



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**AVC  
Asistente Virtual para la  
Comunicación  
Anexos**



Presentado por José Miguel Ramírez Sanz  
en Universidad de Burgos — 22 de junio  
de 2019

Tutores: Dr. José Francisco Díez Pastor y Dr.  
César Represa Pérez



---

# Índice general

---

<b>Índice general</b>	I
<b>Índice de figuras</b>	III
<b>Índice de tablas</b>	V
<b>Apéndice A Plan de Proyecto Software</b>	1
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	2
A.3. Estudio de viabilidad . . . . .	14
<b>Apéndice B Especificación de Requisitos</b>	19
B.1. Introducción . . . . .	19
B.2. Objetivos generales . . . . .	19
B.3. Catalogo de requisitos . . . . .	20
B.4. Especificación de requisitos . . . . .	22
<b>Apéndice C Especificación de diseño</b>	39
C.1. Introducción . . . . .	39
C.2. Diseño de datos . . . . .	40
C.3. Diseño procedimental . . . . .	44
C.4. Diseño arquitectónico . . . . .	48
C.5. Diseño de interfaces . . . . .	49
C.6. Diseño de código de errores . . . . .	53
<b>Apéndice D Documentación técnica de programación</b>	55
D.1. Introducción . . . . .	55

D.2. Estructura de directorios . . . . .	55
D.3. Manual del programador . . . . .	57
D.4. Compilación, instalación y ejecución del proyecto . . . . .	60
D.5. Pruebas del sistema . . . . .	62
<b>Apéndice E Documentación de usuario</b>	<b>65</b>
E.1. Introducción . . . . .	65
E.2. Aplicación de grabación prototipo . . . . .	66
E.3. Aplicación para la recogida de datos . . . . .	71
E.4. Aplicación para la interpretación . . . . .	85
<b>Bibliografía</b>	<b>99</b>

---

# Índice de figuras

---

A.1.	Gráfico del desarrollo del <i>sprint</i> 2 . . . . .	3
A.2.	Gráfico del desarrollo del <i>sprint</i> 3 . . . . .	4
A.3.	Gráfico del desarrollo del <i>sprint</i> 4 . . . . .	6
A.4.	Gráfico del desarrollo del <i>sprint</i> 5 . . . . .	7
A.5.	Gráfico del desarrollo del <i>sprint</i> 6 . . . . .	8
A.6.	Gráfico del desarrollo del <i>sprint</i> 7 . . . . .	9
A.7.	Gráfico del desarrollo del <i>sprint</i> 8 . . . . .	10
A.8.	Gráfico del desarrollo del <i>sprint</i> 9 . . . . .	11
A.9.	Gráfico del desarrollo del <i>sprint</i> 10 . . . . .	12
A.10.	Gráfico del desarrollo del <i>sprint</i> 11 . . . . .	13
B.1.	Caso de Uso 1 . . . . .	23
B.2.	Casos de Uso 2, 3, 4, 5 y 6 . . . . .	24
C.1.	Diagrama de clases . . . . .	42
C.2.	Diagrama de Clases de Alto Nivel . . . . .	43
C.3.	Diagrama de Secuencia aplicación generación de datos . . . . .	45
C.4.	Diagrama de Secuencia de Información . . . . .	46
C.5.	Diagrama de Secuencia del cambio de las opciones . . . . .	46
C.6.	Diagrama de Secuencia de la interpretación de una emoción/estado . . . . .	47
C.7.	Diagrama de Secuencia de la interpretación de una respuesta . . . . .	48
C.8.	Diagrama de Secuencia de Alto Nivel . . . . .	49
C.9.	Diseño interfaz aplicación recogida de datos . . . . .	50
C.10.	Interfaz final de la aplicación recogida de datos . . . . .	50
C.11.	Diseño interfaz aplicación interpretación . . . . .	52
C.12.	Interfaz final de la aplicación AVC . . . . .	53
C.13.	Código de errores . . . . .	54

D.1. Selección de fichero o proyecto en <i>AndroidStudio</i> . . . . .	58
D.2. Estructura de ficheros generada por <i>AndroidStudio</i> . . . . .	58
D.3. Interfaz de <i>noWifi</i> . . . . .	61
D.4. Comando cd a la carpeta <i>/server/final</i> . . . . .	62
D.5. Comando <i>Python</i> para desplegar el servidor. . . . .	62
D.6. Consola del servidor en ejecución. . . . .	63
D.7. Salida de los test unitarios y de integración. . . . .	64
E.1. Logotipo y nombre de la aplicación una vez instalada. . . . .	67
E.2. Petición de permisos para la grabación de audio. . . . .	68
E.3. Petición de permisos para el acceso al almacenamiento multimedia. . . . .	69
E.4. Menú principal de la aplicación prototipo. . . . .	70
E.5. Logotipo y nombre de la aplicación de generación de datos. . . . .	72
E.6. Petición de permisos para la grabación de audio en la aplicación de recogida de datos. . . . .	73
E.7. Petición de permisos para el acceso al almacenamiento multimedia en la aplicación de recogida de datos. . . . .	74
E.8. Menú principal de la aplicación para la recogida de datos. . . . .	75
E.9. Paciente seleccionado, <i>Grabar Audio</i> habilitado. . . . .	76
E.10. Pantalla de Grabación de Audios. . . . .	77
E.11. Pulsamos el botón <i>Grabar</i> . . . . .	78
E.12. Pulsamos el botón <i>Parar</i> . . . . .	79
E.13. Menú principal tras pulsar el botón <i>Seleccionar</i> . . . . .	80
E.14. Pantalla para seleccionar las opciones adicionales. . . . .	81
E.15. Menú principal tras pulsar el botón <i>Seleccionar</i> en la pantalla de opciones. . . . .	82
E.16. Pantalla de selección de emociones o respuesta. . . . .	83
E.17. Mensaje informativo con el error. . . . .	84
E.18. Mensaje de error al intentar enviar sin conexión a Internet. . . . .	85
E.19. Logotipo y nombre de la aplicación de interpretación, AVC. . . . .	86
E.20. Permisos de grabación de audio en AVC. . . . .	87
E.21. Permisos para el acceso a multimedia en AVC. . . . .	88
E.22. Menú principal de la aplicación AVC. . . . .	89
E.23. Pantalla con las opciones cargadas del paciente seleccionado. . . . .	90
E.24. Botón <i>Guardar</i> habilitado. . . . .	91
E.25. Pantalla de selección del tipo de interpretación. . . . .	92
E.26. Pantalla de grabación del audio. . . . .	93
E.27. Empezamos a grabar, tras pulsar <i>Grabar</i> . . . . .	94
E.28. Paramos la grabación, tras pulsar <i>Parar</i> . . . . .	95
E.29. Ejemplo de la pantalla con el resultado de una interpretación. . . . .	96
E.30. Ejemplo de diálogo informativo. . . . .	97

---

# Índice de tablas

---

A.1. Salario trabajador . . . . .	15
A.2. Coste de personal . . . . .	15
A.3. Coste hardware . . . . .	16
A.4. Coste del viaje a Madrid. . . . .	16
A.5. Coste total . . . . .	16
A.6. Licencias de las librerías. . . . .	17
B.1. Caso de uso 1: Generar un dato. . . . .	25
B.2. Caso de uso 1.1: Seleccionar un paciente. . . . .	26
B.3. Caso de uso 1.2: Grabar audio. . . . .	27
B.4. Caso de uso 1.3: Reproducir el audio. . . . .	28
B.5. Caso de uso 1.4: Seleccionar las opciones adicionales. . . . .	28
B.6. Caso de uso 1.5: Seleccionar las emociones o respuestas. . . . .	29
B.7. Caso de uso 2: Seleccionar paciente. . . . .	30
B.8. Caso de uso 3: Obtener información sobre el proyecto. . . . .	31
B.9. Caso de uso 4: Gestionar las opciones adicionales. . . . .	32
B.10.Caso de uso 4.1: Mostrar las opciones. . . . .	33
B.11.Caso de uso 4.2: Modificar y guardar las opciones. . . . .	34
B.12.Caso de uso 5: Interpretar emoción o respuesta. . . . .	35
B.13.Caso de uso 5.1: Grabar audio. . . . .	36
B.14.Caso de uso 5.2: Visualizar y escuchar el resultado. . . . .	37
B.15.Caso de uso 6: Información en la ejecución. . . . .	38



## *Apéndice A*

---

# **Plan de Proyecto Software**

---

## **A.1. Introducción**

La planificación de un proyecto es esencial para su éxito, ya que nos permite ver cuales van a ser las necesidades, los impedimentos y otra serie de eventos o circunstancias relevantes en el desarrollo del proyecto. La planificación de un proyecto software tiene diversas partes que tratan temas como la planificación temporal o el uso de licencias.

La planificación de este proyecto se ha dividido en dos secciones:

- **Planificación temporal:** Sección donde se puede ver la distribución temporal del proyecto, dividido en *sprints*. Cada *sprint* está formado por una serie de tareas relacionadas con una estimación individual de tiempo.
- **Estudio de viabilidad:** Sección orientada a la viabilidad del proyecto, es decir, cuando favorable es su realización, en aspectos económicos y legales.
  - **Viabilidad económica:** En esta subsección comentaré los gastos que ha tenido el desarrollo del proyecto.
  - **Viabilidad legal:** Subsección en la que expondré las distintas licencias de las librerías que han sido necesarias para el desarrollo del proyecto y la licencia final del producto.

En este proyecto he seguido, además, la metodología *SCRUM* gracias a la *GitHub* junto con *ZenHub*, pudiendo así hacer una planificación correcta y desarrollo incremental.

## A.2. Planificación temporal

En esta sección voy a comentar el desarrollo temporal del proyecto divido en distintos *sprints*, por cada uno de ellos voy a comentar:

- Fecha de inicio y de cierre del *sprint*.
- Pequeña descripción de lo que se pretendía hacer en el *sprint*.
- Lista agrupando y resumiendo las tareas correspondientes.
- Gráfico *Burndown* que nos permite ver el desarrollo del *sprint*, estos gráficos en los primeros *sprints* no se ve un progreso claro, debido a que no se ha bien el gráfico al haber estado el repositorio en modo privado, hasta el *sprint* 11.
- Breve comentario sobre el desarrollo del *sprint*.

### Sprint 1

Este *sprint* se inició el 4 de Diciembre de 2018 y se cerró el 22 de Febrero de 2019.

En este *sprint*, que al final por como sucedieron las cosas se alargó al no saber cuando podría cerrarlo, se orientó a la creación del repositorio en Git, a la creación del documento donde poder apuntar los aspectos más relevantes de la investigación junto con el investigador colaborador, Sergio Chico, también se quería empezar con las tareas de investigación y las primeras reuniones con APACE Burgos.

La lista resumen de las tareas que se realizaron en este *sprint* 1 son:

- Investigación de la clasificación de audios.
- Aprender a desarrollar una aplicación de *Android*.
- Aprender a desarrollar una aplicación *Android* que permita grabar [5].
- Implementación del primer prototipo de grabación de audios.
- Investigación y documentación del estudio sobre el audio en *Android* y *MediaRecorder*.
- Continuas reuniones y visitas con APACE Burgos para determinar los objetivos y funcionalidades del proyecto.



Figura A.1: Gráfico del desarrollo del *sprint* 2.

Al ser el primer *sprint* es el más desordenado, tuve en su momento un problema, ya que no sabía exactamente cuando parar y cerrar el *sprint*. Aun así, fue un *sprint* importante donde se definieron los primeros objetivos y funcionalidades, y sirvió para conocer que querían desde APACE.

## Sprint 2

Este *sprint* comenzó el 22 de Febrero de 2019 y terminó el 25 de Febrero de 2019.

En este *sprint* se quería diseñar la interfaz de la aplicación de generación de datos, además de crear las pantallas a partir de la interfaz diseñada y crear su estructura.

La lista resumen de las tareas del *sprint* 2 es:

- Aprender a crear varias pantallas en una aplicación *Android*.
- Diseño con *Pencil* de la interfaz de la aplicación para generar los datos a partir de la información de las reuniones.
- Crear las pantallas según el diseño.



Figura A.2: Gráfico del desarrollo del *sprint* 3.

En este *sprint* A.1 se cometió el error contrario al anterior, este *sprint* quizás sea demasiado corto, aun así se siguió correctamente la planificación inicial.

## Sprint 3

Este *sprint* empezó el 25 de Febrero de 2019 y acabó el 7 de Marzo de 2019.

En este *sprint* se quería implementar las funcionalidades de las diferentes pantallas, además de implementar la navegabilidad entre estas.

Las tareas que se realizaron en este *sprint* se resumen en:

- Crear las funcionalidades de las distintas pantallas.
- Implementar la navegabilidad entre pantallas.
- Corrección de *bugs*.
- Implementar la generación de comprimidos.
- Comentar el código.

En este *sprint A.2*, como en el anterior, se consigue correctamente los objetivos en el tiempo que se quería. Esto era esencial, ya que necesitábamos tener la aplicación que nos permitía generar datos lo antes posible, para contar con el mayor número de ellos al final del proyecto.

## Sprint 4

Este *sprint* comenzó el 7 de Marzo de 2019 y terminó el 18 de Marzo de 2019.

En este *sprint* se quería implementar el método de envío de los comprimidos desde la aplicación según se definiese en la reunión. Y complementar la aplicación con las emociones y las opciones adicionales que nos tenía que facilitar APACE.

Las tareas que se realizaron en este *sprint 4* son:

- Resumir las opciones adicionales facilitadas por APACE.
- Modificación de las pantallas de Estado y de Opciones con los nuevos valores.
- Implementación del envío de los comprimidos por correo.
- Creación del manual de usuario de la aplicación.
- Crear la presentación en *Power Point* para enseñar la aplicación.
- Presentación ante familias y cuidadores de la aplicación.

En este *sprint 4 A.3* se ha realizado más tareas de las que se pretendía en un principio, ya que no se contaba con tener que hacer la presentación en APACE para mostrar y ayudar a instalar la aplicación.

## Sprint 5

Este *sprint* comenzó el 18 de Marzo de 2019 y acabó el 5 de Abril de 2019.

En este *sprint* se pretendía realizar el diseño de la interfaz de la aplicación de interpretación, y generar mis propios datos con la aplicación de generación de datos, ya que como se comentó en la reunión, lo más seguro es que no tuviésemos suficientes datos.

Las tareas del *sprint 5* se pueden resumir en:



Figura A.3: Gráfico del desarrollo del *sprint* 4.

- Presentación de la aplicación de generación de datos a las familias que no pudieron estar en la anterior presentación.
- Diseño de las posibles interfaces de la aplicación de interpretación.
- Diseño de la aplicación.
- Elección junto con APACE de la interfaz final.
- Generación de mis propios datos.

Este *sprint* 5 A.4 costó mucho, ya que la grabación de audios se hizo muy larga, ya que debía de ser precisa. Aun así el *sprint* transcurrió correctamente.

## Sprint 6

Este *sprint* comenzó el 5 de Abril de 2019 y terminó el 11 de Abril de 2019.

En este *sprint* se quería, a partir de los diseños hechos en el *sprint* anterior, crear e implementar las funcionalidades de la aplicación de interpretación.

Las tareas que se realizaron en este *sprint* son:



Figura A.4: Gráfico del desarrollo del *sprint* 5.

- Crear las pantallas y las clases diseñadas.
- Implementar las funcionalidades de las clases.
- Implementar la navegabilidad entre pantallas.
- Modificar las imágenes pasadas por APACE para la accesibilidad.
- Implementar el paciente por defecto.

En el *sprint* 6 A.5 se ha seguido correctamente la idea inicial de los objetivos marcados para este.

## Sprint 7

Este *sprint* va desde el día 11 de Abril de 2019 hasta el día 28 de Abril de 2019.

En el séptimo *sprint* se quería diseñar e implementar diferentes tipos de test que permitan probar las distintas partes de la aplicación, y corregir los *bugs* resultantes.

El resumen de las tareas de este *sprint* es:



Figura A.5: Gráfico del desarrollo del *sprint* 6.

- Diseño de los test unitarios.
- Implementación de los test unitarios.
- Diseño de los test de integración.
- Implementar los test de integración.
- Realizar el *Monkey test*.
- Comentar el código de los test.
- Corrección de los *bugs* detectados.

Este ha sido uno de los *sprints* más costosos y a la vez más favorables para el proyecto A.6, ya que aunque llevó mucho tiempo diseñar e implementar todos los test dio como resultado encontrar algunos *bugs* que no podría haber detectado de otra manera.

## Sprint 8

Este *sprint* se desarrolló desde el 28 de Abril de 2019 hasta el 10 de Mayo de 2019.



Figura A.6: Gráfico del desarrollo del *sprint* 7.

En este *sprint* se quería modificar las cosas que nos comentó APACE sobre la interfaz, añadir la pantalla de información y sus test correspondientes, y por último comenzar con el diseño del servidor.

Las tareas que finalmente se hicieron fueron:

- Cambios en la interfaz pedidos por APACE.
- Implementación de la pantalla de información.
- Cambios en los diseño, para añadir la nueva información.
- Diseño de los test para la pantalla de información.
- Implementación de los test de la pantalla de información.
- Diseñar el servidor.
- Implementar el servidor.
- Probar la implementación del servidor.

En este *sprint* 8 A.7 al final se realizaron todas la tareas que se pretendían y además otras más, sobre todo las tareas de implementación del servidor, que en un principio iba a ser el siguiente *sprint*, pero debido a la necesidad de terminar el servidor lo antes posible se adelanto a este.



Figura A.7: Gráfico del desarrollo del *sprint* 8.

## Sprint 9

Este *sprint* comenzó el 10 de Mayo de 2019 y terminó el 17 de Mayo de 2019.

En este *sprint*, que inicialmente se iba a dirigir a desarrollar el servidor *Flask*, se pretendía acabar la implementación de la aplicación desarrollando los métodos *post* al servidor, y crear la presentación y el guión para la presentación del proyecto en conferencia ASPACENet en Madrid.

El conjunto de tareas que se realizaron fueron:

- Estudiar como hacer los métodos *post* en *Android*.
- Investigar como poder conectarse directamente al servidor.
- Implementar la conexión entre la aplicación y el servidor.
- Comentar el código nuevo.

En este *sprint* A.8, al contrario que en el anterior, no se ha podido realizar todos los objetivos que se tenían, ya que las tareas relacionadas con la creación de la presentación y el guión de Madrid no se han podido hacer.



Figura A.8: Gráfico del desarrollo del *sprint* 9.

## Sprint 10

Este *sprint* comenzó el 17 de Mayo de 2019 y terminó el 11 de Junio de 2019.

En este *sprint*, que quería ser el último en el cual hubiese que implementar cosas tanto en las aplicaciones como en el servidor, se quería implementar la descarga del audio en el servidor, cosa que costó mucho, y se quería añadir la parte de clasificación en el servidor con los algoritmos estudiados en los primeros *sprints* y los estudios realizados por el investigador colaborador, Sergio Chico.

La lista resumen de las tareas de este *sprint* es:

- Desarrollo del paso del audio entre aplicación y servidor con Base64.
- Creación dinámica de nombres en el servidor y eliminación al final de los audios.
- Mejorar la visualización del resultado final.
- Elaboración del método que permite devolver un resultado aleatorio.
- Modificación de los test de la aplicación.



Figura A.9: Gráfico del desarrollo del *sprint* 10.

- Corrección de *bugs*.
- Creación de la presentación y el guión para Madrid.
- Presentación del proyecto en Madrid.
- Añadir el sistema de predicción en el servidor.
- Crear e implementar el estándar de errores.
- Refactorización de código.
- Grabación del vídeo en APACE.
- Presentación de la aplicación de interpretación a los cuidadores de APACE.

Este *sprint* 10 A.9 es sin duda el que más trabajo y problemas ha dado, ya que en él se encuentran varias tareas que han llevado mucho tiempo realizarlas, como por ejemplo la presentación del proyecto en Madrid o el paso del audio al servidor. Aun así, se consiguieron todos los objetivos, sobre todo el objetivo de acabar la implementación de las aplicaciones y del servidor.



Figura A.10: Gráfico del desarrollo del *sprint* 11.

## Sprint 11

Este *sprint* comenzó el 11 de Junio de 2019 y terminó el 18 de Junio de 2019.

En este *sprint* se pretendía presentar el proyecto ante la prensa y empezar la documentación de la memoria.

Las tareas que se realizaron en el *sprint* fueron:

- Presentación del proyecto ante la prensa.
- Documentación de la memoria.

El *sprint* 11 A.10 ha sido uno de los más importantes, ya que se ha cumplido, y con creces, uno de los objetivos que se tenían en el proyecto, este objetivo es la difusión mediática. Ya que con la presentación ante la prensa se obtuvo mucha difusión del proyecto.

## Sprint 12

### ACABAR

Como se puede observar el proyecto, por mi parte, ha tenido más de 400 horas de trabajo de índole muy distinta. Pero cabe destacar que esa no

ha sido el único tiempo invertido en el proyecto, sino que hay que añadir el tiempo de investigación de Sergio Chico y el tiempo invertido en todo el proyecto por mis dos tutores, el doctor César Represa y el doctor José Francisco Díez.

### A.3. Estudio de viabilidad

En este apartado se quiere comprobar si la realización del proyecto es posible, teniendo en cuenta los aspectos económicos y legales del mismo.

#### Viabilidad económica

En este subapartado voy a comentar la viabilidad económica del proyecto, en el cual tenemos que calcular tanto los costes de contratación de las 4 personas del equipo de desarrollo, como los gastos en dispositivos, como los gastos para las presentaciones del proyecto.

##### Coste de personal

El personal que ha trabajado en este proyecto es muy variado, y así hay que tenerlo en cuenta en este apartado. Además, hay que tener en cuenta la duración del proyecto que ha ido desde inicio de Diciembre de 2018 hasta principio de Julio de 2019, un total de 7 meses.

Primero empezaremos con los tutores, donde el coste no lo vamos a calcular por la duración del proyecto sino por los créditos impartidos. En primer lugar tenemos al doctor César Represa Pérez, que es profesor colaborador y por otro lado tenemos al doctor José Francisco Díez Pastor. Al no encontrar datos para los profesores colaboradores se han tomado los valores de de doctor.

El sueldo de un doctor en la Universidad de Burgos es de 2360,12 euros al mes [2], que daría un total de 28321,44 euros al año. Teniendo en cuenta que un doctor tiene que impartir 32 créditos, aunque luego se suele impartir más, el coste por crédito sería de  $28321,44\text{€}/32$  créditos, lo que da un total de 885,04€/crédito. Teniendo en cuenta que el Trabajo Fin de Grado es un total de 12 créditos, al ser dos tutores serían 6 créditos por tutor. Lo que darías un coste total por tutor de 5310,27€.

Después de calcular el coste de los tutores, paso a calcular el coste de mi compañero investigador y el mio, que van a ser el mismo sueldo por el

trabajo de 7 meses. El sueldo mensual neto lo he obtenido del sueldo medio de los empleados en informática [3].

Concepto	Coste
Salario mensual neto	1575€
Retención IRPF (15 %)	428,76€
Seguridad Social (29,9 %) [4]	854,67€
Salario mensual bruto	2858,43€
<b>Total 7 meses</b>	<b>20009,01€</b>

Tabla A.1: Salario trabajador.

El total de los costes por personal es de:

Concepto	Coste
Tutores (x2)	10620,54€
Trabajadores (x2)	40018,02€
<b>Total</b>	<b>50638,56€</b>

Tabla A.2: Coste de personal.

### Coste hardware

En este apartado voy a calcular el coste de los dispositivos hardware que se han usado a lo largo del desarrollo del proyecto junto con su amortización, como la mayoría de los dispositivos hardware, vamos a calcular una amortización de 5 años. En cuanto al coste software no tenemos ningún gasto.

Para el desarrollo he utilizado al menos un dispositivo móvil y dos tablets, y tambiénuento con la tablet que compraron desde APACE para el proyecto. Además, he usado dos portátiles, perouento con el uso de 3 portátiles contando con el ordenador del investigador colaborador.

Las amortizaciones se han calculado por el tiempo usado, es decir, hemos usado todos los dispositivos un total de 7 meses.

Concepto	Coste	Amortización
Dispositivo móvil	300€	35€
Dispositivo tablet (x2)	700€	81,67€
Dispositivo tablet APACE	350€	40,83€
Ordenadores (x3)	2400€	280€
<b>Total</b>	<b>3750€</b>	<b>437,5€</b>

Tabla A.3: Coste hardware.

### Otros gastos

Como ya he comentado, durante el proyecto hemos tenido otro tipo de gastos para realizar las presentaciones, sobre todo en la presentación de Madrid al cual fuimos 3 personas, estos gastos son:

Concepto	Coste
Desplazamiento a Madrid	40€
Dietas (x3)	36€
<b>Total</b>	<b>76€</b>

Tabla A.4: Coste del viaje a Madrid.

### Coste Total

Una vez hemos desglosado todos los costes del proyecto voy a calcular el coste total de este:

Concepto	Coste
Coste de personal	50628,56€
Coste hardware	3750€
Otros costes	76€
<b>Total</b>	<b>54454,56€</b>

Tabla A.5: Coste total.

## Viabilidad legal

En este apartado voy a comentar la viabilidad legal de las licencias de las librerías usadas en el proyecto y la licencia final del proyecto.

Desde el comienzo del proyecto le comunicamos tanto a APACE como a Vodafone que necesitábamos saber la licencia que querían, tanto para el transcurso del desarrollo saber que herramientas podíamos usar y obviamente en esta documentación saber que licencia tener. Y aunque se lo preguntásemos en cada una de las reuniones y tras haber enviado varios correos electrónicos a Vodafone, aun no sabemos la licencia que quieren, solo sabemos que quizás sea *open source*. Esto también a afectado al repositorio en *GitHub*, ya que desde un principio lo he tenido que poner en privado, al no saber la licencia del código, pero actualmente lo he modificado a público tras la no respuesta por parte de APACE o Vodafone y por la necesidad de sacar los gráficos del transcurso del desarrollo.

Después de comentar esto, las licencias de las librerías que usamos son:

Librería	Licencia
Android Support Library	Apache 2.0
OpenCSV	Apache 2.0
JavaMail	GPL v2
JUnit	EPL
Espresso	Apache 2.0
Flask	BSD 3 New
Pandas	BSD 3 New
Numpy	BSD 3 New
Scikit-Learn	BSD 3 New
Matplotlib	PSFL

Tabla A.6: Licencias de las librerías.

Además, hemos usado los pictogramas de la pagina de *ARASAAC* [1], que tiene un licencia CC BY-NC-SA 3.0 ES. Por lo que las imágenes que he tenido que modificar tienen esta misma licencia.

El uso de una licencia EPL, que nos obligaría a descargar las librerías con esta licencia en tiempo de ejecución, ya que no nos permite distribuirlas. Pero, en nuestro caso, la única librería que usa la licencia EPL es *JUnit*, una de las librerías que se usan para probar el código. Como esta no se distribuye junto con el código, no nos obliga a descargarla en tiempo de ejecución.

Tras comprobar las licencias de las librerías que usamos en el proyecto, y ante la negativa de APACE o Vodafone de darnos una respuesta a la pregunta de qué licencia usar, he decidido que la licencia del proyecto sea GPL v3, ya que es la licencia más restrictiva de las herramientas que tengo y me permite en un futuro patentar el software creado.

## *Apéndice B*

---

# **Especificación de Requisitos**

---

## **B.1. Introducción**

En este anexo se voy a comentar un apartado esencial en el desarrollo de un elemento software, esto es los requisitos funcionales y los casos de uso, ya que representan lo que tiene que ser y/o hacer nuestra aplicación, programa o proyecto en general. Estos conceptos son definidos por el usuario con ayuda de los diseñadores del software, para definir, desde el primer momento, lo que tiene que hacer el software a desarrollar.

## **B.2. Objetivos generales**

Los objetivos generales, expuestos y debatidos en las primeras reuniones con APACE, son los siguientes:

- Poder capturar datos con los que poder realizar la investigación.
- Interpretar las emociones de los pacientes a partir de grabaciones de audio y unas opciones adicionales.
- Interpretar las respuestas de los pacientes a partir de grabaciones de audio.
- Toda aplicación desarrollada ha de ser simple y accesible.

### B.3. Catalogo de requisitos

En esta sección voy a comentar los requisitos funcionales y no funcionales del proyecto.

#### Requisitos funcionales

- **RF-1 Selección del paciente:** las aplicaciones deben permitir la selección de los pacientes permitidos.
  - **RF-1.1 Visualización de los pacientes:** en las aplicaciones se tiene que poder ver la lista de posible pacientes a seleccionar.
  - **RF-1.2 Paciente seleccionado único:** solo se puede seleccionar a un paciente a la vez.
  - **RF-1.3 Permanencia del paciente:** en las aplicaciones una vez que se ha seleccionado un paciente este sigue a lo largo de la ejecución, sin variar por ningún tipo de error.
  - **RF-1.4 Paciente por defecto:** en la aplicación de interpretación tiene que poder cargarse un paciente por defecto que aparezca seleccionado al abrir la aplicación.
- **RF-2 Selección de emociones/respuestas:** en la aplicación de recogida de datos, el usuario tiene que poder seleccionar las emociones o respuesta relacionado con el sonido que acaba de grabar.
  - **RF-2.1 Selección de emociones o respuesta:** en la aplicación de generación de datos, el usuario solo puede elegir como resultado de la grabación varias emociones o una respuesta, no puede elegir valores de ambos campos.
  - **RF-2.2 Selección de sí o no:** en la aplicación de generación de datos, si elegimos como resultado una respuesta esta ha de ser sí o no, pero no se puede seleccionar ambas a la vez.
  - **RF-2.3 Selección obligatoria de emociones o respuesta:** en la aplciación de generación de datos, cada grabación tiene que tener asociado al menos una emoción o una respuesta, no se puede enviar una grabación sin resultado.
  - **RF-2.4 Carga de datos:** en la aplicación de generación de datos en una misma ejecución sin enviar ni cancelar la ejecución, si hemos seleccionado ya las emociones o la respuesta relacionada,

si volvemos a entrar a la pantalla de selección de estas, han de estar cargadas.

- **RF-3 Selección de las opciones adicionales:** el usuario tiene que poder modificar las opciones adicionales del paciente seleccionado.
  - **RF-3.1 Carga de valores:** en la aplicación de generación de datos, el usuario podrá volver a ver las selecciones de las opciones que ha elegido si vuelve a entrar en la pantalla.
  - **RF-3.2 Persistencia de las opciones:** en la aplicación de interpretación las opciones de cada cliente se almacenan y se modifican en el servidor, siendo común a todas las ejecuciones.
- **RF-4 Grabación del audio:** el usuario ha de poder grabar los audios.
  - **RF-4.1 Carga del audio grabado:** en la aplicación de generación de datos, un usuario que ya haya grabado una audio lo tendrá cargado en la pantalla de grabación del audio.
  - **RF-4.2 Reproducción del audio:** el usuario tiene que poder reproducir el último audio grabado.
  - **RF-4.3 Reescritura del audio:** el usuario ha de poder volver a grabar el audio, eliminando el anterior.
  - **RF-4.4 Reproducción del audio relacionado con el resultado:** el usuario quiere poder escuchar el audio relacionado con el resultado mostrado.
- **RF-5 Subida de los datos:** el usuario quiere poder subir los datos que han generado, para su posterior investigación.
- **RF-6 Interpretación:** el usuario quiere obtener un resultado para el audio que ha grabado.
  - **RF-6.1 Interpretación con paciente entrenado:** el resultado de una interpretación de un cliente, del cual se tiene los modelos de clasificación, ha de ser calculado por estos.
  - **RF-6.2 Interpretación con pacientes no entrenados:** el resultado de una interpretación para un cliente sin modelos ha de ser una respuesta aleatoria.
- **RF-7 Ayuda en la ejecución:** el usuario quiere tener información o una ayuda durante la ejecución, para saber que hace cada botón.

- **RF-7.1 Ayuda con textos en lectura fácil:** cada botón de la aplicación de interpretación ha de tener un botón de ayuda que abra un diálogo donde se puede leer con texto en lectura fácil lo que hace.
  - **RF-7.2 Reproducción de los diálogos:** el usuario quiere que se reproduzcan los textos de los diálogos.
- **RF-8 Información sobre el proyecto:** el usuario quiere ver en la aplicación una pantalla con información sobre el proyecto.

## Requisitos no funcionales

- **RNF-1 Accesibilidad:** la aplicación de interpretación ha de ser lo más accesible posible para que pueda ser usada por el mayor número de personas.
- **RNF-2 Usabilidad:** las aplicaciones han de ser sencillas e intuitivas.
- **RNF-3 Escalabilidad:** el sistema que engloba las aplicaciones y el servidor ha de poder, de forma sencilla, añadir y eliminar pacientes.
- **RNF-4 Disponibilidad:** el servidor ha de ser siempre accesible para poder cargar los pacientes, ver y/o modificar las opciones y para poder interpretar los sonidos.
- **RNF-5 Seguridad:** no ha de poder acceder a los métodos del servidor desde fuera de la aplicación de interpretación de sonidos.
- **RNF-6 Persistencia:** tanto la lista de posibles pacientes, como las opciones de estos y sus modelos de clasificación deben de permanecer accesibles, aunque ocurran problemas de conexión con el servidor.

## B.4. Especificación de requisitos

Este apartado se va explicar los casos de uso, a partir de diagramas y de tablas.

## Diagramas de Casos de Uso

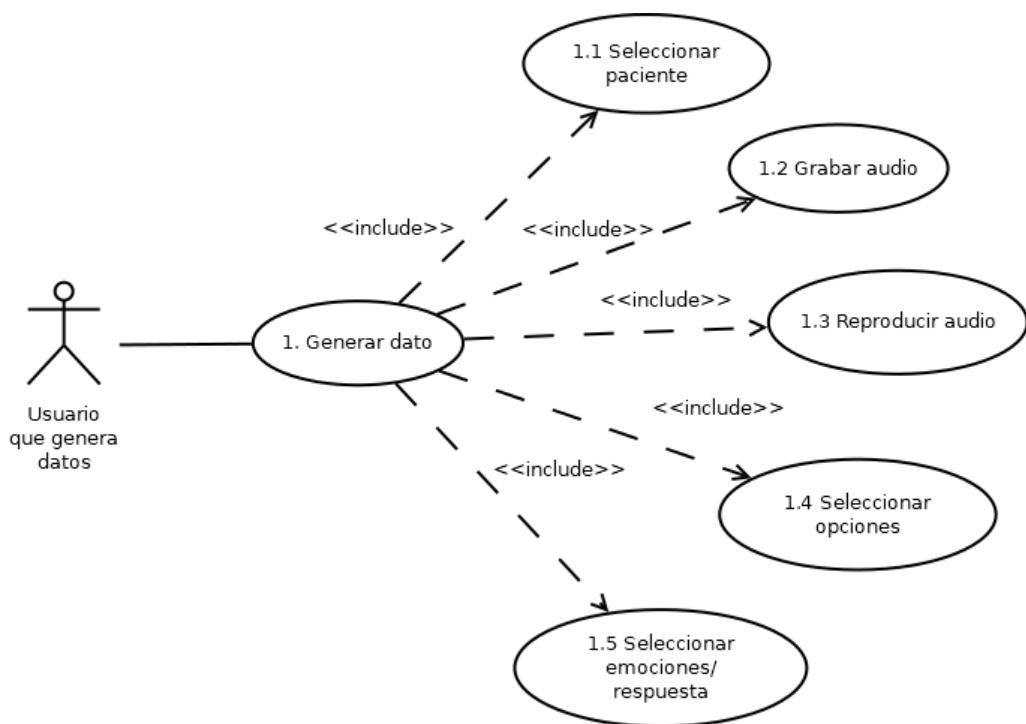


Figura B.1: Caso de Uso 1.

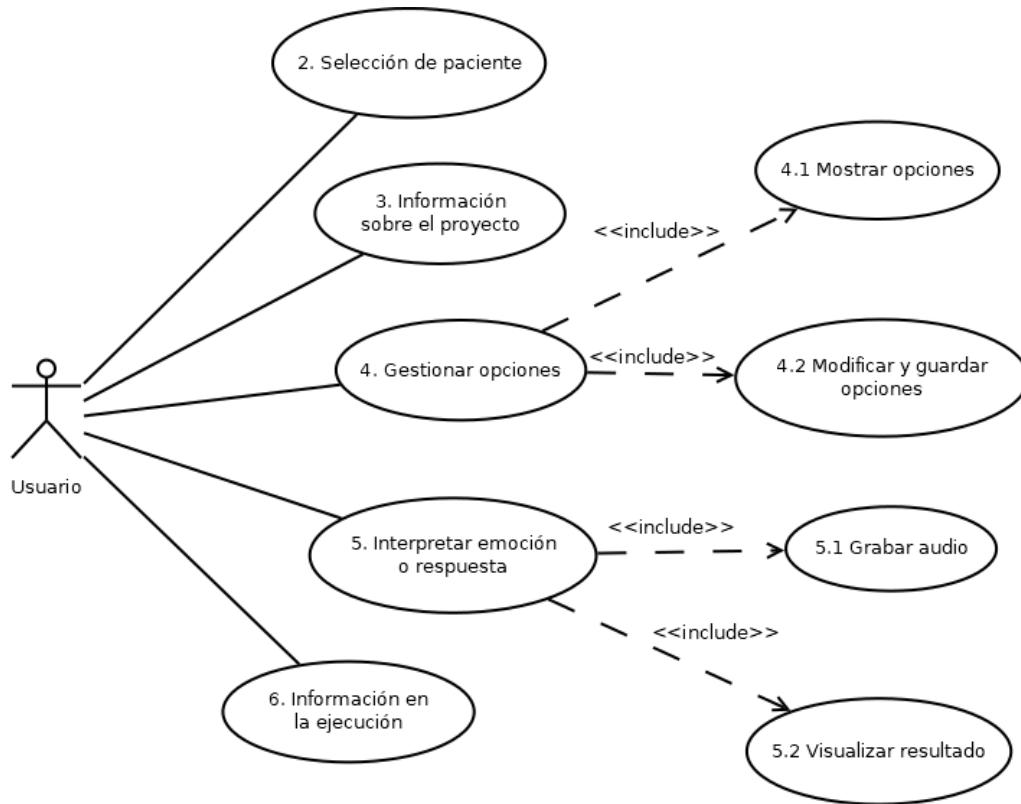


Figura B.2: Casos de Uso 2, 3, 4, 5 y 6.

## Actores

Los actores son las personas que interactúan con los elementos desarrollados y son los que definen los casos de uso de un proyecto. En este desarrollo podemos diferenciar los siguientes actores:

- **Usuario que genera datos:** este usuario es el que interacciona con la aplicación para generar datos. Este usuario será los familiares, amigos y cuidadores más cercanos al paciente, que saben interpretar los sonidos que emite.
- **Usuario:** este usuario es el que interacciona con la aplicación de interpretación, y puede ser cualquier tipo de persona que quiera interpretar a una persona con parálisis cerebral gravemente afectada.

## Casos de uso

<b>CU-1: Generar un dato.</b>		
<b>Descripción</b>	El usuario sube un dato generado.	
<b>Requisitos</b>	RF-5	
<b>Actor</b>	Usuario que genera un dato.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario accede a la aplicación de generación de datos.
	2	El usuario selecciona al paciente.
	3	El usuario rellena el audio, las opciones y las emociones o respuesta.
	4	El usuario sube el dato generado.
<b>Postcondiciones</b>	El correo con el dato se ha enviado correctamente.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4	Si no se tiene conexión a Internet no se envía el dato, pero tampoco se elimina
<b>Frecuencia</b>	Media	
<b>Importancia</b>	Crítico	

Tabla B.1: Caso de uso 1: Generar un dato.

---

**CU-1.1: Seleccionar un paciente.**


---

<b>Descripción</b>	El usuario selecciona y confirma un paciente.										
<b>Precondiciones</b>	El usuario se encuentra en el menú principal de la aplicación de recogida de datos										
<b>Requisitos</b>	RF-1, RF-1.1, RF1.2, RF-1.3.										
<b>Actor</b>	Usuario que genera un dato.										
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>Paso</b></th><th><b>Acción</b></th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario accede a la aplicación de generación de datos.</td></tr> <tr> <td>2</td><td>El usuario despliega el <i>spinner</i> para ver los posibles pacientes.</td></tr> <tr> <td>3</td><td>El usuario selecciona un paciente de entre los posibles.</td></tr> <tr> <td>4</td><td>El usuario confirma el paciente pulsando en el botón <i>SEL..</i></td></tr> </tbody> </table>	<b>Paso</b>	<b>Acción</b>	1	El usuario accede a la aplicación de generación de datos.	2	El usuario despliega el <i>spinner</i> para ver los posibles pacientes.	3	El usuario selecciona un paciente de entre los posibles.	4	El usuario confirma el paciente pulsando en el botón <i>SEL..</i>
<b>Paso</b>	<b>Acción</b>										
1	El usuario accede a la aplicación de generación de datos.										
2	El usuario despliega el <i>spinner</i> para ver los posibles pacientes.										
3	El usuario selecciona un paciente de entre los posibles.										
4	El usuario confirma el paciente pulsando en el botón <i>SEL..</i>										
<b>Postcondiciones</b>	El usuario ha sido seleccionado y confirmado.										
<b>Frecuencia</b>	Alta										
<b>Importancia</b>	Alta										

---

Tabla B.2: Caso de uso 1.1: Seleccionar un paciente.

---

**CU-1.2: Grabar audio.**

---

<b>Descripción</b>	El usuario graba el sonido emitido por un paciente.												
<b>Precondiciones</b>	Hay un paciente seleccionado.												
<b>Requisitos</b>	RF-4, RF-4.1, RF-4.3												
<b>Actor</b>	Usuario que genera un dato.												
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario accede a la aplicación de generación de datos.</td> </tr> <tr> <td>2</td> <td>El usuario selecciona al paciente.</td> </tr> <tr> <td>3</td> <td>El usuario pulsa el botón <i>GRABAR AUDIO</i>.</td> </tr> <tr> <td>4</td> <td>El usuario pulsa el botón <i>GRABAR</i> para comenzar a grabar.</td> </tr> <tr> <td>5</td> <td>El usuario pulsa el botón <i>PARAR</i> para para la grabación.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario accede a la aplicación de generación de datos.	2	El usuario selecciona al paciente.	3	El usuario pulsa el botón <i>GRABAR AUDIO</i> .	4	El usuario pulsa el botón <i>GRABAR</i> para comenzar a grabar.	5	El usuario pulsa el botón <i>PARAR</i> para para la grabación.
Paso	Acción												
1	El usuario accede a la aplicación de generación de datos.												
2	El usuario selecciona al paciente.												
3	El usuario pulsa el botón <i>GRABAR AUDIO</i> .												
4	El usuario pulsa el botón <i>GRABAR</i> para comenzar a grabar.												
5	El usuario pulsa el botón <i>PARAR</i> para para la grabación.												
<b>Postcondiciones</b>	El audio está grabado y puede volver a grabarse pulsando el botón "GRABAR".												
<b>Frecuencia</b>	Alta												
<b>Importancia</b>	Alta												

---

Tabla B.3: Caso de uso 1.2: Grabar audio.

---

**CU-1.3: Reproducir el audio.**

---

<b>Descripción</b>	El último audio grabado es reproducido.				
<b>Precondiciones</b>	Existe un audio grabado, y el usuario se encuentra en la pantalla de grabación de audio.				
<b>Requisitos</b>	RF-4.2				
<b>Actor</b>	Usuario que genera un dato.				
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario pulsa el botón <i>REPRODUCIR</i> y se comienza a reproducir el audio.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario pulsa el botón <i>REPRODUCIR</i> y se comienza a reproducir el audio.
Paso	Acción				
1	El usuario pulsa el botón <i>REPRODUCIR</i> y se comienza a reproducir el audio.				
<b>Postcondiciones</b>	El audio se reproduce, y si se vuelve a pulsar el botón vuelve a comenzar la reproducción.				
<b>Frecuencia</b>	Baja				
<b>Importancia</b>	Media				

---

Tabla B.4: Caso de uso 1.3: Reproducir el audio.

---

**CU-1.4: Seleccionar las opciones adicionales.**

---

<b>Descripción</b>	El usuario quiere seleccionar las opciones relacionadas con el audio grabado.				
<b>Precondiciones</b>	Estamos en la pantalla de selección de opciones y existe un audio grabado.				
<b>Requisitos</b>	RF-3, RF-3.1				
<b>Actor</b>	Usuario que genera un dato.				
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario selecciona las opciones apropiadas en ese momento.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario selecciona las opciones apropiadas en ese momento.
Paso	Acción				
1	El usuario selecciona las opciones apropiadas en ese momento.				
<b>Postcondiciones</b>	Las opciones se guardan, y si se vuelve acceder a la pantalla se cargan.				
<b>Frecuencia</b>	Alta				
<b>Importancia</b>	Alta				

---

Tabla B.5: Caso de uso 1.4: Seleccionar las opciones adicionales.

<b>CU-1: Seleccionar las emociones o respuestas.</b>		
<b>Descripción</b>	El usuario quiere seleccionar las emociones o respuesta relacionadas con el audio.	
<b>Precondiciones</b>	Estamos en la pantalla de selección de emociones o respuesta, con un audio grabado y unas opciones seleccionadas.	
<b>Requisitos</b>	RF-2, RF-2.1, RF-2.2, RF-2.3 y RF-2.4.	
<b>Actor</b>	Usuario que genera un dato.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario selecciona las emociones o respuesta relacionada.
<b>Postcondiciones</b>	Lo seleccionado se guarda, y si se vuelve a entrar en la pantalla se cargarán estos datos.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Si se selecciona emociones y respuestas no se puede aceptar.
	1	Si se seleccionan las respuestas <i>sí</i> y <i>no</i> , no se permite aceptar.
	1	Si no se selecciona nada no deja aceptar y salir de la pantalla.
<b>Frecuencia</b>	Alta	
<b>Importancia</b>	Alta	

Tabla B.6: Caso de uso 1.5: Seleccionar las emociones o respuestas.

---

**CU-2: Seleccionar paciente.**


---

<b>Descripción</b>	El usuario selecciona el paciente con el cual quiere trabajar.							
<b>Precondiciones</b>	Estar en la aplicación de interpretación, con conexión directa al servidor.							
<b>Requisitos</b>	RF-1, RF-1.1, RF-1.2, RF-1.3 y RF-1.4							
<b>Actor</b>	Usuario.							
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario despliega el <i>spinner</i> para ver los posibles pacientes.</td> </tr> <tr> <td>2</td> <td>El usuario selecciona al paciente con el que quiere trabajar.</td> </tr> </tbody> </table>		Paso	Acción	1	El usuario despliega el <i>spinner</i> para ver los posibles pacientes.	2	El usuario selecciona al paciente con el que quiere trabajar.
Paso	Acción							
1	El usuario despliega el <i>spinner</i> para ver los posibles pacientes.							
2	El usuario selecciona al paciente con el que quiere trabajar.							
<b>Excepciones</b>	Paso	Acción						
	1	Si no se puede conectar con el servidor, saldrá un error 1.						
<b>Frecuencia</b>	Alta							
<b>Importancia</b>	Alta							

---

Tabla B.7: Caso de uso 2: Seleccionar paciente.

---

<b>CU-3: Obtener información sobre el proyecto.</b>					
<b>Descripción</b>	El usuario tiene que poder ver información acerca del equipo de desarrollo.				
<b>Precondiciones</b>	Estar en la aplicación de interpretación, con conexión directa al servidor.				
<b>Requisitos</b>	RF-8				
<b>Actor</b>	Usuario.				
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario, en el menú principal, pulsa el botón de información al lado del logo de la aplicación.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario, en el menú principal, pulsa el botón de información al lado del logo de la aplicación.
Paso	Acción				
1	El usuario, en el menú principal, pulsa el botón de información al lado del logo de la aplicación.				
<b>Postcondiciones</b>	El usuario se encuentra en la pantalla de información, puede salir pulsando el botón <i>Aceptar</i> .				
<b>Frecuencia</b>	Baja				
<b>Importancia</b>	Baja				

---

Tabla B.8: Caso de uso 3: Obtener información sobre el proyecto.

---

<b>CU-4: Gestionar las opciones adicionales.</b>		
<b>Descripción</b>	El usuario quiere gestionar las opciones del paciente seleccionado.	
<b>Precondiciones</b>	Estar en la aplicación de interpretación y haber seleccionado al paciente, con conexión directa al servidor.	
<b>Requisitos</b>	RF-3 y RF-3.2	
<b>Actor</b>	Usuario.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario, en el menú principal, pulsa el botón de <i>Registro de Información</i> .
	2	El usuario visualiza las opciones guardadas para el paciente seleccionado.
	3	El usuario puede modificar y guardar las opciones.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si no existe el fichero de opciones en el servidor, este devuelve un error 5.
<b>Frecuencia</b>	Alta	
<b>Importancia</b>	Alta	

---

Tabla B.9: Caso de uso 4: Gestionar las opciones adicionales.

---

**CU-4.1: Mostrar las opciones.**

---

<b>Descripción</b>	El usuario al entrar en la pantalla visualizará las opciones guardadas para el paciente seleccionado.				
<b>Precondiciones</b>	Estar en la pantalla de opciones.				
<b>Requisitos</b>	RF-3 y RF-3.2				
<b>Actor</b>	Usuario.				
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario visualiza las opciones guardadas para el paciente seleccionado.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario visualiza las opciones guardadas para el paciente seleccionado.
Paso	Acción				
1	El usuario visualiza las opciones guardadas para el paciente seleccionado.				
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Si no existe el fichero de opciones en el servidor, este devuelve un error 5.</td> </tr> </tbody> </table>	Paso	Acción	1	Si no existe el fichero de opciones en el servidor, este devuelve un error 5.
Paso	Acción				
1	Si no existe el fichero de opciones en el servidor, este devuelve un error 5.				
<b>Frecuencia</b>	Alta				
<b>Importancia</b>	Alta				

---

Tabla B.10: Caso de uso 4.1: Mostrar las opciones.

---

**CU-4.2: Modificar y guardar las opciones.**

---

<b>Descripción</b>	El usuario modifica, y guarda o cancela los cambios.								
<b>Precondiciones</b>	Estar en la pantalla de opciones.								
<b>Requisitos</b>	RF-5								
<b>Actor</b>	Usuario.								
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario visualiza las opciones para el paciente seleccionado.</td> </tr> <tr> <td>2</td> <td>El usuario modifica alguna de las opciones, habilitando así el botón <i>Guardar</i>.</td> </tr> <tr> <td>3</td> <td>El usuario guarda o cancela los cambios.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario visualiza las opciones para el paciente seleccionado.	2	El usuario modifica alguna de las opciones, habilitando así el botón <i>Guardar</i> .	3	El usuario guarda o cancela los cambios.
Paso	Acción								
1	El usuario visualiza las opciones para el paciente seleccionado.								
2	El usuario modifica alguna de las opciones, habilitando así el botón <i>Guardar</i> .								
3	El usuario guarda o cancela los cambios.								
<b>Postcondiciones</b>	Si el usuario ha decidido guardar, los cambios se realizan en el servidor.								
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Si no existe el fichero de opciones en el servidor, este devuelve un error 5.</td> </tr> <tr> <td>3</td> <td>Si no hay conexión con el servidor, se muestra un error 1.</td> </tr> </tbody> </table>	Paso	Acción	1	Si no existe el fichero de opciones en el servidor, este devuelve un error 5.	3	Si no hay conexión con el servidor, se muestra un error 1.		
Paso	Acción								
1	Si no existe el fichero de opciones en el servidor, este devuelve un error 5.								
3	Si no hay conexión con el servidor, se muestra un error 1.								
<b>Frecuencia</b>	Media								
<b>Importancia</b>	Alta								

---

Tabla B.11: Caso de uso 4.2: Modificar y guardar las opciones.

**CU-5: Interpretar emoción o respuesta.**

<b>Descripción</b>	El usuario quiere interpretar las emociones y las respuestas de los pacientes.										
<b>Precondiciones</b>	Estar en la aplicación de interpretación con un paciente seleccionado, y con conexión directa al servidor.										
<b>Requisitos</b>	RF-6, RF-6.1 y RF-6.2.										
<b>Actor</b>	Usuario.										
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario, en el menú principal, pulsa el botón <i>Qué quiero decir</i>.</td></tr> <tr> <td>2</td><td>El usuario selecciona el tipo de interpretación que quiere hacer.</td></tr> <tr> <td>3</td><td>El usuario graba el audio que quiere interpretar.</td></tr> <tr> <td>4</td><td>El usuario visualiza y escucha el resultado.</td></tr> </tbody> </table>	Paso	Acción	1	El usuario, en el menú principal, pulsa el botón <i>Qué quiero decir</i> .	2	El usuario selecciona el tipo de interpretación que quiere hacer.	3	El usuario graba el audio que quiere interpretar.	4	El usuario visualiza y escucha el resultado.
Paso	Acción										
1	El usuario, en el menú principal, pulsa el botón <i>Qué quiero decir</i> .										
2	El usuario selecciona el tipo de interpretación que quiere hacer.										
3	El usuario graba el audio que quiere interpretar.										
4	El usuario visualiza y escucha el resultado.										
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Si no se puede conectar con el servidor, saldrá un error 1.</td></tr> <tr> <td>4</td><td>Si no se puede conectar con el servidor, saldrá un error 1.</td></tr> <tr> <td>4</td><td>Si no existe un modelo válido para el paciente error 8.</td></tr> </tbody> </table>	Paso	Acción	1	Si no se puede conectar con el servidor, saldrá un error 1.	4	Si no se puede conectar con el servidor, saldrá un error 1.	4	Si no existe un modelo válido para el paciente error 8.		
Paso	Acción										
1	Si no se puede conectar con el servidor, saldrá un error 1.										
4	Si no se puede conectar con el servidor, saldrá un error 1.										
4	Si no existe un modelo válido para el paciente error 8.										
<b>Frecuencia</b>	Alta										
<b>Importancia</b>	Crítica										

Tabla B.12: Caso de uso 5: Interpretar emoción o respuesta.

---

**CU-5.1: Grabar audio.**

---

<b>Descripción</b>	El usuario tiene que poder grabar el audio que quiere interpretar.						
<b>Precondiciones</b>	Haber seleccionado un paciente, y estar en la pantalla de selección de interpretación con conexión directa al servidor.						
<b>Requisitos</b>	RF-5						
<b>Actor</b>	Usuario.						
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El usuario selecciona el tipo de interpretación.</td> </tr> <tr> <td>2</td> <td>El usuario graba el audio que quiere interpretar.</td> </tr> </tbody> </table>	Paso	Acción	1	El usuario selecciona el tipo de interpretación.	2	El usuario graba el audio que quiere interpretar.
Paso	Acción						
1	El usuario selecciona el tipo de interpretación.						
2	El usuario graba el audio que quiere interpretar.						
<b>Postcondiciones</b>	El audio que ha grabado se puede reproducir, pulsando el botón <i>Escuchar</i> , o se puede volver a grabar pulsando <i>Grabar</i> .						
<b>Frecuencia</b>	Alta						
<b>Importancia</b>	Crítica						

---

Tabla B.13: Caso de uso 5.1: Grabar audio.

---

<b>CU-5.2: Visualizar y escuchar el resultado.</b>									
<b>Descripción</b>	El usuario quiere visualizar el resultado y poder volver a escuchar el audio relacionado con este resultado.								
<b>Precondiciones</b>	Estar en la pantalla de grabación del audio, con un audio ya grabado y con conexión directa al servidor.								
<b>Requisitos</b>	RF-6, RF-6.1, RF-6.2 y RF-4.4.								
<b>Actor</b>	Usuario.								
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario pulsa el botón <i>Entender</i> para ver el resultado.</td></tr> <tr> <td>2</td><td>El usuario visualiza y escucha el resultado.</td></tr> <tr> <td>3</td><td>El usuario puede volver a escuchar el audio relacionado con ese resultado pulsando el botón <i>Escuchar</i>.</td></tr> </tbody> </table>	Paso	Acción	1	El usuario pulsa el botón <i>Entender</i> para ver el resultado.	2	El usuario visualiza y escucha el resultado.	3	El usuario puede volver a escuchar el audio relacionado con ese resultado pulsando el botón <i>Escuchar</i> .
Paso	Acción								
1	El usuario pulsa el botón <i>Entender</i> para ver el resultado.								
2	El usuario visualiza y escucha el resultado.								
3	El usuario puede volver a escuchar el audio relacionado con ese resultado pulsando el botón <i>Escuchar</i> .								
<b>Postcondiciones</b>	El correo con el dato se ha enviado correctamente.								
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>2</td><td>Si no se tiene conexión con el servidor, error 1.</td></tr> <tr> <td>2</td><td>Si no existe el modelo entrenado o está dentro del grupo de resultado aleatorios, error 8.</td></tr> </tbody> </table>	Paso	Acción	2	Si no se tiene conexión con el servidor, error 1.	2	Si no existe el modelo entrenado o está dentro del grupo de resultado aleatorios, error 8.		
Paso	Acción								
2	Si no se tiene conexión con el servidor, error 1.								
2	Si no existe el modelo entrenado o está dentro del grupo de resultado aleatorios, error 8.								
<b>Frecuencia</b>	Alta								
<b>Importancia</b>	Crítica								

---

Tabla B.14: Caso de uso 5.2: Visualizar y escuchar el resultado.

---

**CU-6: Información en la ejecución.**

---

<b>Descripción</b>	El usuario quiere tener información sobre las acciones que hacen los botones de la aplicación.
<b>Precondiciones</b>	-
<b>Requisitos</b>	RF-5
<b>Actor</b>	Usuario.
<b>Secuencia normal</b>	<b>Paso</b> <b>Acción</b> 1       Si el usuario pulsa algún botón de información se abrirá un diálogo explicativo en lectura fácil, además se reproducirá por audio.
<b>Frecuencia</b>	Baja
<b>Importancia</b>	Baja

---

Tabla B.15: Caso de uso 6: Información en la ejecución.

## *Apéndice C*

---

# Especificación de diseño

---

## C.1. Introducción

Una vez tenemos bien definidos los requisitos funcionales y los casos de uso, tenemos que realizar las diferentes tareas de diseño necesarias para poder transformar esos requisitos que ha definido el cliente, en software tangible y bien elaborado.

En este apartado voy a comentar los distintos diseños que he realizado para el desarrollo del proyecto, estos tipos de diseño son:

- **Diseño de datos:** en el cual voy a explicar los diseños y decisiones referentes a las estructuras de datos y a los propios datos.
- **Diseño procedimental:** explicación detallada de como se relacionan los apartados en la ejecución, con diagramas de secuencia.
- **Diseño arquitectónico:** explicación de las relaciones entre las distintas partes del sistema.
- **Diseño de interfaces:** donde se podrá ver como se ha diseñado las interfaces y sus versiones iniciales.
- **Diseño de código de errores:** donde explicaré la estandarización que se ha realizado para controlar los errores de la conexión entre la aplicación y el servidor.

## C.2. Diseño de datos

En este subapartado voy a comentar los distintos aspectos de diseños relacionado con los datos, es decir, como se organizan y de que están compuestos.

Cabe destacar que ambas aplicaciones *Android*, trabajan sobre una carpeta raíz base llamada *Apace*, que si no existe al iniciar la aplicación se crea automáticamente.

### Aplicación de Recogida de Datos

En esta aplicación lo que hacemos es generar datos para poder comprimirlos y después enviarlos por correo para su posterior investigación. Estos datos están compuestos de un audio con un sonido de una persona con parálisis cerebral gravemente afectada, unas opciones adiciones y el resultado que la persona que ha grabado cree que es.

Para poder generar todos estos datos ha de existir una forma de poder, dentro de nuestro dispositivo *Android*, crear carpetas donde poder guardar nuestros datos para posteriormente comprimirlos y enviarlos. En esta aplicación, cada vez que seleccionamos un paciente y entramos en la pantalla de grabar un audio se crea una carpeta con el nombre del paciente y la fecha, con hora y segundos. Por cada uno de los ficheros que creamos en una misma ejecución, se almacenan en esta carpeta con el mismo nombre más un valor identificativo. Cabe destacar que para los ficheros en donde se almacenan las opciones adiciones y el resultado (emociones o respuesta) están almacenados en ficheros csv.

Una vez tenemos todos los datos creados en la carpeta, al enviar se genera un comprimido con el mismo nombre de la carpeta donde están los datos y se envía.

Un ejemplo de la estructura de datos comentada sería:

```

Apace
└── JMiguel_10-35-36_19-06-2019
    ├── JMiguel_10-35-36_19-06-2019.mp4
    ├── JMiguel_10-35-36_19-06-2019_Estado.csv
    └── JMiguel_10-35-36_19-06-2019_Opciones.csv
└── JMiguel_10-35-36_19-06-2019.zip

```

## Aplicación para la Interpretación

Esta segunda aplicación es bastante más compleja, en cuanto a la lógica que hay detrás, como al conjunto de datos que maneja.

En primer lugar voy a comentar la estructura de ficheros con la que trabaja la aplicación. En esta aplicación, si no existe algún fichero y/o carpeta en la creación se crea automáticamente gracias al método *inicio()* que se ejecuta nada más crear el primer *Activity*, el *MainActivity*. La estructura de ficheros que da soporte a la lógica de la aplicación es la siguiente:

```
Apace
└── JMiguel_10-35-36_19-06-2019.mp4
    └── config
        └── paciente.csv
```

En esta estructura nos encontramos con una carpeta *config*, que es donde se almacenan los ficheros de configuración, como lo es el fichero *paciente.csv* que nos permite almacenar el último paciente seleccionado en la última ejecución, el cual será cargado en el siguiente inicio de la aplicación, para así tener un paciente por defecto.

Como se puede observar, del directorio raíz también cuelga un fichero *mp4*, este es un ejemplo de donde se almacenan los audios antes de ser enviados al servidor para interpretar. No es hasta que hemos obtenido el resultado de la interpretación cuando este fichero se elimina, para así ocupar el menor almacenamiento posible.

Pero la estructura de datos no es la única parte relevante en el diseño de los datos de una aplicación, sino que también lo son las clases con las que trabajamos, como se puede ver en la figura C.1. Como se puede ver en el diagrama, trabajamos con un conjunto grande de clases, cada una con los datos necesarios para realizar sus funcionalidad.

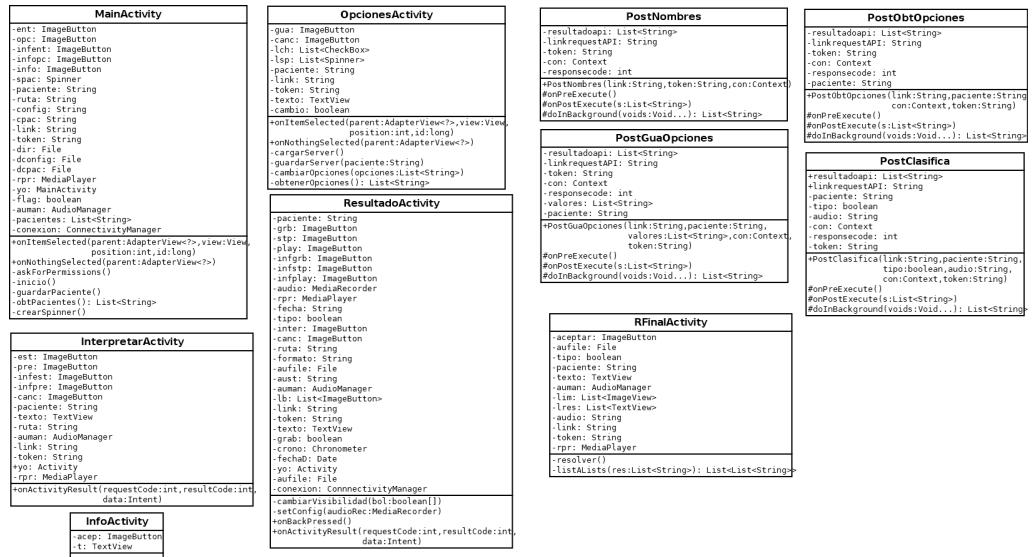


Figura C.1: Diagrama de clases.

Aun con las estructuras de datos de la aplicación y el diagrama de clases no hemos terminado con el diseño de los datos, ya que una parte muy importante de esta es ver la relación que existe entre las diferentes clases, estas relaciones las podemos ver en el diagrama de clases de alto nivel en la figura C.2.

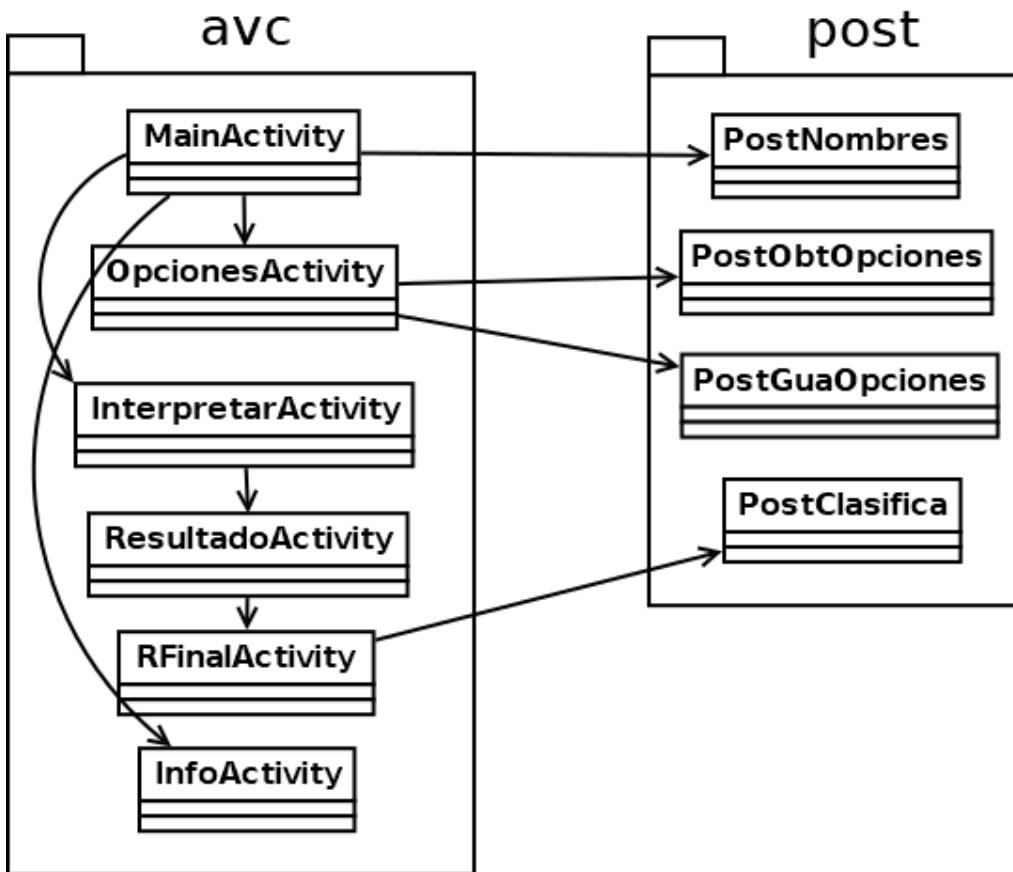


Figura C.2: Diagrama de Clases de Alto Nivel.

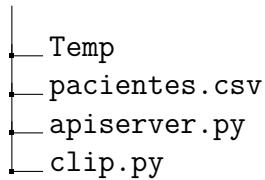
En el diagrama se representan las llamadas entre las diferentes clases, habiendo en estas llamadas distintos parámetros como puede ser la *url* del servidor, el *token* de seguridad o el paciente seleccionado.

## Servidor

Dentro del servidor, como ya se ha comentado en la memoria, se ha puesto un especial interés en el diseño y en la creación de la estructura de datos que nos permita controlar los datos que maneja el servidor. Esta estructura es la siguiente:

```

root
└── Modelos
└── Opciones
  
```



En la carpeta *Modelos*, es donde nos encontramos con los modelos entrenados de los pacientes, tanto los modelos para interpretar estados como los modelos para interpretar respuesta. En la carpeta *Opciones* se encuentran los distintos archivos con los valores actuales de las opciones adicionales de los pacientes. Por último, en la carpeta *Temp* es donde se almacenan los audios subidos desde la aplicación móvil, que una vez el servidor realiza la respuesta al cliente este audio se elimina.

### C.3. Diseño procedimental

#### Aplicación para la recogida de datos

En esta aplicación solo hay un procedimiento de ejecución, ya que es lo que se quería, una aplicación con una ejecución lineal, que resultase fácil de utilizar para los usuarios, como así nos comentaron en las distintas presentaciones de esta aplicación. El procedimiento de esta ejecución se puede ver en el diagrama de secuencia en la figura C.3.

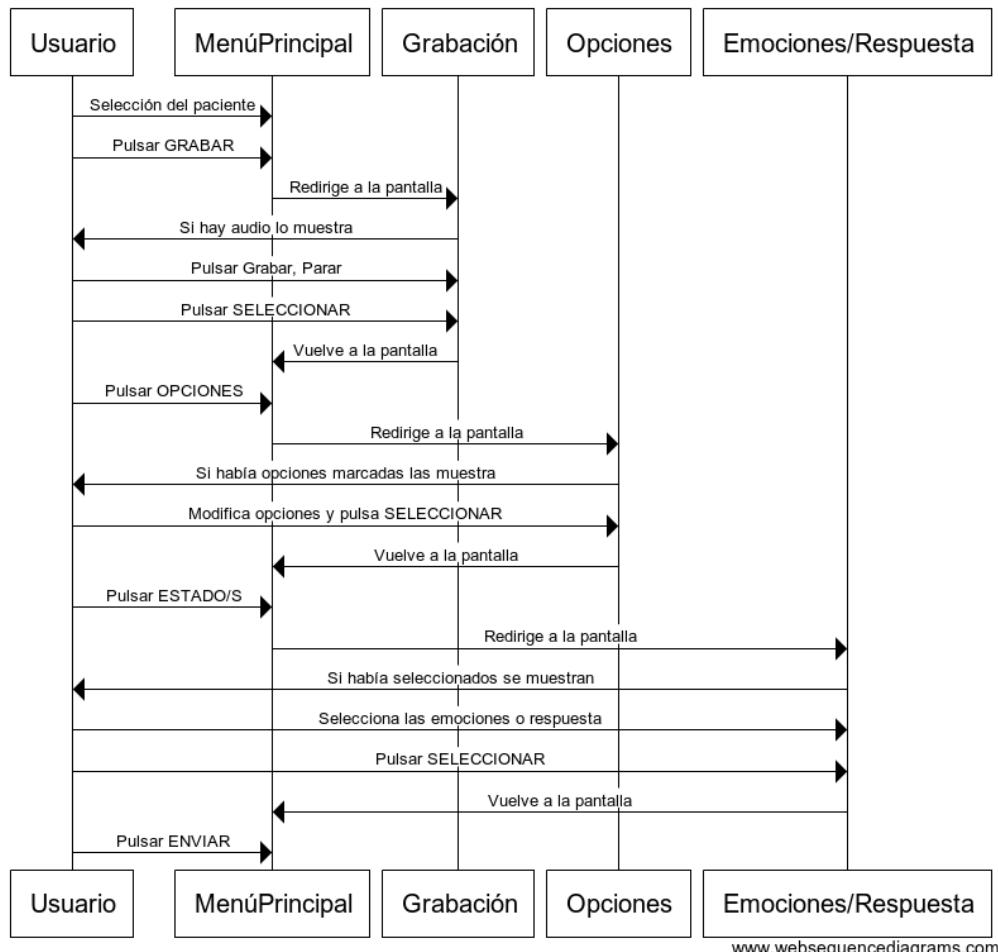


Figura C.3: Diagrama de Secuencia aplicación generación de datos.

## Aplicación para la interpretación

Hay distintas procedimientos que podemos probar en la aplicación de interpretación, estos son unos procedimientos bien definidos que nos permiten realizar las distintas funcionalidades que tiene esta aplicación y el servidor al que está conectado.

Estos procedimientos han sido diseñados siguiendo los diagramas de secuencia que se pueden ver en las figuras C.4 C.5, C.6 y C.7.

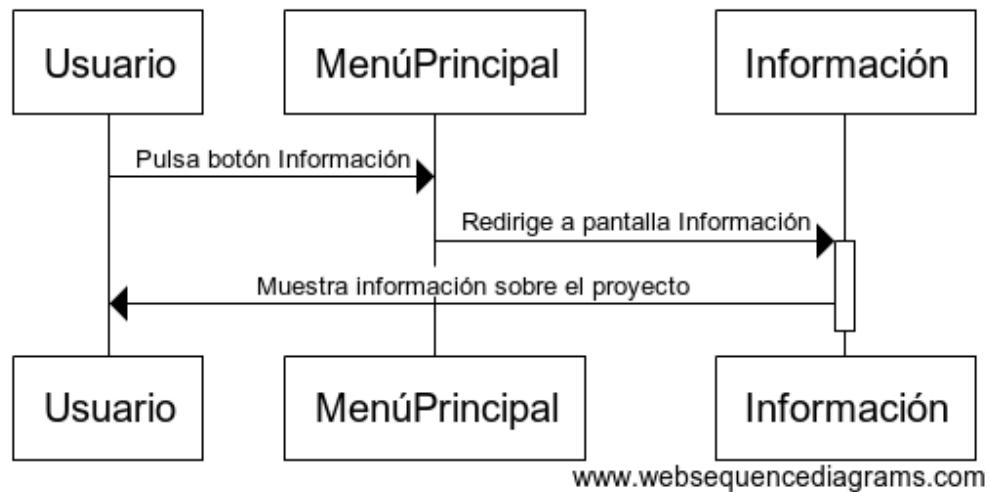


Figura C.4: Diagrama de Secuencia de Información.

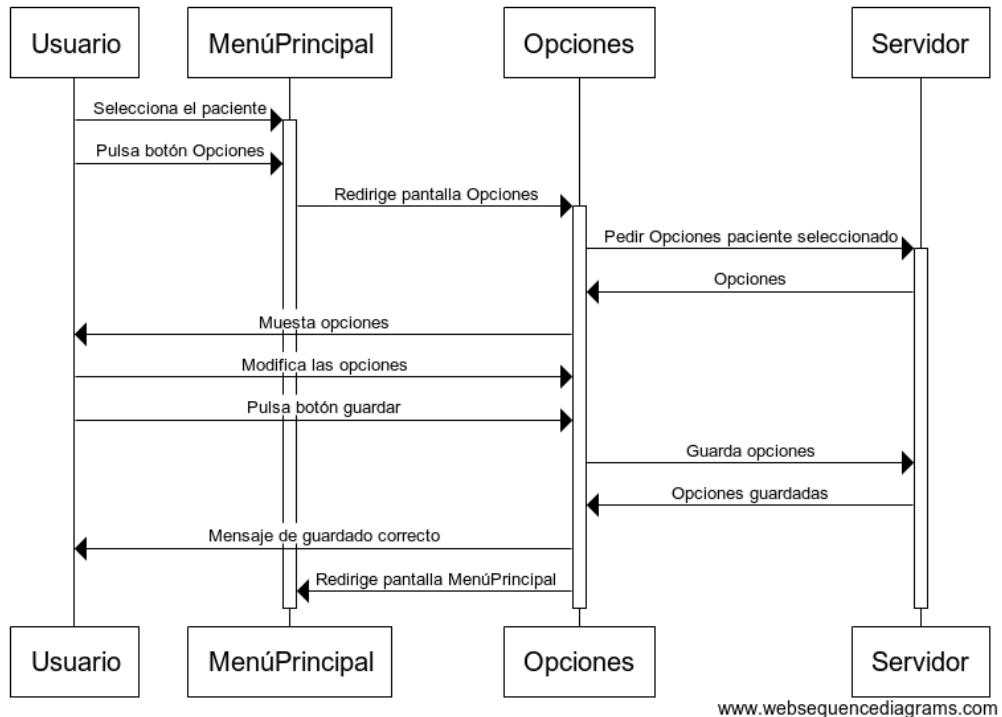


Figura C.5: Diagrama de Secuencia del cambio de las opciones.

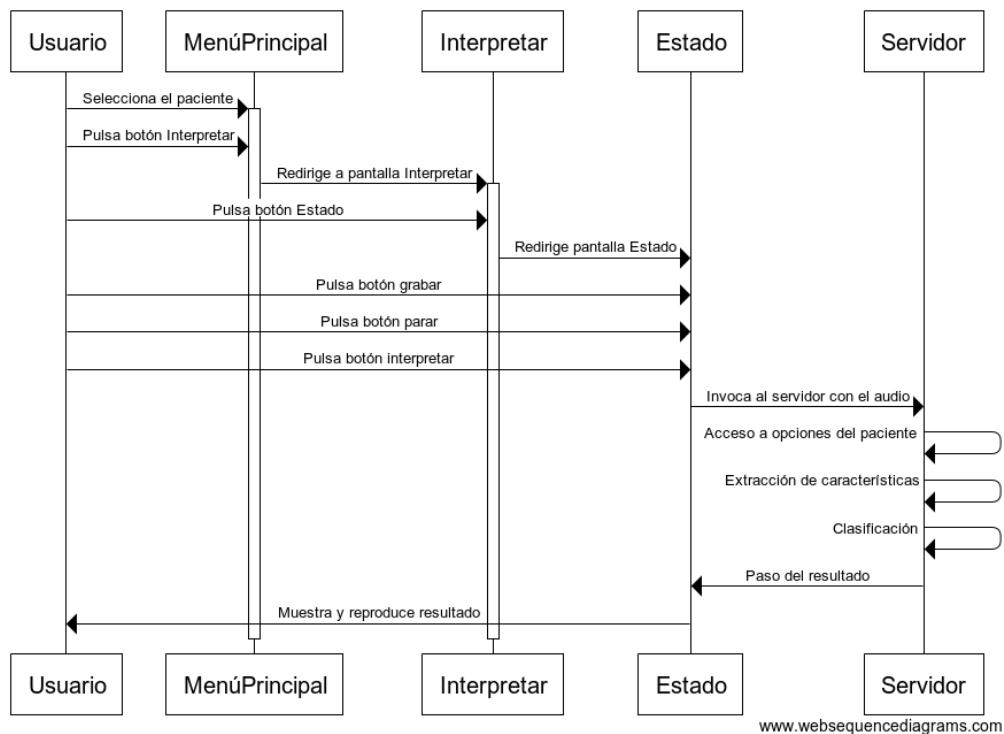


Figura C.6: Diagrama de Secuencia de la interpretación de una emoción/estado.

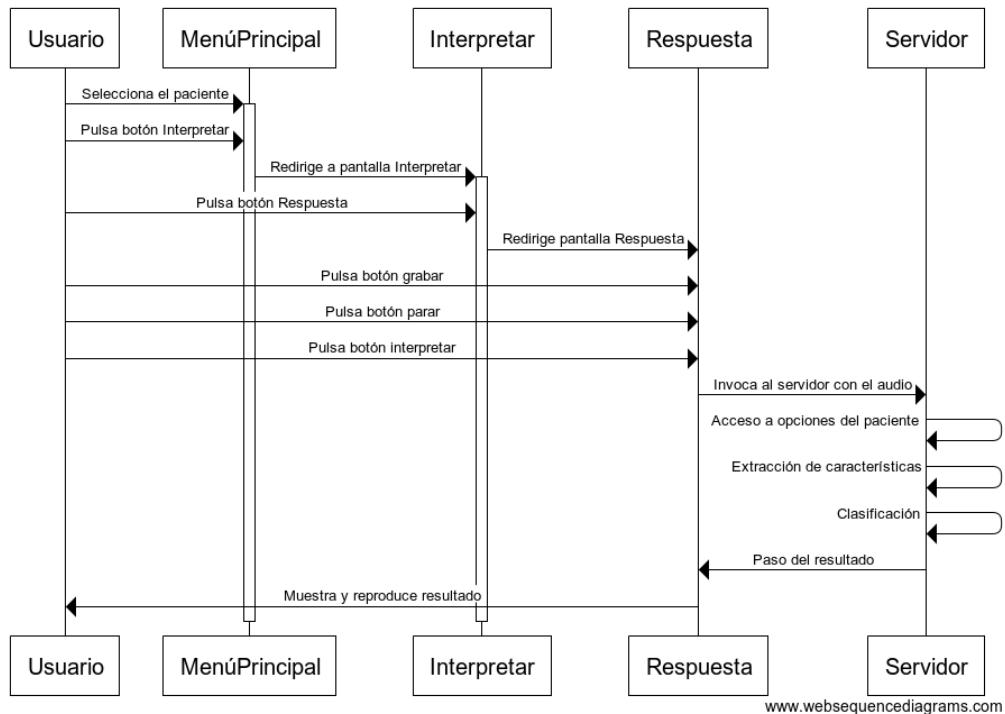


Figura C.7: Diagrama de Secuencia de la interpretación de una respuesta.

Como se puede ver, cada diagrama muestra el diseño de una parte de la aplicación, en el primer diagrama, figura C.4 podemos ver la secuencia para llegar a la pantalla de información del proyecto. En el segundo diagrama, figura C.5, se puede ver la secuencia de procesos que hay que realizar para poder modificar las opciones de un paciente. Por último, podemos ver en los diagramas de la secuencia de pasos a seguir para poder interpretar una emoción, figura C.6, y una respuesta, figura C.7.

## C.4. Diseño arquitectónico

Como se ha comentado a lo largo de la memoria y de los anexos, en el proyecto se han desarrollado distintas aplicaciones y una servidor, que conforman un sistema en el cual una aplicación genera datos que sirven para crear clasificadores con los que poder interpretar en el servidor, sonidos capturados desde la aplicación de interpretación.

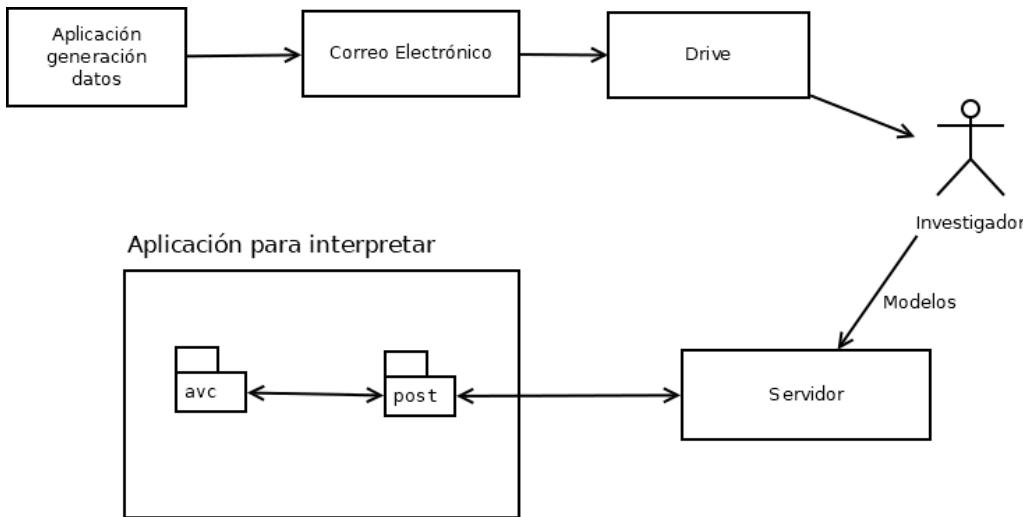


Figura C.8: Diagrama de Secuencia de Alto Nivel.

La relación entre las distintas partes que conforman el sistema se puede ver en la figura C.8.

Como se puede observar, hay dos líneas de ejecución. Una que nos permite generar datos, en la cual partimos de las grabaciones obtenidas de la aplicación de generación de datos, estos datos son enviados por correo y recogidos en drive para que posteriormente el investigador colaborador, Sergio Chico, pueda trabajar y obtener los mejores algoritmos de extracción de características y de clasificación. Por otro lado, tenemos la ejecución orientada a la interpretación, esta comienza con la grabación de los audios en la aplicación para interpretar, en concreto en el paquete *avc*, que luego llama a las clases del paquete *post* para comunicarse con el servidor y obtener el resultado de la interpretación.

## C.5. Diseño de interfaces

Otra parte muy importante en el desarrollo de un proyecto es el diseño de las interfaces de los programas y aplicaciones que se desarrollen. Es más importante aun si tienes un cliente que te contrata para realizar el proyecto, ya que la interfaz tiene que estar cuidadosamente diseñada para cumplir con sus requisitos.



Figura C.9: Diseño interfaz aplicación recogida de datos.



Figura C.10: Interfaz final de la aplicación recogida de datos.

## Aplicación de generación de datos

El diseño de la interfaz que hice de esta aplicación se puede ver en la figura C.9. Y la interfaz final de la aplicación se puede ver en la figura C.10.

## Aplicación para interpretar

En esta aplicación se puso mucho empeño en realizar un buen diseño, y junto con el equipo de APACE Burgos diseñamos alrededor de 8 interfaces, de las cuales se eligió la interfaz que se puede ver en la figura C.11.

A la aplicación final llegaron muchos cambios, algunos pedidos por APACE Burgos y otros por meras necesidades que veía al ver la interfaz. Entre los cambios más significativos están el cambio de color de los botones, o la disposición de alguna pantalla. La interfaz final de la aplicación se puede ver en la figura C.12.



Figura C.11: Diseño interfaz aplicación interpretación.

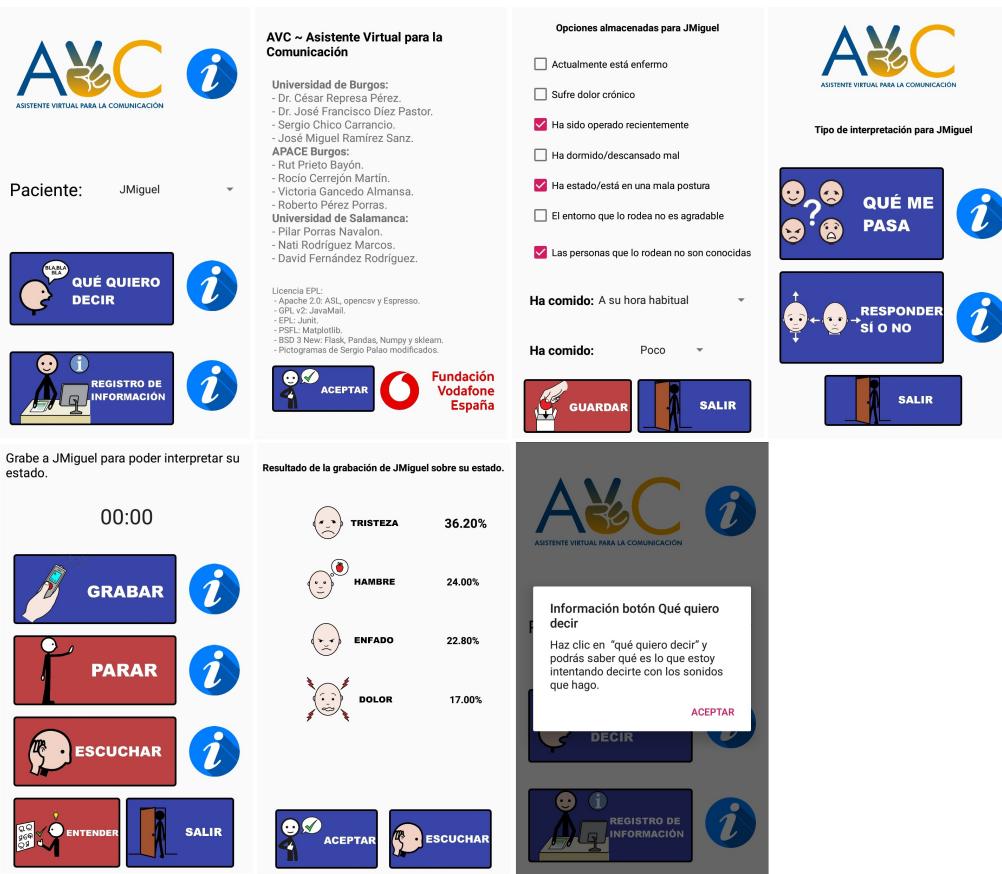


Figura C.12: Interfaz final de la aplicación AVC.

## C.6. Diseño de código de errores

Mientras estaba desarrollando el servidor, e implementando su conexión con la aplicación, me di cuenta de que los errores de los diferentes métodos se repetían continuamente. Es por ello que decidí diseñar un estándar de códigos de errores para así poder responder mejor ante un posible fallo en la conexión entre la aplicación y el servidor, documentando la causa del fallo y como poder arreglarlo. El diseño del código de errores se puede ver en la figura C.13.

Error	Descripción	Causa	Solución
Er1	No se puede acceder al servidor.	Este error se da cuando el servidor está caído.	Avisar al Administrador para que reinicie el servidor.
Er2	Token de seguridad incorrecto.	O se ha corrompido la aplicación cambiando el token de seguridad o el token de seguridad ha cambiado sin ser actualizado en su dispositivo.	Avisar al Administrador para que le pase de nuevo la aplicación.
Er3	Lista de nombres vacía en el servidor.	La lista con los nombres en el servidor está vacía.	Avisar al Administrador para que recupere el archivo.
Er4	Lista de nombres no encontrada.	El fichero con la lista de los pacientes en el servidor no está.	Avisar al Administrador para que recupere el archivo.
Er5	Opciones del paciente no encontradas.	El fichero con las opciones adicionales del paciente no está en el servidor.	Avisar al Administrador para que recupere el archivo.
Er6	No hay conexión a Internet.	No tener conexión ni Wifi ni Datos Móviles.	Conectarse a una red con acceso a Internet.
Er7	Fichero del paciente vacío.	El fichero del paciente con las opciones está vacío en el servidor.	Avisar al Administrador para que recupere el archivo.
Er8	Fichero de clasificación no existe.	No existe el fichero con el que se clasifica en el servidor a ese paciente.	Avisar al Administrador para que recupere el archivo.
Er9	Resultado vacío.	El servidor ha devuelto una solución vacía.	Avisar al Administrador para que compruebe el funcionamiento de la clasificación para ese paciente.

Figura C.13: Código de errores.

## *Apéndice D*

---

# Documentación técnica de programación

---

## D.1. Introducción

En este apartado de los anexos voy a comentar:

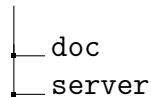
- **Estructura de directorios:** estructura de directorios con la que se ha trabajado en *GitHub*.
- **Manual del programador:** manual en el que se explican los pasos que se ha de realizar para seguir desarrollando el proyecto.
- **Compilación, instalación y ejecución del proyecto:** explicación de los pasos a realizar para la instalación y ejecución del proyecto.
- **Pruebas del sistema:** apartado donde comentaré las distintas pruebas que se han realizado.

## D.2. Estructura de directorios

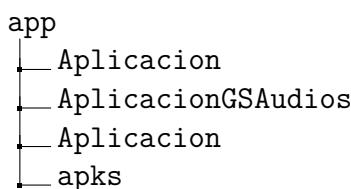
La estructura de directorios que he seguido para la realización de este proyecto es la misma que se puede ver en el repositorio en GitHub, <https://github.com/Josemi/AVC-TFG>. Esta estructura, en primer nivel se puede ver como:

```
/  
└ app
```

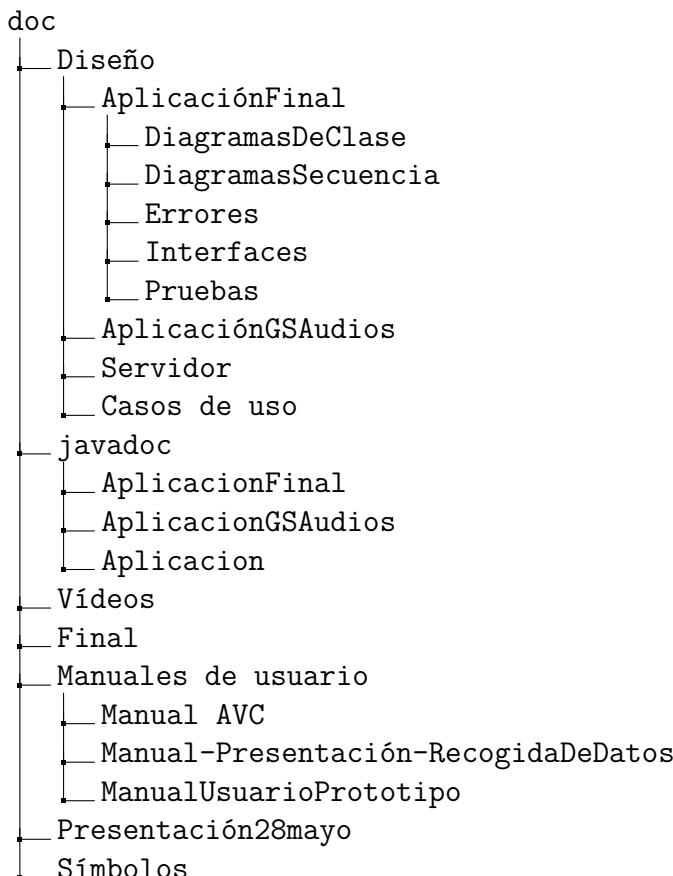
## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN



En la carpeta *app*, nos encontramos con el código de las 3 aplicaciones desarrolladas con *AndroidStudio* en el proyecto, estas son *Aplicacion* que es la aplicación prototipo de grabación de audios, *AplicacionGSAudios* que es la aplicación para la recogida de datos y *AplicacionFinal* que es la aplicación de interpretación AVC. Además, nos encontramos con una carpeta llamada *apks*, donde se encuentran las *apks* de las últimas versiones de cada aplicación. El árbol que cuelga de la carpeta *app* se puede ver a continuación:



En la carpeta *doc* nos encontramos con la estructura de documentación del proyecto, esta se puede ver en el siguiente árbol:

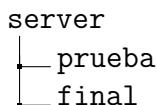


Como se puede observar, de la carpeta *doc* cuelgan muchos elementos, entre ellos están los orientados al diseño de la aplicación en la carpeta *Diseño*, a los manuales de usuario en la carpeta *Manuales de usuario* o por ejemplo a la estructura para generar la documentación final en *LATEX* en la carpeta *Final*.

Dentro de la carpeta *javadoc*, nos podemos encontrar la documentación de las 3 aplicaciones realizadas. En la carpeta *Vídeos* es donde nos podemos encontrar los distintos vídeos explicativos que he creado para enseñar las aplicaciones.

Cabe destacar que en la carpeta *Símbolos* nos encontramos con los pictogramas de *ARASAAC* [1], los cuales hemos modificado para utilizarlos en la aplicación de interpretación final.

Por último en la carpeta *server* nos encontramos con la siguiente estructura:



En la carpeta *prueba*, nos encontramos con la primera prueba que realicé para usar *Flask*, y en la carpeta *final* nos encontramos con la estructura del servidor final ya comentada en el apartado C.2

## D.3. Manual del programador

En este apartado se va a comentar que se tendría que hacer para, a partir del software actual realizar mejoras. Para este apartado se ha decidido no explicar la aplicación prototipo, ya que el resto de aplicaciones son mejoras a partir de esta. Además, se tiene en cuenta que se dispone de la estructura del proyecto tras haber hecho un *clone* del repositorio en *GitHub*.

### Aplicación para la recogida de datos

El código de esta aplicación se encuentra en el directorio */app/AplicaciónGSAudios*. Para poder modificar el código de esta aplicación tenemos que abrir *AndroidStudio*, pulsar *File* y después *Open...*, aquí se nos abrirá un diálogo en el cual podremos elegir la aplicación, *AndroidStudio* se encarga de buscar las aplicaciones y de mostrarlas para poder elegir la aplicación, como se puede ver en la figura D.1.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

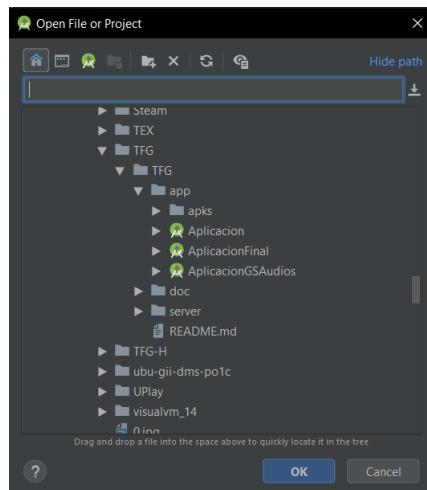


Figura D.1: Selección de fichero o proyecto en *AndroidStudio*.

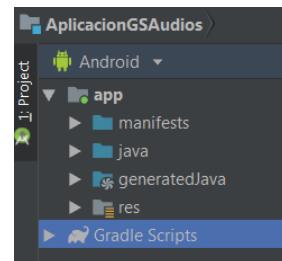


Figura D.2: Estructura de ficheros generada por *AndroidStudio*.

Una vez hayamos seleccionado y abierto el proyecto tendremos en *AndroidStudio* una vista de la estructura del proyecto. Tras procesar y cargar el proyecto, que según el tamaño de este puede durar un rato, *AndroidStudio* nos organiza las carpetas en dos, *app* y *Gradle Scripts*, como se puede ver en la ??.

En la carpeta *app* nos encontramos con la lógica y las interfaces de las pantallas, mientras que en la carpeta *Gradle Scripts* nos encontramos algunas configuraciones importantes del proyecto, como los *imports* de las librerías externas o la API mínima de desarrollo.

Dentro de la carpeta de *app* nos encontramos con la siguiente estructura:

```
app
  manifest
  java
  generatedJava
```



En la carpeta *manifest* nos encontramos con uno de los ficheros más importantes del desarrollo en *Android*, es en el en donde se encuentran los permisos que tiene la aplicación, para el acceso a Internet, para la grabación de audios... En la carpeta *java* nos encontramos con la lógica de la aplicación y con los test para probar esta lógica. Y en la carpeta *res* nos encontramos con la estructura que nos permite administrar y utilizar los archivos como imágenes, sonidos y sobre todo los *layouts* que son las interfaces de las pantallas.

Una de las posibles modificaciones que se podrían querer hacer a esta aplicación sería añadir algún paciente al cual poder grabar, para ello tenemos que acceder a la carpeta */res/values* y en concreto al archivo *strings.xml*, donde podemos encontrar el *array* con los pacientes, el cual se carga al principio de la aplicación en el *spinner* de selección de paciente.

## Aplicación para la interpretación

Para abrir el proyecto con esta aplicación tenemos que hacer los pasos que en la aplicación anterior, pero a la hora de abrir el proyecto tenemos que seleccionar *AplicaciónFinal*.

La estructura de este proyecto es similar a la anterior, pero más completa, ya que en este nos encontramos con más imágenes, sonidos e incluso ficheros donde se definen las formas y los colores de los botones, todo ello en la carpeta *res*.

## Servidor

La estructura del servidor ha sido explicada en el apartado C.2. Si se quiere modificar algún apartado de su lógica tendríamos que modificar el fichero *apiserver.py* donde se encuentra el servidor *Flask*.

Una de las posibles modificaciones que se pueden hacer sobre el servidor es, como en la aplicación de generación de datos, añadir un paciente al sistema. Para ello tenemos que abrir el fichero *paciente.csv* y añadir el nombre del paciente nuevo. Además, tenemos que crear el fichero csv, puede ser vacío, en la carpeta *Opciones* con el mismo nombre que se ha puesto en el fichero *pacientes.csv*. Por último, si se quiere que la interpretación no se haga de forma aleatoria tenemos que añadir los modelos de clasificación correspondientes en la carpeta *Modelos*, además de modificar el flujo del método *clasifica* del servidor.

## **APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN**

Aun así, si metemos un nuevo paciente en el fichero *pacientes.csv* y no se crean los ficheros necesarios, una vez queramos acceder desde la aplicación para la interpretación nos devolverá el error correspondiente que nos dirá que documentos nos faltan en el servidor.

### **D.4. Compilación, instalación y ejecución del proyecto**

En este apartado se va a comentar la instalación y ejecución del servidor, ya que la instalación y ejecución de las aplicaciones se van a comentar en el apartado de *Manual de usuario*.

#### **Requisitos**

- Tener una tarjeta de red que nos permita a partir del programa *noWifi* crear una red, con esa tarjeta como *router*.
- Tener instalado *Python 3*, en concreto la ejecución se va a mostrar con *Anaconda*.
- Tener instaladas las librerías de *sklearn*, *numpy*, *flask*, *base64*, *csv*, *datetime*, *os*, *librosa*, *random*, *glob*, *pydub* y *matplotlib*, si no se tiene instalada algunas de estas librerías tendríamos que instalarlas en su última versión.

#### **Instalación**

El único software que hay que instalar, a parte de *Python* y las librerías, son las aplicaciones que ya se menciona su instalación en los manuales de usuario.

#### **Ejecución**

Para la ejecución primero tenemos que hacer es desactivar el *FireWall* de sistema operativo *Windows*, después tenemos que abrir el programa *noWifi* y crear nuestra propia red cumpliendo el *SSID*, con el nombre que queremos que tenga nuestra red, y *Password* con la contraseña que queremos que tenga la red. Una vez hemos cumplido eso, tenemos que pulsar el *checkbox* llamado *Access Point* para iniciar la red, como se ve en la figura D.3.

#### D.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

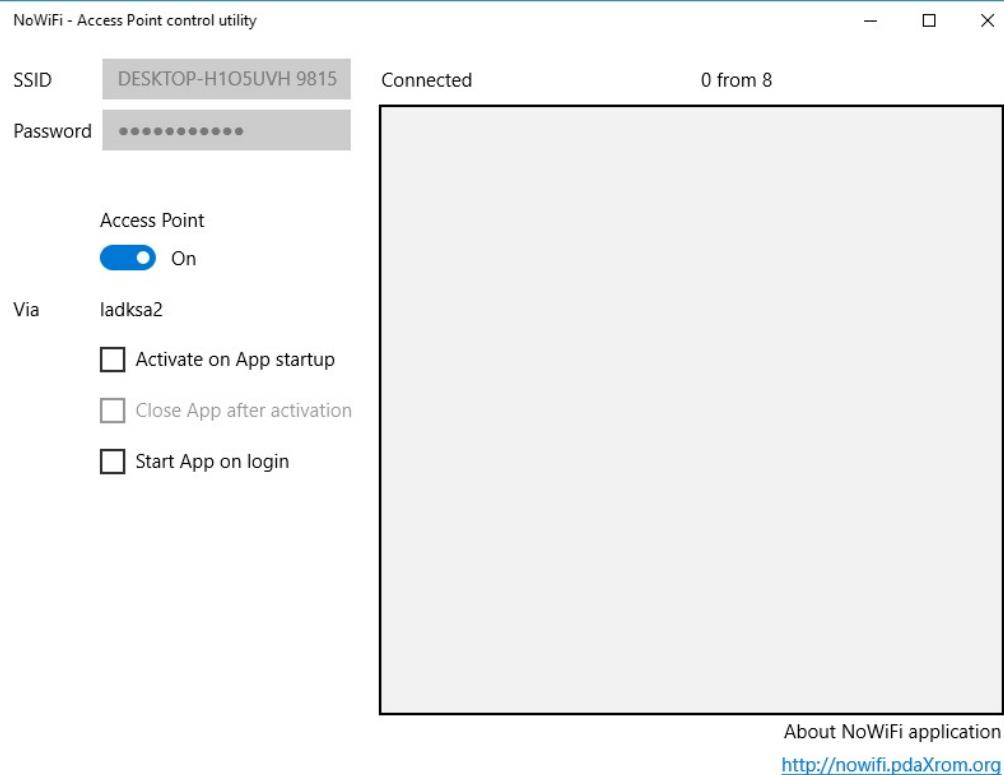


Figura D.3: Interfaz de *no WiFi*.

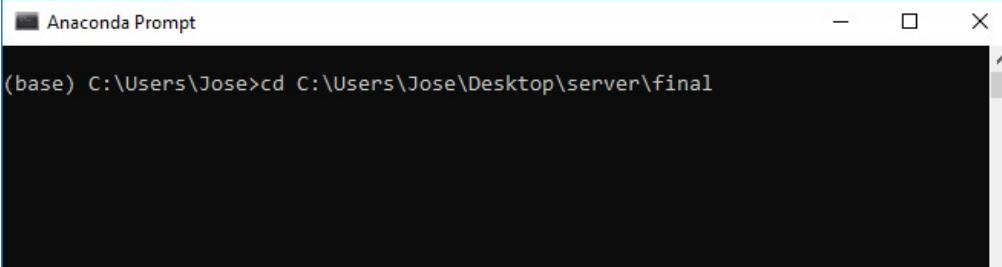
Es a esta red a la cual nos tenemos que conectar desde el dispositivo que quiere ejecutar la aplicación para la interpretación, AVC. Como se puede ver en la figura D.3, hay un apartado llamado *Connected* donde podemos observar los dispositivos conectados a nuestra red.

Después de crear y configurar nuestra red, tenemos que abrir *Anaconda Prompt*, para desplegar el servidor, para ello lo primero que tenemos que hacer en esta consola es movernos a las carpeta */server/final* del servidor, como se ve en la figura D.4.

Tras movernos a la carpeta donde se encuentra el código del servidor *Flask*, solo tenemos que ejecutar el fichero *Python* con el comando *python apiserver.py*, como se puede ver en la figura D.5.

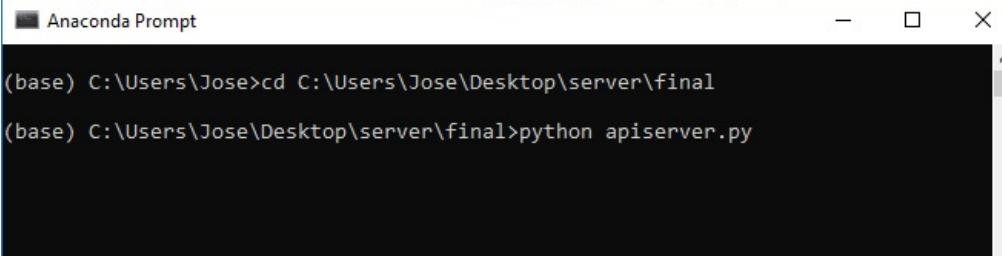
Tras ejecutar el comando empezará a desplegarse el servidor en *Flask*, esto puede llevar un tiempo debido a que *Flask*, como nos dice cada vez que hacemos una ejecución, no es un *framework* para desarrollar servidores

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN



```
Anaconda Prompt
(base) C:\Users\Jose>cd C:\Users\Jose\Desktop\server\final
```

Figura D.4: Comando cd a la carpeta */server/final*.



```
Anaconda Prompt
(base) C:\Users\Jose>cd C:\Users\Jose\Desktop\server\final
(base) C:\Users\Jose\Desktop\server\final>python apiserver.py
```

Figura D.5: Comando *Python* para desplegar el servidor.

de producción, sino para desarrollar pruebas de estos. Una vez cargado el servidor, sabemos que está cargado porque no pone la dirección *IP* sobre la que se está ejecutando, podremos ver las distintas llamadas al servidor, junto con la dirección *IP* de la llamada, el día y la hora, el método del servidor al cual se ha llamado y por último el código de respuesta que se ha devuelto. Todo esto se puede ver en la figura D.6.

Los *Deprecation Warnings* que se ven en la ejecución son debido a la librería que ha decidido usar el investigador colaborador, Sergio Chico, para almacenar los modelos.

## D.5. Pruebas del sistema

Las pruebas en un proyecto software son esenciales, ya que nos permiten probar las funcionalidades y el comportamiento del software desarrollado. Su finalidad es detectar fallos, es decir, una conjunto de pruebas tienen éxito cuando descubren algún *bug* o mal comportamiento en el software.

```

(base) C:\Users\Jose>cd C:\Users\Jose\Desktop\server\final
(base) C:\Users\Jose\Desktop\server\final>python apiserver.py
C:\Users\Jose\Anaconda3\lib\site-packages\sklearn\externals\joblib\_init_.py:15: DeprecationWarning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with scikit-learn 0.21+.
    warnings.warn(msg, category=DeprecationWarning)
* Serving Flask app "apiserver" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
C:\Users\Jose\Anaconda3\lib\site-packages\sklearn\externals\joblib\_init_.py:15: DeprecationWarning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with scikit-learn 0.21+.
    warnings.warn(msg, category=DeprecationWarning)
* Debugger is active!
* Debugger PIN: 469-067-008
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
192.168.137.231 - - [20/Jun/2019 21:21:14] "POST /Nombres HTTP/1.1" 200 -
192.168.137.231 - - [20/Jun/2019 21:21:16] "POST /ObtOpciones HTTP/1.1" 200 -
192.168.137.231 - - [20/Jun/2019 21:21:22] "POST /ObtOpciones HTTP/1.1" 200 -
192.168.137.231 - - [20/Jun/2019 21:21:38] "POST /Clasifica HTTP/1.1" 200 -

```

Figura D.6: Consola del servidor en ejecución.

En este proyecto se han hecho pruebas sobre la aplicación más importante, la aplicación de interpretación. Debido a la necesidad de tener la aplicación para la recogida de datos lo antes posible, para poder trabajar con el mayor número de estos, no fue posible la elaboración de test, aun así la aplicación se probó en diversos dispositivos para encontrar los *bugs* más graves y significativos.

Se han elaborado distintos tipos de pruebas en la aplicación de interpretación y por ende en el servidor, estas se pueden clasificar en dos tipos:

- **Unitarias:** prueban la que la lógica y la creación de las clases es correcta.
- **Integración:** prueban que la colaboración entre las distintas clases es correcta.
- **Estrés:** prueban el comportamiento de la aplicación en situaciones complicadas de mucho trabajo.

En total se han realizado 82 test entre test unitarios y de integración, en ellos he utilizado diferentes herramientas orientadas a los test como son *Espresso* y *JUnit*. El resultado de los 82 test se puede ver en la figura D.7.

Además, he realizado pruebas de test con la herramienta *MonkeyTest*, que permite en su ejecución probar las aplicaciones en situaciones de estrés, con muchas pulsaciones, todas ellas aleatorias para poder probar todos los apartados de la aplicación. Debido a la velocidad de esta prueba, solo se

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

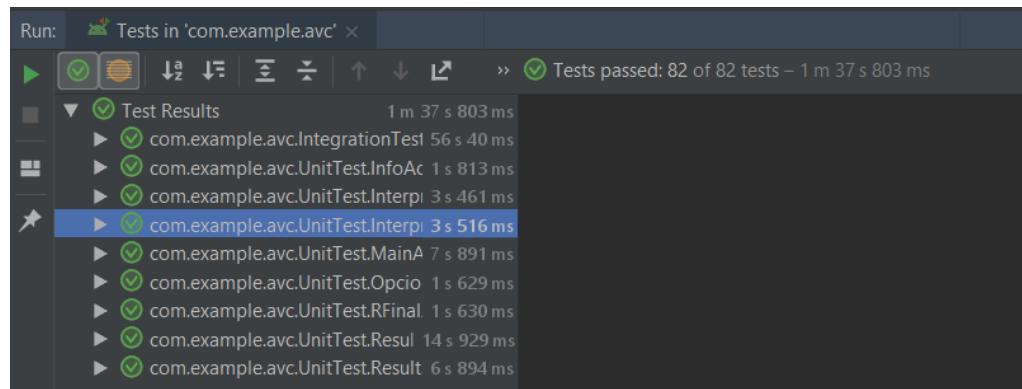


Figura D.7: Salida de los test unitarios y de integración.

ha usado en la versión de la aplicación de interpretación sin conexión al servidor.

Para ejecutar el *MonkeyTest* tenemos que hacer los siguientes pasos en el mismo orden:

- Abrir una consola *cmd*.
- Movernos a la carpeta *C:\Users\Usuario\AppData\Local\Android\Sdk\platform-tools*
- Ejecutar el comando *adb -e(simulador)/-d(dispositivo conectado) shell monkey -p [package de la app] -v*(para obtener más información de cómo ha ido la ejecución) [Número de eventos aleatorios].

Un ejemplo del comando de ejecución sería, *adb -d shell monkey -p com.example.avc -v 1000 >test.txt*, en el cual decimos que lo queremos ejecutar en un dispositivo conectado, que el paquete donde se encuentra nuestra aplicación es *com.example.avc*, que queremos obtener información adicional sobre la ejecución, que queremos que haga 1000 eventos aleatorios y por último que la salida por consola la desvíe al fichero *test.txt*.

Tras la ejecución podremos ver en el fichero el conjunto de acciones que ha ido haciendo, así como su resultado final, es decir, si la ejecución se ha detenido o no.

## *Apéndice E*

---

# Documentación de usuario

---

## E.1. Introducción

En este apartado se comenta una parte esencial dentro de la documentación de un proyecto, los manuales de usuario. Estos son documentos donde los desarrolladores tienen que poner toda la información necesaria para instalar y ejecutar los programas y/o aplicaciones que han desarrollado. Estos documentos tienen que tener también los requisitos que se necesitan para poder instalar y ejecutar los programas, e incluso pueden incluir comentarios que nos dicen que hacer en caso de error de la aplicación.

Este apartado ha sido dividido por los distintos manuales de las distintas aplicaciones que he hecho, de cada una se comentarán los requisitos de los usuarios, la instalación y como se ejecuta. Estas aplicaciones de las que vamos a hablar son:

- **Aplicación de grabación prototipo:** aplicación que un principio iba a ser la aplicación con la cual generar datos. Nos permite grabar audios con el nombre que elijamos.
- **Aplicación para la recogida de datos:** aplicación final para la recogida que nos permite grabar un audio y seleccionar tanto las opciones adicionales como las emociones o respuesta relacionada.
- **Aplicación para la interpretación:** aplicación final AVC que nos permite modificar las opciones adicionales de los pacientes comunes a todos los dispositivos y nos permite interpretar una emoción o una respuesta.

## E.2. Aplicación de grabación prototipo

### Uso

Este es un prototipo de la aplicación final, en este prototipo se permite grabar y dar nombre a audios y reproducirlos.

Estos audios se almacenan en un carpeta llamada Apace localizada en la raíz.

### Requisitos

El requisito principal es contar con un dispositivo *smartphone Android* actualizado, con versión *Android 6.0* o superior.

### Instalación

Para poder instalar la aplicación el único requisito que se necesita es conexión a Internet para poder descargarse la aplicación.

### Ejecución

Para la ejecución del prototipo solo se requiere que la aplicación esté instalada y que se acepten los permisos de grabación de audio y de escritura que se preguntan al principio de la ejecución.

### Instalación

La instalación del prototipo es sencilla, solo hay que descargar el archivo *apk* desde donde se haya sido suministrado (correo, *WhatsApp*, *Github*) y realizar los ajustes necesarios para poder descargarla al ser una aplicación ajena a *Play Store*.

Estos ajustes adicionales consisten simplemente aceptar que se descargue esta aplicación exterior a *Play Store*.

Una vez descargada el *apk* de la aplicación aparecerá en su menú de aplicaciones instaladas con el nombre de Apace, como se puede ver en la figura E.1.

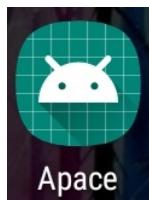


Figura E.1: Logotipo y nombre de la aplicación una vez instalada.

## Ejecución

Para la ejecución solo tenemos que pulsar el logotipo anteriormente mostrado. En la primera ejecución que hagamos del prototipo nos saldrá el siguiente diálogo, para pedirnos permisos para poder usar tanto el sistema de grabación de audio, como para poder almacenar archivos. Debemos dar permisos a ambos pulsando en la opción *Permitir*. Es muy importante aceptar estos permisos ya que sino la aplicación se cerrará en la primera vez que queramos grabar un audio. En cualquier caso si por algún error se ha denegado los permisos y la aplicación se ha cerrado se puede volver a abrir y nos volverá a pedir aceptar los permisos. La petición de estos permisos se pueden ver en las figuras E.2 y E.3.



Figura E.2: Petición de permisos para la grabación de audio.



Figura E.3: Petición de permisos para el acceso al almacenamiento multimedia.

Después de aceptar los permisos tendremos acceso al menú principal de la aplicación, como se ve en la figura E.4, en el se pueden ver 3 botones (*RECORD*, *STOP* y *PLAY*) y una caja de texto con un botón *GEN*.



Figura E.4: Menú principal de la aplicación prototipo.

Los tres primeros botones sirven para manejar la grabación de los audios, al principio solo tendremos habilitado el botón *RECORD* que nos permite comenzar a grabar el audio siempre que la caja de texto no esté vacía o que en ella haya un nombre que ya existe (es decir, que ya exista un audio con ese nombre).

Una vez pulsemos el botón *RECORD* este quedará deshabilitado y se habilitará el botón *STOP* que nos permite parar y guardar la grabación.

Cuando hayamos finalizado la grabación se nos habilitarán los botones de *RECORD* para poder volver a grabar y de *PLAY* que nos permite reproducir el audio con el nombre que esté en la caja de texto de la ruta (nombre del archivo).

Por último tenemos la caja de texto con la ruta donde podremos elegir el nombre que le queremos dar al archivo o podemos generarla automáticamente con el botón *GEN* que generará un nombre con la hora y fecha de la grabación.

## E.3. Aplicación para la recogida de datos

### Uso

Esta aplicación sirve para recolectar datos necesarios para realizar la investigación y la creación de la aplicación final de predicción de emociones y respuestas. Esta aplicación nos permite grabar a los pacientes y tras llenar dos formularios, genera un archivo comprimido con la grabación de audio y las respuestas de los formularios, y lo envía para que se pueda realizar la investigación.

### Requisitos

El requisito principal es contar con un dispositivo *smartphone Android* actualizado, con versión *Android 6.0* o superior, y conexión a Internet ya sea vía *Wi-Fi* o datos móviles.

### Instalación

Para poder instalar la aplicación el único requisito que se necesita es conexión a Internet para poder descargarse la aplicación.

### Ejecución

Para la ejecución del prototipo se requiere que la aplicación esté instalada, que se acepten los permisos de grabación de audio y de escritura que se preguntan al principio de la ejecución y conexión a Internet para poder subir los archivos grabados.

### Instalación

La instalación del prototipo es sencilla, solo hay que descargar el archivo *apk* desde donde se haya sido suministrado (correo, *WhatsApp*, *Github*) y realizar los ajustes necesarios para poder descargarla al ser una aplicación ajena a *Play Store*.

Estos ajustes adicionales consisten simplemente aceptar que se descargue esta aplicación exterior a *Play Store*.

Una vez descargada la *apk* la aplicación aparecerá en su menú de aplicaciones instaladas con el nombre de AplicaciónGSAudios, como se puede ver en la figura E.5.



Figura E.5: Logotipo y nombre de la aplicación de generación de datos.

## Ejecución

Para la ejecución solo tenemos que seleccionar el logotipo anteriormente mostrado. En la primera ejecución que hagamos del prototipo nos saldrá el siguiente diálogo para pedirnos permisos para poder usar tanto el sistema de grabación de audio como para poder almacenar archivos, debemos dar permisos a ambos pulsando en la opción *Permitir*, es muy importante aceptar estos permisos ya que sino la aplicación se cerrará en la primera vez que queramos grabar un audio. En cualquier caso, si por algún error se ha denegado los permisos y la aplicación se ha cerrado se puede volver a abrir y nos volverá a pedir aceptar los permisos. La petición de estos permisos se puede ver en las figuras E.6 y E.7.

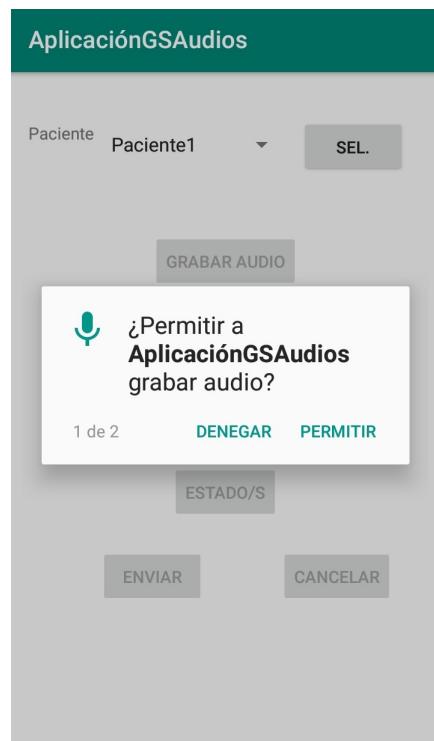


Figura E.6: Petición de permisos para la grabación de audio en la aplicación de recogida de datos.

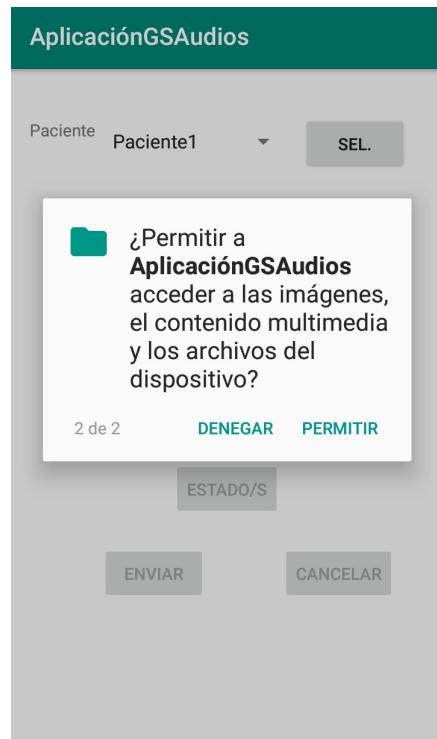


Figura E.7: Petición de permisos para el acceso al almacenamiento multimedia en la aplicación de recogida de datos.

Después de aceptar los permisos tendremos acceso al menú principal de la aplicación, como se puede ver en la figura E.8, en la parte superior de este, se puede ver la selección del paciente sobre el cual vamos a realizar la grabación.

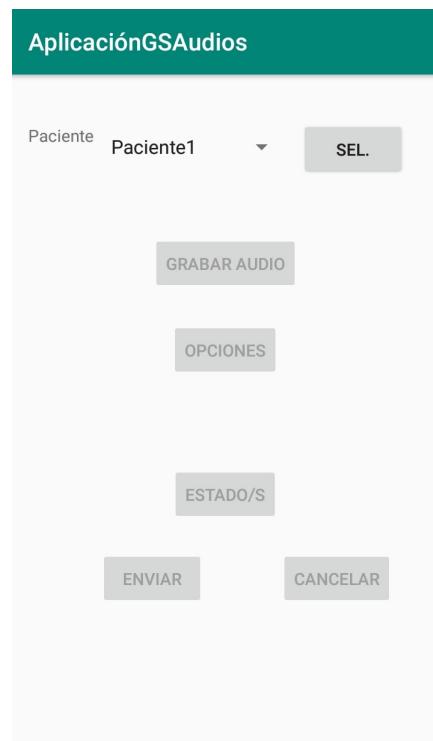


Figura E.8: Menú principal de la aplicación para la recogida de datos.

Tras haber seleccionado al paciente sobre el cual se va a recoger los datos debemos confirmar la selección del paciente pulsando el botón *Sel.* (Seleccionar). Tras pulsar el botón *Sel.* se nos habilitará el botón *Grabar Audio*, en todo momento desde la pantalla del menú principal podremos cancelar la grabación y la selección del paciente pulsando el botón *Cancelar*, como se puede ver en la figura E.9.

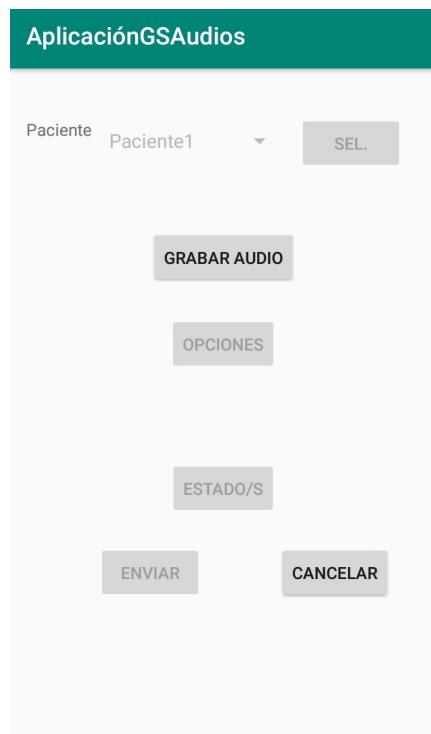


Figura E.9: Paciente seleccionado, *Grabar Audio* habilitado.

Cuando pulsamos el botón *Grabar Audio* se abrirá una nueva pantalla donde podremos grabar el audio correspondiente, como se puede ver en la figura ??.



Figura E.10: Pantalla de Grabación de Audios.

Como se puede observar esta pantalla tiene 4 botones, pero nada más abrir la pantalla solo tenemos habilitado el botón de *Grabar* que su función es comenzar a grabar el audio una vez lo pulsemos. Una vez pulsado se deshabilitará (para no poder pulsar *Grabar* dos veces) y se habilitará el botón *Parar*, como se puede ver en la figura E.11.



Figura E.11: Pulsamos el botón *Grabar*.

Si pulsamos el botón *Parar* se parará la grabación que habíamos comenzado, además, se habilitarán los botones de *Reproducir* que nos permite reproducir el audio que se acaba de grabar, el botón *Seleccionar* que nos permite seleccionar el audio y volver al menú principal y el botón *Grabar* que nos permite eliminar la grabación actual y comenzar de nuevo una grabación. La interfaz tras pulsar el botón *Parar* se puede ver en la figura E.12.



Figura E.12: Pulsamos el botón *Parar*.

Es importante que si se quiere seleccionar el audio se pulse el botón *Seleccionar* tras grabar el audio, ya que si volvemos al menú principal con el botón de ir atrás propio del móvil no se habilitará la siguiente pantalla, la pantalla de las Opciones. El menú principal tras pulsar el botón *Seleccionar* se muestra en la figura E.13

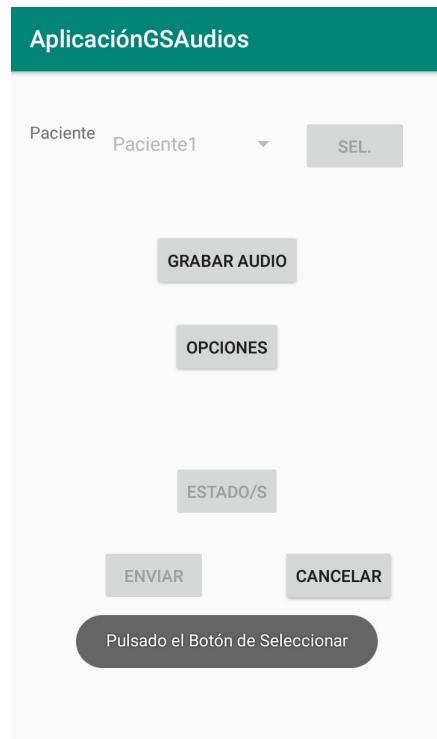


Figura E.13: Menú principal tras pulsar el botón *Seleccionar*.

Como podemos ver, tras pulsar el botón de *Seleccionar* en la pantalla de Grabación de Audio nos dirige hacia la pantalla del menú principal con el botón *Opciones* habilitado, si le pulsamos iremos a la pantalla de Opciones donde podremos seleccionar las opciones adicionales, esta pantalla la podemos ver en la figura E.14.

The screenshot shows a mobile application titled "AplicaciónGSAudios". At the top, there is a green header bar with the title. Below it, the main content area has a white background. A sub-header reads "Seleccione los apartados y opciones para el paciente:" followed by "Paciente1". There is a list of symptoms with checkboxes:

- Actualmente está enfermo
- Sufre dolor crónico
- Ha sido operado recientemente
- Ha dormido/descansado mal
- Ha estado/está en una mala postura
- El entorno que lo rodea no es agradable
- Las personas que lo rodean no son conocidas

Below this, there are two dropdown menus:

- Ha comido: A su hora habitual ▾
- Ha comido: Normal ▾

At the bottom center is a grey button labeled "SELECCIONAR".

Figura E.14: Pantalla para seleccionar las opciones adicionales.

Una vez hayamos llenado el formulario tenemos que pulsar el botón *Seleccionar* que nos volverá a la pantalla del menú principal. Como pasaba en la pantalla anterior, debemos volver al menú principal a partir de el botón *Seleccionar*, ya que sino no se desbloqueará el botón de *Estado* en el menú principal. El menú principal tras pulsar el botón *Seleccionar* en la pantalla de las opciones se puede ver en la figura E.15.



Figura E.15: Menú principal tras pulsar el botón *Seleccionar* en la pantalla de opciones.

Como podemos observar, al pulsar el botón *Seleccionar* en la pantalla de Opciones nos redirige al menú principal con el botón *Estado/s* habilitado. Si pulsamos este botón *Estado/s* nos llevará a la pantalla de selección de estados donde podremos elegir la emoción o respuesta que creemos que tiene el paciente (se puede seleccionar más de uno, pero de distintas columnas, es decir, se puede seleccionar por ejemplo dolor y hambre, pero no se puede seleccionar dolor y sí). Esta pantalla se puede ver en la figura ??.

The screenshot shows a mobile application interface titled 'AplicaciónGSAudios'. At the top, there is a green header bar with the title. Below it, the main content area has a light gray background. A heading reads 'Seleccione el/los estados para el paciente: Paciente1'. The interface is organized into two columns: 'Estado' (State) and 'Si/No' (Yes/No). Under 'Estado', there are five items: 'Dolor' (Pain), 'Enfado' (Anger), 'Tristeza' (Sadness), 'Hambre' (Hunger), and 'Normal'. Each item is preceded by an empty square checkbox. To the right of each checkbox is a small text label: 'Si' for 'Dolor', 'No' for 'Enfado', and 'Normal' for the last two. Below the list of states is a large, rounded rectangular button labeled 'SELECCIONAR' in capital letters.

Figura E.16: Pantalla de selección de emociones o respuesta.

Como podemos observar en esta pantalla nos permite seleccionar una o más emociones de la primera columna o la respuestas sí o no en la segunda. Con el botón *Seleccionar* guardaremos y volveremos al menú principal, si y solo si hemos cumplido las siguientes reglas en la selección de estados:

- Hemos seleccionado elemento/s de solo una de las dos columnas.
- Hemos seleccionado algún elemento.
- No hemos seleccionado sí y no a la vez.

Si no se cumplen todas estas reglas aparecerá un mensaje informativo, y no se nos permitirá volver al menú principal, como se puede ver en la figura E.17.

Seleccione el/los estados para el paciente: Paciente1

Estado	Si/No
<input checked="" type="checkbox"/> Dolor	<input checked="" type="checkbox"/> Si
<input type="checkbox"/> Enfado	<input type="checkbox"/> No
<input type="checkbox"/> Tristeza	
<input type="checkbox"/> Hambre	

**Normal**  
No se puede dejar vacío, ni llenar valores de ambas columnas, ni marcar "sí" y "no" a la vez

**SELECCIONAR**

Figura E.17: Mensaje informativo con el error.

Si por el contrario, no ha ocurrido ningún error al pulsar el botón *Seleccionar* iremos al menú principal teniendo ya todo completado para poder enviar.

Desde el menú principal, tras volver de la pantalla de Estado/s dando al botón *Seleccionar* se nos habrá habilitado el botón de *Enviar* que nos permite comprimir nuestros archivos y enviarlos para la investigación, siempre que haya conexión a Internet, si no tienes conexión a Internet le saldrá el siguiente mensaje (podrá volverse a conectar a Internet y volver a pulsar el botón de *Enviar*), como se puede ver en la figura E.18.



Figura E.18: Mensaje de error al intentar enviar sin conexión a Internet.

Si tenemos conexión el comprimido se generará y se enviará para su posterior investigación.

Cabe destacar que en cualquier momento se puede volver a ir a una pantalla (Grabar Audios, Opciones y Estado/s) una vez visitada, para observar lo que se tiene marcado o para modificarlo.

## E.4. Aplicación para la interpretación

### Uso

Esta aplicación sirve para interpretar las emociones y las respuestas de las personas con parálisis cerebral de las cuales se han obtenido los datos suficientes como para poder haber creado los algoritmos correspondientes y necesarios. Si el paciente al que desea interpretar no está en la lista de pacientes disponibles ha de seguir grabando datos válidos para ese paciente.

## Requisitos

El requisito principal es contar con un dispositivo *smartphone Android* actualizado con una versión *Android* 6.0 o superior y conexión a directa vía *Wi-Fi* al servidor.

### Instalación

Para poder instalar la aplicación el único requisito que se necesita es conexión a Internet para poder descargarse la aplicación.

### Ejecución

Para la ejecución de la aplicación se requiere que esta esté instalada, que se acepten los permisos de grabación de audio y de escritura que se preguntan al principio de la ejecución y conexión directa con el servidor.

### Instalación

La instalación del prototipo es sencilla, solo hay que descargar el archivo *apk* desde donde se haya sido suministrado (correo, *WhatsApp*, *Github*) y realizar los ajustes necesarios para poder descargarla al ser una aplicación ajena a *Play Store*.

Estos ajustes adicionales consisten simplemente aceptar que se descargue esta aplicación exterior a *Play Store*.

Una vez descargada la *apk* la aplicación aparecerá en su menú de aplicaciones instaladas con el nombre de *AVC*, como se puede ver en la figura E.19.



Figura E.19: Logotipo y nombre de la aplicación de interpretación, *AVC*.

### Ejecución

Para la ejecución solo tenemos que seleccionar el logotipo anteriormente mostrado. En la primera ejecución que hagamos del prototipo nos saldrá el siguiente diálogo para pedirnos permisos para poder usar tanto el sistema de grabación de audio como para poder almacenar archivos, debemos dar

permisos a ambos pulsando en la opción *Permitir*, es muy importante aceptar estos permisos ya que sino la aplicación se cerrará en la primera vez que queramos grabar un audio. En cualquier caso, si por algún error se ha denegado los permisos y la aplicación se ha cerrado se puede volver a abrir y nos volverá a pedir aceptar los permisos. La forma en la que se piden estos permisos se puede ver en las figuras E.20 y E.21.

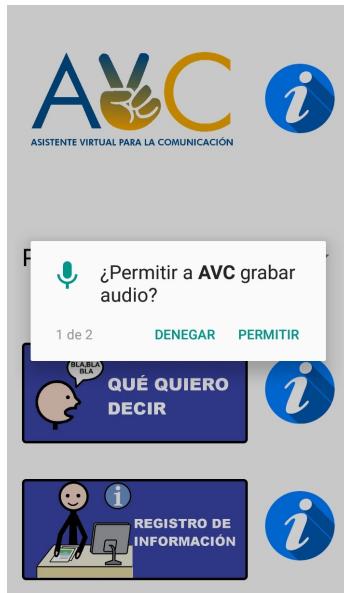


Figura E.20: Permisos de grabación de audio en AVC.

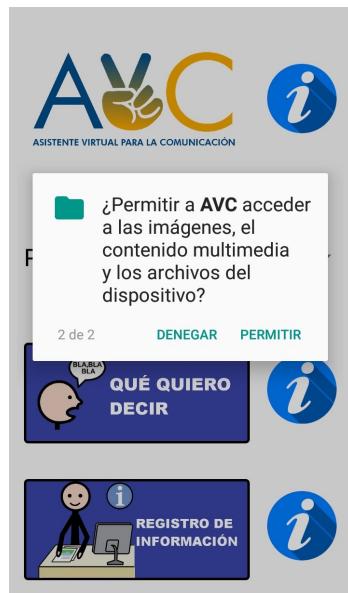


Figura E.21: Permisos para el acceso a multimedia en AVC.

Después de aceptar los permisos tendremos acceso al menú principal de la aplicación, aquí podremos seleccionar el paciente con el cual queremos trabajar (el último paciente que se seleccionó antes de cerrar la aplicación será el paciente que aparezca como seleccionado cuando volvamos a ejecutar la aplicación). El menú principal de la aplicación se puede ver en la figura ??.



Figura E.22: Menú principal de la aplicación AVC.

Tras haber seleccionado a nuestro paciente podemos hacer dos cosas diferentes, o podemos modificar las opciones adicionales relacionadas con ese paciente o podemos interpretar una emoción o una respuesta.

Si queremos modificar o ver las opciones adicionales del paciente seleccionado tenemos que pulsar el botón de *Registro de Información*, que nos llevará a la pantalla donde podremos ver las opciones guardadas para ese paciente, como se puede ver en la figura E.23.

**Opciones almacenadas para JMiguel**

<input type="checkbox"/>	Actualmente está enfermo
<input type="checkbox"/>	Sufre dolor crónico
<input checked="" type="checkbox"/>	Ha sido operado recientemente
<input type="checkbox"/>	Ha dormido/descansado mal
<input checked="" type="checkbox"/>	Ha estado/está en una mala postura
<input type="checkbox"/>	El entorno que lo rodea no es agradable
<input checked="" type="checkbox"/>	Las personas que lo rodean no son conocidas

**Ha comido:** A su hora habitual ▾

**Ha comido:** Poco ▾


GUARDAR


SALIR

Figura E.23: Pantalla con las opciones cargadas del paciente seleccionado.

Al cargar la pantalla se verán las opciones almacenadas para el paciente seleccionado, aquí podemos salir si no queremos modificar ninguna opción pulsando el botón *Salir* o podemos modificar alguna de las opciones, entonces el botón *Guardar* se pondrá en azul por lo que estaría disponible, y pulsándolo haría que se guarden las opciones. El cambio en el botón *Guardar* se puede observar en la siguiente figura E.24.

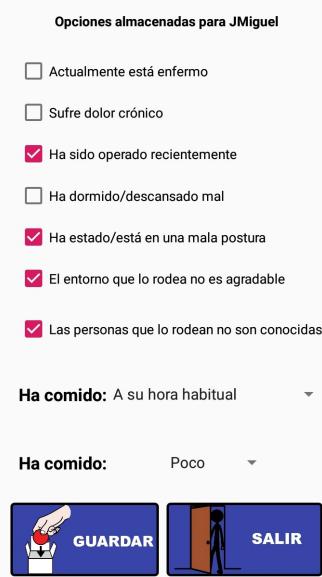


Figura E.24: Botón *Guardar* habilitado.

Una vez pulsados el botón *Salir* o *Guardar* volveremos a la pantalla principal.

Si por el contrario lo que queremos es interpretar una emoción o una respuesta tenemos que pulsar en el menú principal *Qué quiero decir*, este nos llevará a una nueva pantalla, la pantalla de selección de interpretación, que se puede ver en la figura E.25.



Figura E.25: Pantalla de selección del tipo de interpretación.

En esta pantalla podemos pulsar el botón *Salir* que nos llevaría de nuevo al menú principal, o podemos pulsar en los botones de *Qué me pasa* si queremos interpretar una emoción, o *Responder sí o no* si queremos interpretar una respuesta.

Cualquier tipo de interpretación que pulsemos nos llevará a la pantalla de grabación del audio, que se puede ver en la figura E.26.

Grabe a JMiguel para poder interpretar su estado.



Figura E.26: Pantalla de grabación del audio.

En esta pantalla podemos grabar el audio que queremos interpretar. Como en el resto de las pantallas, tenemos el botón *Salir* que nos devuelve a la pantalla anterior, la pantalla de selección de interpretación.

Si por el contrario queremos grabar un audio solo tenemos que pulsar el botón *Grabar* para comenzar a grabar, entonces el contador empezará a sumar mostrándonos el tiempo de grabación, como se puede ver en la figura E.27.

Grabe a JMiguel para poder interpretar su estado.



Figura E.27: Empezamos a grabar, tras pulsar *Grabar*.

Si queremos parar de grabar entonces tenemos que pulsar el botón *Parar*, entonces el audio se guardará y podremos escucharlo pulsando el botón *Escuchar* o intepretarlo pulsando el botón *Entender*, como se puede ver en la figura E.28.

Grabe a JMiguel para poder interpretar su estado.



Figura E.28: Paramos la grabación, tras pulsar *Parar*.

Una vez grabado el audio podemos volver a grabarlo pulsando de nuevo el botón *Grabar*, que iniciaría de nuevo la grabación. Si por el contrario el audio es correcto y es el que queremos interpretar tenemos que pulsar en el botón *Entender* que nos llevará a la pantalla donde encontraremos el resultado de la interpretación. Un ejemplo de un resultado de una interpretación se puede ver en la figura E.29.

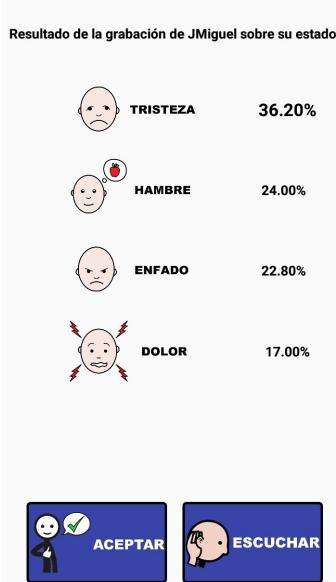


Figura E.29: Ejemplo de la pantalla con el resultado de una interpretación.

En esta pantalla podremos ver el resultado con porcentajes de la interpretación, siendo el que viene más arriba el resultado con mayor porcentaje. En esta pantalla podemos volver a escuchar el audio que hemos interpretado pulsando el botón *Escuchar*, o podemos salir al menú principal pulsando el botón *Aceptar*.

Además, a lo largo de la aplicación nos podemos encontrar con distintos botones de información a la derecha de otros botones, estos botones de información nos abren un dialogo en el cual se nos comunica vía texto en lectura fácil y por audio lo que hace el botón que tiene a la izquierda, un ejemplo de estos diálogos se puede ver en la figura E.30.

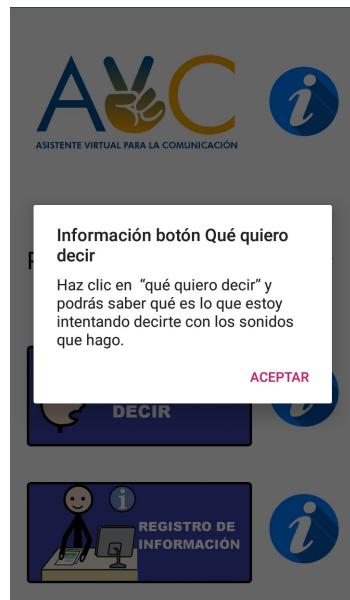


Figura E.30: Ejemplo de diálogo informativo.



---

## Bibliografía

---

- [1] Gobierno de Aragón. Portal aragonés de la comunicación aumentativa y alternativa. <http://www.arasaac.org/>, 2019.
- [2] Universidad de Burgos. Retribuciones segÚn ii convenio de pdi laboral aÑo 2018. [https://www.ubu.es/sites/default/files/portal\\_page/files/pdi\\_laboral\\_2018\\_hasta\\_junio.pdf](https://www.ubu.es/sites/default/files/portal_page/files/pdi_laboral_2018_hasta_junio.pdf).
- [3] Indeed. Salarios para empleos de informatico/a en españa. <https://www.indeed.es/salaries/Informatico/a-Salaries>.
- [4] David Miguel Lozano. Gobees. <https://github.com/davidmigloz/go-bees/blob/master/docs/latex/anexos.pdf>, feb 2017.
- [5] Sylvain Saurel. Create an audio recorder for android. <https://medium.com/@ssaurel/create-an-audio-recorder-for-android-94dc7874f3d>.