que pueda ayudar al mayor número de personas posibles. Esta es la plataforma que probablemente, debido a su extensión y diversidad, pueda llegar al mayor número de ellas.

A continuación se citan los componentes más reseñables que presenta este sistema operativo:

- Un núcleo *Linux* que proporciona una capa de abstracción de *hardware* básica así como servicios como gestión de procesos, memoria y sistema de archivos. En el núcleo se implementan controladores de *hardware* específicos, funciones como *Wi-Fi* y *Bluetooth*. La pila de *Android* tiene un diseño flexible, con diferentes componentes opcionales que dependen de la disponibilidad de *hardware* concreto en cada dispositivo, como por ejemplo pantallas táctiles, cámaras, receptores *GPS* o acelerómetros.
- Entre las principales bibliotecas de código destacan las siguientes:
 - Tecnología de navegador de *WebKit*, que se basa en el motor de renderizado *KHTML* del navegador web del proyecto *KDE*.
 - Compatibilidad con bases de datos a través de *SQLite*, la cual es una base de datos *SQL* simplificada y sencilla de manejar.
 - Compatibilidad gráfica avanzada, con 2D, 3D, animación *SGL* y *OpenGL ES*.
 - Compatibilidad con audio y video a través de *OpenCore* de *Packet Video*.
 - Funciones *SSL* del proyecto *Apache*.
- Diferentes administradores de servicios para:
 - Actividades y vistas.

- Telefonía.
- Ventanas.
- Recursos.
- Servicios basados en ubicación.
- El entorno de ejecución de *Android* proporciona lo siguiente:
 - Paquetes *Java* para obtener un entorno de programación *Java* prácticamente completo. No es un entorno *J2ME*.
 - La máquina virtual *Dalvik* utiliza servicios del núcleo basado en *Linux* para proporcionar un entorno de alojamiento para las aplicaciones *Android*.

Las aplicaciones básicas y las de terceros se ejecutan en la máquina virtual *Dalvik*, sobre los componentes anteriores y que, para finalizar, podemos ver en la figura 1:

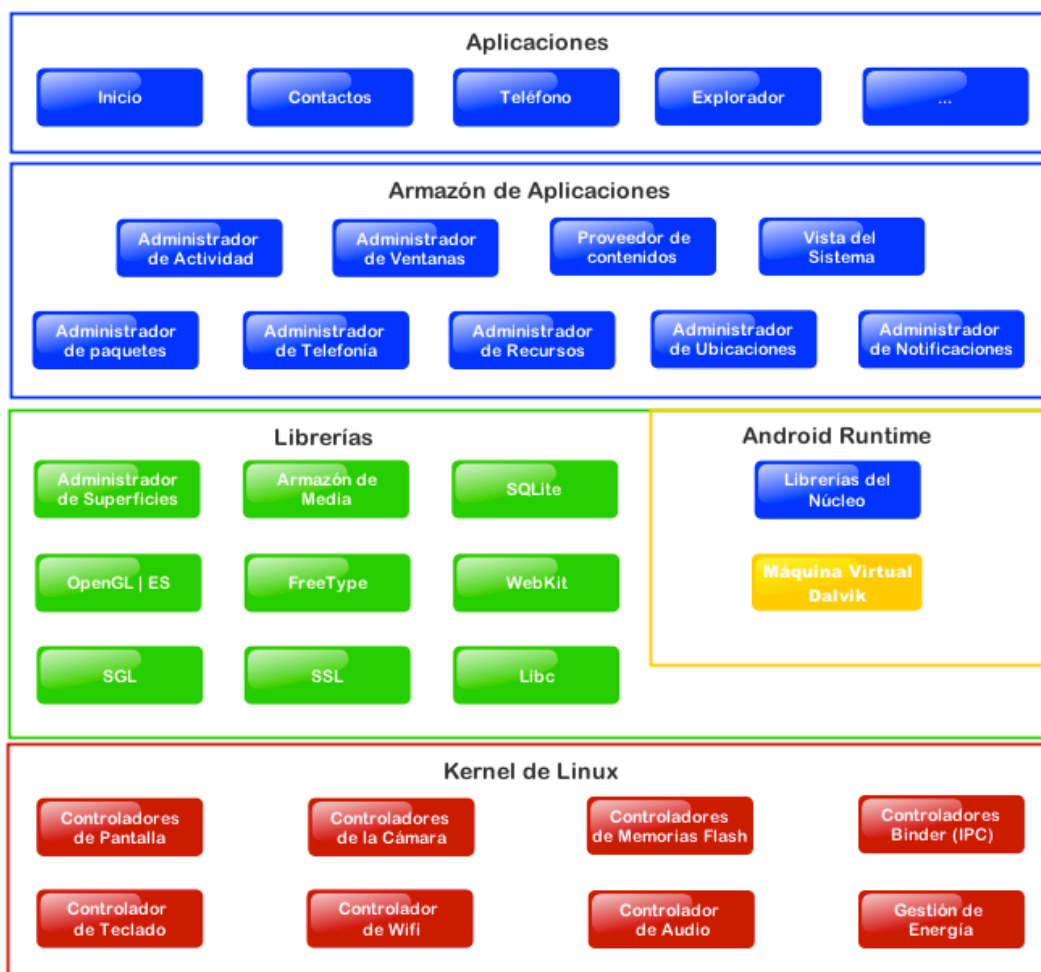


Figura 1: Componentes de la arquitectura *Android*

D. Metodología empleada

Una aplicación *Android* está siempre formada por al menos uno de los siguientes 4 componentes:

- **Activity:** lo podemos relacionar con una pantalla visible. Cada *Activity* muestra una interfaz de usuario capaz de responder a eventos iniciados por el sistema o por el usuario.
- **Service:** es usado cuando el ciclo de vida de la aplicación es prolongado. Son similares a los “demonios” de *UNIX*⁹, es decir, código que se ejecuta en segundo plano esperando a ser llamado para llevar a cabo una operación.
- **BroadcastReceiver:** es usado cuando una aplicación desea recibir y responder a un evento global.
- **ContentProvider:** es usado cuando una aplicación debe gestionar datos y mostrarlos a otras aplicaciones ejecutadas en el entorno de *Android*. Básicamente actúa como servidor de base de datos pudiendo usar cualquier mecanismo de almacenamiento de datos disponible en la plataforma *Android*, como archivos o bases de datos *SQL*.

Por otro lado tenemos los siguientes componentes que son usados habitualmente en el desarrollo de la aplicación y que quizás sean el aspecto más relevante que tiene la programación en *Android*:

- **Intent:** básicamente es una declaración de necesidades, describe lo que queremos hacer. Está formado por fragmentos de información que describen esta acción o servicio deseados. De esta manera los podemos usar tanto para buscar el registro de un contacto o abrir un sitio *Web* como para cambiar entre *Activities* o llamar a un *Service*.
- **IntentFilter:** es una declaración de capacidad e interés por ofrecer asistencia a los que la necesitan. Define la relación entre *Intent* y la aplicación. De esta manera, al enviar un *Intent*, el sistema evalúa los elementos *Activity* y *Service* disponibles, y los elementos *BroadcastReceiver* registrados y dirige el *Intent* al receptor más indicado.

Una vez analizada la estructura que va a presentar una aplicación *Android* podemos encontrar que encaja bastante bien con el patrón de diseño Modelo Vista Controlador (*MVC*). Este patrón está ideado para obtener una aplicación que sea fácil de mantener y reutilizable debido

⁹ UNIX: Sistema operativo portable, multitarea y multiusuario.

a la separación de la misma en tres módulos, que son los que le dan el nombre. Estos módulos son los siguientes:

- **Modelo:** Accede a la capa de almacenamiento de datos y define las reglas de negocio, es decir, la funcionalidad del sistema. Por tanto se encarga de encapsular los datos y las funcionalidades. Podemos relacionar este módulo con el componente *ContentProvider* de *Android*.
- **Vista:** Muestra la información al usuario. Podemos relacionar este módulo con el componente *Activity* de *Android* ya que éste nos proporciona, entre otras cosas, la interfaz de usuario.
- **Controlador:** Recibe los eventos de entrada y gestiona dichos eventos pudiendo cambiar la vista o realizando peticiones al modelo. Podemos relacionar este módulo nuevamente con el componente *Activity* de *Android*, ya que son estos componentes son capaces de cambiar de vista a través de los *Intent*, o de realizar peticiones al modelo.

Podemos observar que el componente *Activity* está presente en dos módulos, lo cuál no es coherente con la filosofía de este patrón, que precisamente persigue separar las partes de una aplicación en estos tres módulos. Sin embargo tiene una explicación que, pese a esto, hayamos indicado que la aplicación se adapta al patrón *MVC*. La respuesta es que efectivamente un *Activity* presenta la interfaz de usuario pero es un fichero *XML*, adjunto al mismo, quien se encarga de definir cómo se muestran los elementos e información de este *Activity*. Por lo tanto ya tenemos la separación que queríamos, los ficheros *XML* adjuntos a los *Activities* se corresponde con el módulo Vista, y los *Activities* como tal, se corresponden con el módulo Controlador. A continuación se muestra un gráfico representando este patrón *MVC* (ver figura 2).

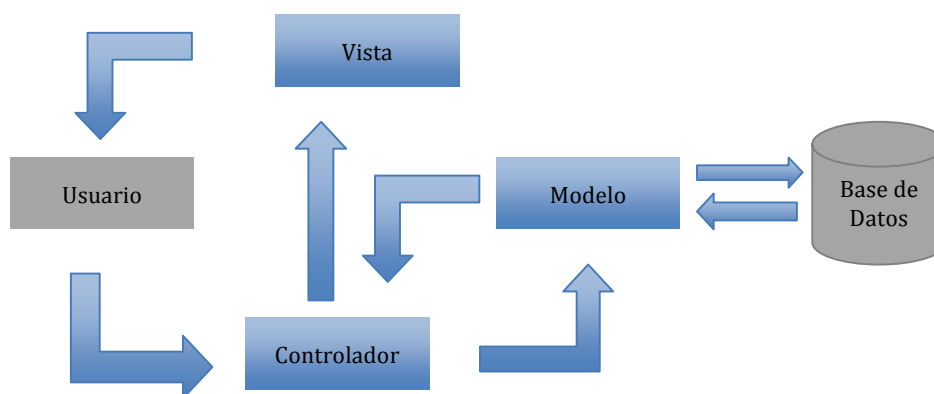


Figura 2: Diagrama de patrón MVC



3. DESCRIPCIÓN INFORMÁTICA

Una vez se ha expuesto en la sección anterior, el problema sobre el cuál se quiere proporcionar una solución en forma de aplicación para dispositivos móviles, de analizar los requisitos que debe cumplir la misma tanto a nivel *software* como a nivel *hardware*, y de haber escogido la plataforma sobre la cuál implementar la aplicación, llega el momento de centrarse en el diseño e implementación de la aplicación en sí misma. Para ello se realizarán una serie de pasos que permitan ir acercando la propia idea que se tiene de la aplicación a algo más tangible y definido sobre el cuál ya poder proceder a su implementación en un lenguaje de programación. Los pasos a realizar son los siguientes:

- Identificación y explicación de los diferentes casos de uso y de los diagramas de actividad.
- Plasmar los requisitos de la aplicación en forma de bocetos de las diferentes pantallas.
- Diseño de la base de datos sobre la cuál va a girar la principal funcionalidad que va a poseer la aplicación (almacenamiento de palabras, frases y consultas sobre las mismas).
- Implementación de la aplicación.

A. Casos de uso

En este punto se va a tratar de capturar todas las situaciones o escenarios que se le van a plantear al usuario así como la interacción del mismo con el sistema.

a. Diagrama casos de uso

A continuación se muestra el diagrama indicando los casos de uso para un usuario de la aplicación (ver figura 3). Tal y como se puede observar se dispone de un único rol de usuario, el cuál tiene acceso a todas y cada una de las funcionalidades de la aplicación.

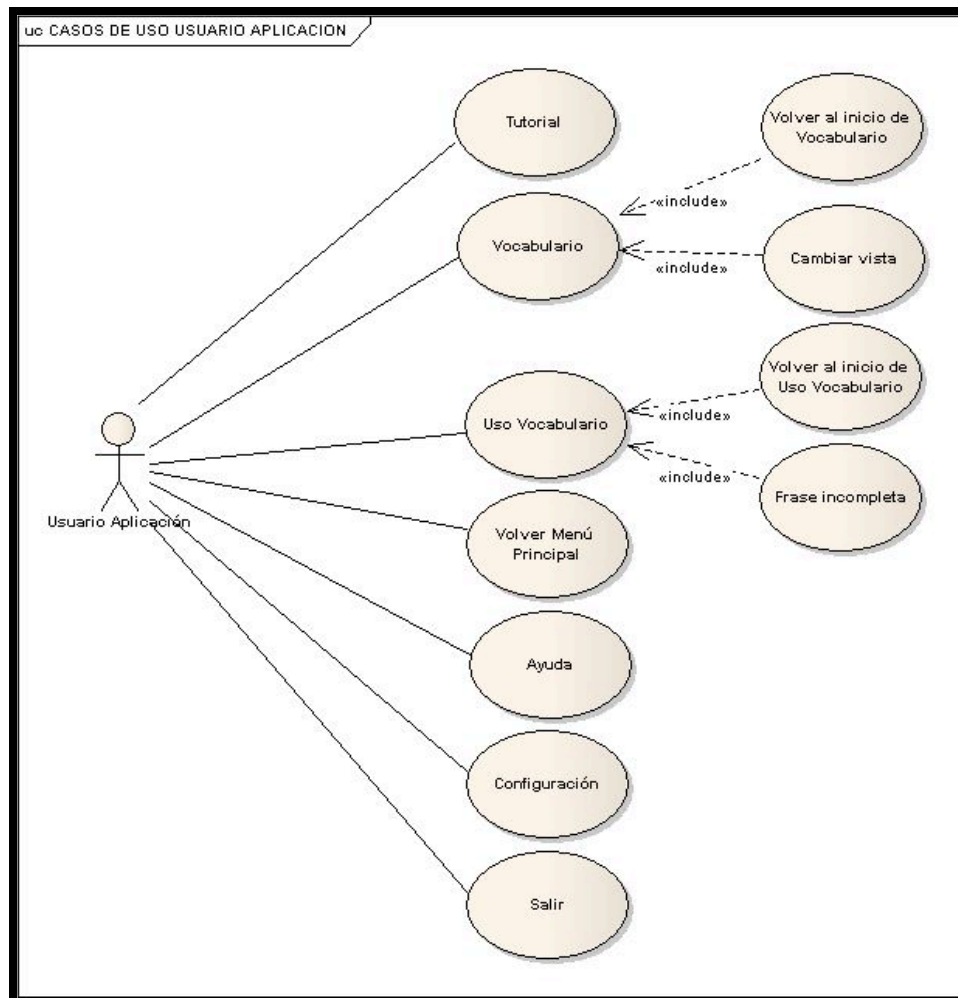


Figura 3: Casos de uso

b. Descripción casos de uso

A continuación se detallan cada uno de los casos de uso indicados en el apartado anterior.

Nombre: Tutorial.

Descripción: Muestra al usuario una descripción indicando para qué sirve la aplicación y cómo se maneja.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse en el menú principal de la aplicación “Dispositivo de Ayuda a la Comunicación”.

Flujo normal:

1. El actor pulsa sobre el botón en forma de androide.

Postcondiciones: El sistema muestra en pantalla la ayuda general de la aplicación.

Nombre: Vocabulario.

Descripción: Permite visualizar las palabras disponibles en el dispositivo móvil para obtener el símbolo pictográfico y/o traducción en otros idiomas de la palabra seleccionada.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse en el menú principal de la aplicación “Dispositivo de Ayuda a la Comunicación”.

Flujo normal:

1. El actor pulsa sobre el botón “Vocabulario”.
2. El sistema muestra todas las palabras disponibles. Para facilitar la visualización de las palabras, el usuario podrá seleccionar dos vistas: “Vista Clásica” (la cuál se encuentra activada por defecto) y “Vista Grupos”. Estas opciones se encuentran detalladas en el caso de uso “Cambiar Vista”.

Flujo alternativo:

- El actor selecciona la opción “Vista Clásica” o “Vista Grupos”, cuyo comportamiento será descrito en el caso de uso “Cambiar Vista”.
- El actor selecciona la opción “Volver Menú Principal”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.
- El actor solicita “Ayuda”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.
- El actor selecciona la opción “Volver al inicio de Vocabulario”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.

Postcondiciones: El sistema muestra la palabra escogida en los idiomas seleccionados a través del caso de uso “Configuración”.

Nombre: Cambiar vista.

Descripción: Permite cambiar el modo de visualizar las palabras dentro de “Vocabulario”.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse dentro de la opción “Vocabulario” de la aplicación. Si no se ha realizado un cambio de vista previo la vista que se encuentra activa por defecto es “Vista Clásica”.

Flujo normal:

1. El actor selecciona la opción “Vista Clásica” o “Vista Grupos”.

Postcondiciones: Si la opción seleccionada no era la que se encontraba activa, se cambia de vista modificando el modo de visualizar las palabras de “Vocabulario”. Si la vista activa es “Vista Clásica” se mostrarán las palabras ordenadas alfabéticamente. Por el contrario si la vista activa es “Vista Grupos” las palabras se encontrarán agrupadas por una temática común.

Nombre: Volver al inicio de Vocabulario.

Descripción: Permite regresar a la pantalla inicial de la opción “Vocabulario”.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse dentro de la opción “Vocabulario” de la aplicación. Dentro de esta opción, “Volver al inicio de Vocabulario” estará disponible si el actor ha seleccionado previamente la “Vista Grupos” y seleccionado uno de ellos, o bien se encuentra en la pantalla donde se muestra la palabra en los diferentes idiomas seleccionados.

Flujo normal:

1. El actor elige la opción “Volver Vocabulario”.

Postcondiciones: El sistema regresa a la pantalla inicial de “Vocabulario”, esto implica que, pese a que se tuviera activa la “Vista Grupos”, al regresar a la pantalla inicial la vista activa será “Vista Clásica”.

Nombre: Uso Vocabulario.

Descripción: Permite seleccionar una de las frases disponibles en el dispositivo móvil para poder obtener su traducción en otros idiomas y/o representarla en forma de secuencia de símbolos pictográficos.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse en el menú principal de la aplicación “Dispositivo de Ayuda a la Comunicación”.

Flujo normal:

1. El actor selecciona la opción “Uso de Vocabulario”.
2. El sistema muestra todas las situaciones posibles. Una situación engloba a una serie de frases que pueden resultar útiles para la misma.
3. El actor selecciona una situación.
4. El sistema muestra las frases disponibles para esa situación.
5. Las frases pueden ser completas o incompletas. Las incompletas se distinguen porque finalizan con el carácter “:”.
 - El actor selecciona una frase completa.
 - El actor selecciona una frase incompleta. Esto implica que el actor debe realizar una acción extra, detallada en el caso de uso “Frase Incompleta”.

Flujo alternativo:

- El actor selecciona la opción “Volver Menú Principal”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.
- El actor solicita “Ayuda”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.
- El actor selecciona la opción “Volver al inicio de Uso Vocabulario”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.

Postcondiciones: El sistema muestra la frase escogida en los idiomas seleccionados a través del caso de uso “Configuración”.

Nombre: Frase incompleta.

Descripción: Las frases incompletas son más versátiles que las completas puesto que permiten ser personalizadas con la palabra que mejor convenga.

Actores: Usuario (rol único).

Precondiciones: El actor debe haber seleccionado una frase incompleta (acabada en “:”) dentro de una “Situación” de la opción “Uso de Vocabulario”.

Flujo normal:

1. El sistema muestra la frase incompleta seleccionada junto con una serie de palabras que son compatibles con la misma.
2. El actor selecciona una de las palabras para completar la frase seleccionada.

Flujo alternativo:

- El actor selecciona la opción “Volver Menú Principal”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.
- El actor solicita “Ayuda”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.
- El actor selecciona la opción “Volver al inicio de Uso Vocabulario”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.

Postcondiciones: La frase incompleta seleccionada se convierte en una frase completa.

Nombre: Volver al inicio de Uso Vocabulario.

Descripción: Permite regresar a la pantalla inicial de la opción “Uso de Vocabulario”.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse dentro de la opción “Uso de Vocabulario” de la aplicación. Dentro de esta opción, “Volver al inicio de Uso Vocabulario” estará disponible si el actor ha seleccionado previamente una frase, ya sea completa o incompleta.

Flujo normal: El actor pulsa sobre el botón “Volver Uso Vocabulario”.

Postcondiciones: El sistema regresa a la pantalla inicial de “Uso de Vocabulario”.

Nombre: Volver Menú Principal.

Descripción: Permite regresar a la pantalla inicial de aplicación.

Actores: Usuario (rol único).

Precondiciones: Debe encontrarse en una pantalla distinta a la del menú principal.

Flujo normal: El actor selecciona la opción “Volver Menú Principal”.

Postcondiciones: El sistema regresa a la pantalla inicial de la aplicación.

Nombre: Ayuda.

Descripción: Permite mostrar al usuario una descripción indicando qué es lo que está apareciendo en pantalla, qué opciones tiene para interactuar con la misma y qué resultados va a obtener.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse en cualquier pantalla de la aplicación.

Flujo normal:

1. El actor selecciona la opción de “Ayuda”.

Postcondiciones: El sistema muestra la ayuda solicitada.

Nombre: Configuración.

Descripción: Permite indicar los idiomas con los que se quiere trabajar.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse en el menú principal de la aplicación.

Flujo normal:

1. El actor selecciona la opción “Configuración”.
2. El sistema muestra la pantalla desde la cual activar o desactivar los idiomas.
3. El actor selecciona los idiomas deseados.

Flujo alternativo:

- El actor selecciona la opción “Volver Menú Principal”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.
- El actor solicita “Ayuda”, cuyo comportamiento será descrito en el caso de uso del mismo nombre.

Postcondiciones: El sistema almacena qué idiomas se encuentran activos y por tanto deben ser usados cuando se solicite traducir una palabra o frase.

Nombre: Salir.

Descripción: Permite salir de la aplicación y regresar a la pantalla donde seleccionar las aplicaciones instaladas en el dispositivo móvil.

Actores: Usuario (rol único).

Precondiciones: El actor debe encontrarse en el menú principal de la aplicación.

Flujo normal:

1. El actor selecciona la opción “Salir”.
2. El sistema muestra un mensaje preguntando si se quiere salir y los botones “Sí” y “No”.
3. El actor selecciona una de las opciones.

Postcondiciones: El sistema regresa a la pantalla donde se encuentran las aplicaciones instaladas del dispositivo móvil si se ha pulsado sobre el botón “Sí”, por el contrario si se ha pulsado sobre el botón “No” el sistema regresa al menú principal de la aplicación.

B. Diagrama de actividad

Una vez analizados los escenarios y determinado cómo el usuario va a interactuar con el sistema se va a analizar el comportamiento dinámico del mismo. De esta manera se podrá observar de una forma rápida y sencilla el flujo de trabajo que va a presentar la aplicación. Esto se consigue describiendo la secuencia de actividades que el usuario deberá realizar para obtener los diferentes resultados posibles que va a proporcionar la aplicación. La manera de representar esto gráficamente es mediante un diagrama de actividad.

Al igual que en el diagrama de casos de uso, en éste vamos a disponer de un único rol de usuario, por tanto la secuencia de actividades comenzará en un punto, que representará el inicio de flujo de ejecución, y según el usuario tome un camino u otro se llegará a los diferentes estados finales de los que consta la aplicación. Se ha determinado como estado final de actividad aquellos que representan el final de cada una de las principales acciones que puede realizar el usuario y que se encuentran en el menú principal de la aplicación, así como salir de la aplicación que también se ha considerado como tal. Por tanto los estados finales

serán los siguientes: i) ejecutar el tutorial, ii) traducir la palabra, iii) traducir la frase y iv) salir de la aplicación. Por otro lado se ha considerado como final de flujo los puntos donde se selecciona la ayuda de la que se dispone en cualquier pantalla de la aplicación y la selección, en configuración, de los idiomas en los que queremos que aparezcan traducidas las palabras y frases.

A continuación se muestra el diagrama de actividad de la aplicación. Para una mayor legibilidad y facilidad de comprensión se ha dividido en tres partes. El primero de ellos (véase la figura 4) representa las actividades principales que puede realizar el usuario. Mientras que los dos siguientes muestran en profundidad el comportamiento de las dos grandes actividades de la aplicación: “Vocabulario” (figura 5) y “Uso de Vocabulario” (figura 6).

La figura 4 muestra que las acciones principales que puede realizar el usuario al iniciar la aplicación son la visualización del tutorial de la aplicación, ejecutar el módulo de uso de vocabulario o ejecutar el módulo de vocabulario. Por otro lado si el usuario pulsa el botón del menú del móvil se despliega un submenú con otras tres opciones: salir (a través de la cuál se sale de la aplicación), ayuda (se visualiza una explicación sobre la pantalla actual) y configuración (permite seleccionar los idiomas en los que se traducirán las palabras o frases).

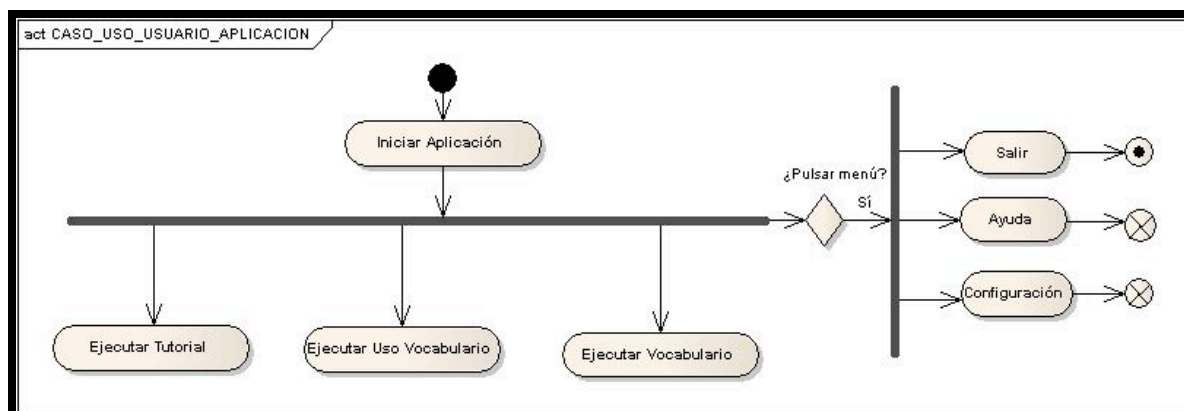


Figura 4: Actividades principales de la aplicación

La figura 5 muestra la secuencia de actividades que el usuario puede realizar una vez que ha ejecutado el módulo “Vocabulario”. Como se puede observar, nada más ejecutar el módulo, el usuario puede cambiar el modo en el que se muestran las palabras. Inicialmente se encuentran ordenadas alfabéticamente, si se opta por cambiar de visualización pasarían a mostrarse agrupadas por temática. Tanto si el usuario opta por cambiar de vista como si no, si pulsa el

botón menú del móvil emerge un submenú con las opciones volver al menú principal (regresa a la pantalla inicial de la aplicación) o ayuda (visualiza una explicación sobre la pantalla actual). Una vez el usuario se encuentre en la vista deseada, puede seleccionar una palabra (si se encuentra en la vista clásica) o escoger previamente un grupo antes de seleccionar ésta (si se encuentra en vista de grupo). Al pulsar sobre la palabra aparecerá traducida en los idiomas seleccionados. En ese punto, pulsando sobre el botón menú del móvil, podrá seleccionar volver al menú principal (regresa a la pantalla inicial de la aplicación), volver al vocabulario (regresa a la pantalla inicial del módulo vocabulario) o ayuda (visualiza una explicación sobre la pantalla actual).

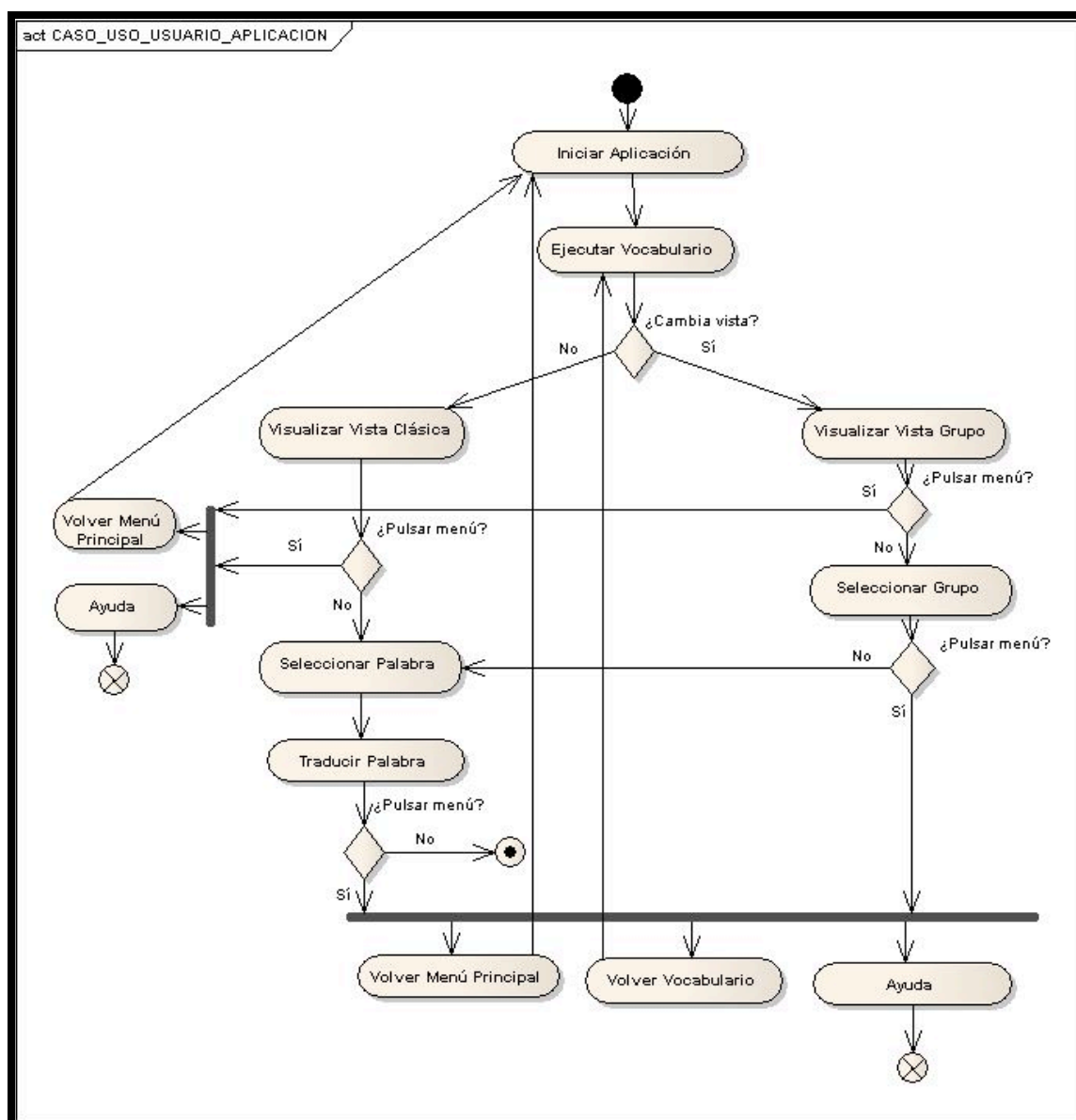


Figura 5: Actividades dentro del caso de uso "Vocabulario"

Por último, la figura 6 muestra la secuencia de actividades que el usuario puede realizar si ha seleccionado el módulo “Uso de Vocabulario”. Al ejecutar este módulo el usuario puede escoger una situación para visualizar sus respectivas frases. Si pulsa el botón menú del móvil, ya sea antes o después de seleccionar la situación, se muestran las opciones para volver al menú principal (regresa a la pantalla inicial de la aplicación) y ayuda (visualiza una explicación sobre la pantalla actual). Si ha seleccionado una situación podrá seleccionar una de sus frases para obtener su traducción. Si la frase seleccionada es incompleta debe completarla con una de las palabras compatibles que se le muestran. En cualquiera de estos puntos si pulsa sobre el botón menú del móvil podrá seleccionar las opciones volver al menú principal (regresa a la pantalla inicial de la aplicación), volver a la pantalla inicial del uso de vocabulario y ayuda (visualiza una explicación sobre la pantalla actual).

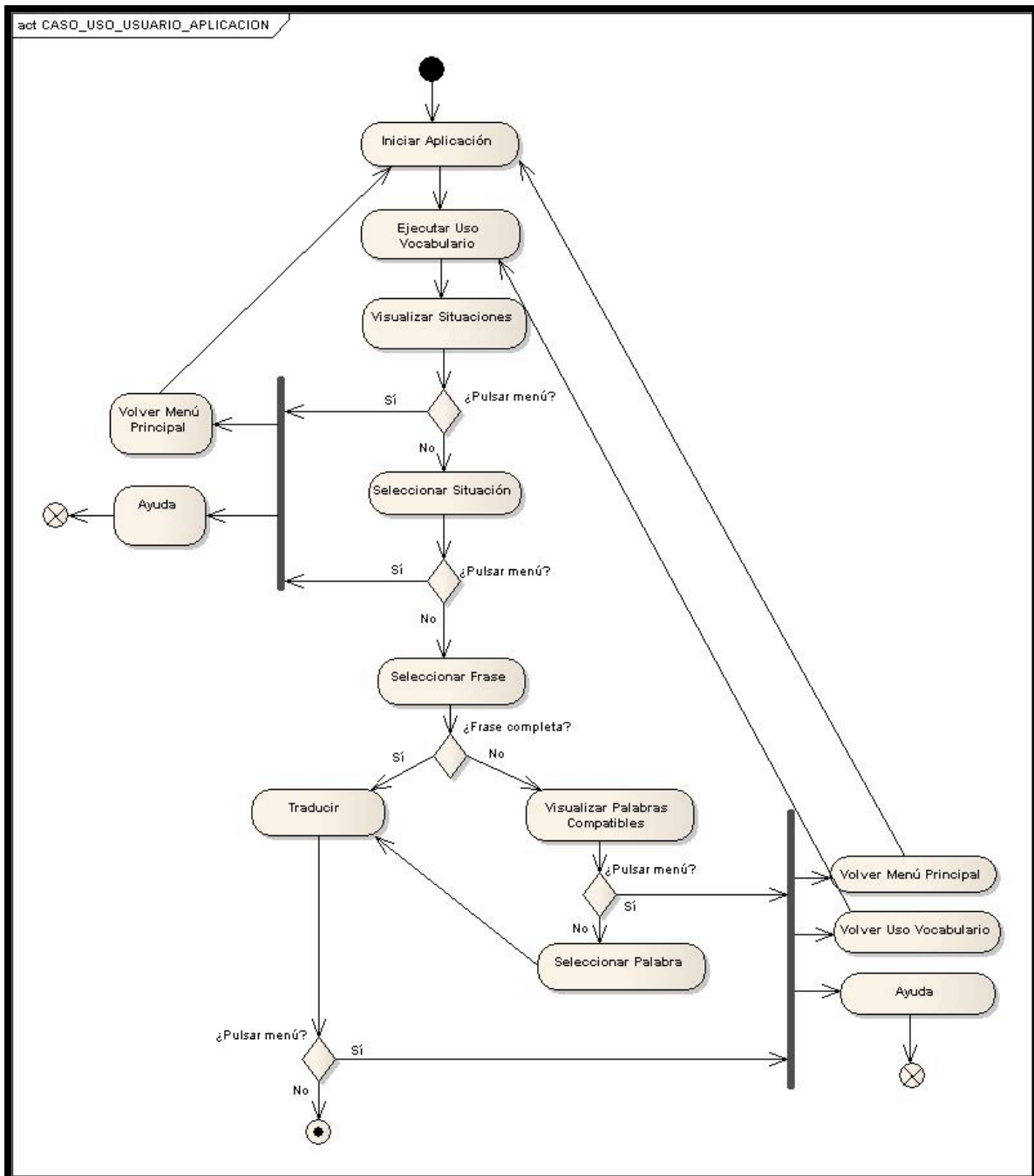


Figura 6: Actividades dentro del caso de uso “Uso de Vocabulario”

C. Interfaz de usuario de la aplicación

Tras la realización de los anteriores diagramas, que nos han permitido adquirir una idea mucho más clara y concisa de cómo va a estar estructurada y cómo va a funcionar la aplicación, es momento de diseñar su interfaz de usuario. En este punto se pretende mostrar

unos bocetos que nos van a servir para aproximarnos al diseño final que presentará la aplicación. Por tanto, se darán las primeras pinceladas de cómo puede quedar la aplicación.

a. Menú aplicación “Dispositivo de Ayuda a la Comunicación”

El menú principal de la aplicación permitiría acceder a los dos grandes bloques de la aplicación: “Vocabulario” y “Uso de Vocabulario” (ver figura 7).

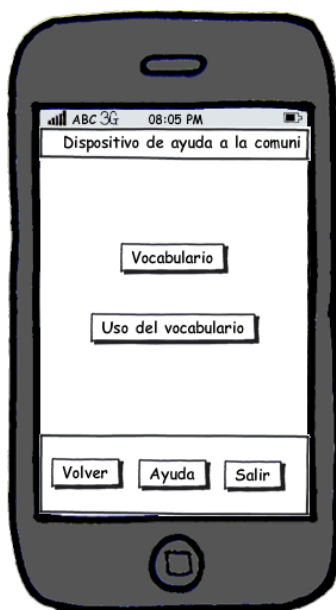


Figura 7: Boceto del menú principal de la aplicación

b. Vocabulario

Si se selecciona la opción “Vocabulario” nos encontraríamos con una serie de palabras ordenadas alfabéticamente (“Vista Clásica”) o agrupadas por temáticas (“Vista Grupos”). En la siguiente secuencia de imágenes se pueden apreciar las diferencias entre estas dos vistas. En la parte izquierda de la figura 8 se muestra la pantalla principal de la “Vista Clásica”, donde aparecería el listado de palabras que contiene la aplicación ordenadas por orden alfabético. La parte derecha de la figura 8 muestra el boceto de la pantalla que aparecería si se selecciona una de las palabras del listado de la parte izquierda (traducción a diferentes idiomas.)



Figura 8: Boceto de la vista clásica del módulo Vocabulario

La parte izquierda de la figura 9 mostraría la vista por grupos. En el ejemplo mostrado se puede apreciar tres grupos: comida, ropa y transporte. Cuando el usuario seleccionase un determinado grupo, se le mostrarían las palabras que contiene esta temática. En la parte de la derecha de la figura 9 se puede observar un ejemplo de la pantalla para el caso en que el usuario hubiera seleccionado el grupo comida.

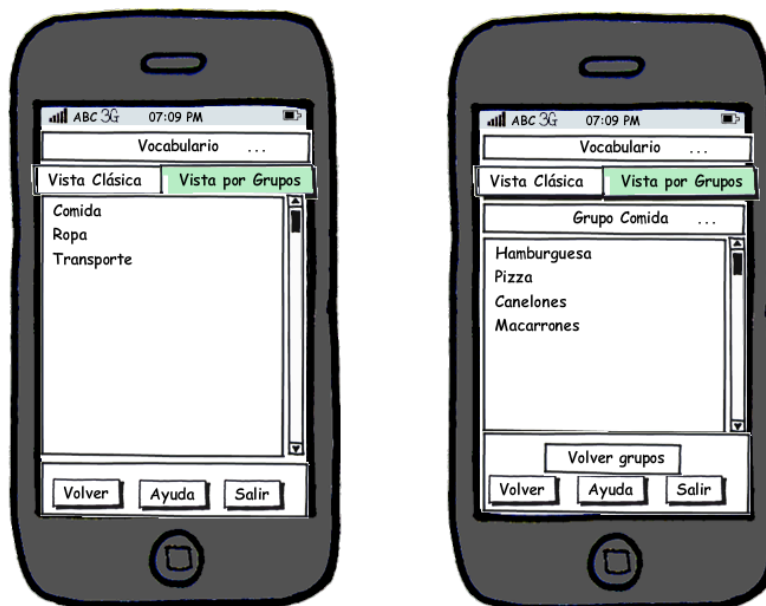


Figura 9: Boceto de la vista de grupos del módulo Vocabulario

c. Uso de Vocabulario

Si se selecciona la opción “Uso de Vocabulario” nos encontraríamos con una serie de situaciones que contienen frases útiles para las mismas. La figura 10 muestra la pantalla principal de esta opción con las posibles situaciones (parte izquierda) y las posibles frases que aparecerían si se seleccionase una de las situaciones (parte derecha).

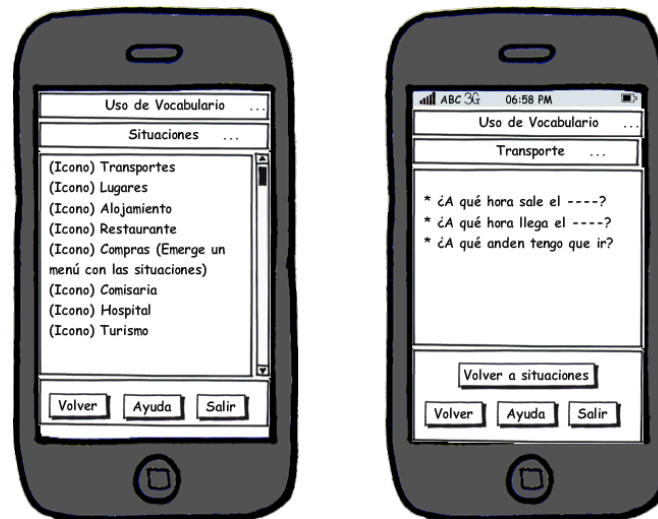


Figura 10: Boceto de la pantalla de selección de situación del módulo Uso de Vocabulario

La figura 11 presenta la pantalla que aparecería si el usuario selecciona una frase incompleta (parte izquierda). En ese caso, se muestran una serie de palabras que son compatibles con la frase seleccionada (en el ejemplo, tren, coche, metro y tranvía). La parte derecha muestra el boceto de la pantalla que aparecería si se selecciona una de las palabras compatibles que permiten completar la frase (traducción a diferentes idiomas.)

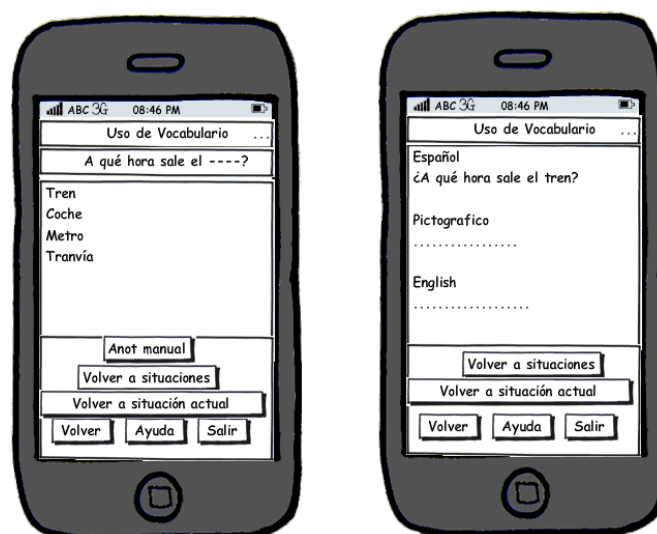


Figura 11: Boceto de la pantalla de selección de palabra para completar una frase incompleta

D. Diseño de la base de datos

Toda la funcionalidad de la aplicación pasa por la base de datos. La base de datos se encarga de mostrar al usuario una serie de palabras o frases almacenadas y, según las elecciones que éste tome, realizar consultas a la misma para mostrarle la información solicitada.

a. SQLite

La plataforma *Android* incorpora la base de datos relacional *SQLite*. Ésta es una base de datos simplificada y sencilla de manejar que cumple los estándares SQL-92. Por tanto, no va a disponer de toda la funcionalidad de una base de datos SQL cliente/servidor comercial, pero sí que va a proporcionar todo lo necesario para lograr un almacenamiento local de datos. Entre sus características podemos destacar las siguientes:

- Se integra con la aplicación por lo que no requiere una instalación previa de la misma. Esto también conlleva que se pueda acceder a ella directamente a través de subrutinas, lo que se traduce en un mayor rendimiento.
- Encapsula toda la información necesaria (tablas, índices, definiciones, datos) en un único archivo que queda guardado en el dispositivo.
- El código fuente es de dominio público.

b. Sqlite3

Desde el punto de vista del desarrollador, las consultas y actualizaciones de la base de datos se harán a través de *sqlite3*. Ésta es una herramienta que, desde línea de comandos, va a permitir gestionar la base de datos. De esta manera, desde un terminal se podrán crear tablas, añadir elementos o simplemente consultar la misma. A continuación se detalla cómo podríamos observar el contenido de la base de datos usando esta herramienta:

Accedemos al directorio *Tools* del SDK de *Android*

```
cd /Application/Android/Tools
```

Desde allí obtenemos la lista de dispositivos conectados, ya sean reales o virtuales.

```
adb devices
```

Accedemos al dispositivo que estemos usando (en el ejemplo el emulador-5554)

```
adb -s emulator emulator-5554 shell
```

Una vez dentro del dispositivo, accedemos al directorio donde se encuentran las bases de datos de la aplicación

```
cd data/data/mi_aplicacion/databases
```

Ejecutamos la herramienta sqlite3 sobre la base de datos que nos interese

```
sqlite3 nombre_bd
```

Ejecutamos el comando que queramos realizar sobre ella, por ejemplo observar su contenido

```
.dump
```

c. Diseño

La aplicación es capaz de mostrar palabras y frases en diferentes idiomas y como secuencia de símbolos pictográficos. Esta información es análoga en ambos casos, puesto que palabras y frases comparten los mismos idiomas, pero también existen diferencias. Por tanto, es necesario que la información relativa a frases y palabras sea almacenada en distintas tablas dentro de la base de datos. La diferencia más reseñable que encontramos entre ellas es que la frase puede ser completa o incompleta, siendo en este último caso necesario poder relacionarla con una serie de palabras que sean compatibles con ella. Puesto que no todas las frases son incompletas se ha optado por almacenar esta relación frase-palabra en una tercera tabla. Por tanto, la base de datos de la aplicación va a constar de las siguientes tablas:

- **Tabla palabras:** contendrá toda la información relativa a las palabras, incluyendo la temática a la que pertenecen, la traducción a los diferentes idiomas disponibles y un símbolo pictográfico.
- **Tabla frases:** contendrá toda la información relativa a las frases, incluyendo la situación a la que pertenecen, si es una frase completa o incompleta, su traducción a los diferentes idiomas disponibles y su secuencia de símbolos pictográficos.
- **Tabla frases-palabras compatibles:** contendrá la correspondencia entre una frase incompleta y una palabra compatible.

E. Implementación

Para proceder a la implementación, previamente se han de tener instaladas todas las herramientas necesarias para este proceso. En el anexo A de este documento se indica cuáles son los pasos a seguir para conseguir instalar y configurar todo lo necesario correctamente.

En un anterior apartado se comentó que el patrón de diseño que mejor encajaba en una aplicación *Android* era el Modelo Vista Controlador (*MVC*) y por tanto, es el que se va a seguir en esta aplicación. Sin embargo habría que matizar que no se va a disponer de un *ContentProvider* actuando como modelo, si no que simplemente se va a disponer de una base de datos con una funcionalidad determinada. El motivo de que no exista un *ContentProvider* es que no es necesario compartir los datos de la aplicación con otras aplicaciones. Por tanto dispondremos de una base de datos que proporciona los datos y la funcionalidad (modelo), una interfaz a través de los ficheros *XML* adjuntos a las *Activities* (vista) y las propias *Activities* que van a servir de puente entre el usuario y la aplicación permitiendo cambiar de vista y solicitar peticiones al modelo (controlador). El controlador se apoyará en los *Intent* para realizar su trabajo.

A continuación se detallará la estructura de un proyecto *Android* seguido de la explicación sobre los diferentes elementos característicos que va a contener la aplicación.

a. Estructura del proyecto

La creación de un proyecto *Android*, si se ha instalado el entorno de desarrollo tal y como se indica en el Anexo A, no dista mucho del que hay que realizar para crear un proyecto *Java*. Los campos a rellenar serán:

- **Project name:** nombre del proyecto.
- **Package name:** nombre del paquete donde va a residir el código fuente.
- **Create Activity:** nombre de la clase que será generada automáticamente y que servirá como punto de entrada a la aplicación. Se trata de una subclase de la clase *Activity*.
- **Application name:** nombre de la aplicación que se mostrará al usuario.
- **Build Target:** versión de la plataforma sobre la que se va a implementar la aplicación (en nuestro caso, versión 2.2 (*Froyo*) de *Android*, que corresponde a la API de nivel 8).

Una vez creado el proyecto se va a generar la estructura que podemos observar en la figura 12.

De ella, podemos destacar lo siguiente:

- **Versión de Android:** Contiene la biblioteca de clases correspondiente a la versión de *Android* para la que estemos implementando la aplicación, en nuestro caso la 2.2.
 - **src:** Contiene todo el código fuente que generemos para la aplicación.
 - **gen:** Contiene los ficheros *Java* generados automáticamente por el compilador. En esta carpeta podemos encontrar el archivo *R.java* que contiene los identificadores de todos los recursos empleados en la proyecto y que se encuentran en la carpeta */res*.
 - **res:** Contiene todos los ficheros de recursos necesarios para el proyecto. Dentro de esta carpeta, los recursos se encuentran organizados en subcarpetas:
 - **res/drawable:** Contiene las imágenes de la aplicación. Según la resolución que contengan deberán ser situadas en su correspondiente carpeta: *drawable-ldpi* (*low density screen*, 36x36 px), *drawable-mdpi* (*medium density screen*, 48x48 px) y *drawable-hdpi* (*high density screen*, 72x72 px). Esto es útil porque no todos los dispositivos tienen la misma resolución de pantalla. Almacenando las imágenes en estos tres formatos nos aseguramos una total compatibilidad, sea cuál sea la pantalla del dispositivo donde vaya a ser ejecutada la aplicación. Una vez almacenadas las imágenes, la propia plataforma *Android* se encarga de comprobar las características de la pantalla del dispositivo y de emplear las imágenes adecuadas para la misma.
 - **res/layout:** Contiene los ficheros de definición de las diferentes vistas/pantallas en formato *XML*.
 - **res/values:** Contiene otros recursos de la aplicación como cadenas de texto, colores, estilos, etc. También son ficheros *XML*.
 - **AndroidManifest.xml:** Fichero *XML* con toda la información necesaria para poder ejecutar la aplicación: nombre de la aplicación, icono, paquete, punto de entrada de la aplicación, componentes de la aplicación (*Activities*, *Services*, *Broadcast receivers* y *Content providers*), permisos, etc. En nuestro caso, es importante indicarle tanto las clases que contiene así como el componente que implementa (en nuestro caso sólo utilizamos *Activities*). Al mismo tiempo es necesario indicar cuáles de esas *Activities* van a llamarse desde otros puntos de la aplicación a través de *Intents*. La manera de hacerlo es declarando un *Intent-filter* para cada una de estas *Activities*, lo que permitirá que puedan recoger el *Intent* correspondiente.
-

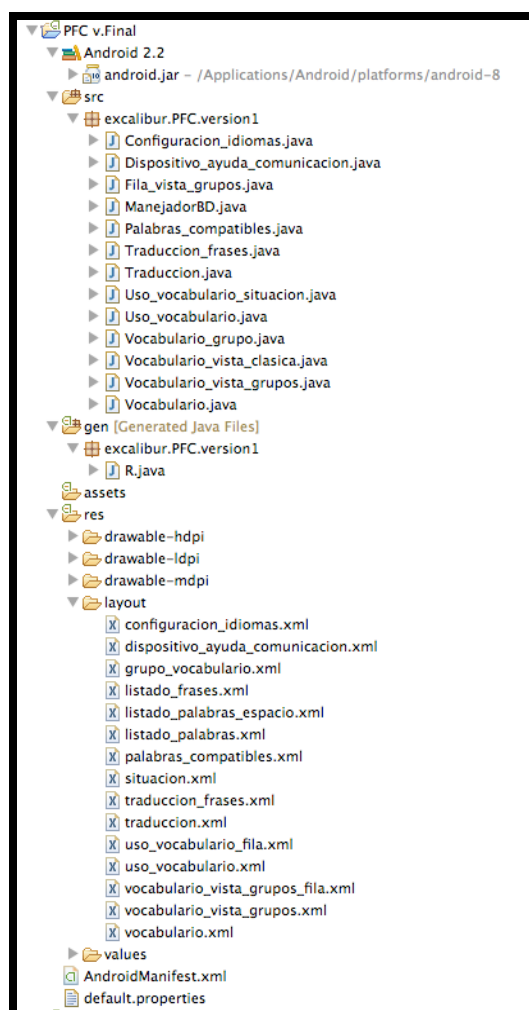


Figura 12: Estructura del proyecto implementado

b. Componentes de la aplicación

En este punto se van a van a detallar algunos de los elementos más característicos de *Android* que se han usado para la implementación de la aplicación.

i. Activity

El componente *Activity* es el más usado en aplicaciones *Android*, el motivo de ello es que podemos establecer una equivalencia entre un *Activity* y una pantalla visible por el usuario. Por tanto cualquier aplicación que tenga IU tendrá también, al menos, un *Activity* que se encargará de mostrarla y que responderá a eventos iniciados por el sistema o por el usuario. Para mostrar los diferentes elementos de los que estará compuesta la IU el componente *Activity* emplea las vistas (*View*). Más adelante se explicará más detalladamente estos

componentes de *Android*, analizando algunos de los que han sido empleados en la aplicación. También se explicará otro de los elementos que hemos hablado y que se trata de otra piedra angular en los *Activity*, los *Intent* que nos permitirán cambiar entre ellos. Pero antes vamos a analizar un *Activity* de la manera más básica que lo podemos encontrar que es implementando su método *onCreate()*. Este método se invoca al crear el *Activity* y es capaz de recuperar cualquier estado almacenado previamente gracias a su argumento. Esto es importante porque un *Activity* tiene un ciclo de vida complejo. Para añadir la funcionalidad que deseemos es necesario sobreescribirlo. En el fragmento de código que podemos observar en la figura 13 se está cargando una interfaz de usuario, la cuál está contenida en un fichero *XML*. Para obtenerla se emplea el fichero *R.java*, del que ya hablamos anteriormente, y que nos sirve para poder acceder a los recursos de la aplicación, en este caso uno del tipo *layout*.

```
public class Dispositivo_ayuda_comunicacion extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        //Establecemos la IU  
  
        setContentView(R.layout.dispositivo_ayuda_comunicacion);  
    }  
}
```

Figura 13: Activity - Carga de la IU

ii. Elementos gráficos y contenedores

Los elementos gráficos (*Views*) son esenciales para construir la interfaz de usuario (IU) puesto que representan elementos de la pantalla. *Android* proporciona una gran cantidad de vistas pero también se pueden combinar o personalizarlos para obtener el deseado. Pese a que podríamos definir todo el diseño y la configuración de estos elementos como objetos Java en cada *Activity*, lo normal es tener asociado un fichero *XML* al mismo. En este fichero *XML* se definirán los objetos *View* que serán establecidos posteriormente en el *Activity*. La manera que tenemos de controlar la distribución, posición y dimensiones de estos elementos es a través de los contenedores *Layout*. Un contenedor se encarga de organizar los elementos gráficos que serán mostradas en pantalla. Para entender cómo funcionan hay que pensar que la distribución de los elementos se va a realizar creando un árbol jerárquico. Existen valores muy importantes, como por ejemplo el ancho (*width*) o el alto (*height*), de los contenedores o de los *View*: *fill_parent* (indica que debe tomar el tamaño que le está dejando el padre) y

wrap_content (indica que debe tomar el tamaño mínimo en el que entren sus hijos). De esta manera se va repartiendo el espacio de la pantalla entre los diferentes elementos que lo forman. *Android* proporciona varios tipos de contenedores, cada uno con unas características propias, para ajustarse a todas las necesidades que se puedan tener a la hora de realizar la IU. Los contenedores son: **FrameLayout** (sus hijos se alinean en la esquina superior izquierda, por tanto es útil para contener un único elemento), **LinearLayout** (apila los hijos uno tras otro de forma horizontal o vertical), **TableLayout** (sus hijos se distribuyen en filas y columnas dentro de una tabla) y **RelativeLayout** (permite distribuir cada hijo de manera individual y tomando como referencia el propio padre u otros elementos incluidos en el *Layout*).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:padding="30dip"
    android:orientation="vertical">
    <RelativeLayout
        android:orientation="vertical"
        android:layout_height="fill_parent"
        android:layout_width="fill_parent">
        <Button
            android:id="@+id/android_ayuda"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@drawable/android_azul"
            android:layout_marginTop="20dip"
            android:layout_marginBottom="50dip"/>
        <TextView
            android:id="@+id/texto_android_ayuda"
            android:layout_toRightOf="@id/android_ayuda"
            android:layout_marginTop="10dip"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:text="@string/contenido_ayuda_android"
            android:textStyle="bold"/>
        <Button
            android:id="@+id/app_vocabulario"
            android:layout_below="@id/android_ayuda"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/label_vocabulario"
            android:textSize="15.5sp"
            android:layout_marginBottom="22.5dip" />
        <Button
            android:id="@+id/app_uso_vocabulario"
            android:layout_below="@id/app_vocabulario"
            android:layout_width="fill_parent"
```

Figura 14: Activity - Carga de la IU

La figura 14 presenta el contenido del fichero *XML* al que se hacía referencia en el código mostrado en la figura 13. La interfaz de usuario del *Activity* “Dispositivo de ayuda a la comunicación” estará formada por 4 elementos gráficos (1 *TextView* y 3 *Buttons*) organizados dentro de un *RelativeLayout*. Este contenedor está a su vez contenido en el contenedor padre *LinearLayout*. Pese a que podríamos prescindir de él en este caso, se ha decidido incluirlo para unificar el diseño con el resto de interfaces de la aplicación que sí lo emplean. También nos aporta flexibilidad en el caso de que necesitáramos añadir un nuevo elemento a la interfaz. El resultado final es el que podemos ver en la siguiente figura:

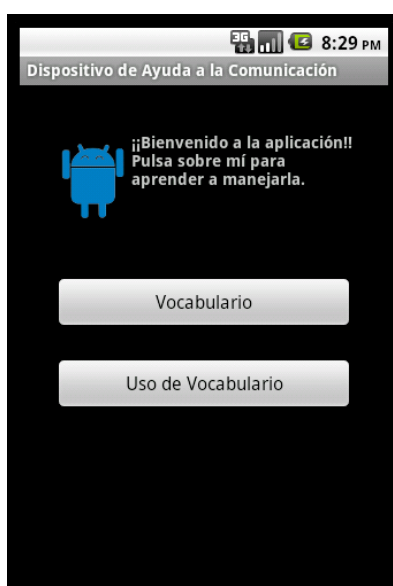


Figura 15: Interfaz de usuario del menú principal

Una vez se ha expuesto de manera general qué son y cómo funcionan los elementos *gráficos* de *Android* se va a analizar algunos de los empleados en la aplicación.

iii. Button

Representa un botón en pantalla, el cuál puede ser pulsado por el usuario para acceder a la funcionalidad que tenga asociada dentro del código correspondiente a su *Activity*. Para ello debemos establecer un vínculo entre el diseño del botón en sí (definido en el fichero *XML*) y la parte del código. Si observamos la figura 14, en la que se definía la IU de la pantalla inicial de la aplicación, podemos ver como además de los atributos que proporcionan un determinado aspecto a los *View*, hay uno que nos permite asignarles un identificador (*android:id*). El nombre que usemos en este atributo será el que empleemos para poder establecer el vínculo. Una vez establecido, podremos usar los métodos que dispone este elemento. Un método

especialmente útil en los botones es el método `setOnClickListener()`. Este método permite definir un oyente que se encargue de responder de una forma determinada cuando se pulse sobre él. En la figura 16 podemos ver cómo se ha establecido desde el código, la vinculación con la parte gráfica de los *Button* y cómo se le ha añadido funcionalidad a uno de ellos, en este caso iniciar un nuevo *Activity*.

```
//Asociamos los botones con su representación gráfica
boton_android = (Button)findViewById(R.id.android_ayuda);
boton_vocabulario = (Button)findViewById(R.id.app_vocabulario);
boton_uso_vocabulario = (Button)findViewById(R.id.app_uso_vocabulario);

//Añadimos funcionalidad al botón Vocabulario.
//Cuando se pulse sobre él se llamará a la actividad Vocabulario
boton_vocabulario.setOnClickListener(new OnClickListener(){
    public void onClick(View arg0) {
        intent_vocabulario = new Intent (Dispositivo_ayuda_comunicacion.this,
            Vocabulario.class);
        startActivity(intent_vocabulario);
    }
});
```

Figura 16: Interfaz de usuario - Botones

iv. *ListView*

Se encarga de mostrar por pantalla un listado de ítems y, en el caso de que no se puedan mostrar todos estos ítems en una sola pantalla, se permite realizar *scrolling* sobre los mismos. Por defecto, los elementos son cadenas de texto, pero podemos modificarlo para que se liste lo que nosotros deseemos. Por otro lado, los *ListView* se van a apoyar en otros elementos para poder funcionar, los *Adapter*. Éstos van a hacer de puente entre los datos que se quieren visualizar y su interfaz gráfica para que puedan ser mostrados a través del *ListView*. Existen diferentes tipos de ellos como pueden ser *ListAdapter*, *ArrayAdapter*, *SpinnerAdapter* y *SimpleCursorAdapter*, pero también podemos crear *Adapters* personalizados para que se ajusten a los ítems que queremos mostrar. En las siguientes figuras, se muestra la parte de la interfaz (figura 17) y del código (figura 18) correspondiente a la visualización de palabras del vocabulario usando la vista clásica. Podemos ver cómo obtenemos previamente las palabras de la base de datos, preparamos los datos con *SimpleCursorAdapter()*, apuntamos al *ListView* con *setListAdapter()* y desde el método *onCreate()* del *ListActivity* lo obtenemos con *getListView()*. Remarcar que en este caso no es necesario tener definido dentro del fichero *XML* un elemento *ListView* como tal ya que éste se generará automáticamente cuando

llamemos a `getListView()`, simplemente definiremos el formato de la pantalla y de las filas de la lista. También indicar que la actividad debe extender de `ListActivity`.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/palabra_con_espacio"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="5dip"
        android:textStyle="bold"
        android:textSize="28dip" />
</LinearLayout>
```

Figura 17: Interfaz de usuario - ListView

```
public class Vocabulario_vista_clasica extends ListActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //Obtenemos las palabras que queremos que se muestren dentro de la pestaña Vista
        Clásica
        obtenerPalabras();

        //Obtenemos el ListView local de nuestra actividad, que recoge los datos del
        adapter creado anteriormente, para visualizarlo
        getListView();
    }

    private void obtenerPalabras() {
        // Apuntamos a la información de la tabla palabras de la BD
        miCursorPalabras =
        Dispositivo_ayuda_comunicacion.miManejadorBaseDeDatos.apuntarTablaPalabras();

        startManagingCursor(miCursorPalabras);

        // Creamos un array con los campos de la tabla que queremos mostrar en la lista
        String[] from = new String[]{ManejadorBD.KEY_ESPANOL};
        // Creamos un array con los campos vinculados a los anteriores dentro del fichero xml
        correspondiente a su IU
        int[] to = new int[]{R.id.palabra_con_espacio};
        //Creamos un adapter que nos relacione todo lo anterior para poder visualizarlo
        adapterListaVocabulario = new SimpleCursorAdapter(this,
        R.layout.listado_palabras_espacio, miCursorPalabras, from, to);
        //Apuntamos al adapter creado para que pueda ser usado más adelante en esta actividad
        setListAdapter(adapterListaVocabulario);
    }
}
```

Figura 18: Interfaz de usuario - ListView

v. Tab Layout

Se encarga de proporcionar a la IU una distribución de la aplicación mediante pestañas. Para conseguir esto no basta únicamente con un *View* si no hay que usar dos elementos: *TabHost* y *TabWidget*. *TabHost* puede considerarse el contenedor de todas las pestañas que se quieran agregar a la IU, mientras que *TabWidget* se encarga de representar a la pestaña en sí junto con su contenido. El contenido será representado como un todo por lo que se usará un *FrameLayout* para contenerlo. Esto no significa que tengamos que limitarnos a un contenido “simple”, ya que estamos usando el *Layout* más sencillo, si no que el contenido puede ser el que queramos ya que incluso podemos invocar a otra *Activity* para que se encargue de gestionarlo. La diferencia está en que esta nueva *Activity* no va a disponer de todo el espacio de la pantalla si no únicamente del espacio reservado por el *FrameLayout*. A continuación podemos observar tanto la parte de la interfaz *XML* (figura 19) como del código usado para implementar el sistema de pestañas de nuestra aplicación (figura 20), las cuáles van a permitir cambiar entre “Vista Clásica” y “Vista Grupos” dentro de “Vocabulario”.

```
<?xml version="1.0" encoding="utf-8"?>
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:padding="5dip">

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content" />

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:padding="5dip" />

    </LinearLayout>
</TabHost>
```

Figura 19: Interfaz de usuario - TabLayout

En la figura 20, podemos ver que el *Activity* encargado de implementar las pestañas extiende de *TabActivity*. El proceso de implementación consiste en definir un *TabHost* que sirva de contenedor de las pestañas y un *TabSpec* que facilite la creación de las mismas. Este constructor nos permitirá definir el aspecto de las pestañas y su contenido/funcionalidad. En este caso su funcionalidad es llamar a otra *Activity*.

```
public class Vocabulario extends TabActivity {

    //Declaramos el objeto con el que podremos almacenar pestañas y el contenido de las
    mismas
    private TabHost mi_tabHost;
    //Declaramos el objeto que nos va a ayudar a definir nuevas pestañas con su respectivo
    contenido
    private TabHost.TabSpec constructor_tab;

    //Declaramos el intent que nos va a permitir dirigirnos a una actividad u otra según
    pulsemos las pestañas
    private Intent intent_tab;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.vocabulario);

        //Obtenemos los recursos de la aplicación y el tabHost donde almacenar nuestras
        pestañas
        recursos = getResources();
        mi_tabHost = getTabHost();

        //Creamos un intent para poder llamar a la actividad indicada cuando se pulse la
        pestaña (Vocabulario Vista Clásica)
        intent_tab = new Intent().setClass(Vocabulario.this,
            Vocabulario_vista_clasica.class);

        //Construimos la plantilla de la pestaña y le añadimos sus características
        (nombre/icono, intent)
        constructor_tab = mi_tabHost.newTabSpec("clasica");
        constructor_tab.setIndicator("Vista Clásica",
            recursos.getDrawable(R.drawable.vista_clasica));
        constructor_tab.setContent(intent_tab);

        //Añadimos la pestaña a nuestro tabHost
        mi_tabHost.addTab(constructor_tab);

        //Se repite el proceso para la otra pestaña (Vista Grupos)

        //Establecemos la pestaña a visualizar inicialmente (en este caso la primera,
        Vista Clásica)
        mi_tabHost.setCurrentTab(0);
    }
}
```

Figura 20: Interfaz de usuario - TabLayout

vi. Menu

Los menús son elementos muy importantes dentro de la interfaz de usuario ya que nos van a permitir acceder a opciones, funcionalidad o información que a primera vista no se encuentran en pantalla. *Android* proporciona tres tipos de menús: **Options Menu** (se muestran los diferentes ítems del menú del *Activity* al pulsar sobre el botón menú del dispositivo), **Context Menu** (se muestran, mediante una lista flotante, los diferentes ítems del menú cuando se mantiene pulsado un *View* al que se le ha añadido esta funcionalidad) y **Submenu** (se muestran, mediante una lista flotante, los diferentes ítems del menú al pulsar sobre un ítem de otro menú). En la aplicación se ha implementado un *Options Menu* por cada pantalla de la misma. A continuación se muestra el código correspondiente a la implementación del *Options Menu* de la pantalla principal de la aplicación (véase figura 21). La creación del menú en sí correrá a cargo del método *onCreateOptionsMenu()*, mientras que la funcionalidad de cada uno de sus ítems será tarea de *onOptionsItemSelected()*.


```
public class Dispositivo_ayuda_comunicacion extends Activity {

    //Asignamos valor numérico a las constantes que representarían botones del menú
    private final int BOTON_SALIR = Menu.FIRST;
    private final int BOTON_AYUDA = Menu.FIRST + 1;
    private final int BOTON_CONFIGURACION = Menu.FIRST + 2;

    //Creamos el menú con las opciones Salir, Ayuda y Configuración
    @Override
    public boolean onCreateOptionsMenu(Menu menu_disp_ayuda_comunic) {
        super.onCreateOptionsMenu(menu_disp_ayuda_comunic);

        menu_disp_ayuda_comunic.add(Menu.NONE, BOTON_SALIR, 0,
            R.string.label_menu_salir).setIcon(R.drawable.icono_salir);

        menu_disp_ayuda_comunic.add(Menu.NONE, BOTON_AYUDA, 1,
            R.string.label_menu_ayuda).setIcon(R.drawable.icono_ayuda);

        menu_disp_ayuda_comunic.add(Menu.NONE, BOTON_CONFIGURACION, 2,
            R.string.label_menu_configuracion).setIcon(R.drawable.icono_confiracion);

        return true;
    }

    //Definimos qué acción realizar según la opción seleccionada del menú
    @Override
    public boolean onOptionsItemSelected(MenuItem opcion_seleccionada) {
        super.onOptionsItemSelected(opcion_seleccionada);

        switch (opcion_seleccionada.getItemId()) {

            case BOTON_AYUDA:
                mostrarAyuda(R.string.titulo_ayuda_principal,
                    R.string.contenido_ayuda_principal, R.drawable.icono_ayuda,
                    R.string.aceptacion_ayuda);
                return true;
            case BOTON_SALIR:
                salir(R.string.salir_preguntar, R.string.salir_si,
                    R.string.salir_no);
                return true;
            case BOTON_CONFIGURACION:
                configuracion();
                return true;
            default:
                return super.onOptionsItemSelected(opcion_seleccionada);
        }
    }
}
```

Figura 21: Interfaz de usuario – Menú

vii. Intent

Las clases *Intent* son uno de los pilares en cualquier aplicación *Android* porque se trata de su red de comunicación. Una de las características principales de *Android* es que se puede indicar que la aplicación realice una determinada tarea sin ni siquiera saber quién se va a encargar de la misma. Esto responde a un enfoque *SOA* (Arquitectura Orientada a Servicios) en el cuál la filosofía es dividir los objetivos concretos y centrarse en el problema a resolver. Los *Intent* se encargan de enviar solicitudes de tareas que serán recogidas gracias a las clases *IntentFilter*, las cuáles quedarán recogidas en el *AndroidManifest.xml*. Los *Intent* pueden ser convocados explícitamente o implícitamente. La primera forma es la más sencilla y es la que usaremos en la aplicación. Consiste en indicarle directamente el nombre del *Activity* que queremos llamar. Si se quiere enviar datos añadidos también se puede hacer a través de objetos *Bundle*. La segunda forma consiste en usar una especie de lenguaje propio en el que participan los elementos del *Intent* (acción, categoría y datos) para resolver qué componente es el más adecuado para encargarse de la tarea solicitada. A continuación se muestra un ejemplo de utilización de los *Intent* (véase las figuras 22 a 24). La figura 22 presenta la parte de código que se encarga de llamar a un *Activity*, enviándole el identificador de un elemento pulsado en pantalla. En la figura 23 se observa cómo se recoge este dato desde el *Activity* invocado. Para finalizar, en la figura 24, se muestra la parte del *AndroidManifest.xml* donde aparece el *IntentFilter* necesario para que el *Intent* pueda ser recogido por el *Activity* que se desea invocar.

```
public class Vocabulario_vista_clasica extends ListActivity {

//Definimos que accion realizar cuando se pulsa sobre un elemento de la lista
@Override
protected void onItemClick(ListView l, View v, int position, long id){
    super.onItemClick(l, v, position, id);

    //Creamos un intent para poder llamar a la actividad traducción una vez se ha
    //seleccionado un elemento de la lista
    Intent intent_traducccion = new Intent(Vocabulario_vista_clasica.this,
    Traducccion.class);

    //Obtenemos el _id del elemento seleccionado, lo guardamos y lo añadimos al
    intent //para poder ser recuperado desde la actividad Traducción
    Long itemSeleccionado = adapterListaVocabulario.getItemId(position);
    Bundle bundle = new Bundle();
    bundle.putLong("ITEM", itemSeleccionado);
    intent_traducccion.putExtras(bundle);
    startActivity(intent_traducccion);
}
}
```

Figura 22: Intent – Emisión

```
public class Traducccion extends Activity {

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

//Recuperamos el _id de la palabra seleccionado en la actividad Vocabulario_vista_clasica
    Bundle bundle = getIntent().getExtras();
}
```

Figura 23: Intent – Recepción

```
<activity android:name=".Traducccion"
    android:label="@string/label_traducccion">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
```

Figura 24: Intent – Declaración en AndroidManifest



4. EVALUACIÓN

Una vez implementada la aplicación es el turno de comprobar si efectivamente cumple con los objetivos para los que fue desarrollada y satisface las necesidades de las personas a las que va dirigida. Es decir, es necesario conocer si la aplicación es capaz de facilitar la comunicación entre personas, ya sean oyentes o no, que no conozcan un idioma común. Para obtener esta información es necesario realizar una serie de tareas previas:

- Instalación de la aplicación en dispositivos móviles reales.
- Realización de un manual de la aplicación junto con tests donde recoger la opinión de los usuarios.
- Estudio de los resultados recogidos en los formularios rellenados por los usuarios.

A. Instalación de la aplicación

El SDK de *Android* junto con el *plugin* ADT del entorno de desarrollo *Eclipse* proporciona las herramientas necesarias para probar el funcionamiento de la aplicación. Entre otras cosas permite emular un dispositivo móvil, con la versión de *Android* que se quiera, sobre el cuál cargar la aplicación y probarla (véase la figura 25).

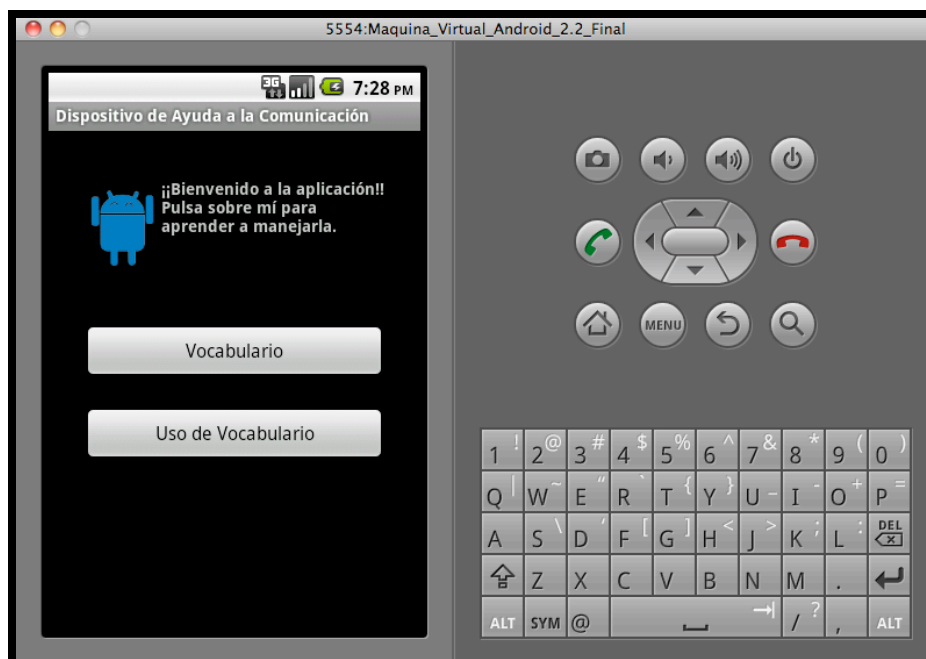


Figura 25: Emulador Android

Durante el desarrollo de la aplicación se ha usado este emulador para comprobar el funcionamiento de la aplicación. Sin embargo, una vez finalizada ésta, es necesario instalarla en un dispositivo real. De esta forma, por un lado nos aseguramos que el comportamiento de la aplicación es idéntico en el emulador y en un dispositivo real, y por el otro podemos ofrecer la posibilidad al usuario que vaya a evaluar la aplicación de que la pruebe en un dispositivo existente en el mercado. Los móviles en los que se ha instalado la aplicación disponían de diferentes tamaños de pantalla, diferentes versiones de *Android* (2.2 y 2.3) y diferentes capas de personalización por parte del fabricante. En todos ellos el comportamiento de la aplicación fue óptimo. Los móviles usados han sido: *HTC Desire*, *HTC Desire HD* y *Samsung Galaxy S*. Ambas marcas (*HTC* y *Samsung*) han colaborado estrechamente con *Google*, siendo los fabricantes de sus móviles “modelo”: *Nexus One* (fabricado por *HTC*) y *Nexus S* (fabricado por *Samsung*).

B. Tutorial y formularios

El tutorial de la aplicación, el cual puede encontrarse en el anexo B de este documento, muestra el funcionamiento de la misma pantalla a pantalla y proporciona dos comics en los que se representan dos hipotéticas situaciones reales en los que apreciar la utilidad y uso de la aplicación. Por otro lado también se confeccionaron varios formularios (ver anexo B) que permitieron recoger tanto información personal del usuario con los que obtener su perfil como su evaluación de la aplicación. Debido a que las pruebas fueron realizadas tanto a oyentes como a discapacitados auditivos, se realizaron dos tipos de formularios siguiendo en cada caso las necesidades de cada grupo. La gran mayoría de las respuestas a las preguntas realizadas siguen una escala de *Likert*, a excepción de las preguntas abiertas y algunas preguntas realizadas al colectivo con discapacidad auditiva que estaban adaptadas para su mejor entendimiento.

C. Estudio de los resultados obtenidos en la evaluación

En este apartado, se encuentra un resumen de los resultados de la evaluación realizada. Debido al espacio limitado de esta memoria, se muestran los resultados de las preguntas y comentarios más relevantes.

La evaluación de la aplicación fue realizada por 13 personas oyentes y 11 con discapacidad auditiva. Las características generales de los dos colectivos implicados en la evolución de la aplicación, en cuanto a edad y familiarización con el ordenador y las nuevas tecnologías se pueden observar en las siguientes gráficas (figuras 26 y 27). Como se puede apreciar, el perfil del colectivo oyente es más joven y por tanto más familiarizado con las nuevas tecnologías que el colectivo con discapacidad auditiva.

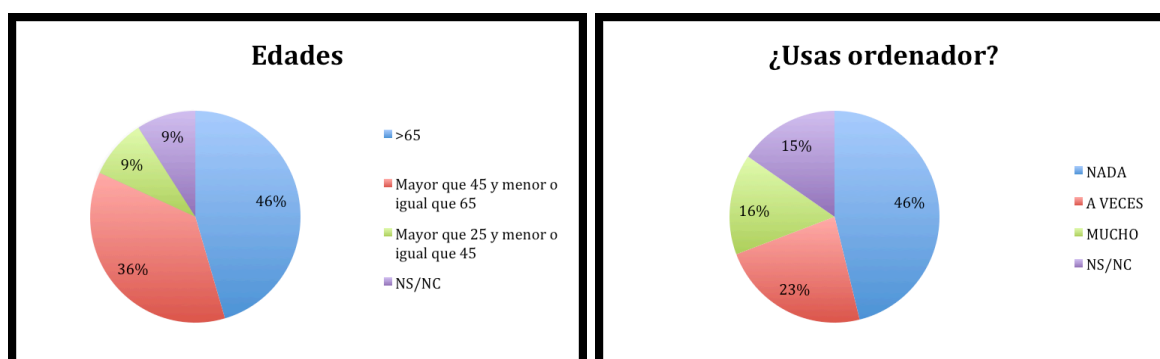


Figura 26: Perfil discapacitados auditivos

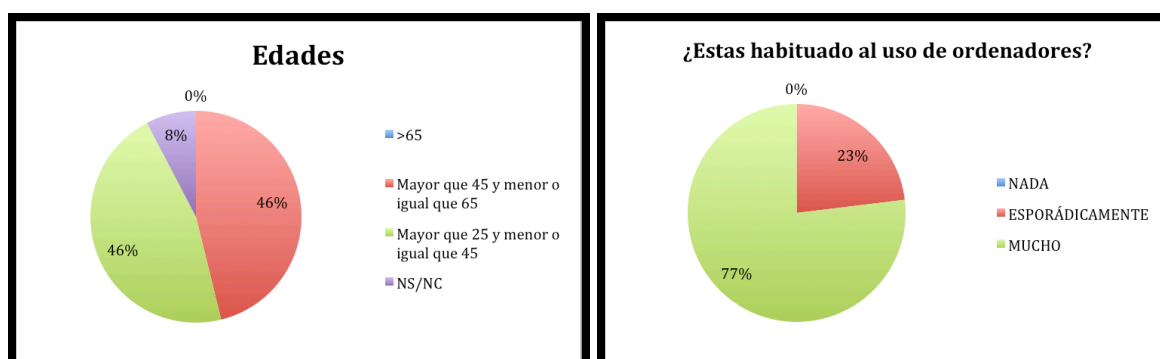


Figura 27: Perfil oyentes

No obstante lo que de verdad distingue a estos grupos son las motivaciones que presentan para usar la aplicación. Las personas con discapacidad auditiva esperan de ella una herramienta que les permita comunicarse con otras personas en el día a día. Por otro lado, a las personas oyentes les resultaría útil una herramienta que les permitiera comunicarse en el extranjero, sin importar el idioma que se emplee en el país de destino.

Estudiando las contestaciones realizadas por el colectivo con discapacidad auditiva habría que indicar que, pese a que en su mayoría no estaban familiarizados con las nuevas tecnologías y

que incluso algunos no disponían de un teléfono móvil, al 100% de los encuestados les gustó la aplicación. Igualmente a la gran mayoría le pareció adecuada como se encontraba organizada la información dentro de la aplicación. Otro de los aspectos que más les agradó de la aplicación fueron los símbolos pictográficos. A continuación se muestran unas gráficas (figuras 28 y 29) que detallan la contestación a tres de las preguntas más significativas que se les formularon.

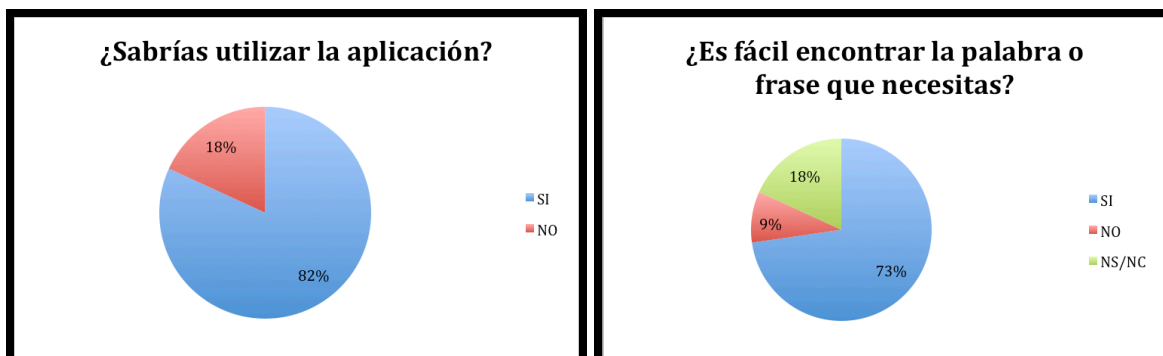


Figura 28: Resultados sobre la utilización de la aplicación y facilidad de uso - Discapacitados auditivos

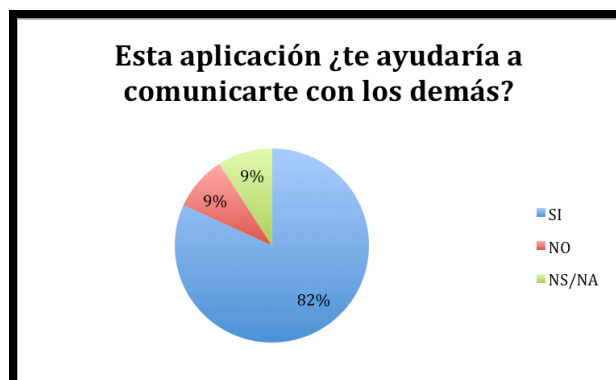


Figura 29: Resultados sobre la ayuda que proporciona a la comunicación - Discapacitados auditivos

En estas preguntas se pueden apreciar que la gran mayoría de estas personas sabrían usar la aplicación, aunque también es cierto que la mayoría indicó que al principio necesitarían que alguien les ayudara hasta que cogieran soltura. La mayor parte encontrarían lo que están buscando en ese momento (un 73%) y creen que la aplicación les ayudaría a comunicarse con los demás (un 82%). Sin embargo también hay aspectos en la aplicación que deberían mejorar o ser modificados tal y como se nos manifestó. Las personas hipoacúsicas (discapacidad auditiva moderada) no tienen problemas para hablar, pero lo que sí necesitan es ayuda a la hora de recibir el mensaje por lo que se nos indicó que echaban en falta un apartado de respuestas a las preguntas que contiene la aplicación. Por otro lado las personas con más

grado de discapacidad, y más aún teniendo en cuenta su edad y la educación que habían recibido, tienen dificultades para entender frases largas. Esto es debido a que la estructura sintáctica de la lengua de signos es diferente a la del castellano. Por ello nos sugirieron que las frases debían ser más cortas y, si fuera posible, incluso con la estructura sintáctica que ellos emplean. De esta forma no tendrían ningún problema para navegar a través de las frases, del mismo modo que no presentaron problemas para encontrar la palabra que deseaban.

Respecto al colectivo oyente también muestran su conformidad respecto a cómo está estructurada la aplicación, les resulta sencillo encontrar la información que necesitan, entienden la información dada por el sistema y consideran que es efectiva a la hora de ayudarles en la comunicación. La figura 30 presenta el resultado de dos de las preguntas más generales que resumen bastante bien la impresión general del sistema. En la parte izquierda, un 80% de los oyentes indicaron que la aplicación desarrollada les parecía muy fácil de utilizar y un 20% fácil. No hubo ninguna persona que indicase que la aplicación era complicada de utilizar. La parte derecha muestra el grado de satisfacción con la aplicación desarrollada y como se puede observar, el total de los oyentes se mostraba satisfecho con el sistema (un 77% de ellos muy satisfecho).

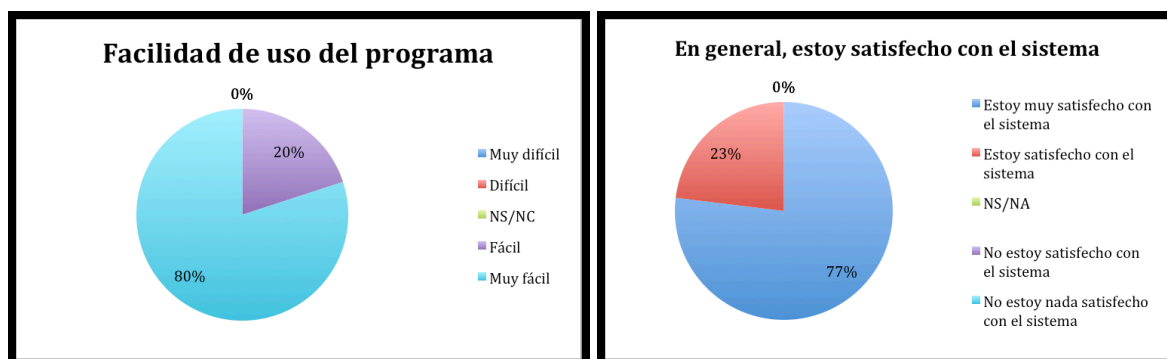


Figura 30: Resultados sobre la facilidad de uso y la satisfacción - Oyentes

Aparte de estos resultados, en este colectivo cabe resaltar que ni siquiera indicaron que necesitaran ayuda para aprender a manejarla. Como sugerencias, se indicó que sería conveniente añadir más palabras y frases, especialmente aquellas relacionadas con el turismo y servicios que puedan necesitar en el extranjero (p.e. salud, aduanas, legalidad, etc.). También hubo comentarios respecto a que la aplicación fuera más llamativa y que se rediseñara algún símbolo pictográfico. Para finalizar este apartado de evaluación, transcribir el



comentario de una especialista en deficiencia auditiva e intérprete de lengua de signos: *“Me parece muy práctica la aplicación para transmitir información, ya que evita posibles errores de interpretación. También pienso que es de gran ayuda el uso de pictogramas pues permite, que personas sin lenguaje funcional puedan recibir y/o dar información. Es un instrumento muy válido para que estas personas se comuniquen en su vida diaria.”*

5. CONCLUSIONES Y POSIBLES AMPLIACIONES

Si estas líneas estuvieran escritas por símbolos inteligibles para el lector, probablemente le causarían una sensación de vacío e impotencia al saber que hay algo ahí, delante de sus ojos, que no logra entender qué es, qué significa. Esa es la sensación que sienten, demasiado a menudo, personas con discapacidad auditiva. Esa barrera, provocada por la incompreensión y que puede llevar incluso a la soledad, es la que se ha querido derribar o al menos disminuir en este proyecto fin de carrera.

Ayudar a personas con discapacidad auditiva u oyentes que no pueden comunicarse con otras personas, porque no existe un idioma común entre ellos, ha sido el objetivo de este trabajo. Y conseguir proporcionar esta ayuda es, por tanto, el principal logro que se ha alcanzado con el desarrollo de la “Aplicación de ayuda a la comunicación, para personas con discapacidad auditiva, sin intérprete, u oyentes con desconocimiento del idioma del receptor.” Sin embargo, esta aplicación no sólo va a ayudar en la comunicación a personas que lo necesiten si no que, durante las tareas necesarias para su desarrollo, ya ha estado ayudando a otra persona, al autor de la misma. Esta ayuda ha consistido en que, a medida que la aplicación iba avanzando en su desarrollo, también los conocimientos del autor han avanzado con ella. Adquiriendo conocimientos que no poseía y asentando los que ya habían sido aprendidos previamente. Entre todos ellos, destaca el aprendizaje de la plataforma para dispositivos móviles *Android*.

Por otro lado, gracias a la decisión de desarrollar la aplicación para un dispositivo móvil, se han debido de seguir una serie de pasos que han abarcando desde la propia percepción del problema hasta la implantación de la solución en un dispositivo real sobre el que realizar las pruebas pertinentes. Por tanto, se ha adquirido una visión completa del modo de desarrollar una aplicación a través de todo su ciclo de vida: definición de necesidades, análisis, diseño, codificación, pruebas, validación y mantenimiento y evolución. En estas fases, entre otras cosas, se han tenido que realizar diagramas UML (casos de uso y de actividad) con los que describir la aplicación, seleccionar un patrón de diseño *software* sobre el cuál proceder a la implementación (Modelo Vista Controlador), diseñar interfaces gráficas y usar un entorno de desarrollo integrado (*Eclipse*) en el que codificar la aplicación en el lenguaje *Java* usando las librerías que proporciona el SDK de *Android*.

Uno de los puntos más importantes ha sido el de presentar la aplicación al usuario final. Esto es debido a que, en el momento en el que entra en contacto la hipotética solución del problema con los afectados del mismo, se averigua si la aplicación y el esfuerzo que ha llevado consigo ha tenido sentido o no. Y en este aspecto las valoraciones han sido positivas porque la idea ha resultado atractiva y la aplicación ha sido considerada válida y útil. Una de las características que ha hecho que la aplicación tuviera esta acogida ha sido su usabilidad. Tanto a las personas con discapacidad auditiva como a las que no padecían ningún tipo de discapacidad les ha resultado fácil su uso. Algo realmente importante si se tiene en cuenta que un gran número de las personas discapacitadas encuestadas eran personas de avanzada edad, que no se encontraban familiarizadas con las nuevas tecnologías, y que incluso en algún caso, no disponían de móvil. Sin embargo, la aplicación aún debe evolucionar para adaptarse perfectamente a las necesidades de las personas a las que va dirigida. Estas conclusiones han sido extraídas de entre un grupo de personas que han aportado su opinión al respecto de la aplicación. Entre estas personas se encuentran discapacitados auditivos de asociaciones de sordos los cuáles han aportado un gran número de sugerencias para perfeccionarla. Personal que trabaja en la asociación de daño cerebral, también ha manifestado su interés en la aplicación, por lo que en un futuro se tendrá en cuenta sus requerimientos para intentar adaptarla a los mismos.

No obstante, ya se dispone de una serie de solicitudes a implementar, que serían factibles, y que permitirían satisfacer en gran medida las necesidades del potencial usuario de la aplicación, entre las cuales, podemos destacar las siguientes:

- Aumentar el vocabulario y las frases, reforzando especialmente aquellas que atañen a situaciones importantes como pueden ser las relativas a la salud y las judiciales.
- Separar la aplicación en dos modos: oyente y discapacitado auditivo. El modo oyente sería el que ha sido desarrollado y el modo discapacitado auditivo presentaría la misma funcionalidad pero adaptando al usuario las frases presentadas teniendo en cuenta que éstas deben ser más cortas y a ser posible estar escritas según las reglas sintácticas de la lengua de signos española.
- Añadir frases que sean posibles respuestas a las preguntas que contiene la aplicación.
- Añadir frases de carácter informativo, es decir que no pregunten nada ni esperen respuesta.

- Añadir la opción de escribir o dibujar la pregunta o respuesta si ésta no está incluida en la aplicación. Permitir también grabar esta pregunta o respuesta dentro de una de las situaciones de la aplicación.
- Crear una opción “favoritos” y “recientes”, donde queden almacenadas las frases o palabras que más se usen y las últimas que hayan sido utilizadas.

Estas posibles mejoras han sido extraídas de las evaluaciones realizadas con la aplicación tanto por el colectivo de oyentes como el colectivo con discapacidad auditiva.



BIBLIOGRAFÍA

A. Referencias

- [1] GERONET “Informática e Ingeniería”: <http://www.geronet.com.ar/?p=67>
- [2] Asociación EUNATE elkartea: http://www.eunate.org/tel_texto.htm
- [3] SVisual Servicio de Videointerpretación en LSE: <http://www.svisual.org/>
- [4] XBOX Kinect: <http://www.xbox.com/es-ES/kinect>
- [5] Wikipedia Smartphone: <http://es.wikipedia.org/wiki/Smartphone>
- [6] Apple Facetime: <http://www.apple.com/es/iphone/features/facetime.html>
- [7] Skipe: <http://www.skype.com/intl/es/home/>
- [8] Buttussi F., Chittaro L., Carchietti E., Coppo M. (2010) Using Mobile Devices to Support Communication Between Emergency Medical Responders and Deaf People, Proceedings of MOBILE HCI 2010: 12th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 7-16. ACM Press, New York.
- [9] Apple APPS: <http://www.apple.com/es/iphone/apps-for-iphone/>
- [10] CNN Expansión: <http://www.cnnexpansion.com/tecnologia/2011/02/09/nokia-apple-google-iphone-android-symbia>
- [11] Fundación Linux: <http://www.linuxfoundation.org/>
- [12] Meego: <https://meego.com/>

B. Libros y sitios web

- FIAPAS (Jáudenes, C. Et ál.) “Manual básico de formación especializada sobre discapacidad auditiva.
- Servicio de Información sobre Discapacidad (SID): <http://sid.usal.es/default.aspx>
- Sociedad española para el desarrollo de sistemas de comunicación aumentativos y alternativos:
http://www.esaac.org/index.php?option=com_content&task=view&id=35&Itemid=45
- Equipo específico de deficiencia auditiva Ministerio de Educación y Ciencia, “Índice temático de los símbolos pictográficos para la comunicación (no vocal)”.

- Nicolas Gramlich, “Handbook! Android Programming”: <http://andbook.anddev.org/>
- Chris Haseman, “Android Essentials”. Editorial Apress.
- J.F. Dimarzio “Android A Programmer’s Guide”. Editorial McGrawHill
- Rick Rogers, John Lombardo, Zigurd Mednieks and Blake Meike, “Android Application Development”. Editorial O’Reilly.
- Anthony Stevens, Mark Murphy, Gabor Paller, “Hello, Android”. Editorial Burnette.
- Frank Ableson, Charlie Collins y Robi Sen, “Android, guía para desarrolladores”. Editorial Anaya.

ANEXO A: INSTALACIÓN

En este punto se pretende explicar cómo instalar y configurar el entorno de desarrollo para desarrollar o emular una aplicación *Android*. Además, se explicará cómo instalar la aplicación en el propio dispositivo móvil.

A. Instalación del entorno de desarrollo

La instalación del entorno de desarrollo necesario para poder desarrollar, depurar y emular una aplicación *Android* va a constar de 3 fases: instalación de un entorno de desarrollo integrado, instalación de las propias herramientas de desarrollo de *Android* e instalación de un *plugin* que permite integrar los dos anteriores. Como se vio en puntos anteriores, uno de los motivos por los que se escogió la plataforma *Android* para la realización de la aplicación es que dispone de unas buenas herramientas de desarrollo multiplataforma gratuitas. Por ello se harán las indicaciones necesarias para la instalación de las mismas en los 3 sistemas operativos más extendidos: *Windows*, *MacOS X* y *GNU/Linux* (distribución *Ubuntu*).

a. Entorno de desarrollo integrado

Antes de instalar el SDK (*Software Development Kit*) de *Android* vamos a proceder a la instalación de un IDE (*Integrated Development Environment*). Realmente sólo es necesario tener instalado el JDK (*Java Development Kit*) en nuestro equipo, sin embargo disponer de un IDE y más concretamente de *Eclipse*, va a facilitar la tarea de desarrollo. El motivo de esto es que *Eclipse* permite integrarse perfectamente con el SDK de *Android* gracias a su *plugin* ADT (*Android Development Tools*).

1. Se debe tener instalada la última versión disponible del JDK. Es importante que sea el JDK y no el JRE ya que tener instalado éste último no es suficiente.

Windows: Descargarlo desde la web oficial de *Oracle* e instalarlo:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

MacOS X: Java viene integrado en el propio S.O. por lo que no es necesario descargarlo ya que el sistema de actualizaciones automáticas se encargará de esta tarea, si se detecta

una nueva versión del JDK disponible. Sin embargo si se quiere conocer la versión que está instalada actualmente en el equipo se puede hacer lo siguiente:

```
cd /System/Library/Frameworks/JavaVM.framework/Versions
```

```
ls -la
```

Ubuntu (paso opcional): Si se tiene activado los repositorios *Multiverse* desde un terminal hacer lo siguiente:

```
sudo aptitude update
```

```
sudo aptitude install sun-java6-jdk (o la versión que corresponda)
```

No obstante esto se puede realizar también a través de interfaz gráfica usando *Synaptic*, tanto para activar el repositorio *Multiverse* como para instalar el paquete.

2. Descargar e instalar *Eclipse*.

Windows / MacOs X: Descargarlo desde la siguiente página:

<http://www.eclipse.org/downloads/>. En esta página, tendremos que escoger *Eclipse IDE for Java Developers* o *Eclipse IDE for Java EE Developers* e instalarlo.

Ubuntu: En este S.O. se pueden realizar los pasos 1 y 2 directamente en uno sólo (por eso el paso anterior se indicó que era opcional). Desde un terminal hacer lo siguiente:

```
sudo aptitude install eclipse sun-java6-jdk
```

Con esto instalaremos los dos paquetes a la vez, sin embargo *Eclipse* seguirá usando la versión GNU de Java que venía incorporada en el S.O. Esto hará que *Eclipse* funcione más lento que si usa el paquete oficial de *Sun* que hemos instalado. Para ello debemos hacer lo siguiente:

```
sudo update-java-alternatives -s java-6-sun
```

y ahora editamos el archivo de configuración de la máquina virtual de Java para indicar cuál queremos usar:

```
sudo -b gedit /etc/jvm
```

y agregamos la siguiente línea al principio:

```
/usr/lib/jvm/java-6-sun
```

Según la versión de *Eclipse* instalada puede que posea un bug que hace que se ignore la configuración de Java que acabamos de indicarle al S.O., para solucionarlo hacemos lo siguiente:

```
sudo -b gedit /etc/eclipse/java_home
```

y agregamos al principio la misma línea que habíamos empleado antes:

```
/usr/lib/jvm/java-6-sun
```

b. SDK (Software Development Kit)

Una vez instalado el IDE *Eclipse* vamos a proceder a la instalación de las herramientas de desarrollo *Android*.

1. Descargar e instalar el SDK.

Windows / MacOS X / Ubuntu: Descargarlo desde la siguiente página:

<http://developer.android.com/sdk/index.html>. Una vez descargado, extraer el contenido del archivo en la ruta deseada. El directorio creado tendrá los siguientes subdirectorios: *tools*, *samples* y *docs*.

2. Actualizar el PATH, ya que es recomendable que el subdirectorio *tools* se encuentre en él.

Windows: Pulsar con el botón derecho del ratón sobre “Mi PC” y acceder a “Propiedades”. Desde ahí seleccionamos “Configuración avanzada del sistema” y posteriormente “Variables de entorno”. En el recuadro inferior (“Variables del Sistema”) buscamos la entrada llamada “Path”, la seleccionamos y pulsamos “Modificar”. Desde ahí agregamos la ruta del subdirectorio.

MacOS X: Para ver las variables de entorno en este S.O., podemos escribir desde un terminal:

```
env
```

o si queremos ver sólo la variable de entorno “PATH”:

```
echo $PATH
```

Para añadir la ruta del subdirectorio podemos hacer lo siguiente:

```
nano .profile
```

y escribir:

```
export PATH=$PATH:/ruta_del_subdirectorio_tools
```

Ubuntu: Editar los archivos `~/.bash_profile` y `~/.bashrc` y añadir la ruta del subdirectorio a la variable PATH en ambos. Si ésta no aparece en los ficheros se puede poner:

```
export PATH=$PATH:/ruta_del_subdirectorio_tools
```

c. ADT (Android Development Tools)

Para finalizar con la instalación del entorno de desarrollo vamos a proceder a la instalación de un *plugin* de *Eclipse*. Este *plugin* va a permitir a *Eclipse* convertirse en un IDE perfectamente adaptado para el desarrollo de aplicaciones *Android*, desde la creación de proyectos hasta la exportación de nuestras aplicaciones como ficheros .apk, pasando por creación de interfaces de usuario y procesos de depuración.

1. Descargar e instalar el *plugin* de *Eclipse* ADT.

Windows / MacOS X / Ubuntu: Todo el proceso se realiza desde *Eclipse*, por ello es común a los tres S.O. Una vez abierto *Eclipse* se debe pulsar sobre el botón “Help” y seleccionar la opción “Install New Software...”. En la pantalla que se nos muestra, debemos pulsar sobre el botón “Add” y rellenar la información que aparece en esta pantalla de la siguiente manera:

Name: ADT Plugin

Location: <https://dl-ssl.google.com/android/eclipse/>

Una vez insertados estos datos, pulsamos el botón OK y se nos mostrará una pantalla con el *software* disponible. Seleccionamos “Developer Tools” y pulsamos sobre el botón “Next”. En la siguiente pantalla se puede observar la lista de herramientas que van a ser descargadas. Pulsamos de nuevo el botón “Next” y por último, el botón de “Finish”.

Para completar la configuración del *plugin* ADT, reiniciamos *Eclipse* y en el menú de preferencias (desde el menú “Window” o desde el menú “Eclipse” si se está trabajando con MacOS X), seleccionamos “Android” en la columna de la izquierda e indicamos la ruta donde se encuentra el SDK.

2. Crear un AVD (*Android Virtual Device*), es decir, un dispositivo virtual desde el cuál podremos probar las aplicaciones que desarrollemos.

Windows / MacOS X / Ubuntu: Desde *Eclipse* seleccionamos “Window” y la opción “Android SDK and AVD Manager”. Se nos muestra una pantalla en la que podemos comprobar los paquetes instalados, los disponibles y una opción desde la cuál podemos crear dispositivos virtuales de *Android* con la versión que queramos. Si hemos creado varios de estos AVDs, podemos indicar con cuál de ellos queremos que se ejecute nuestra aplicación. Para ello pulsamos con el botón derecho del ratón sobre el proyecto creado y

seleccionamos “Properties”. Se nos muestra una pantalla en la que debemos seleccionar en la columna de la izquierda “Android”. Hecho esto vemos una lista con las versiones disponibles en las cuales podemos ejecutar la aplicación. Se selecciona la deseada. Una vez indicada la versión de la plataforma sobre la cual queremos ejecutar la aplicación, debemos indicarle la máquina virtual que queremos emplear. Nuevamente pulsamos con el botón derecho del ratón sobre el proyecto, seleccionamos “Run As” y a continuación “Run Configurations...”. En la pantalla que se nos muestra seleccionamos la pestaña “Target” y desde ahí escogemos la máquina virtual.

3. Es muy recomendable también aumentar la fecha de caducidad de los certificados utilizados para firmar las aplicaciones creadas. Por defecto estos certificados tienen un año de validez, por lo que cumplido este tiempo puede producirse el siguiente error al compilar nuestra aplicación:

Error generating final archive: Debug Certificate expired on xx/xx/xxxx

Si ocurre esto la solución es borrar el fichero *debug.keystore* (este fichero es el lugar donde se almacenan los certificados). La ruta donde se encuentra la podemos ver pulsando “Preferences” de *Eclipse*, seleccionando “Android” en la columna de la izquierda y en el desplegable que aparece pulsando “Build”. Para borrar este fichero, tendremos que escribir en un terminal uno de los dos siguientes comandos dependiendo del S.O. que estemos utilizando:

Windows:

del C:\user\NombreUsuario\.android\debug.keystore

MacOs X / Ubuntu:

rm /Users/NombreUsuario/.android/debug.keystore

Una vez borrado, y dentro del directorio, vamos a aumentar los días que deben de pasar antes de que vuelva a expirar el certificado, por ejemplo 99999. En un terminal, escribimos el siguiente comando (independientemente del S.O. con el que estemos trabajando):

keytool -genkey -keypass android -keystore debug.keystore -alias androiddebugkey -storepass android -validity 99999 -dname "CN=Android Debug,O=Android,C=US".

Una vez borrado el fichero ya aumentado los días en los que expirará nuestro certificado, simplemente hay que hacer un “clean” de nuestro proyecto con error y reiniciar *Eclipse*. Automáticamente, se generará el nuevo *debug.keystore* con una nueva fecha para el certificado.

B. Instalación de la aplicación en un dispositivo móvil

Para instalar la aplicación en un dispositivo móvil real es necesario generar previamente un archivo *.apk*. Estos archivos son una variante del formato JAR de *Java* y son usados para distribuir e instalar componentes empaquetados para *Android*. Por ello en este punto se va a tratar cómo obtener el fichero *.apk* de nuestra aplicación y cómo instalarla en un móvil *Android*.

a. Generación del fichero *.apk*

La generación de un fichero *.apk* consiste en generar una aplicación firmada, creando un *keyStore*. Los pasos a realizar son los siguientes:

- Pulsar con el botón derecho sobre la carpeta raíz del proyecto y seleccionar la opción “Android Tools”. Desde ahí seleccionar “Export Signed Application Package...”.
- Seleccionar el proyecto deseado y pulsar el botón “Next”.
- Si no hay un *keyStore* creado previamente y que queramos usar debemos crear uno. Para ello se debe seleccionar “Create new keystore”, indicar la ruta donde queremos que se guarde e introducir una contraseña para el mismo. Pulsar “Next”.
- Para crear la clave hay que rellenar unos campos del formulario que aparece. Pulsar “Next”.
- Por último se debe indicar la ruta donde queremos que se guarde el *.apk*. Pulsar “Finish”.

b. Instalación del fichero *.apk* en el dispositivo móvil

Una vez generado el fichero *.apk*, simplemente hay que llevarlo a un móvil e instalarlo desde allí. Hay terminales como por ejemplo el “*Nexus one*” (móvil fabricado bajo las directrices de *Google* y que sirve como terminal prototipo para desarrolladores) que incluso si reciben el *.apk* desde correo lo detectan como tal y se instala directamente. En el caso de otros

terminales, se requiere que el fichero .apk sea pasado al dispositivo móvil a través de USB.

Los pasos serían los siguientes:

- Se debe habilitar la opción de poder instalar archivos que no provengan del *Android Market*. Para ello hay que activar, desde el móvil, lo siguiente:
Ajustes → *Aplicaciones* → *Orígenes desconocidos*
- Una vez se ha dado permiso para instalar aplicaciones externas al *Android Market* podemos usar un cable USB para guardar la aplicación, en el propio móvil o en la tarjeta de memoria externa del mismo.
- Desde el gestor de archivos del móvil accedemos a la aplicación y la ejecutamos para instalarla.
- Si se desea desinstalarla se debe hacer desde:
Ajustes → *Aplicaciones* → *Administrar aplicaciones*



ANEXO B: TUTORIAL Y CUESTIONARIOS

En este punto se pretende presentar la “Aplicación de ayuda a la comunicación, para personas con discapacidad auditiva, sin intérprete, u oyentes con desconocimiento del idioma del receptor” al usuario final. Para ello se mostrarán capturas de pantalla de la propia aplicación junto con unos breves comentarios sobre las mismas. De esta manera el usuario podrá tener una percepción general del funcionamiento de la aplicación. No obstante, para que la comprensión sea total, se presentarán dos posibles situaciones de la vida real en las que se hace uso de la aplicación. En estas situaciones se explicarán cuáles son las acciones que realizan sobre la aplicación los protagonistas de las mismas.

Antes de comenzar el tutorial el usuario debe rellenar un cuestionario, sobre una determinada información personal, que se le habrá proporcionado previamente. Con este cuestionario se podrá englobar al usuario dentro de un determinado perfil. Una vez rellenado este cuestionario y leído el tutorial, con lo que habrá adquirido los conocimientos básicos sobre funcionamiento y uso de la aplicación, se le proporcionará otro. Con este segundo cuestionario se pretende recoger su impresión general de la aplicación. Con los datos obtenidos se podrá tener una valoración, desde el punto de vista del perfil de cada usuario, de la aplicación. De esta forma se averiguarán los puntos fuertes y débiles de la misma y, por tanto, se podrá ir mejorando la aplicación.

A. Tutorial

A continuación se muestran todas y cada una de las pantallas que forman parte de la aplicación. Acompañando las capturas de pantalla se indican todos los aspectos relevantes de las mismas como pueden ser: acceso a esas pantallas, explicación de los elementos que se muestran, para qué sirven, acción por parte del usuario, resultados que se van a obtener, etc.

a. Pantallas de la aplicación

i. Menú principal

Esta es la pantalla que se muestra una vez se ha ejecutado la aplicación (véase la parte izquierda de la figura 31). Desde ella podemos acceder al “Vocabulario” y al “Uso de Vocabulario”. También es posible acceder a una guía de la aplicación pulsando sobre el Androide azul (véase la parte izquierda de la figura 32). Al pulsar sobre la tecla “Menú” del móvil aparece el menú emergente (véase la parte derecha de la figura 31).



Figura 31. Pantalla inicial de la aplicación.

Indicar que en este tutorial se presentan las pantallas tal cuál aparece en el móvil y, al lado de ellas, como aparecerían si se pulsa sobre el botón menú. Las opciones que se presentan en este menú emergente son: “Salir” (sale de la aplicación), “Ayuda” (opción común a todas las pantallas donde se muestra una explicación de la pantalla en la que nos encontramos, véase el ejemplo de la parte de la derecha de la figura 32) y “Configuración” (desde esta opción se podrá seleccionar los idiomas en los que se quiere que se realice la traducción, véase la figura 33).

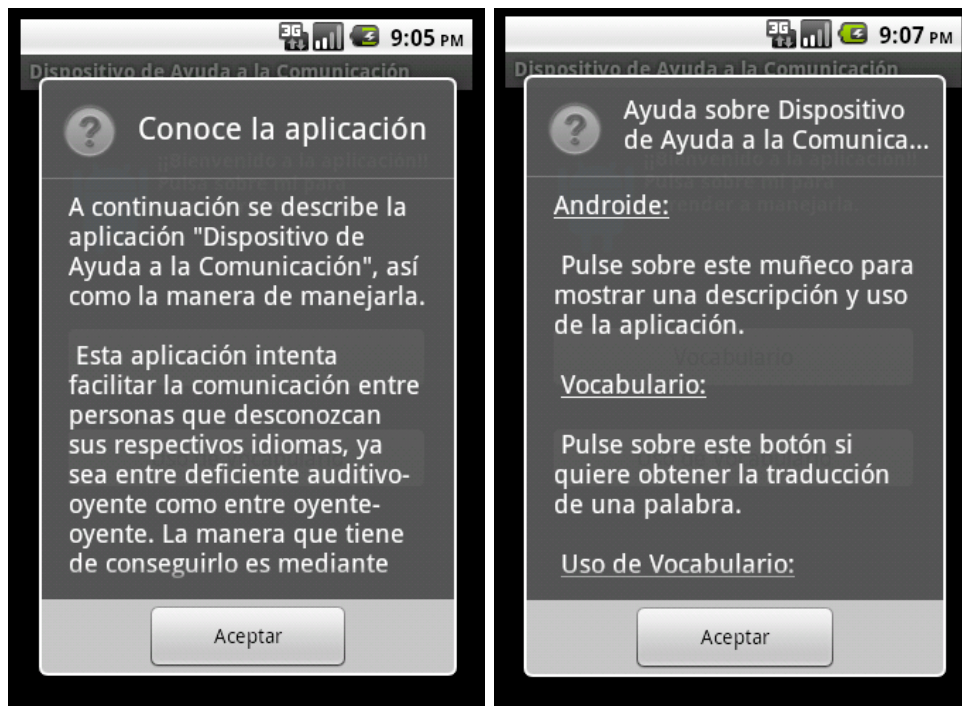


Figura 32. Ayuda sobre la aplicación

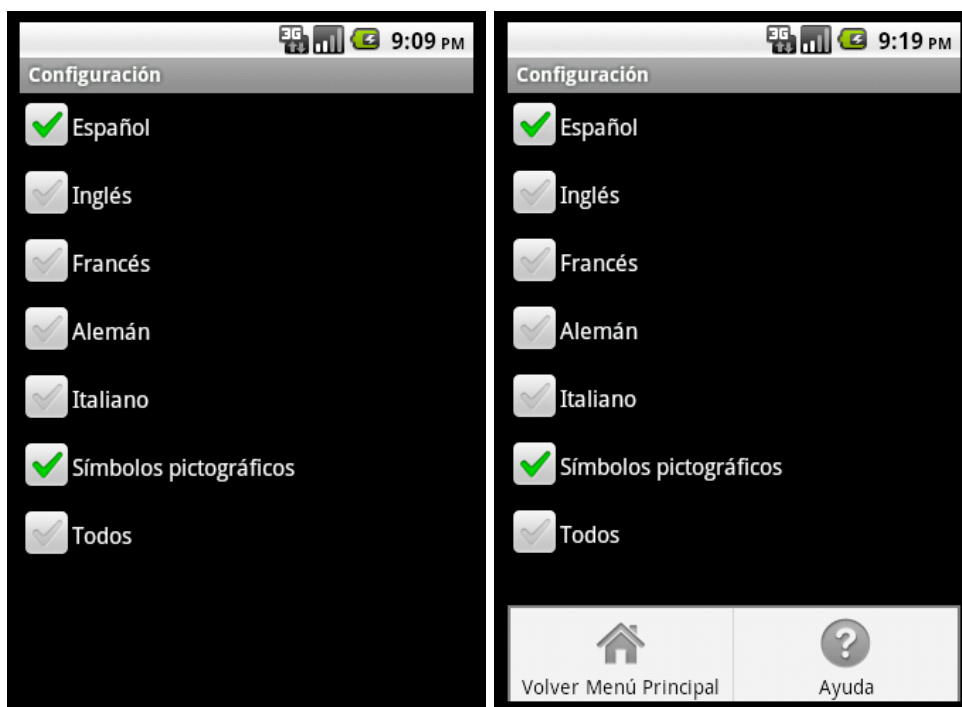


Figura 33. Configuración de los idiomas y símbolos utilizados por la aplicación

ii. Vocabulario

Para acceder a la opción “Vocabulario” se debe pulsar sobre el botón del mismo nombre que aparece en el “Menú principal” (véase la figura 31). Dentro de esta opción el usuario puede seleccionar una de las palabras que se le presentan y obtener su representación en otro idioma y/o símbolo pictográfico (para seleccionar los idiomas en los que se desea la palabra se debe realizar desde la pantalla que muestra la figura 33, que aparece al pulsar en las opciones de configuración de la pantalla principal).

Las palabras del vocabulario pueden mostrarse de dos formas: vista clásica o vista por grupos. La opción por defecto es la “Vista Clásica” y en ella se muestran las palabras en orden alfabético (véase la figura 34). En esta opción el usuario busca la palabra que está buscando y al seleccionarla le aparece la traducción de la misma. La otra posible opción es la “Vista por Grupos” donde las palabras se encuentran agrupadas por una temática común (véase la figura 35). Para cambiar entre vistas simplemente hay que pulsar sobre una de las dos pestañas, que se encuentran en la parte superior de la pantalla. Estas pestañas tienen el nombre de las vistas y un icono representativo de la manera en que se encuentran ordenadas sus palabras.



Figura 34. Vista clásica del vocabulario



Figura 35. Vista por grupos dentro de la opción de vocabulario.

Cuando un usuario selecciona un determinado grupo estando en la “Vista por grupos”, se le muestra un listado de palabras, ordenadas por orden alfabético, pertenecientes a ese grupo (véase la figura 36).



Figura 36. Grupos de palabras del vocabulario de la situación Lugares

Una vez se ha escogido una palabra, ésta se muestra en los idiomas que se hayan seleccionado, así como su representación como símbolo pictográfico, si es que ésta ha sido también seleccionada (véase la figura 37). Por defecto la aplicación está configurada para mostrar la palabra en español y presentar su símbolo pictográfico.



Figura 37. Traducción de una palabra determinada al idioma seleccionado e icono correspondiente.

Respecto a los botones que podemos encontrar en cada una de las pantallas cuando se pulsa sobre el botón menú del móvil tenemos: “Ayuda” (explicación de la pantalla), “Volver Menú Principal” (como indica su nombre se regresa al menú principal o lo que es lo mismo, a la pantalla que se muestra en la figura 31) y “Volver Vocabulario” (se regresa a la pantalla inicial de la opción de “Vocabulario” o figura 34 de este tutorial).

iii. Uso de Vocabulario

Dentro de la opción “Uso de Vocabulario”, el usuario puede seleccionar una de las frases que se le presentan y obtener su representación en otro idioma y/o como una secuencia de símbolos pictográficos (para seleccionar los idiomas en los que se desea la frase es necesario

configurarlo desde la pantalla que se muestra en la figura 33, que aparece al pulsar en las opciones de configuración de la pantalla principal).

Las frases se encuentran agrupadas por una temática común (véase la figura 38). Una vez se ha seleccionado un grupo, se muestra un listado con las frases disponibles para esa situación. Por ejemplo, la figura 39 muestra las frases asociadas a la situación “Alojamiento”.



Figura 38. Menú principal sobre el uso de vocabulario o frases de la aplicación.



Figura 39. Frases asociadas a la temática alojamiento.

Las frases pueden ser de dos tipos: completas e incompletas. Las frases completas son aquellas que una vez seleccionada no requieren ninguna acción por parte del usuario para ser traducidas (ejemplos de las figuras 39 y 40).



Figura 40. Traducción de una frase determinada al idioma seleccionado y a los símbolos pictográficos.

Las frases incompletas se distinguen de las completas en que finalizan en “:.”. La figura 41 muestra las frases incompletas relacionadas con los medios de transporte.



Figura 41. Frases asociadas a la temática transporte.

Cuando un usuario selecciona una frase incompleta, la aplicación requiere de la acción del usuario para completarla. El usuario debe escoger una de las palabras que se le muestran. La figura 42 muestra un ejemplo donde se presentan las posibles opciones para la frase “A qué hora sale el: ”. En este caso el usuario tendrá que escoger entre una de las siguientes opciones: barco, bus, metro, tranvía o tren.

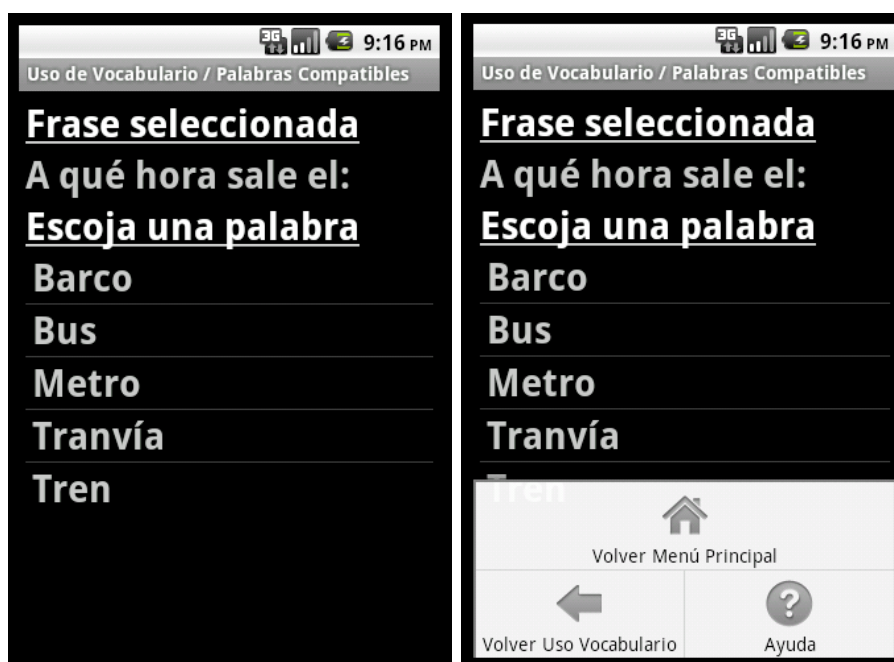


Figura 42. Presentación de la frase incompleta seleccionada y de las palabras compatibles.

Una vez seleccionada la frase (completándola en el caso de que sea incompleta), ésta se muestra en los idiomas que se hayan seleccionado, así como su representación como secuencia de símbolos pictográficos, si es que ésta ha sido también seleccionada. Por defecto la aplicación está configurada para mostrar la frase en español y presentar su secuencia de símbolos pictográficos. La figura 43 muestra un ejemplo de traducción mediante símbolos pictográficos de la frase “¿A qué hora sale el tren?”.

Respecto a los botones que podemos encontrar en cada una de las pantallas cuando se pulsa sobre el botón menú del móvil tenemos: “Ayuda” (explicación de la pantalla), “Volver Menú Principal” (como indica su nombre se regresa al menú principal, véase la figura 31) y “Volver Uso Vocabulario” (se regresa a la pantalla inicial de Uso de Vocabulario, véase la figura 38).



Figura 43. Traducción de una frase incompleta al idioma seleccionado y a los símbolos pictográficos.

B. Ejemplos de uso en posibles situaciones reales

A continuación se presentan, en forma de comic, dos posibles situaciones de la vida real en las cuales se podrá observar la utilidad de la aplicación así como su uso.

a. En el restaurante

En esta figura 44 podemos observar una situación en la que una pareja, que se encuentra en otro país, va a cenar a un restaurante. No entienden el idioma en el que les habla el camarero por lo que optan por usar la aplicación.



Figura 44. Situación de entrada en un restaurante.

Primero la configuran para añadir el idioma inglés (ver figuras 45 y 46), por si el camarero conoce ese idioma, a los activados por defecto (español y símbolos pictográficos). Una vez hecho esto seleccionan, dentro de “Uso de Vocabulario”, la situación “Restaurante” (ver figura 47). Dentro de esta situación se muestran una serie de frases útiles para cuando nos encontremos en uno de ellos. A esta pareja le interesa indicar al camarero que quieren una mesa. Esta frase es incompleta, por lo que tienen que completarla con el número, en este caso dos (ver figura 48). Una vez completada la frase ya pueden enseñar la pantalla de su móvil al camarero para que sepa qué es lo que quieren.



Figura 45. Pantalla inicial de la aplicación y selección del menú de configuración.

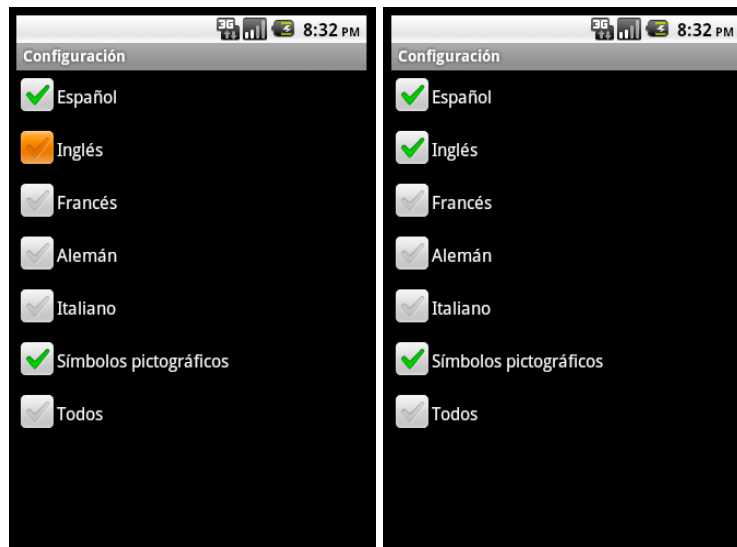


Figura 46. Configuración de los idiomas de la aplicación incluyendo el inglés.



Figura 47. Selección de la situación restaurante dentro del módulo uso del vocabulario.

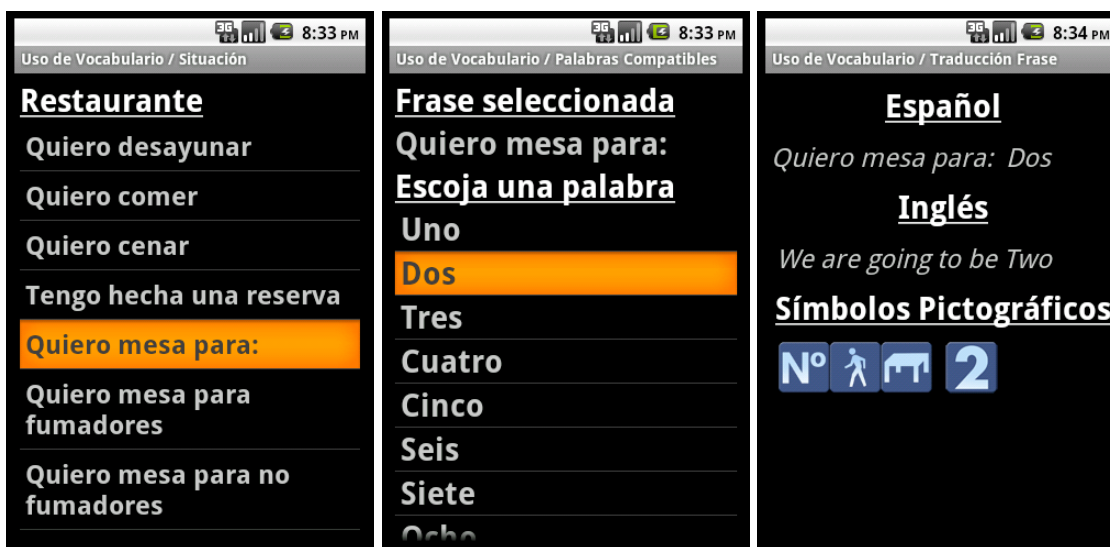


Figura 48. Selección de la frase "Quiero una mesa para dos" y su correspondiente traducción.

Una vez han cenado se disponen a pedir la cuenta tal y como se muestra en la figura 49. Para ello, situándose de nuevo en la situación “Restaurante” del “Uso de Vocabulario”, seleccionan la frase adecuada (ver figura 50).



Figura 49. Situación en la que se pide la cuenta al camarero.



Figura 50. Selección de la frase “Quiero la cuenta” y posterior traducción.

Sin embargo, en la figura 51 se puede observar que el chico se da cuenta que no ha traído dinero en metálico, por lo que necesita preguntar si aceptan pagar con tarjeta. Nuevamente en la situación “Restaurante” del “Uso de Vocabulario” selecciona la frase correspondiente (ver figura 52).



Figura 51. Situación en la que se pregunta sobre el pago con tarjeta.



Figura 52. Selección y traducción de la frase “¿Puedo pagar con tarjeta?”

Desafortunadamente este restaurante no acepta pago con tarjeta. Ante la situación desesperada que se les plantea, optan por una no muy buena opción con consecuencias aún peores tal y como se puede observar en la figura 53. Pero no todo iban a ser malas noticias y la aplicación dispone de una situación que seguro que ellos hubieran preferido no necesitar. Seleccionan la situación “Comisaría” dentro de “Uso de Vocabulario” y ahí la frase que no les queda más remedio que usar (ver figura 54).

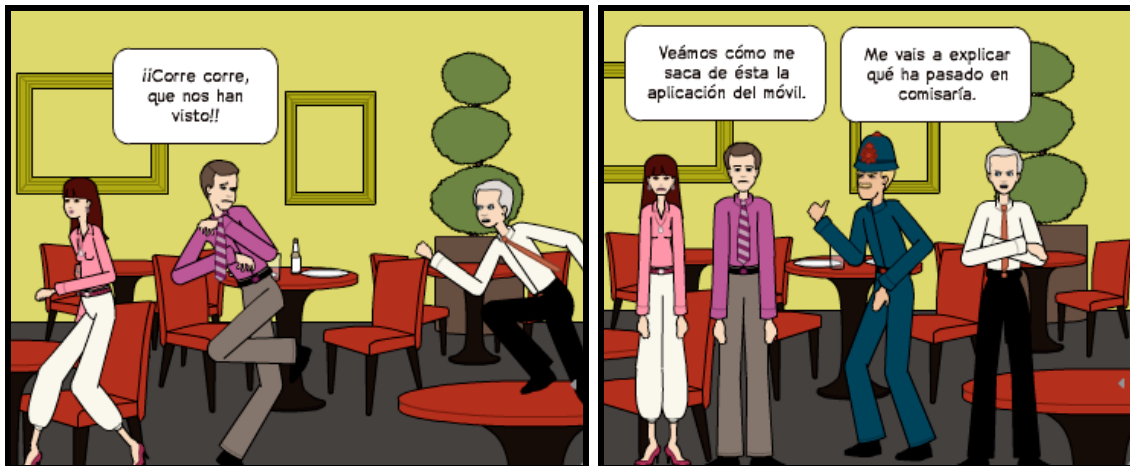


Figura 53. Situación en la que los clientes intentan irse sin pagar y acude la policía.



Figura 54. Traducción de la frase "¿Puedo hablar con un abogado?" dentro de la temática comisaria.

b. Buscando el teatro

En la figura 55 se muestra a una chica que quiere llegar al teatro pero no sabe cómo hacerlo desde donde se encuentra. Debido a que tiene una deficiencia auditiva prefiere preguntarle al niño mediante la aplicación.



Figura 55. Medios de transporte - Teatro

Selecciona la situación “Lugares” dentro de “Uso de Vocabulario” y ahí la frase que le interesa, la cual tiene que completar con la palabra “Teatro” (ver figuras 56 y 57). La chica le da el móvil al niño para que use también la aplicación para indicarle cómo llegar al teatro, tal y como se muestra en la figura 58.



Figura 56. Selección de la frase incompleta “Como llego a el/la” dentro del grupo “Lugares”.



Figura 57. Selección de una de las palabras compatibles junto con su posterior traducción.



Figura 58. Situación en la que se pregunta por el teatro a otra persona utilizando la aplicación.

El chico selecciona la opción “Vocabulario” y, pese a que podía haber buscado la palabra alfabéticamente en la “Vista Clásica”, opta por seleccionar la “Vista Grupos”. Una vez seleccionada esta vista escoge el grupo adecuado, en este caso “Transporte”, selecciona el medio de transporte con el que puede llegar al teatro y le devuelve el móvil a la chica (ver figuras 59 y 60).

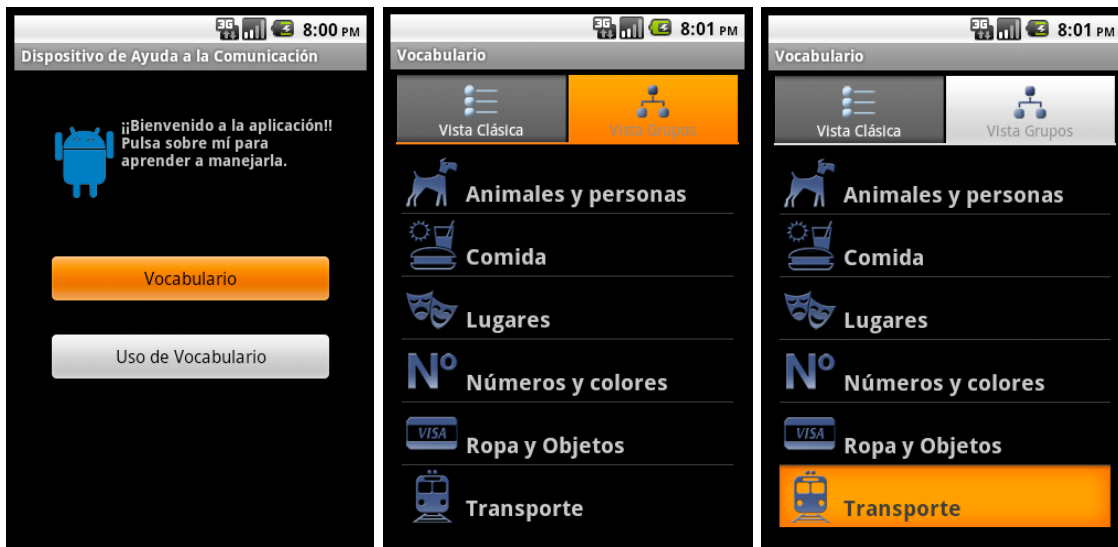


Figura 59. Selección de la temática transporte en respuesta a la pregunta realizada.



Figura 60. Selección del autobús como medio de transporte para ir al teatro.

La chica necesita más información sobre el transporte que debe coger así que selecciona la situación “Transporte” de “Uso de Vocabulario” y ahí selecciona una frase que le de más información (ver figuras 61 y 62).



Figura 61. De nuevo, selección de la temática transporte.

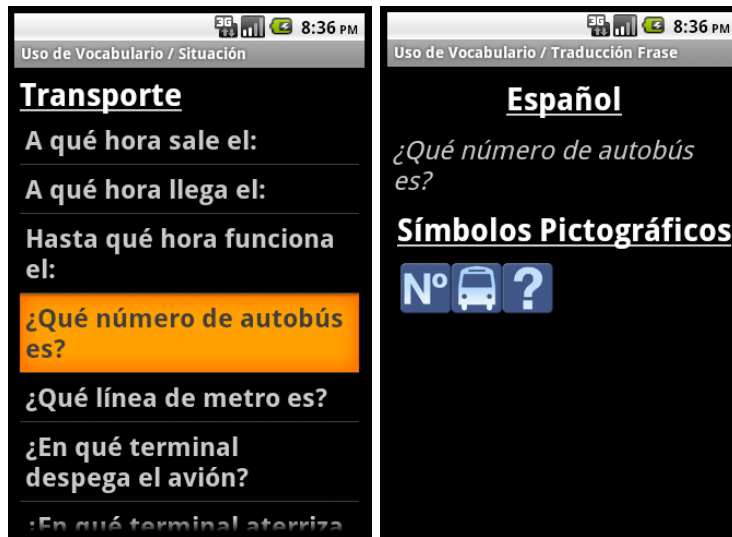


Figura 62. Pregunta sobre el número del autobús que hay que coger para ir al teatro

El niño opta por acompañarla a la parada del autobús (ver figura 63).



Figura 63. Situación en la que una persona acompaña a otra a la parada de autobús apropiada.

Una vez han llegado a la parada (ver figura 64), para que no haya confusión sobre el autobús que debe coger, le indica mediante la aplicación que tiene que ser de la línea azul el número 9. Para ello el niño usa las dos vistas de “Vocabulario”. La “Vista clásica” para seleccionar la palabra “Azul” (ver figura 65) y la “Vista Grupos” para el número “9” (ver figura 66).



Figura 64. Situación en la que se encuentran en la parada.



Figura 65. Indicación del color del autobús utilizando la aplicación.



Figura 66. Indicación del número de autobús.

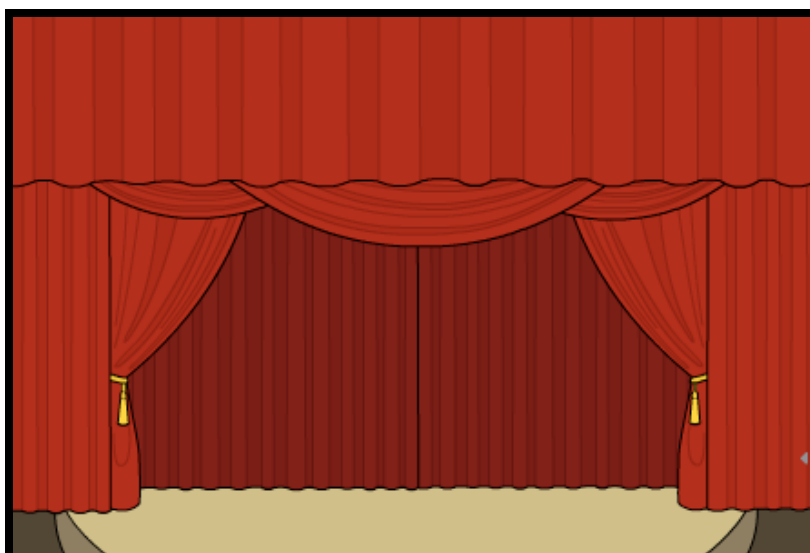


Figura 67. Fin del tutorial.

C. Cuestionarios

A continuación se presentan los cuestionarios que deberán ser rellenados por las personas que se van a encargar de evaluar la aplicación. Estos cuestionarios van a permitir recoger tanto información personal del usuario para obtener el perfil del mismo (se rellenará antes de leer el tutorial) como la evaluación de la aplicación (se rellenará una vez leído el tutorial). Debido a que la evaluación de la aplicación la van a realizar tanto oyentes como discapacitados auditivos, se han realizado dos tipos de formularios siguiendo en cada caso las necesidades de cada grupo.

a. Cuestionario inicial personas sin discapacidad auditiva

SUJETO: Mujer / Hombre

EDAD:

IDIOMA:

1. ¿Estás habituado al uso de ordenadores?

NADA ESPORÁDICAMENTE MUCHO

2. ¿Dispones de conexión a Internet en casa?

SI NS/NC NO

3. Discapacidad auditiva

SI NS/NC NO

4. ¿Te gusta viajar?

SI NS/NC NO

5. ¿Viajas a menudo?

SI NS/NC NO

6. ¿Viajas por tu país de origen o por el extranjero?

ORIGEN EXTRANJERO AMBOS

7. Cuando viajas por el extranjero, ¿te es fácil comunicarte con los demás?

SI NS/NC NO

8. ¿Cómo te has comunicado cuando has viajado fuera?

9. En caso de no haber viajado al extranjero, ¿cómo te comunicarías si te fuera necesario?

10. ¿Te gustaría o te sería de utilidad, un programa para tu móvil, que te ayude a comunicarte con los demás gracias a iconos universales que representan las palabras y frases hechas más comunes que se utilizan para viajar?

SI NS/NC NO

11. ¿Por qué te gustaría? O ¿Por qué no?

12. ¿Qué aspectos crees que son necesarios que la aplicación tuviera?

13. ¿A qué tareas necesitarías que la aplicación te ayudase?

14. ¿Qué contextos crees que serían en los que más necesitarías ayuda?

b. Cuestionario final personas sin discapacidad auditiva

1. Forma en la que el sistema le permite realizar las tareas solicitadas

Confusa 1 2 3 4 5 Muy relevante

2. La navegación a través del sistema resulta

Confusa 1 2 3 4 5 Muy relevante

3. Apariencia general del sistema

Confusa 1 2 3 4 5 Muy relevante

4. Representatividad de los iconos de la aplicación respecto a la función de los mismos

Confusa 1 2 3 4 5 Muy relevante

5. Estructura y organización del sistema

Confusa 1 2 3 4 5 Muy relevante

6. Los nombres y descripciones que aparecen en el sistema son

Confusos 1 2 3 4 5 Muy relevantes

7. En general, ¿te fue fácil realizar las tareas solicitadas?

Confusa 1 2 3 4 5 Muy relevante

8. En general, aprender a utilizar este sistema me ha parecido ...

Difícil 1 2 3 4 5 Fácil

9. Facilidad de uso del programa, en general

Confusa 1 2 3 4 5 Muy relevante

10. Pienso que voy a necesitar a alguien que me ayude a manejar esta aplicación

Seguro que si 1 2 3 4 5 Seguro que no

11. Pienso que con un poco de uso el sistema será muy potente

Seguro que si 1 2 3 4 5 Seguro que no

12. La interacción con el sistema fue

Confusa 1 2 3 4 5 Clara

13. Es fácil encontrar la información que necesito.

Seguro que sí 1 2 3 4 5 Seguro que no

14. La información dada por el sistema es fácil de entender.

Seguro que sí 1 2 3 4 5 Seguro que no

15. La información es efectiva a la hora de ayudarme a comunicarme con los demás.

Seguro que sí 1 2 3 4 5 Seguro que no

16. La organización de la información en la pantalla de este sistema es clara.

Seguro que sí 1 2 3 4 5 Seguro que no

17. La interfaz de este sistema es agradable.

Seguro que sí 1 2 3 4 5 Seguro que no

18. Este sistema tiene todas las funciones y habilidades que esperaba que tuviera.

Seguro que sí 1 2 3 4 5 Seguro que no

19. En general, estoy satisfecho con el sistema.

Seguro que sí 1 2 3 4 5 Seguro que no

Comentarios sobre los aspectos más negativos de la aplicación que hay que mejorar

Comentarios sobre los aspectos más positivos de la aplicación que quieres resaltar

c. Cuestionario inicial personas con discapacidad auditiva

SUJETO: Mujer / Hombre

EDAD:

IDIOMA:

1. ¿Usas ordenador?

- A) NADA
- B) A VECES
- C) MUCHO

2. ¿Tienes Internet en casa?

- A) SÍ
- B) NO SÉ
- C) NO

3. ¿Tienes discapacidad auditiva?

- A) SÍ
- B) NO

4. ¿Te gusta viajar?

- A) SÍ
- B) NO

5. ¿Viajas mucho?

- A) SÍ
- B) NO

6. ¿Viajas por España o por el extranjero?

- A) POR ESPAÑA
- B) POR EL EXTRANJERO
- C) POR ESPAÑA Y POR EL EXTRANJERO

7. Cuando viajas por el extranjero, ¿te comunicas bien con los demás?

- A) SÍ
- B) NO

8. Cuando viajas al extranjero, ¿cómo te comunicas?

- A) CON SIGNOS
- B) VOCALIZANDO Y LEYENDO LOS LABIOS
- C) ESCRIBIENDO
- D) OTROS

9. ¿Te gustaría tener en tu móvil, un programa que te ayude a comunicarte con los demás?

- A) SÍ
- B) NO SÉ
- C) NO

10. Si la respuesta es no, ¿por qué?

- A) PORQUE NO SABRÍA UTILIZARLA
- B) PORQUE NO ME GUSTA O NO SÉ USAR EL MÓVIL
- C) OTRAS

11. Este programa tiene que ser:

- A) FÁCIL
- B) QUE SE ENTIENDA BIEN
- C) QUE LO PUEDA UTILIZAR SIEMPRE QUE LO NECESITE
- D) OTROS

12. ¿En qué situaciones utilizarías este programa?

- A) BANCO
- B) COMISARÍA
- C) VIAJES
- D) COMPRAS
- E) RESTAURANTE
- F) TRANSPORTES
- G) HOSPITAL
- H) EN TODAS LAS SITUACIONES ANTERIORES
- I) EN OTRAS SITUACIONES

d. Cuestionario final personas con discapacidad auditiva

1. ¿Es difícil el uso del programa?

- i. SÍ
- ii. NO

2. ¿Te gustan las imágenes?

- i. SÍ
- ii. NO

3. ¿Te gusta como está organizado el programa?

- i. SÍ
- ii. NO

4. ¿Entiendes las palabras y frases del programa?

- A) SÍ
- B) NO
- C) ALGUNAS

5. ¿Sabrías usar el programa?

- A) SÍ
- B) NO

6. ¿Necesitarías a alguien que te enseñe a usarla?

- A) SÍ
- B) NO

7. ¿Es fácil encontrar la palabra o frase que necesitas?

- i. SÍ
- ii. NO

8. ¿Este programa te ayudaría a comunicarte con los demás?

- i. SÍ
- ii. NO

9. ¿Necesitarías algo más en el programa para comunicarte?

- i. SÍ
- ii. NO

10. ¿Te gusta el programa?

- i. SÍ
- ii. NO

11. ¿La usarías para comunicarte?

- i. SÍ
- ii. NO