



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

UBUassistant  
Documentación Técnica



Presentado por Carlos González Calatrava  
en Universidad de Burgos — 25 de junio de 2018  
Tutor: Pedro Renedo Fernández

---

# Índice general

---

Índice general	I
Índice de figuras	III
Índice de tablas	V
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	2
A.3. Estudio de viabilidad . . . . .	5
<b>Apéndice B Especificación de Requisitos</b>	<b>13</b>
B.1. Introducción . . . . .	13
B.2. Objetivos generales . . . . .	13
B.3. Catalogo de requisitos . . . . .	14
B.4. Especificación de requisitos . . . . .	16
<b>Apéndice C Especificación de diseño</b>	<b>28</b>
C.1. Introducción . . . . .	28
C.2. Diseño de datos . . . . .	28
C.3. Diseño procedimental . . . . .	30
C.4. Diseño arquitectónico . . . . .	34
<b>Apéndice D Documentación técnica de programación</b>	<b>42</b>
D.1. Introducción . . . . .	42
D.2. Estructura de directorios . . . . .	42
D.3. Manual del programador . . . . .	44
D.4. Compilación, instalación y ejecución del proyecto . . . . .	51
D.5. Pruebas del sistema . . . . .	57

<b>Apéndice E Documentación de usuario</b>	<b>61</b>
E.1. Introducción . . . . .	61
E.2. Requisitos de usuarios . . . . .	61
E.3. Instalación . . . . .	61
E.4. Manual del usuario . . . . .	63
<b>Bibliografía</b>	<b>73</b>

---

# Índice de figuras

---

B.1. Diagrama de casos de uso . . . . .	16
C.1. Diagrama E/R . . . . .	29
C.2. Diagrama de secuencia para una palabra reservada (Ej. <i>Buenos días</i> ). .	30
C.3. Diagrama de secuencia para una pregunta con una única respuesta. .	31
C.4. Diagrama de secuencia para una pregunta con múltiples respuestas. .	32
C.5. Diagrama de secuencia para una pregunta que no tiene respuesta. .	33
C.6. Modelo-vista-controlador . . . . .	34
C.7. Diagrama MVC de una aplicación Android . . . . .	35
C.8. Diagrama MVC en una aplicación web JSP . . . . .	35
C.9. Arquitectura de una aplicación Android . . . . .	36
C.10.Arquitectura de una aplicación web basada en JSP . . . . .	37
C.11.Diagrama de estructura de los paquetes en Android . . . . .	37
C.12.Diagrama de estructura de los recursos . . . . .	38
C.13.Diagrama de estructura de los paquetes en el servidor . . . . .	39
C.14.Diagrama de navegabilidad . . . . .	40
C.15.Colores . . . . .	41
D.1. Configuración .dektop Eclipse . . . . .	45
D.2. Interfaz Android Studio . . . . .	47
D.3. Ventana inicio Tomcat . . . . .	48
D.4. Configuración fichero de servicio de Tomcat . . . . .	49
D.5. Ventana de administración de MySQL . . . . .	50
D.6. Descarga del proyecto desde GitHub . . . . .	51
D.7. Icono Android Studio . . . . .	52
D.8. Proyecto importado en Android Studio . . . . .	52
D.9. Selección del tipo de proyecto . . . . .	53
D.10.Selección del proyecto . . . . .	54
D.11.Proyecto importado en Eclipse JEE . . . . .	54
D.12.Configuración del PATH en Eclipse JEE . . . . .	55

D.13. Configuración de librerías en el fichero WAR . . . . .	56
D.14. Exportar servidor a fichero WAR . . . . .	56
D.15. Test unitarios de la base de datos . . . . .	58
D.16. Ejecución de una clase con test unitarios . . . . .	59
E.1. Opciones de seguridad Android . . . . .	62
E.2. Instalación UBUassistant . . . . .	63
E.3. Interfaz aplicación Android . . . . .	64
E.4. Reconocimiento de voz . . . . .	65
E.5. Interacción con el asistente virtual . . . . .	66
E.6. Respuesta única . . . . .	67
E.7. Múltiples respuestas . . . . .	68
E.8. Pregunta sin respuesta . . . . .	69
E.9. Continuar búsqueda . . . . .	70
E.10. Reiniciar búsqueda . . . . .	71
E.11. Respuesta consultada . . . . .	72

---

# Índice de tablas

---

A.1. Costes Humanos . . . . .	6
A.2. Costes Hardware . . . . .	7
A.3. Costes Alojamiento . . . . .	7
A.4. Costes Publicación . . . . .	7
A.5. Costes Licencias . . . . .	8
A.6. Costes Redes . . . . .	8
A.7. Costes Oficina . . . . .	9
A.8. Costes Varios . . . . .	9
A.9. Costes totales . . . . .	10
A.10.Licencias de software . . . . .	11
B.1. CU-01 Preguntar . . . . .	17
B.2. CU-02 Obtener respuesta . . . . .	17
B.3. CU-03 Obtener recomendación . . . . .	18
B.4. CU-04 Visualizar respuesta . . . . .	19
B.5. CU-05 Visualizar recomendación . . . . .	19
B.6. CU-06 Valorar respuesta . . . . .	20
B.7. CU-07 Reiniciar búsqueda . . . . .	21
B.8. CU-08 Gestión del log . . . . .	21
B.9. CU-09 Listar log . . . . .	22
B.10.CU-10 Gestión de casos . . . . .	23
B.11.CU-11 Listar casos . . . . .	24
B.12.CU-12 Añadir caso . . . . .	24
B.13.CU-13 Editar caso . . . . .	25
B.14.CU-14 Gestión de aprendizaje . . . . .	26
B.15.CU-15 Aprender recomendación . . . . .	27
D.1. Pruebas manuales . . . . .	57
D.2. Pruebas manuales web de administración . . . . .	60

# Plan de Proyecto Software

---

## A.1. Introducción

Una correcta planificación del proyecto es muy importante. Esto se debe por que de ello dependerá de que nuestro proyecto vaya evolucionando de una forma correcta y que se cumplan los tiempos de entrega, evitando así todos los retrasos posibles. Esta fase de planificación está compuesta por dos apartados:

- **Planificación temporal:** esta planificación consta principalmente en establecer los diferentes tiempos del proyecto. En este apartado se establece la fecha de inicio y la fecha de finalización del proyecto. A parte se establece la duración de cada una de las fases del proyecto, aunque estas pueden adaptarse algo a lo largo del proyecto en función del trabajo realizado a lo largo de cada uno de los sprints.
- **Estudio de viabilidad:** este estudio es muy importante, ya que es lo que nos va a permitir valorar si el proyecto nos va a permitir continuar hacia delante con él o no. Dentro de este estudio nos encontramos con dos fases;
  - **Viabilidad económica:** es el estudio en el que vamos a estudiar los costes del proyecto, tanto de tiempo, como de personal así como los recursos de terceros que son necesarios para realizarlo.
  - **Viabilidad legal:** este estudio es el más importante de todos, ya que es el que necesitamos para saber si el proyecto que vamos a hacer está permitido por ley. A parte de esto, también se estudia todo lo relacionado con la adquisición de las diferentes licencias que sean necesarias para el desarrollo del proyecto.

## A.2. Planificación temporal

Para el desarrollo del proyecto se optó por seguir la metodología de desarrollo ágil *Scrum* [1].

Debido al tamaño del proyecto realizado, no se ha podido seguir a la totalidad los rasgos de esta metodología, pero se ha intentado adaptar todo lo posible, manteniendo la esencia de la misma. Las diferentes pausas que se siguieron fueron las siguientes:

- Desarrollo incremental de la aplicación. Para ello se realizó a través de sprints.
- Los sprints se realizaron de una duración de una semana cada uno.
- Reuniones al final de cada uno de los sprint, en los cuales se valoraba el trabajo realizado en el sprint y se establecían los objetivos para el próximo sprint.

Las tareas realizadas en cada uno de los sprints se comenta a continuación.

### Sprint 0 (31/02/2018 - 06/03/2018)

En esta primera reunión lo que se hizo fue hablar sobre la propuesta del tutor para realizar el TFG. En ella se me comentó en que consistió el TFG que me ha tocado mejorar, como un poco las ideas que se quería para mejorar el citado TFG.

La tarea que se me encomendó en este primer sprint fue el de que hiciera una breve investigación sobre el funcionamiento de los algoritmos de razonamiento basado en casos.

### Sprint 1 (07/03/2018 - 13/03/2018)

En esta reunión el tutor me ofreció todo el material relacionado con el TFG a mejorar, así como una definición formal de los requisitos a cumplir durante el desarrollo del proyecto.

El objetivo que se estableció en este sprint es el de leerme la documentación del TFG e ir empezando la comprensión del proyecto para ir recogiendo todas las dudas posibles para que pudieran ser respondidas las dudas en la próxima reunión.

### Sprint 2 (14/03/2018 - 20/03/2018)

En esta reunión, lo primero que se hizo, fue solucionar todas aquellas dudas que fueron surgiendo a lo largo del sprint anterior y se establecieron los objetivos para el próximo sprint.



El objetivo de este sprint fue el de analizar todas aquellas herramientas que creyera que fueran necesarias para el desarrollo del proyecto y dejar constancia de porque la decisión de emplear esas herramientas en vez de usar otras herramientas.

### **Sprint 3 (21/03/2018 - 27/03/2018)**

Lo primero que se realizó fue valorar las herramientas elegidas para el proyecto, así como alguna recomendación de otras herramientas por parte del tutor, dejando la valoración de esas herramientas para el próximo sprint.

A parte de ese objetivo, también se estableció como objetivo investigar la forma de implementar la introducción de texto mediante voz en nuestra aplicación de Android.

### **Sprint 4 (28/03/2018 - 10/04/2018)**

Al comienzo de la reunión se terminó de valorar las herramientas sugeridas por el profesor en la reunión anterior. Además también se valoró la decisión tomada respecto al reconocimiento de voz dentro de la aplicación, tomando como buena la solución encontrada para ello.

Debido a que este sprint duró dos semanas a raíz de que la Universidad de Burgos cerraba por vacaciones de Semana Santa se establecieron más objetivos para cubrir trabajo en las dos semanas. El primer objetivo que se estableció fue el de investigar la viabilidad de realizar el cambio de framework de razonamiento basado en casos, ya que es uno de los objetivos del proyecto. A parte de esto, también se estableció como objetivo que fuera creando el repositorio en GitHub y que se fuera importando el anterior TFG para poder empezar a realizar cambios en el código.

### **Sprint 5 (11/04/2018 - 17/04/2018)**

Lo primero que se realizó en esta reunión fue valorar junto al tutor la viabilidad sobre un posible cambio en el framework de persistencia. A parte de esto, también se comentó el trabajo realizado en el repositorio de GitHub.

Como objetivos para este sprint se estableció terminar de organizar correctamente el repositorio en GitHub, ya que se consideró que una buena organización desde el primer momento, evitaría problemas mayores durante el desarrollo del código de la aplicación.

### **Sprint 6 (18/04/2018 - 24/04/2018)**

En esta reunión se comentó la forma en la que finalmente quedó organizado el repositorio de GitHub.

Tras valorar positivamente la forma en la que se había organizado por parte del tutor, se estableció como objetivo empezar con la aplicación Android, teniendo que dar los primeros pasos con ella.

### **Sprint 7 (25/04/2018 - 01/05/2018)**

En la reunión de este sprint se mostró los progresos que se habían realizado en la aplicación Android al tutor, teniendo en este momento un prototipo de la pantalla de inicio de nuestra aplicación.

Como objetivo se estableció que la pantalla de inicio de la aplicación tuviera una animación, haciéndola así que fuera más dinámica. A parte de esto, también se marcó como objetivo comenzar con las pruebas cliente-servidor de nuestra aplicación.

### **Sprint 8 (02/05/2018 - 08/05/2018)**

Al comienzo de la reunión se mostró la forma en la que aparecía la pantalla de inicio y tras la valoración del tutor. Además se comentaron todos los problemas que fueron surgiendo al intentar realizar las pruebas de conexión cliente-servidor.

El objetivo que se estableció para este sprint fue utilizar otra tipo de conexiones cliente-servidor recomendada por el tutor.

### **Sprint 9 (09/05/2018 - 15/05/2018)**

En esta reunión se comentaron los resultados positivos que se obtuvieron al utilizar las recomendaciones dadas por el tutor.

Como objetivo de este sprint fue terminar de realizar la conexión cliente-servidor de nuestra aplicación Android, sin tener que entrar en el tratamiento de datos.

### **Sprint 10 (16/05/2018 - 22/05/2018)**

Durante esta reunión, lo primero que se hizo fue mostrar los resultados de la conexión cliente-servidor con la aplicación Android.

Como objetivo de esta reunión se estableció en realizar la conexión de tal forma en que la aplicación recibiera los datos en forma de un ficheros JSON, para facilitar así el tratamiento de los datos.

### **Sprint 11 (23/05/2018 - 29/05/2018)**

En esta reunión se mostró como la aplicación trataba los datos y los mostraba por pantalla, pudiendo acceder ya a las respuestas dadas por el asistente.

Como objetivo se estableció cambiar el diseño de la aplicación, para dar un aspecto más similar a una conversación, dándole similitudes a aplicaciones de ese estilo como pueden ser *Whatsapp* o *Telegram*.

#### **Sprint 12 (30/05/2018 - 05/06/2018)**

En esta reunión se mostró la nueva interfaz de la aplicación. A parte de esto se mostró una primera versión en la que los enlaces que nos da el asistente, nos los abre dentro de la propia aplicación.

Como objetivos de este sprint, se estableció afinar la forma en la que se abren los enlaces y añadir la valoración de las repuestas, así como el comienzo con la documentación del proyecto.

#### **Sprint 13 (06/06/2018 - 12/06/2018)**

En esta reunión se mostró la nueva forma en la que se muestran los enlaces, así como una primera versión de como funcionaría la ventana de valoración de las respuestas.

Como objetivo, se estableció principalmente continuar de una forma más continuada con los anexos del proyecto.

#### **Sprint 14 (13/06/2018 - 22/06/2018)**

En esta reunión se mostró el estado de los anexos en aquel momento.

Como objetivo, se estableció tener terminada toda la documentación para el día 25 de junio, así como tener implementado el mecanismo de valoración de las respuestas.

#### **Sprint 15 (23/06/2018 - 27/06/2018)**

En esta reunión se mostró el estado final de la aplicación desarrollada, obteniendo la aprobación del tutor para presentar el proyecto.

### **A.3. Estudio de viabilidad**

#### **Viabilidad económica**

En este apartado se van a realizar los diferentes estudios de costes y beneficios del proyecto que se ha realizado.

Este estudio lo vamos a hacer de forma en la que suponemos que lo va a realizar una empresa, por lo que vamos a tener en cuenta tanto los costes humanos, como de los diferentes recursos que se han empleado para realizarlo.

### Costes humanos

Vamos a considerar que la empresa dedica a un empleado a desarrollar la aplicación UBUassistant. Para ello consideramos que el tiempo necesario para el desarrollo de la aplicación es la misma que ha necesitado el alumno para el desarrollo de la misma, es decir, cuatro meses.

Vamos a establecer que el programador encargado de desarrollar nuestro software, es una persona con poca experiencia laboral y que acaba de entrar al mercado laboral. Según un estudio realizado por la página web *www.michaelpage.es*, el salario mínimo para una persona de estas características es de 18000 € al año, lo que supone 1500 € al mes, siendo este valor lo que paga la empresa por él [2]. Según el régimen de la Seguridad Social para el año 2018, de ese salario, la empresa tiene que cotizar un 23,60 %, mientras que el trabajador tiene que cotizar un 4,70 % [3]. Teniendo en cuenta estos datos podemos saber que la empresa hace una cotización a la Seguridad Social de 286.41 €, mientras que el trabajador la hace de 67.34 €, siendo los valores obtenidos cotizados de forma mensual. Con estos datos, sabemos que el sueldo neto del trabajador es de 1169.13 €.

Con todos estos datos, podemos saber lo que le cuesta a la empresa el trabajador dedicado al proyecto durante 4 meses:

Concepto	Coste
Salario mensual bruto	1169.13 €
Seguridad Social (28.3 %)	330.87 €
Total al mes	1500 €
Total 4 meses	6000 €

Tabla A.1: Costes Humanos

**Costes hardware**

En este apartado se tienen en cuenta el costo de las herramientas hardware empleadas para el desarrollo del proyecto.

Concepto	Coste
Ordenador HP [4]	970.42 €
Pantalla [5]	199.00 €
Ratón [6]	19.36 €
Teclado [7]	18.58 €
Smartphone Android para pruebas [8]	109.00 €
Total	1316.36 €

Tabla A.2: Costes Hardware

**Costes alojamiento servidor**

Aquí se tiene en cuenta los costes mensuales del servidor Microsoft Azure donde está alojado el servidor de nuestro asistente virtual.

Concepto	Coste
Servidor Microsoft Azure [9]	33.44 €
Total al mes	33.44 €
Total 4 meses	133.76 €

Tabla A.3: Costes Alojamiento

**Costes publicación**

En este apartado se tiene en cuenta el coste que supone publicar la aplicación para poder descargarla desde *Google Play Store*.

Concepto	Coste
Cuenta desarrollador Google [10]	21.45 €
Total	21.45 €

Tabla A.4: Costes Publicación

**Costes licencias**

En este apartado se tienen en cuenta aquellos softwares que poseen licencias de pago. Los precios hacen referencia al costo mensual.

Concepto	Coste
Adobe Creative Cloud <a href="#">[11]</a>	69.99 €
IVA (21 %)	14.70 €
Total al mes	84.69 €
Total 4 meses	338.76 €

Tabla A.5: Costes Licencias

**Costes redes**

Aquí se tienen en cuenta los gastos relacionados con la conexión de Internet de la oficina [\[12\]](#).

Concepto	Coste
Fibra Óptica 100Mb	33.06 €
Cuota de línea	14.38 €
Total (sin IVA)	47.44 €
IVA (21 %)	9.96 €
Total al mes	57.40 €
Total 4 meses	229.60 €

Tabla A.6: Costes Redes

**Costes oficina**

En este apartado se tiene en cuenta el alquiler de la oficina donde se realizan las actividades de la empresa.

Concepto	Coste
Alquiler oficina	700 €
Total 4 meses	2800 €

Tabla A.7: Costes Oficina

**Costes varios**

En esta sección entran aquellos gastos que no están calificados en ninguno de los apartados anteriores.

Concepto	Coste
Impresión memoria y cartel	30 €
Material oficina	10 €
Total	40 €

Tabla A.8: Costes Varios

**Coste total**

Con todos los costes especificado anteriormente, podemos calcular el coste total del proyecto para la empresa:

Concepto	Coste
Costes humanos	6000.00 €
Costes hardware	1316.36 €
Costes alojamiento servidor	133.76 €
Costes publicación	21.45 €
Costes licencias	338.76 €
Costes redes	229.60 €
Costes oficina	2800.00 €
Costes varios	40.00 €
Total proyecto	10879.93 €

Tabla A.9: Costes totales

### Beneficios

Unas de las ventajas en el desarrollo de esta aplicación es que en el desarrollo del mismo, solo se ha tenido que pagar la *suite* de Adobe CC para la creación de las imágenes del producto. El resto de herramientas de software que se han empleado han sido totalmente gratuitas.

La búsqueda de beneficios de esta aplicación se puede orientar desde diferentes puntos de vista.

- **Venta a terceros:** debido a la fácil escalabilidad de esta aplicación a otros ámbitos, nos permite poder vendérsela a otras empresas, cobrando el precio que se considere oportuno. A parte se puede cobrar a esas empresas un dinero al mes por mantener la aplicación.
- **Uso en la página de la UBU:** gracias a la facilidad de encontrar los resultados con el asistente, va a generar más visitas a la página de la Universidad. Este mayor número de visitas puede repercutir indirectamente en un beneficio económico en la Universidad de Burgos.

### Viabilidad legal

En el caso del desarrollo de una aplicación software, tal y como es este caso, hay que tener en cuenta las diferentes licencias del software que se han empleado para el desarrollo del mismo.

En este proyecto, a excepción de la licencia de Adobe Creative Cloud, todas las licencias empleadas son gratuitas. A continuación se detalla una tabla con el tipo de licencia utilizada por cada software.



Software	Descripción	Licencia
jCOLIBRI 2 [13]	Framework CBR	LGPL
Eclipse IDE [14]	IDE de desarrollo para JAVA	EPL
Android Studio [15]	IDE de desarrollo de aplicaciones Android	Apache 2.0
Apache Tomcat [16]	Servidor JAVA	Apache 2.0
MySQL [17]	Gestor base de datos SQL	GNU GPL
JAVA JDK y JRE [18]	Lenguaje de programación	GNU GPL
Apache Commons [19]	Librería JAVA	Apache 2.0
JSON [20]	Formato de texto de intercambio de datos	The JSON License
Jersey [21]	Servicios web RESTful en JAVA	GNU GPL
JUnit [22]	Librería de test para Java	EPL
Hibernate [23]	Herramienta de mapeo objeto-relacional	LGPL
Log4j [24]	Herramienta de logging en JAVA	Apache 2.0
Selenium [25]	Framework de pruebas automáticas web	Apache 2.0
Guava [26]	Conjunto de bibliotecas para JAVA	Apache 2.0
Maven [27]	Herramienta de gestión de proyectos JAVA	Apache 2.0

Tabla A.10: Licencias de software

En la tabla anterior, podemos ver diferentes tipos de licencia, las cuales tienen diferentes cláusulas, las cuales vamos a explicar ahora [28] [29]:

- **LGPL (Lesser General Public License):** GPL sin copyleft la cuál permite enlazar con módulos no libres.
- **EPL (Eclipse Public License):** Licencia libre, pero con patentes.
- **Apache 2.0:** Licencia libre y abierta, pero con patentes.
- **GNU GPL:** Licencia libre, abierta y con copyleft.
- **The JSON License:** Licencia igual que Expat (libre, simple, permisiva y sin copyleft), pero con la cláusula añadida «*El software deberá ser utilizado para el Bien y no para el Mal*».

En algunas licencias aparece el término *copyleft*, este término hace referencia a que si un software es libre y se hace una modificación sobre él, este también tiene que ser libre.

## Documentación

Debido a que este software es una modificación de uno anterior, hay que mantener el mismo tipo de licencia. La licencia elegida para este proyecto fue *Creative Commons*, eligiendo de todas las variantes la *Reconocimiento-NoComercial-CompartirIgual 3.0 España (CC BY-NC-SA 3.0 ES)*. Esta licencia permite la evolución del software con reconocimiento de autoría, im-

pidiendo el uso comercial del mismo y sus derivados y obliga a que todas las obras derivadas tienen que utilizar esta misma licencia [\[30\]](#).

---

## Especificación de Requisitos

---

### B.1. Introducción

En este apartado se va a proceder a explicar de forma detalla como funciona nuestra aplicación y el comportamiento que tiene la misma mientras se usa.

Se va a detallar los diferentes casos de uso que existen para cada una de las actividades que puede realizar un usuario.

Para facilitar la comprensión en este apartado, se va a intentar emplear un lenguaje informal, intentando llegar a cualquier tipo de público.

Los requisitos que nos podemos encontrar son de dos tipos:

- **Requisitos funcionales:** son aquellos requisitos que dictaminan los diferentes servicios que debe de ofrecer un software. Estos requisitos están relacionados con los diferentes casos de uso.
- **Requisitos no funcionales:** son aquellos que marcan las pautas y las restricciones de diseño y/o implementación del software.

### B.2. Objetivos generales

Los objetivos marcados para el desarrollo de este proyecto fueron:

- Utilizar la voz para introducir el texto de las búsquedas.
- Ampliar el número máximo de vocablos utilizados para realizar las búsquedas.
- Disponer de la aplicación en dispositivos móviles con sistema operativo Android.

- Analizar otros frameworks de Razonamiento Basado en Casos, como se sugirió en la defensa del proyecto e implementarlo en función del resultado.
- Independizar la capa de aplicación de la interfaz de presentación que permita implementar el sistema cliente en distintos dispositivos móviles y distintas tecnologías como HTML5.
- Minimizar los posibles ataques de denegación de servicio.
- Mantener la imagen corporativa.

### B.3. Catalogo de requisitos

El proyecto que se ha desarrollado posee diferentes requisitos de uso, tanto funcionales como no funcionales. Dentro de todos estos requisitos, hay uno que es fundamental en el desarrollo del proyecto. Este requisito, es que la aplicación cliente que se desarrolle, tiene que tener los mismos requisitos que poseía el asistente web realizado por Daniel Santidrián en la versión del proyecto que se ha mejorado.

A parte de este requisito, otro muy importante durante el desarrollo de la aplicación, se ha tenido en cuenta que ésta sea compatible con cualquier dispositivo Android con versión igual o superior a 4.4.

Con esta premisa, los requisitos funcionales de nuestra aplicación son:

- **RF-1 Reconocimiento de texto mediante voz:** La aplicación debe de ser capaz de reconocer la voz del usuario y convertirlo a texto.
- **RF-2 Reconocimiento de preguntas:** La aplicación tiene que ser capaz de reconocer las preguntas del usuario.
- **RF-3 Obtener respuesta:** El sistema debe de ser capaz de obtener respuestas a la pregunta realizada por el usuario.
- **RF-4 Realizar recomendaciones:** El sistema debe de ser capaz de sugerir respuestas recomendadas cuando el asistente obtiene varias respuesta o no obtiene respuestas.
- **RF-5 Navegador web:** La aplicación cliente tiene que ser capaz de abrir los enlaces ofrecidos por el asistente sin recurrir a un navegador web externo.
- **RF-6 Valoración de respuestas:** La aplicación tiene que ser capaz de mostrarle al usuario la posibilidad de valorar la respuesta dada por parte del asistente.
- **RF-7 Reiniciar palabras de búsqueda:** La aplicación debe de permitir al usuario que reinicie las palabras con las que se efectúan la búsqueda, permitiendo así, empezar una nueva búsqueda desde cero.
- **RF-8 Gestión del log:** El administrador de la aplicación debe de ser capaz de gestionar el registro de las búsquedas realizadas por el usuario.

- **RF-8.1 Listar log:** La aplicación debe ser capaz de listar el registro de las búsquedas realizadas.
- **RF-9 Gestión de casos:** La aplicación tiene que permitir gestionar a través de una interfaz los diferentes casos que se encuentran almacenados.
  - **RF-9.1 Listar casos:** La interfaz tiene que ser capaz de listar todos los casos almacenados correctamente.
  - **RF-9.2 Añadir caso:** El sistema debe permitir añadir nuevos casos a la base de datos
  - **RF-9.3 Editar caso:** La aplicación tiene que permitir al administrador modificar los casos almacenados en la base de datos.
- **RF-10 Gestión de aprendizaje:** El sistema tiene que permitir que el administrador pueda gestionar el aprendizaje del algoritmo.
  - **RF-10.1 Aprender recomendación:** La aplicación tiene que ser capaz de aprender las diferentes recomendaciones que hayan sido aprobadas por el administrador.

A parte de los anteriores requisitos funcionales, tenemos requisitos no funcionales:

- **RNF-1 Seguridad:** El servidor tiene que estar protegido ante los posibles ataques de denegación de servicio.
- **RNF-2 Mejorar búsqueda:** Ampliar el número de vocablos de 5 a 7, para facilitar así al usuario las búsquedas.
- **RNF-3 Adaptabilidad:** La aplicación tiene que ser adaptable fácilmente a diferentes plataformas.
- **RNF-4 Imagen corporativa:** Mantener la imagen corporativa de la Universidad de Burgos.
- **RNF-5 Registro de actividad:** La aplicación debe de ser capaz de registrar los diferentes enlaces accedidos por un usuario.
- **RNF-6 Fácil manejo:** La aplicación tiene que ser sencilla de utilizar, tanto como para poder enviar las preguntas, como de visualizar las respuestas ofrecidas por el asistente.
- **RNF-7 Servicio virtualizado:** El servidor de la aplicación tiene que estar virtualizado en la nube, como *Microsoft Azure*, permitiendo así que sea accesible desde cualquier localización.

## B.4. Especificación de requisitos

### Diagrama de casos de uso

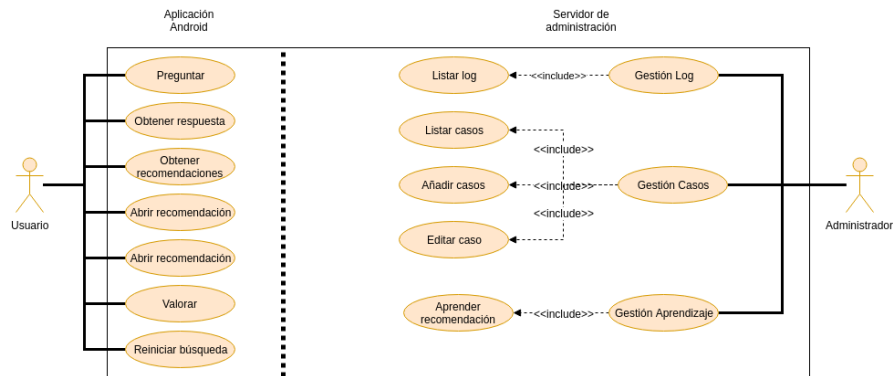


Figura B.1: Diagrama de casos de uso

### Actores

En el sistema que se ha desarrollado, nos podemos encontrar con dos perfiles actores totalmente diferenciados:

- **Usuario:** El usuario es el que se encarga de utilizar la aplicación Android para realizar las diferentes búsquedas que considere pertinentes.
- **Administrador:** El administrador utiliza la interfaz web con sus credenciales, para gestionar los diferentes ámbitos del asistente de búsqueda.

## Casos de uso

CU-01	Preguntar
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-1 & RF-2
<b>Descripción</b>	Permite al usuario introducir texto en la interfaz del asistente para realizar una pregunta.
<b>Precondiciones</b>	Se ha conectado con la base de datos.
<b>Acciones</b>	El usuario abre la aplicación Android. Introduce texto o lo dicta por voz. Enviar la pregunta.
<b>Postcondiciones</b>	Obtener resultados de la respuesta.
<b>Excepciones</b>	Servidor web no disponible.

Tabla B.1: CU-01 Preguntar

CU-02	Obtener respuesta
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-3
<b>Descripción</b>	El usuario obtiene una respuesta a la pregunta realizada.
<b>Precondiciones</b>	Se ha conectado con la base de datos. Existe respuesta única para la pregunta realizada.
<b>Acciones</b>	El usuario abre la aplicación Android. Introduce texto o lo dicta por voz. Enviar la pregunta. Obtener la respuesta a la pregunta.
<b>Postcondiciones</b>	Se almacena la respuesta en el log del servidor
<b>Excepciones</b>	Servidor web no disponible.

Tabla B.2: CU-02 Obtener respuesta

<b>CU-03</b>	<b>Obtener recomendación</b>
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-3 & RF-4
<b>Descripción</b>	El usuario obtiene múltiples posibles respuestas a la pregunta realizada o recomendaciones por que la pregunta no tiene respuesta.
<b>Precondiciones</b>	Se ha conectado con la base de datos. Existe múltiples respuestas o no existen respuestas para la pregunta realizada.
<b>Acciones</b>	El usuario abre la aplicación Android. Introduce texto o lo dicta por voz. Enviar la pregunta. El servidor no obtiene respuesta o encuentra múltiples respuestas.
<b>Postcondiciones</b>	Se muestran las recomendaciones a la pregunta.
<b>Excepciones</b>	Se almacena la respuesta en el log del servidor Servidor web no disponible.

Tabla B.3: CU-03 Obtener recomendación

<b>CU-04</b>	<b>Visualizar respuesta</b>
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-5
<b>Descripción</b>	El usuario puede visualizar la página web dada como respuesta desde la propia aplicación.
<b>Precondiciones</b>	Se ha conectado con la base de datos. Existe respuesta única para la pregunta realizada.
<b>Acciones</b>	El usuario abre la aplicación Android. Introduce texto o lo dicta por voz. Enviar la pregunta. Obtener la respuesta a la pregunta. El usuario selecciona la respuesta.

continúa en la página siguiente



continúa desde la página anterior

<b>CU-04</b>	<b>Visualizar respuesta</b>
<b>Postcondiciones</b>	La respuesta se abre dentro de la propia aplicación.
<b>Excepciones</b>	Se valora la respuesta recibida. Servidor web no disponible.

Tabla B.4: CU-04 Visualizar respuesta

<b>CU-05</b>	<b>Visualizar recomendación</b>
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-5
<b>Descripción</b>	El usuario puede visualizar la página web respuesta que seleccione entre las diferentes recomendaciones realizadas por el asistente.
<b>Precondiciones</b>	Se ha conectado con la base de datos. Existe múltiples respuestas o no existen respuestas para la pregunta realizada.
<b>Acciones</b>	El usuario abre la aplicación Android. Introduce texto o lo dicta por voz. Enviar la pregunta. El servidor no obtiene respuesta o encuentra múltiples respuestas. Se muestran las recomendaciones a la pregunta. El usuario selecciona una de las recomendaciones realizadas. La recomendación seleccionada se abre dentro de la propia aplicación.
<b>Postcondiciones</b>	Se valora la recomendación seleccionada.
<b>Excepciones</b>	Servidor web no disponible.

Tabla B.5: CU-05 Visualizar recomendación

<b>CU-06</b>	<b>Valorar respuesta</b>
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-6
<b>Descripción</b>	El usuario puede valorar la respuesta o la recomendación que ha seleccionado.
<b>Precondiciones</b>	Se ha conectado con la base de datos. Existe múltiples respuestas o no existen respuestas para la pregunta realizada.
<b>Acciones</b>	La página web de la respuesta se encuentra disponible. El usuario abre la aplicación Android. Introduce texto o lo dicta por voz. Enviar la pregunta. El servidor no obtiene respuesta o encuentra múltiples respuestas (recomendación) o tiene respuesta única. Se muestran las recomendaciones o la respuesta a la pregunta. El usuario selecciona una de las recomendaciones realizadas o la respuesta. La recomendación seleccionada o respuesta se abre dentro de la propia aplicación. El usuario vuelve al asistente.
<b>Postcondiciones</b>	Valora la respuesta que se le ha otorgado.
<b>Excepciones</b>	Se almacena la valoración en la base de datos. Servidor web no disponible.

Tabla B.6: CU-06 Valorar respuesta

<b>CU-07</b>	<b>Reiniciar búsqueda</b>
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-7
<b>Descripción</b>	El usuario debe de ser capaz de reiniciar las palabras que se están utilizando para hacer la búsqueda.

continúa en la página siguiente

continúa desde la página anterior

CU-07	Reiniciar búsqueda
<b>Precondiciones</b>	Se han realizado búsquedas con anterioridad.
<b>Acciones</b>	El usuario abre la aplicación Android. Introduce texto o lo dicta por voz. Pulsa sobre el botón con el texto <i>Nueva</i> .
<b>Postcondiciones</b>	Se realiza la búsqueda con las nuevas palabras.
<b>Excepciones</b>	No se han realizado búsquedas con anterioridad.

Tabla B.7: CU-07 Reiniciar búsqueda

CU-08	Gestión del log
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-8 & RF-8.1
<b>Descripción</b>	Permite al administrador gestionar el registro de las búsquedas realizadas con el asistente virtual.
<b>Precondiciones</b>	La base de datos se encuentra accesible. El administrador se ha autenticado en el sistema con sus credenciales.
<b>Acciones</b>	El usuario accede a la página web. El usuario abre la interfaz de administración. El usuario se autentica en el sistema con sus credenciales de administrador. El administrador accede a la página de gestión del log. Se listan las diferentes búsquedas realizadas con el asistente virtual. Se permite ordenar las búsquedas por diferentes campos. Se permite exportar la lista a diferentes formatos. Se permite borrar la lista de búsquedas realizadas.
<b>Postcondiciones</b>	El número de logs que se muestran es igual al número de logs que se encuentra en la base de datos.
<b>Excepciones</b>	La base de datos no se encuentra accesible.

Tabla B.8: CU-08 Gestión del log

CU-09	Listar log
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-8.1
<b>Descripción</b>	Permite al administrador visualizar en forma de tabla el registro de todas las búsquedas que ha realizado un usuario utilizando el asistente virtual.
<b>Precondiciones</b>	La base de datos se encuentra accesible. El administrador se ha autenticado en el sistema con sus credenciales.
<b>Acciones</b>	El usuario accede a la página web. El usuario abre la interfaz de administración. El usuario se autentica en el sistema con sus credenciales de administrador. El administrador accede a la página de gestión del log. Se listan las diferentes búsquedas realizadas con el asistente virtual.
<b>Postcondiciones</b>	El número de logs que se muestran es igual al número de logs que se encuentra en la base de datos.
<b>Excepciones</b>	La base de datos no se encuentra accesible.

Tabla B.9: CU-09 Listar log

CU-10	Gestión de casos
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-9, RF-9.1, RF-9.2 & RF-9.3
<b>Descripción</b>	Permite al administrador gestionar los diferentes casos que son utilizados por el asistente virtual.
<b>Precondiciones</b>	La base de datos se encuentra accesible. El administrador se ha autenticado en el sistema con sus credenciales.
<b>Acciones</b>	El usuario accede a la página web. El usuario abre la interfaz de administración. El usuario se autentica en el sistema con sus credenciales de administrador.

continúa en la página siguiente

continúa desde la página anterior

CU-10	Gestión de casos
	<p>El administrador accede a las páginas de la sección <i>Editor de casos</i></p> <p>Si se accede a la página <i>Añadir Caso</i>, se muestra un formulario, con el cual se puede añadir un nuevo caso a la base de datos para que lo pueda utilizar el asistente virtual.</p> <p>Si se accede a la página <i>Modificar Casos</i>, se muestra en forma de tabla los diferentes casos que utiliza el asistente virtual.</p> <p>Se muestran diferentes menús con los que se permite editar o eliminar cada uno de los diferentes casos.</p> <p>Se muestra una opción que permite exportar la tabla a diferentes formatos.</p>
<b>Postcondiciones</b>	El número de casos que se muestran es igual al número de casos que se encuentra en la base de datos.
<b>Excepciones</b>	La base de datos no se encuentra accesible.

Tabla B.10: CU-10 Gestión de casos

CU-11	Listar casos
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-9.1
<b>Descripción</b>	Permite al administrador ver en forma de tabla los casos almacenados en la base de datos.
<b>Precondiciones</b>	<p>La base de datos se encuentra accesible.</p> <p>El administrador se ha autenticado en el sistema con sus credenciales.</p>
<b>Acciones</b>	<p>El usuario accede a la página web.</p> <p>El usuario abre la interfaz de administración.</p> <p>El usuario se autentica en el sistema con sus credenciales de administrador.</p> <p>El administrador accede a la página <i>Modificar Casos</i>.</p>
<b>Postcondiciones</b>	<p>Se muestran todos los casos almacenados en forma de tabla.</p> <p>El número de casos que se muestran es igual al número de casos que se encuentra en la base de datos.</p>

continúa en la página siguiente

continúa desde la página anterior

CU-11	Listar casos
Excepciones	La base de datos no se encuentra accesible.

Tabla B.11: CU-11 Listar casos

CU-12	Añadir caso
Versión	1.0
Autor	Carlos González Calatrava
Requisitos asociados	RF-9.2
Descripción	Permite al administrador añadir un nuevo caso a la base de datos para que lo pueda utilizar el asistente virtual.
Precondiciones	La base de datos se encuentra accesible. El administrador se ha autenticado en el sistema con sus credenciales.
Acciones	El usuario accede a la página web. El usuario abre la interfaz de administración. El usuario se autentica en el sistema con sus credenciales de administrador. El administrador accede a la página <i>Añadir Caso</i> . Se muestra el formulario para poder añadir un nuevo caso. El administrador rellena los campos obligatorios y aquellos opcionales que considere oportunos.
Postcondiciones	Pulsa sobre el botón de <i>Aceptar</i> para añadir el nuevo caso. El caso que se ha añadido se encuentra en la base de datos.
Excepciones	La base de datos no se encuentra accesible. Los campos marcados como obligatorios no se han rellenado.

Tabla B.12: CU-12 Añadir caso

CU-13	Editar caso
Versión	1.0
Autor	Carlos González Calatrava
Requisitos asociados	RF-9.3

continúa en la página siguiente

continúa desde la página anterior

CU-13	Editar caso
<b>Descripción</b>	Permite al administrador editar un caso que ya existe en la base de datos.
<b>Precondiciones</b>	La base de datos se encuentra accesible. El administrador se ha autenticado en el sistema con sus credenciales.
<b>Acciones</b>	El caso que se quiere modificar, existe. El usuario accede a la página web. El usuario abre la interfaz de administración. El usuario se autentica en el sistema con sus credenciales de administrador. El administrador accede a la página <i>Modificar Caso</i> . Se muestra la tabla con todos los casos almacenados en la base de datos. El administrador pulsa sobre el botón <i>Editar</i> del caso que quiere modificar. Se muestra un formulario con los datos del caso a modificar. El administrador cambia o rellena los campos que considere oportuno, dejando siempre rellenos los campos que son obligatorios. El administrador pulsa sobre el botón <i>Aceptar</i> para modificar el caso.
<b>Postcondiciones</b>	El caso que se ha añadido se encuentra en la base de datos.
<b>Excepciones</b>	La base de datos no se encuentra accesible. Los campos marcados como obligatorios no se han rellenado.

Tabla B.13: CU-13 Editar caso

CU-14	Gestión de aprendizaje
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-10 & RF-10.1
<b>Descripción</b>	Permite al administrador gestionar los diferentes casos que son utilizados por el asistente virtual.
<b>Precondiciones</b>	La base de datos se encuentra accesible. El administrador se ha autenticado en el sistema con sus credenciales.
<b>Acciones</b>	El usuario accede a la página web. El usuario abre la interfaz de administración. El usuario se autentica en el sistema con sus credenciales de administrador. El administrador accede a la página <i>Aprendizaje</i> . Se muestra una tabla con todas las recomendaciones que ha ido recogiendo el asistente virtual. Se muestran opciones para aprender o descartar la recomendación de aprendizaje. Se muestra una opción para poder exportar la tabla de recomendaciones a diferentes formatos.
<b>Postcondiciones</b>	El número de recomendaciones que se muestran es igual al número de recomendaciones que se encuentra en la base de datos.
<b>Excepciones</b>	La base de datos no se encuentra accesible.

Tabla B.14: CU-14 Gestión de aprendizaje

CU-15	Aprender recomendación
<b>Versión</b>	1.0
<b>Autor</b>	Carlos González Calatrava
<b>Requisitos asociados</b>	RF-10.1
<b>Descripción</b>	Permite al administrador considerar como buena la recomendación de aprendizaje del asistente virtual.
<b>Precondiciones</b>	La base de datos se encuentra accesible.

continúa en la página siguiente



continúa desde la página anterior

CU-15	Aprender recomendación
<b>Acciones</b>	<p>El administrador se ha autenticado en el sistema con sus credenciales.</p> <p>Existe al menos una recomendación de aprendizaje.</p> <p>El usuario accede a la página web.</p> <p>El usuario abre la interfaz de administración.</p> <p>El usuario se autentica en el sistema con sus credenciales de administrador.</p> <p>El administrador accede a la página <i>Aprendizaje</i>.</p> <p>Se muestra una tabla con todas las recomendaciones que ha ido recogiendo el asistente virtual.</p> <p>El administrador pulsa sobre la opción <i>Aprender</i> de la recomendación que desee.</p>
<b>Postcondiciones</b>	<p>El número de recomendaciones que se muestran es igual al número de recomendaciones que se encuentra en la base de datos.</p> <p>El nuevo caso se encuentra disponible en la base de datos.</p> <p>La recomendación ha sido eliminada de la interfaz de aprendizaje.</p>
<b>Excepciones</b>	<p>La base de datos no se encuentra accesible.</p>

Tabla B.15: CU-15 Aprender recomendación

---

## Especificación de diseño

---

### C.1. Introducción

En este apartado se va a detallar como se han estructurado los datos, los procedimientos y la arquitectura del proyecto que se ha desarrollado.

### C.2. Diseño de datos

En este proyecto, nos podemos encontrar con dos entidades que se relacionan entre ellas.

- **CaseDescription:** esta entidad es la encargada de almacenar los diferentes casos descritos para el algoritmo de IA que lleva el asistente. Para definir el caso lleva un identificador, así como siete palabras clave que describen cada uno de los diferentes casos.
- **CaseSolution:** esta entidad se encarga de almacenar las diferentes respuestas que da el algoritmo de IA a cada uno de los diferentes casos descriptivos. Los atributos de esta entidad son un identificador y una respuesta, que por defecto en todos los casos es una dirección web.

La relación entre estas dos entidades está realizada a través de Hibernate, sin necesidad de emplear ninguna clave externa para poder hacerlo.

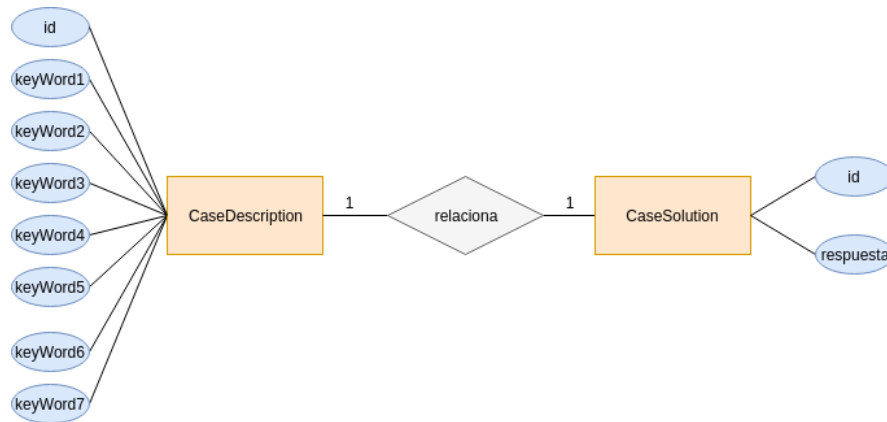


Figura C.1: Diagrama E/R

A parte de las dos entidades que hemos mencionado con anterioridad, existen otras entidades representadas en forma de tablas.

- **aprendizaje:** Esta entidad se encarga de almacenar los diferentes casos que puede aprender nuestro algoritmo. Para almacenar los diferentes casos, se emplea un identificador (*id*), el identificador del usuario que ha propuesto el aprendizaje (*userid*) y dos palabras que definen el caso (*palabra1* & *palabra2*).
- **logger:** La función de esta entidad es almacenar un registro de todas las respuestas accedidas por los usuarios. Para ello, se utiliza un identificador (*id*), el identificador del usuario que ha accedido a la respuesta (*userid*), la fecha de acceso a la respuesta (*fecha*), las siete posibles palabras que pueden describir el caso (*keyWord1* - *keyWord7*), la categoría de la respuesta dada (*categoria*), la respuesta accedida (*respuesta*), el número de veces que se ha accedido a la respuesta (*num\_busquedas*), el número de veces que se ha votado esa respuesta (*num\_votos*) y la valoración total que ha recibido la respuesta (*valoracion\_total*).
- **frases:** Esta entidad se encarga de almacenar una serie de frases que se otorga junto a la respuesta, para dar así un aspecto más humano a nuestro asistente. Para ello, se utiliza un identificador (*id*) y la frases que se ofrecen junto a la pregunta (*frase*).
- **saludos:** Esta entidad se encarga de recoger los diferentes saludos que puede ofrecer el asistente virtual al usuario. Para almacenar estos saludos, se emplea un identificador (*id*), un campo que es el saludo introducido por el usuario (*saludo*) y la respuesta que da el asistente virtual (*respuesta*).

### C.3. Diseño procedimental

En este apartado se detalla como pueden interacciones los diferentes participantes con el sistema.

Vamos a explicar cuál es el funcionamiento del usuario con la aplicación Android para realizar las búsquedas. Para ello, vamos a ver como funciona el algoritmo de búsqueda de respuestas, empleando diferentes diagramas de secuencia.

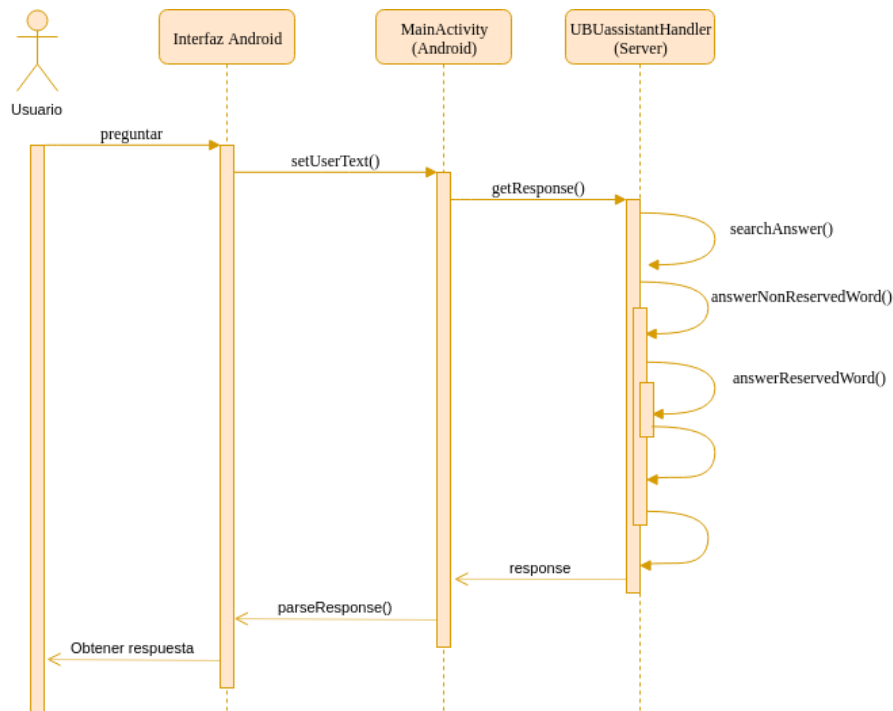


Figura C.2: Diagrama de secuencia para una palabra reservada (Ej. *Buenos días*).

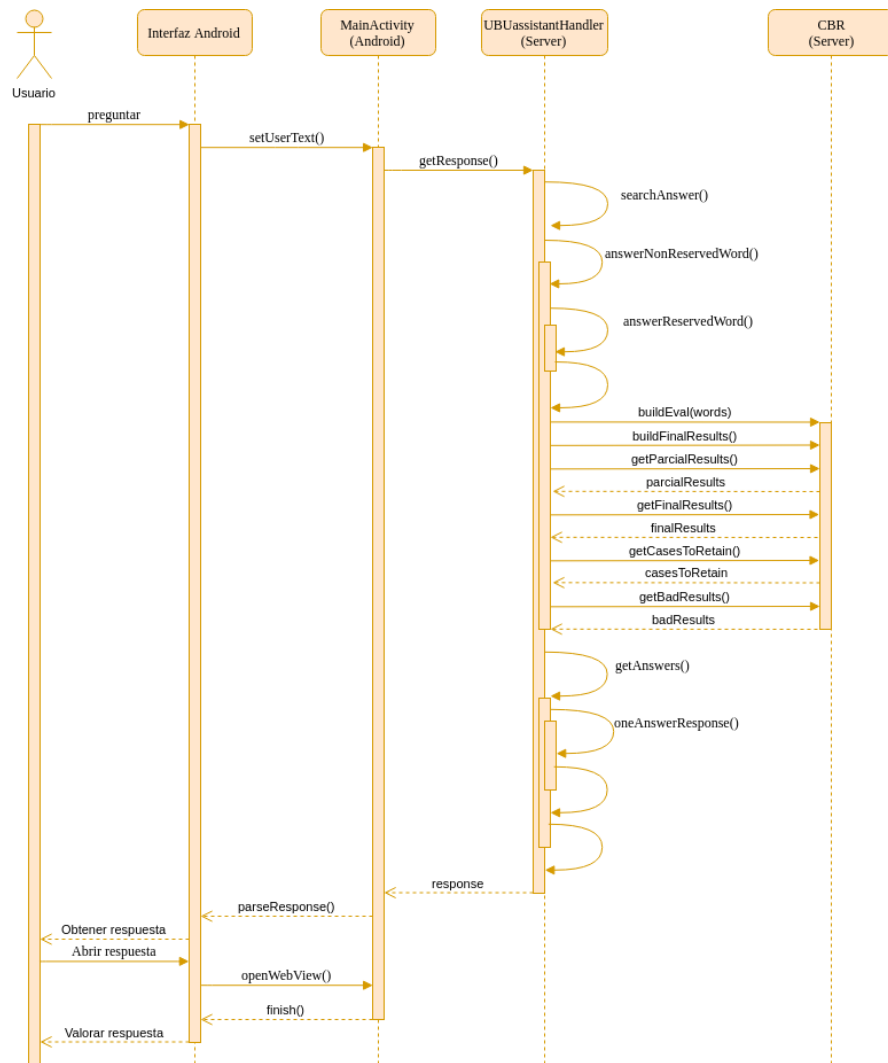


Figura C.3: Diagrama de secuencia para una pregunta con una única respuesta.

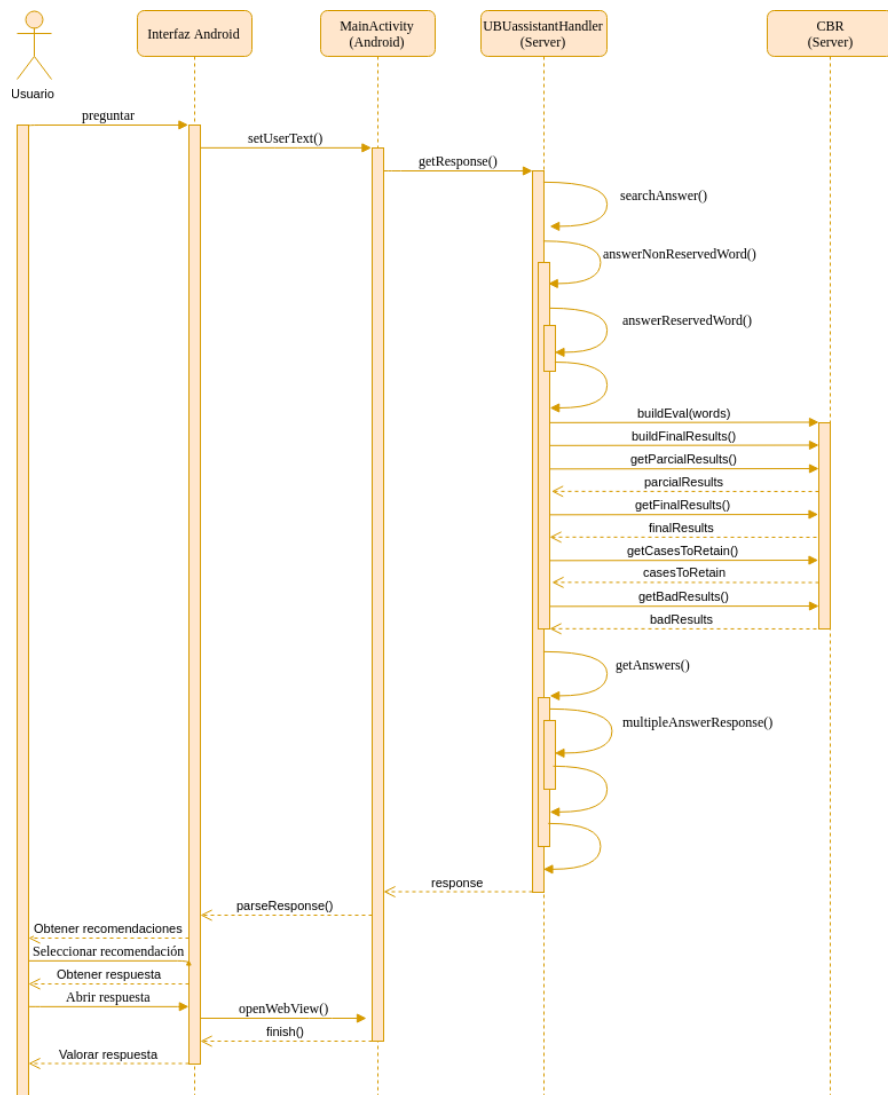


Figura C.4: Diagrama de secuencia para una pregunta con múltiples respuestas.

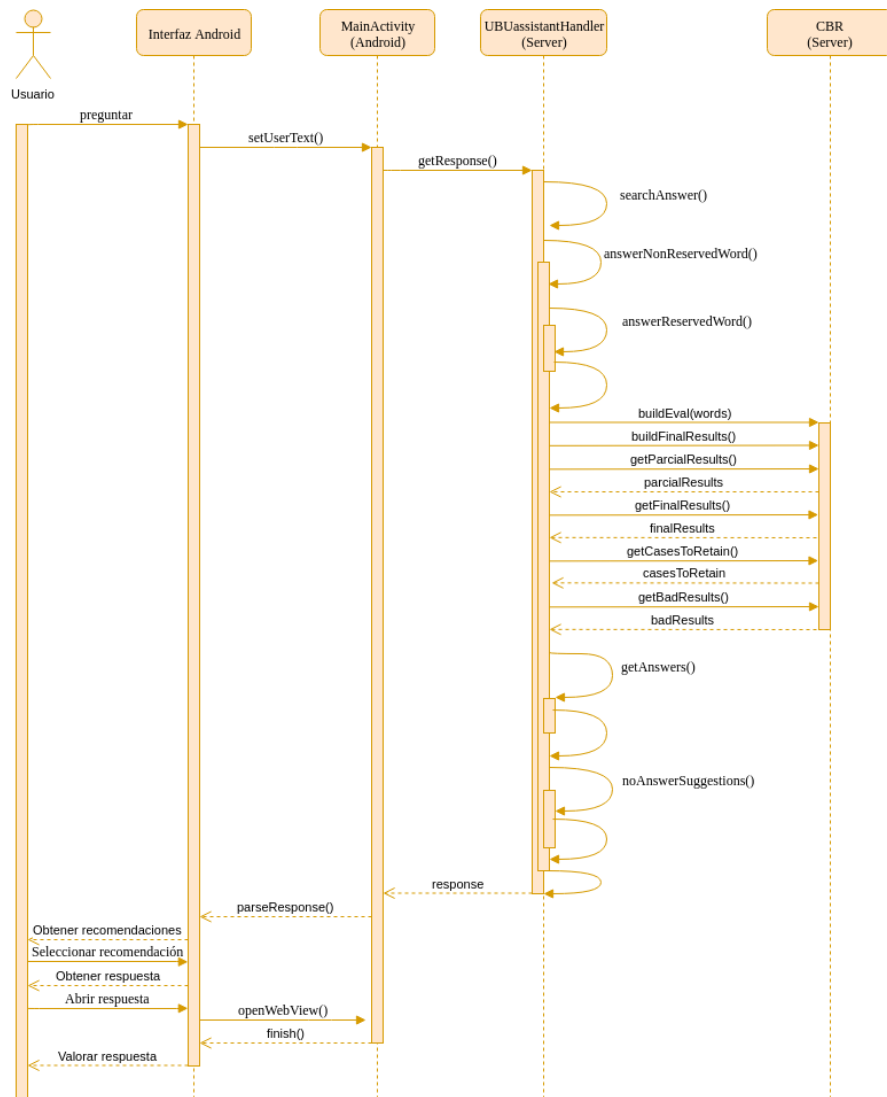


Figura C.5: Diagrama de secuencia para una pregunta que no tiene respuesta.

## C.4. Diseño arquitectónico

Debido a que uno de los objetivos del proyecto es que la aplicación sea fácilmente adaptable a diferentes plataformas, desde un primer momento se decidió emplear *MVC*. De esta manera, se tiene bien diferenciadas la capa de datos, la capa lógica y la capa de interfaz.

### Modelo Vista Controlador

El Modelo-vista-controlador (*MVC*) es un patrón de arquitectura, el cual separa los datos de la lógica de negocio y de la interfaz, para así poder realizar de forma correcta la reutilización de código y la separación de conceptos [31].

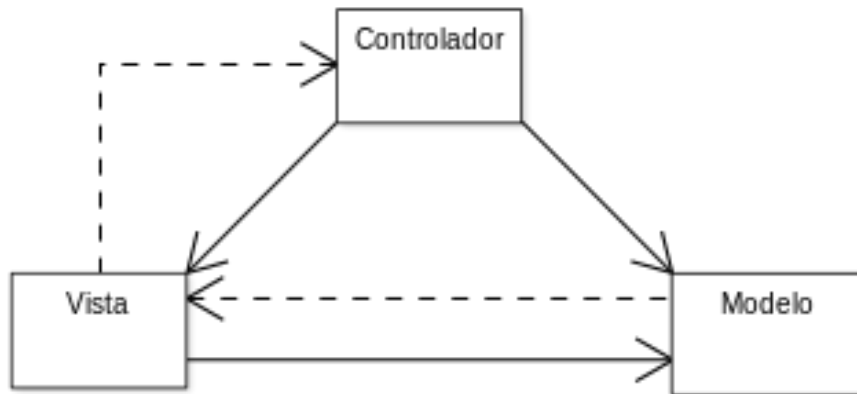


Figura C.6: Modelo-vista-controlador

El significado de cada uno de los componentes de este diseño arquitectónico es el siguiente:

- **Modelo:** Representa los diferentes datos con los que trabaja la aplicación, por lo que esta parte se encarga de gestionar todos los accesos a dicha información.
- **Vista:** Representa la lógica de negocio y la información de una forma correcta al usuario, por lo que normalmente es una interfaz de usuario.
- **Controlador:** Representa los diferentes eventos que se pueden dar en la aplicación, por lo que se encarga de realizar las invocaciones pertinentes al modelo, además también puede comunicarse con la vista para realizar modificaciones de la representación de datos sobre ella. En definitiva, este componente es el que se encarga de hacer de intermediario entre la vista y el modelo.



Para la aplicación de usuario, se ha empleado una aplicación para sistema operativo Android. Aplicando el concepto de MVC a la aplicación, obtenemos un sistema que cumple el siguiente diagrama:

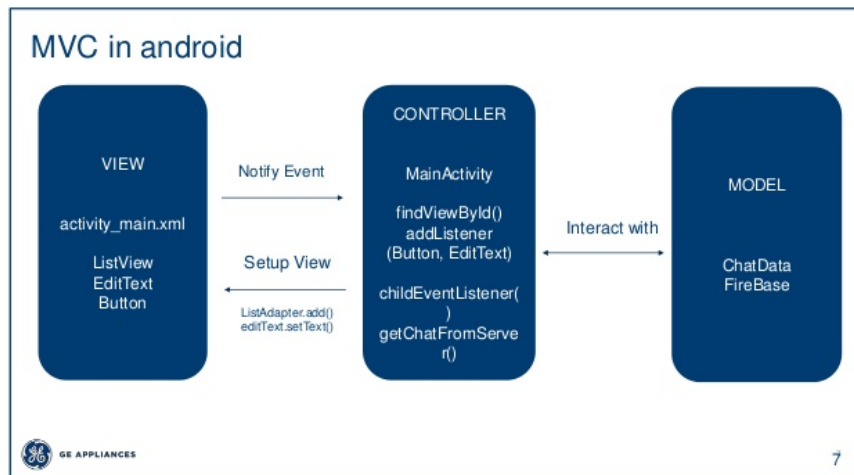


Figura C.7: Diagrama MVC de una aplicación Android

En cambio, el administrador emplea una aplicación web diseñada en JSP, con la cual realiza diferentes tareas. Si aplicamos el concepto de MVC a esta aplicación web, obtenemos el siguiente diagrama:

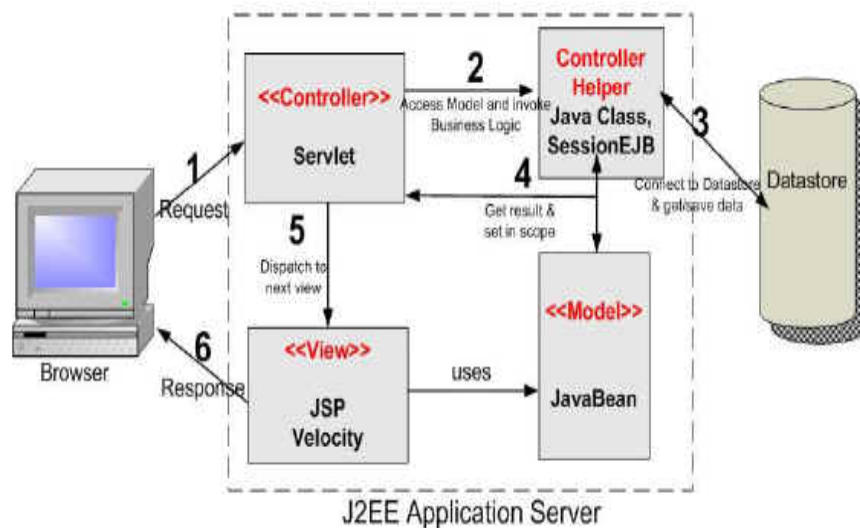


Figura C.8: Diagrama MVC en una aplicación web JSP

### Arquitectura de la aplicación de usuario

Para la realización de la aplicación de usuario, se ha realizado una aplicación Android.

Dentro de las aplicaciones Android, se pueden emplear diferentes lenguajes para realizar la programación de la misma, pero para este proyecto se ha decidido que la aplicación esté realizada en JAVA, debido a que el servidor se encuentra programado en ese mismo lenguaje.

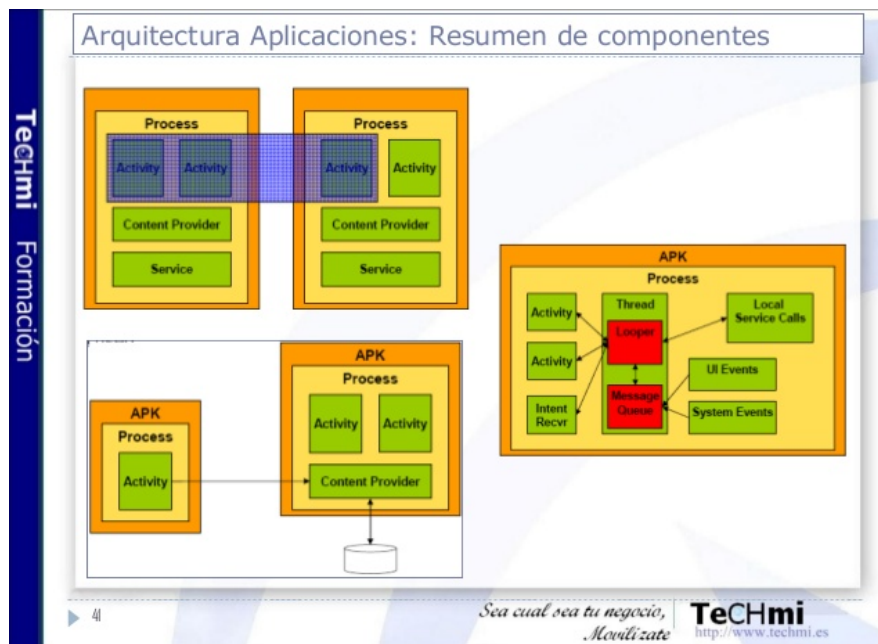


Figura C.9: Arquitectura de una aplicación Android

### Arquitectura de la aplicación web

Para la aplicación web de administración se ha desarrollado utilizando JSP.

JSP (JavaServer Pages) es una tecnología que permite desarrollar páginas web dinámicas basadas en HTML y XML, en las cuales se permite alojar clases JAVA, las cuales manejan la lógica de negocio y el acceso a los datos.

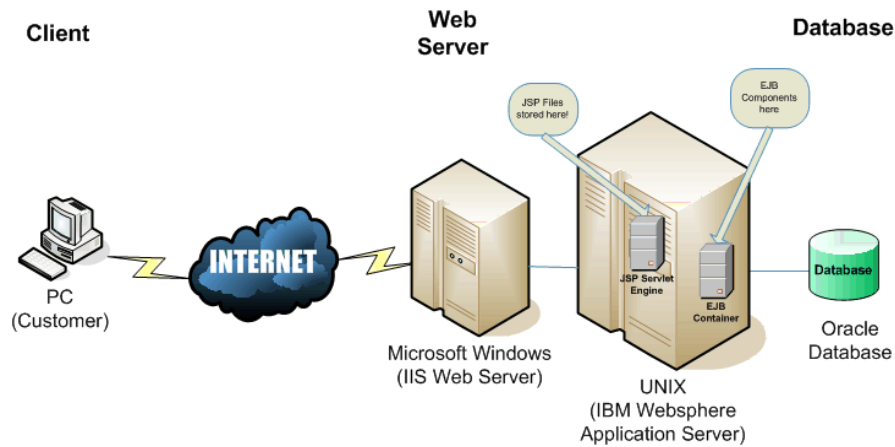


Figura C.10: Arquitectura de una aplicación web basada en JSP

## Estructura de paquetes

El sistema está agrupado en paquetes, que comparten funcionalidades, facilitando así la comprensión del mismo. Hay que tener en cuenta que el sistema está separado en dos, por una parte la aplicación de usuario, mientras por otro lado está el servidor de la aplicación junto a la web de administración.

## Aplicación de usuario

La aplicación de usuario, tiene las clases dentro de un mismo paquete, mientras que en otro directorio, podemos encontrar los diferentes recursos que se han empleado para la confección de la aplicación. A parte, también tiene todos los recursos utilizados para la aplicación agrupados en diferentes carpetas, clasificados según su finalidad.

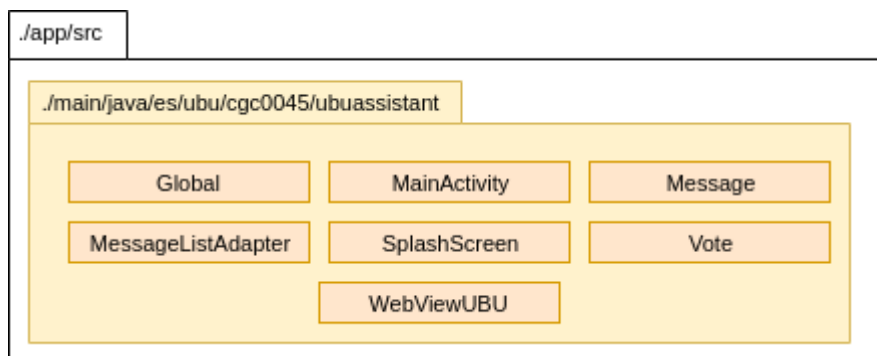


Figura C.11: Diagrama de estructura de los paquetes en Android

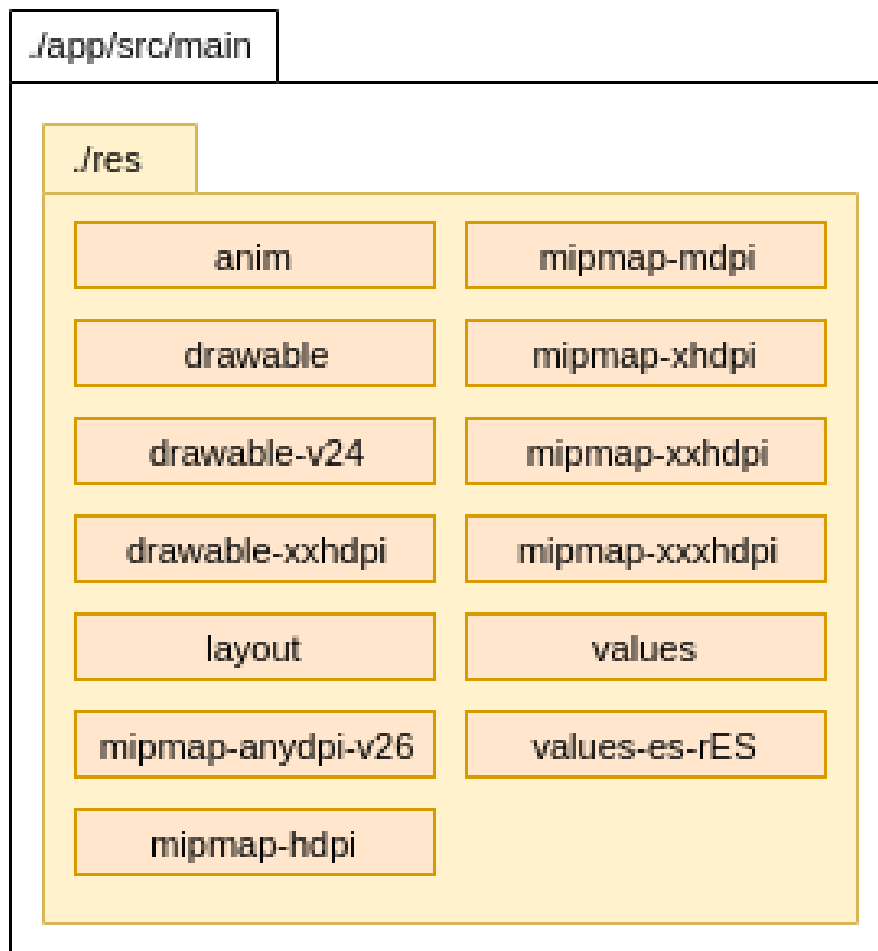


Figura C.12: Diagrama de estructura de los recursos

### Servidor web

Al igual que en la aplicación Android, en el servidor, se han agrupado las clases según las que comparten funcionalidad, permitiendo así, que sea más fácil la comprensión del sistema que se ha desarrollado. Teniendo en cuenta esto, obtenemos el siguiente diagrama de paquetes de nuestro servidor.

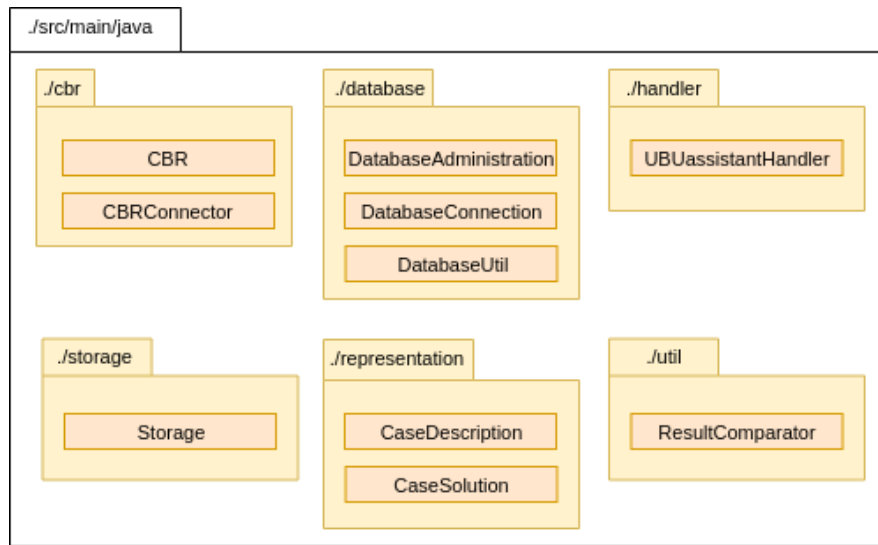


Figura C.13: Diagrama de estructura de los paquetes en el servidor

El contenido de cada uno de los paquetes es el siguiente:

- **cbr:** Contiene la lógica del algoritmo de razonamiento basado en casos.
- **database:** Contiene las clases que se encargan de la comunicación con la base de datos MySQL. Tenemos separado en dos clases las conexiones encargadas para que funcione el asistente virtual (*DatabaseConnection*), de las conexiones encargadas de la administración del algoritmo (*DatabaseAdministration*).
- **handler:** Contiene la clase que se encarga de hacer de intermediario entre la aplicación del usuario y el algoritmo de razonamiento basado en casos.
- **storage:** Contiene la clase que se encarga de almacenar el texto que se le ha mostrado al administrador cuando utiliza la interfaz web del asistente.
- **representation:** Contiene las diferentes clases que representan el modelo y que son utilizadas por el algoritmo para poder conectarse correctamente con la base de datos.
- **util:** Contiene una clase que es la encargada de comparar los diferentes resultados que nos devuelve el algoritmo de razonamiento basado en casos.

## Navegabilidad

El sistema que se ha desarrollado tiene dos apartados bien diferenciados, por un lado es la aplicación de usuario para Android y por otro lado la página web que se emplea para la administración del algoritmo. Dentro de la página web, podemos diferenciar dos interfaces, la página de administración propiamente dicha y la posibilidad de utilizar el asistente virtual desde la página web, permitiendo así que el administrador pueda hacer las pruebas pertinentes para el correcto funcionamiento del algoritmo.

Teniendo en cuenta, esto que se ha mencionado, se puede obtener el siguiente diagrama de navegabilidad.

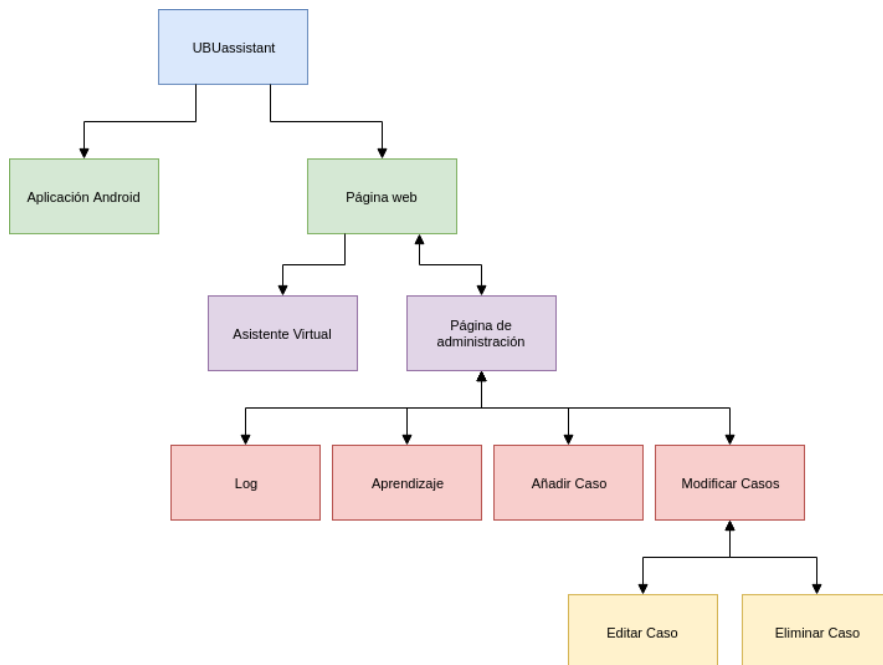


Figura C.14: Diagrama de navegabilidad

## Colores

Los colores que se han elegido para el desarrollo del proyecto, han sido los de la Universidad de Burgos, ya que uno de los objetivos del proyecto es mantener la imagen corporativa de la universidad, siendo los colores un azul de tono medio y el granate. Además, como tercer color, se ha elegido un tono grisáceo bastante cercano al negro. En las situaciones en las que se necesita un color de fondo, se ha optado por utilizar el blanco.



Figura C.15: Colores

---

## Documentación técnica de programación

---

### D.1. Introducción

En este apartado se va a explicar la estructura de directorios del proyecto que se ha desarrollado. También se incluyen los pasos necesarios para poder compilar, instalar y ejecutar el proyecto de una forma adecuada. A parte de todo esto, se incluye un manual de programador, para configurar correctamente los diferentes entornos que se han utilizado durante el proyecto.

### D.2. Estructura de directorios

Los directorios que posee nuestro proyecto es el siguiente:

- `/`: el directorio raíz del proyecto posee el fichero *README.md*, el cual contiene la licencia del proyecto. A parte de esto, tiene dentro todas las demás carpetas del proyecto, las cuales se detallan a continuación.
- **/Código Fuente**: este directorio posee los diferentes ficheros fuente del proyecto, tanto de la aplicación Android como del servidor web. En este directorio también podemos encontrar el fichero SQL necesario para poder crear correctamente la base de datos con todos los casos base de nuestro algoritmo.
- **/Código Fuente/Android/UBUassistant**: el contenido de este directorio es los ficheros necesarios para poder compilar la aplicación sin necesidad de utilizar *Android Studio* para ello. Además incluye los directorios que poseen los ficheros fuente de la aplicación.



- **/Código Fuente/Android/UBUassistant/app/build:** en este directorio se almacenan las diferentes clases compiladas de la aplicación en diferentes directorios.
- **/Código Fuente/Android/UBUassistant/app/build/outputs/apk/debug:** en este directorio podemos encontrar el fichero APK de nuestra aplicación Android.
- **/Código Fuente/Android/UBUassistant/app/src/main:** el contenido de este directorio es el código fuente de la aplicación Android junto a los recursos utilizados.
- **/Código Fuente/Android/UBUassistant/app/src/main/java:** código fuente de la aplicación Android.
- **/Código Fuente/Android/UBUassistant/app/src/main/res:** recursos utilizados en la aplicación Android.
- **/Código Fuente/Tomcat App:** en este directorio encontramos los ficheros de nuestro servidor web. En este directorio nos encontramos con las carpetas *UBUassistant* y *UBUassistant\_admin*, estas dos carpetas poseen la misma estructura de directorios, con la diferencia que en *UBUassistant* los ficheros JSP están programados para poder comunicarse con la aplicación Android, mientras que en *UBUassistant\_admin*, los ficheros JSP están programados para mostrar la página de administración del algoritmo. Como referencia a estas dos carpetas, vamos a usar siempre la nomenclatura de *UBUassistant*.
- **/Código Fuente/Tomcat App/doc/javadoc:** en este directorio encontramos la documentación del código de nuestro proyecto generada con JavaDoc.
- **/Código Fuente/Tomcat App/lib:** en este directorio nos encontramos con las diferentes librerías en formato JAR que son necesarias para el correcto funcionamiento de nuestro servidor.
- **/Código Fuente/Tomcat App/rsc:** el contenido de este directorio son los diferentes recursos necesarios para nuestro servidor, como pueden ser los ficheros para realizar correctamente la conexión con la base de datos.
- **/Código Fuente/Tomcat App/src:** en este directorio nos encontramos con el código fuente de nuestro servidor.
- **/Código Fuente/Tomcat App/src/main:** clases del servidor, ficheros de configuración y archivos JSP/HTML.
- **/Código Fuente/Tomcat App/src/main/java:** en este directorio encontramos los diferentes ficheros Java de nuestro servidor.
- **/Código Fuente/Tomcat App/src/main/resources:** los diferentes ficheros de configuración de nuestro servidor.
- **/Código Fuente/Tomcat App/src/main/webapp:** ficheros JSP/HTML para que funcione nuestro servidor. Como hemos mencionado antes, dependiendo del directorio en el que nos encontremos, los ficheros JSP sirven para la conexión de nuestro servidor con el cliente en Android o

- para ejecutar la aplicación web de administración.
- **/Código Fuente/Tomcat App/target:** clases y librerías Maven compiladas del servidor web.
- **/Documentación:** en este directorio nos podemos encontrar con la documentación del proyecto realizada con *LaTeX*.
- **/Documentación/img:** diferentes imágenes utilizadas en la documentación del proyecto.
- **/Documentación/tex:** ficheros utilizados para la generación de la documentación.
- **/Ejecutables:** en este directorio nos encontramos con los ficheros WAR para poder utilizarlos directamente en el servidor, como la aplicación APK para poder instalarla directamente en nuestro dispositivo Android.

### D.3. Manual del programador

Las herramientas que se han utilizado para el desarrollo del proyecto y que son necesarias tener instaladas en el sistema son:

- **Java JDK 8**
- **Eclipse JEE**
- **Android Studio**
- **Tomcat**
- **MySQL**

No todas las herramientas que se han instalado tienen que ir instaladas en el mismo ordenador, sino que unas se instalan en el ordenador que se ha utilizado para programar el sistema, mientras que otras se instalan en la máquina remota de Microsoft Azure que se utiliza como servidor de la aplicación.

Debido a que se ha utilizado como sistema operativo Linux, se va a explicar como instalar las siguientes herramientas en este sistema operativo.

#### Herramientas de programación

##### Java JDK 8

Java JDK (*Java Development Kit*) es un conjunto de herramientas necesarias para el desarrollo de aplicaciones Java. Para instalarlo en un sistema Linux, hay que abrir una sesión en el terminal e introducir los siguientes comandos [32]:

```
sudo add-apt-repository ppa:webupd8team/java  
sudo apt update
```

```
sudo apt install oracle-java8-set-default
```

Con estos comandos, el paquete se instala y configura automáticamente, incluyendo las diferentes variables de entorno.

### Eclipse JEE

Eclipse JEE es una versión de Eclipse que con todas las herramientas que son necesarias para el desarrollo de aplicaciones web basadas en Java.

Para la instalación, primero tenemos que descargarnos la aplicación desde la página web oficial [33]. Una vez que tengamos el fichero descargado, lo descomprimos en la carpeta que deseemos. Desde esa carpeta, ya se puede iniciar la aplicación directamente si abrimos el fichero *eclipse*. En caso de que queramos crear un acceso en nuestra barra de inicio, introducimos el siguiente comando desde una ventana del terminal:

```
nano /.local/share/applications/opt_eclipse.desktop
```

En el editor que se nos ha abierto, introducimos lo siguiente y guardamos el fichero.

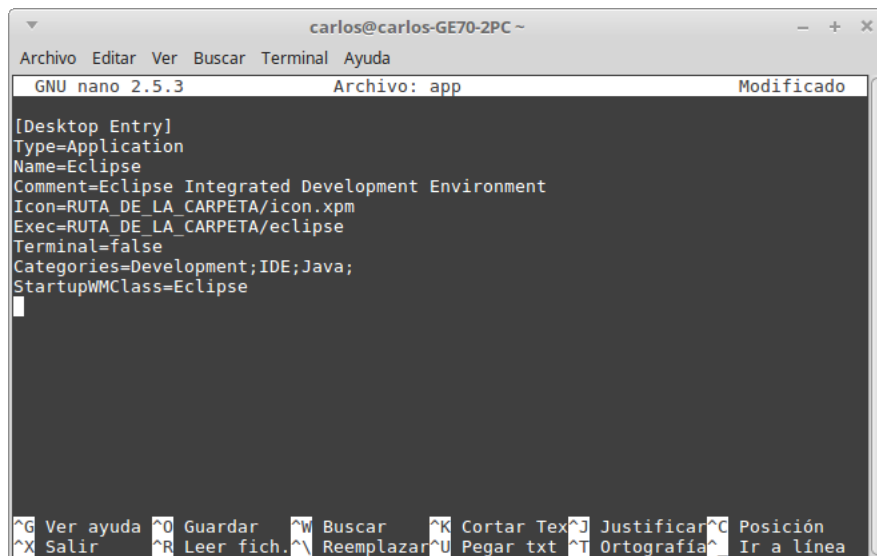


Figura D.1: Configuración .desktop Eclipse

Finalmente, para que se pueda ejecutar el programa desde el acceso directo correctamente, introducimos el siguiente comando:

```
chmod +x /.local/share/applications/opt_eclipse.desktop
```

## Android Studio

Android Studio es el IDE oficial que nos ofrece Google para el desarrollo de aplicaciones para Android. Para descargar la última versión, lo podemos hacer desde su página web oficial [34]. Una vez que tenemos descargado el fichero, lo podemos descomprimir en la localización que nosotros deseemos, al igual que en Eclipse. Una vez que lo tenemos descomprimido, lo podemos ejecutar a través del fichero *CARPETA\_ANDROID\_STUDIO/bin/studio.sh*.

Si queremos crear un acceso a nuestra barra de inicio, introducimos el siguiente comando en una terminal:

```
nano /.local/share/applications/androidstudio.desktop
```

Dentro del editor, introducimos el siguiente texto y lo guardamos:

```
[Desktop Entry]  
Version=1.0  
Type=Application  
Name=Android Studio  
Exec=CARPETA_ANDROID_STUDIO/bin/studio.sh %f  
Icon=CARPETA_ANDROID_STUDIO/bin/studio.png  
Categories=Development;IDE;  
Terminal=false  
StartupNotify=true  
StartupWMClass=android-studio
```

Finalmente, para que se puede ejecutar desde la aplicación, introducimos el siguiente comando desde una terminal:

```
chmod +x /.local/share/applications/androidstudio.desktop
```

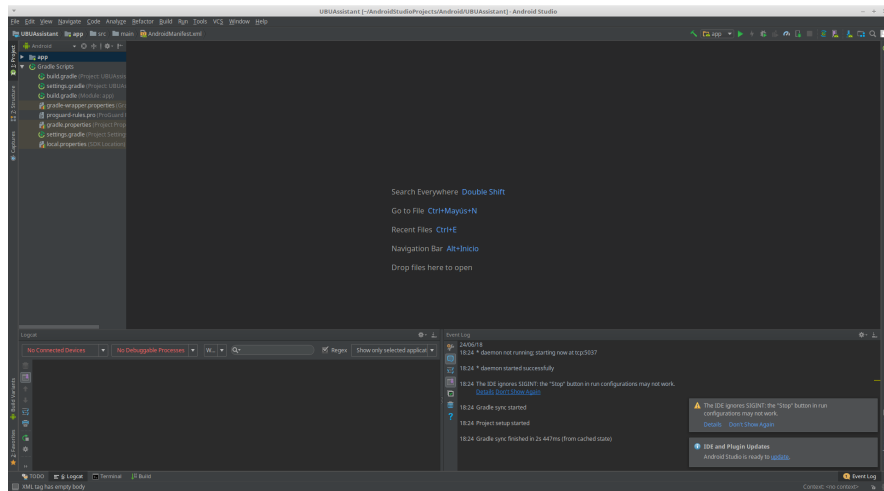


Figura D.2: Interfaz Android Studio

## Herramientas del servidor

### Java JDK 8

En el servidor, también es necesario que tengamos instalado Java JDK 8, por lo que seguimos los mismos pasos que se han mencionado con anterioridad en la sección D.3.

### Tomcat

Tomcat es una plataforma que nos permite ejecutar aplicaciones web basadas en Java que se encuentren en formato WAR. Para la instalación, primero descargamos la última versión de Tomcat 9 desde la página oficial de Tomcat [35]. Una vez que tengamos descargado el fichero, abrimos una terminal e introducimos los siguientes comandos [36], recomendando la ejecución del terminal con permisos de administrador:

```
groupadd tomcat
useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
cd /CARPETA_FICHEROS_TOMCAT
cp ./FICHERO_TOMCAT.tar.gz /opt
tar -xzf ./FICHERO_TOMCAT.tar.gz
mv apache-tomcat-9.X.X/ tomcat/
chown -R tomcat:tomcat /opt/tomcat
chmod +x /opt/tomcat/bin/*
nano /.bashrc
```

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN<sup>48</sup>

En el fichero que se ha abierto para editar, introducimos la siguiente línea al final del fichero y lo guardamos:

```
export CATALINA_HOME=/opt/tomcat
```

Sin cerrar la terminal, seguimos introduciendo los siguientes comandos:

```
source /.bashrc  
$CATALINA_HOME/bin/startup.sh
```

Con esto, tendríamos Tomcat totalmente configurado y funcionando. Para ver que funciona correctamente, podemos acceder introduciendo el ip del servidor en un navegador (*http://192.168.x.x:8080/*) y nos debería de aparecer lo siguiente:

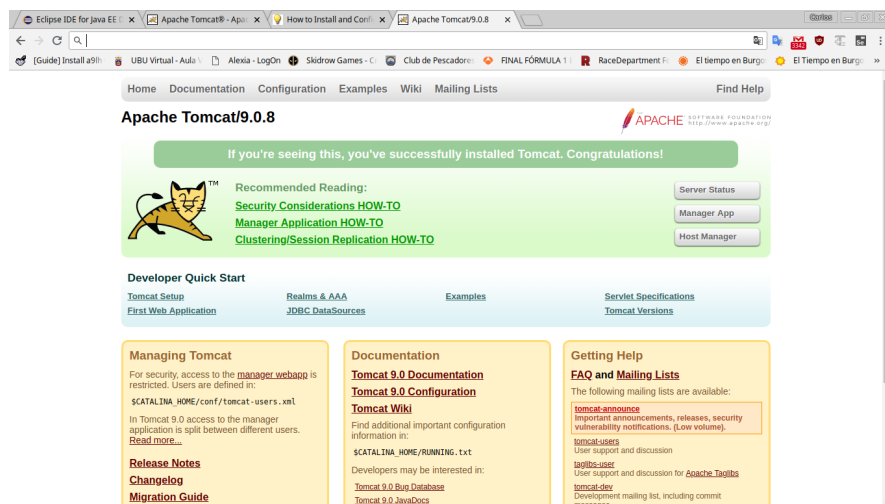
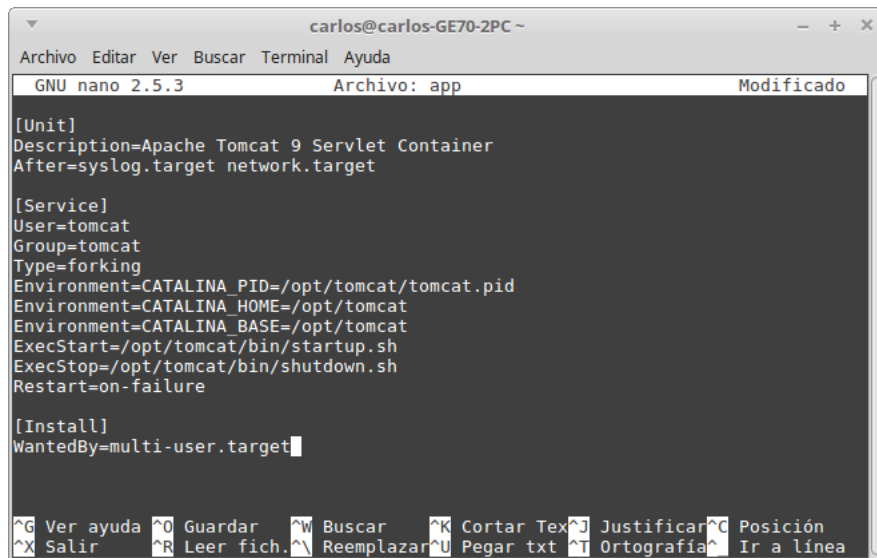


Figura D.3: Ventana inicio Tomcat

Finalmente, hay que configurar el inicio de Tomcat como servicio, para que este se inicie automáticamente cada vez que se inicia el servidor. Para ello, abrimos una ventana del terminal e introducimos los siguientes comandos:

```
$CATALINA_HOME/bin/shutdown.sh  
chown -hR tomcat:tomcat /opt/tomcat/  
cd /etc/systemd/system/  
nano apache-tomcat.service
```

En el fichero que se nos abre para modificar, introducimos las siguientes líneas y guardamos el fichero:



```

carlos@carlos-GE70-2PC ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.5.3 Archivo: app Modificado

[Unit]
Description=Apache Tomcat 9 Servlet Container
After=syslog.target network.target

[Service]
User=tomcat
Group=tomcat
Type=forking
Environment=CATALINA_PID=/opt/tomcat/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea

```

Figura D.4: Configuración fichero de servicio de Tomcat

Sin cerrar la terminal, introducimos los siguientes comandos:

```

systemctl daemon-reload
systemctl start apache-tomcat
systemctl enable apache-tomcat

```

Con todo esto, ya estaría correctamente configurado nuestro servidor de Tomcat y en caso de que tengamos que reiniciar nuestro servidor, nos garantizamos que el servicio se inicia automáticamente.

## MySQL

MySQL es un sistema de gestión de base de datos. Para la instalación, tenemos que abrir una ventana del terminal e introducir los siguientes comandos [37]:

```

sudo apt install mysql-server
sudo mysql_secure_installation

```

Estas acciones, mostrarán una serie de mensajes para configurar el usuario *root* de la base de datos. En la primera opción que nos aparece, le indicamos que no queremos usar *Validate Password Plugin*. Lo siguiente que haremos es introducir la contraseña del usuario *root*, la cuál para este proyecto el valor que

se la ha establecido es *1234*. El resto de opciones que nos va dando, las aceptamos todas para que finalice la configuración de MySQL. Finalmente, para que todo quede correctamente inicializado, introducimos el siguiente comando en una terminal:

***mysqld --initialize***

Con estos pasos, ya estaría todo configurado correctamente. Para administrar las bases de datos MySQL, solamente tenemos que introducir el siguiente comando en una ventana del terminal:

***mysql -u root -p***

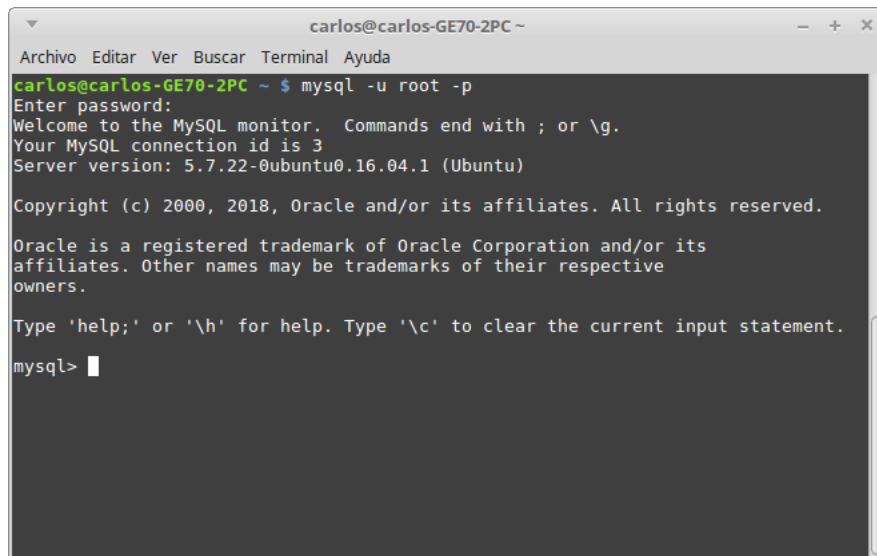


Figura D.5: Ventana de administración de MySQL



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

### Obtención del código

Lo primero que tenemos que hacer, es obtener el código de la aplicación. Para ello, se puede descargar desde el repositorio GitHub del proyecto [38].

Una vez que nos encontramos dentro de la página del repositorio, lo podemos descargar pulsando primero sobre el botón *Clone or download* y posteriormente sobre *Download ZIP*. Una vez que se termine de descargar el fichero, lo extraemos en la ruta que el usuario prefiera.

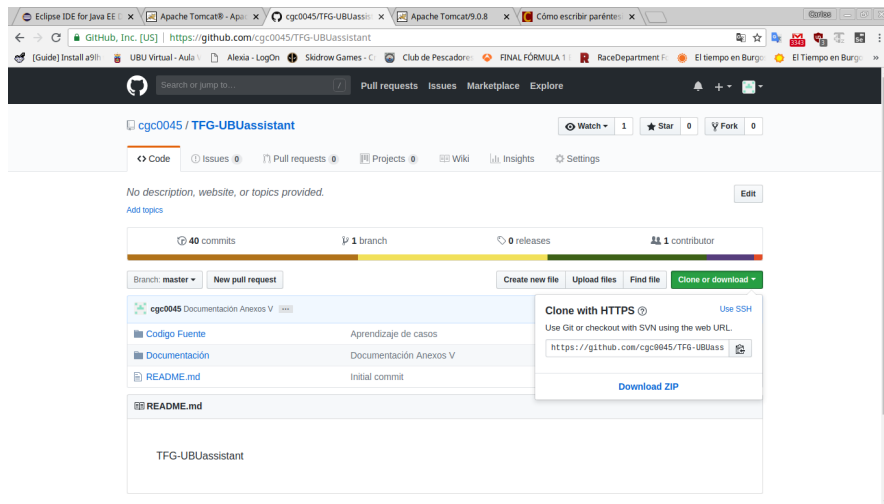


Figura D.6: Descarga del proyecto desde GitHub

Una vez que hemos realizado esto, lo que tenemos que hacer es importar las diferentes partes del proyecto en sus respectivos IDE's.

### Importar proyecto en Android

Lo primero que tenemos que hacer es abrir la aplicación de Android Studio. Una vez que la tenemos abierta, nos dirigimos al menú *File* y pulsamos sobre la opción *Open*. Se nos abrirá una nueva ventana, en la cuál podemos navegar hasta el directorio de nuestra aplicación. Android Studio nos facilita encontrar los proyectos que han sido creados con la misma herramienta, ya que cuando identifica uno, cambia el icono de carpeta por el icono de Android Studio. Una vez que tenemos seleccionada la carpeta, pulsamos sobre el botón *OK*.



Figura D.7: Icono Android Studio

Si todo ha sido importado correctamente, en el menú de la izquierda nos debería de aparecer los diferentes componentes que conforman la aplicación.

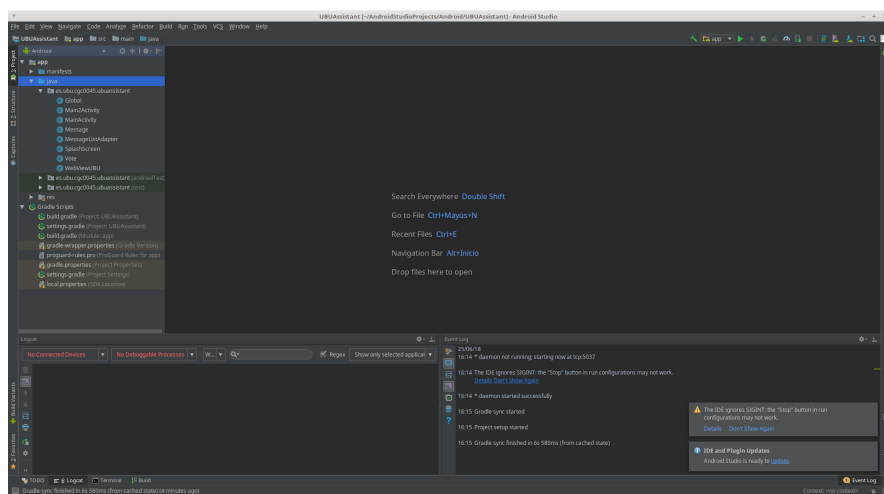


Figura D.8: Proyecto importado en Android Studio

Para exportar la aplicación a un fichero APK, en la barra de opciones superior abrimos el menú *Build* y seleccionamos la opción *Build APK(s)*. Esta opción nos genera un fichero APK en la ruta:

***CARPETA\_APLICACIÓN\_ANDROID/app/build/outputs/apk/debug***

## Importar proyecto en Eclipse JEE

Lo primero que hacemos es iniciar Elcipse JEE y seleccionamos la carpeta donde queremos alojar nuestro *Workspace*. Una vez que se termina de abrir el programa, sobre el menú de la izquierda abrimos un menú pulsando sobre el botón derecho de nuestro ratón y en el desplegable de la opción *Import* seleccionamos *Import...* En la ventana que se abre, nos dirigimos a la opción llamada *Maven*, seleccionamos la opción *Existing Maven Projects* y pulsamos sobre el botón *Next*.

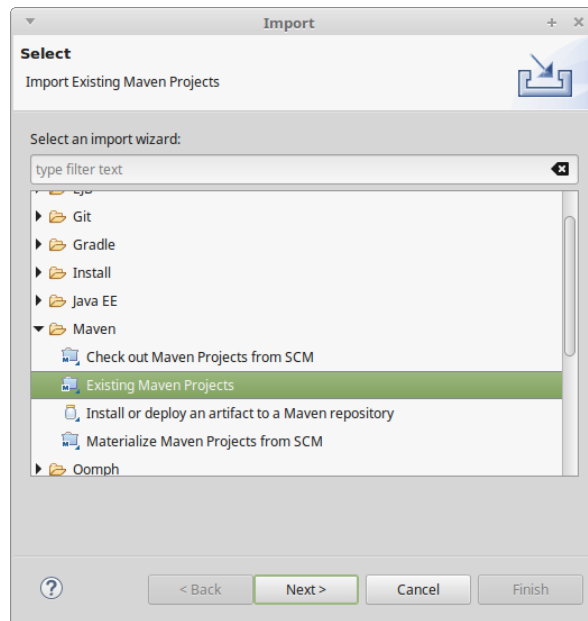


Figura D.9: Selección del tipo de proyecto

En la siguiente ventana que nos aparece, hacemos seleccionamos la opción *Browse...*, navegamos hasta el directorio donde se aloja la parte del código del servidor web y seleccionamos la opción *OK*. De nuevo a la ventana que teníamos con anterioridad, seleccionamos el proyecto y para terminar la importación, seleccionamos la opción *Finish*.

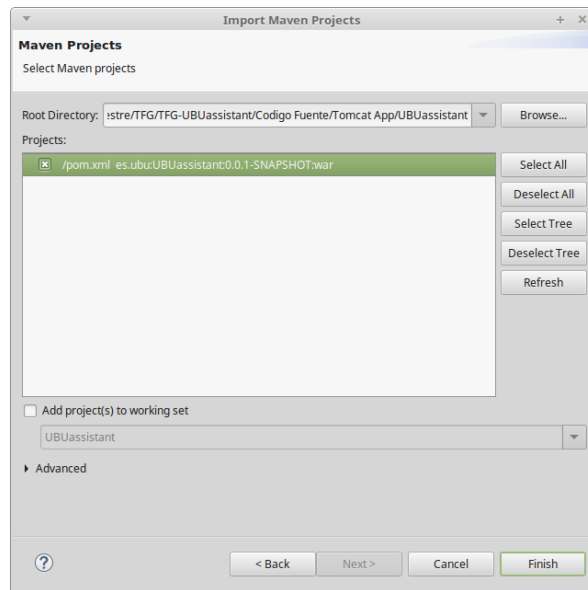


Figura D.10: Selección del proyecto

Si todo se ha realizado de forma correcta, en el menú de la izquierda nos tiene que aparecer el proyecto y si lo desplegamos, nos aparecerán los diferentes directorios y ficheros del proyecto.

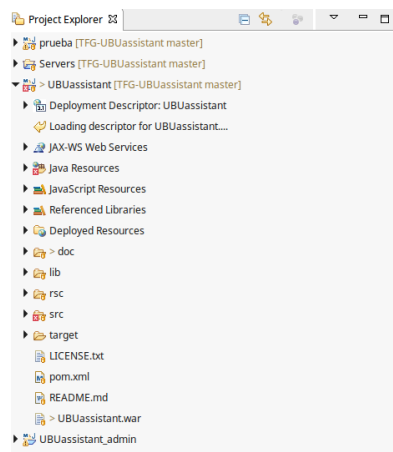


Figura D.11: Proyecto importado en Eclipse JEE

Ahora, hay que configurar el *PATH* del proyecto, para que se importen correctamente las diferentes librerías que se usan en la aplicación. Para ello, seleccionamos el proyecto y con el botón derecho del ratón desplegamos el menú de opciones. Con el ratón, nos dirigimos a la opción *Build Path* y en el desplegable seleccionamos *Configure Build Path*. En la ventana que ha aparecido,

seleccionamos la opción *Add JARs...*. A continuación, se abre una ventana, la cual permite desplazarse por los directorios del proyecto, abrimos la carpeta *lib* y seleccionamos todos los ficheros JAR que hay en su interior. Para finalizar la configuración, seleccionamos la opción *Apply and Close*.

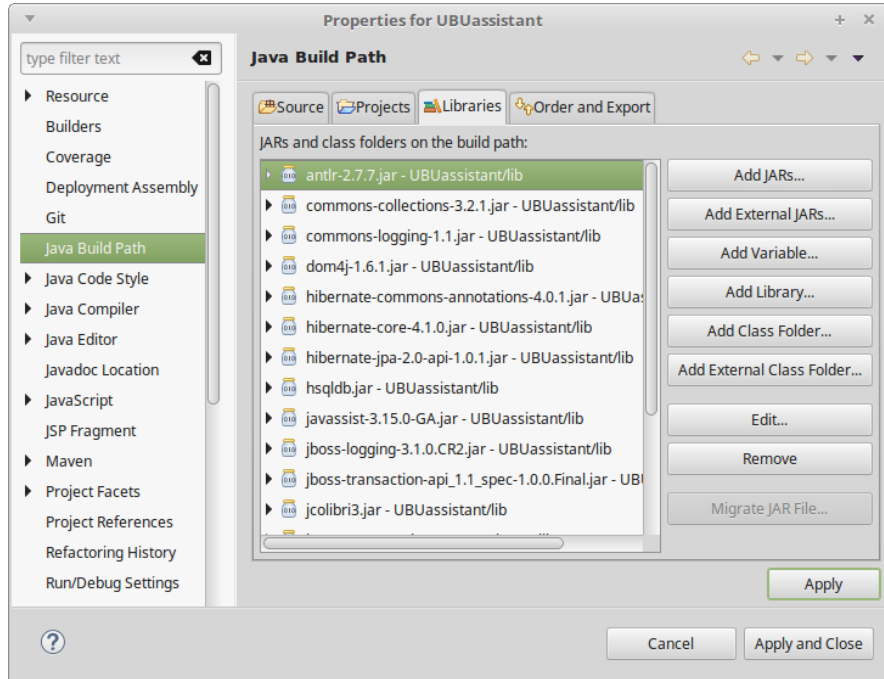


Figura D.12: Configuración del PATH en Eclipse JEE

Finalmente, hay que configurar el proyecto para que cuando exportemos la aplicación al fichero WAR para usarlo en nuestro servidor Tomcat, incluya todas las librerías que se utilizan en el proyecto.

Para ello, seleccionamos nuestro proyecto y en el menú de opciones que abrimos pulsando el botón derecho del ratón, seleccionamos la opción *Properties*. En la ventana que nos aparece, seleccionamos en el menú de la izquierda la opción *Deployment Assembly*. En la parte derecha de la ventana nos aparecerá un listado con carpetas y ficheros. Para añadir nuestras librerías a esta lista, pulsamos sobre *Add...*. En la ventana que nos ha aparecido, seleccionamos *Java Build Path Entries* y seleccionamos todas las librerías que nos aparecen. Para finalizar la configuración, pulsamos sobre *Apply and Close*.

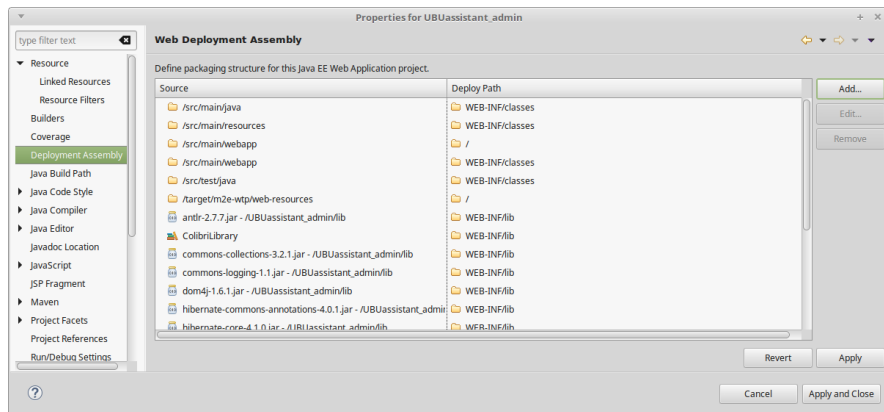


Figura D.13: Configuración de librerías en el fichero WAR

Con todo esto, tendríamos el proyecto configurado correctamente para poder modificarlo y exportarlo correctamente.

Para exportar el proyecto a un fichero WAR, seleccionamos el proyecto, abrimos el menú de opciones con el botón derecho de nuestro ratón y sobre el desplegable *Export* seleccionamos la opción *WAR file*. En la ventana que se abre, seleccionamos el directorio donde queremos guardar el fichero junto a su nombre y pulsamos sobre *Finish*.

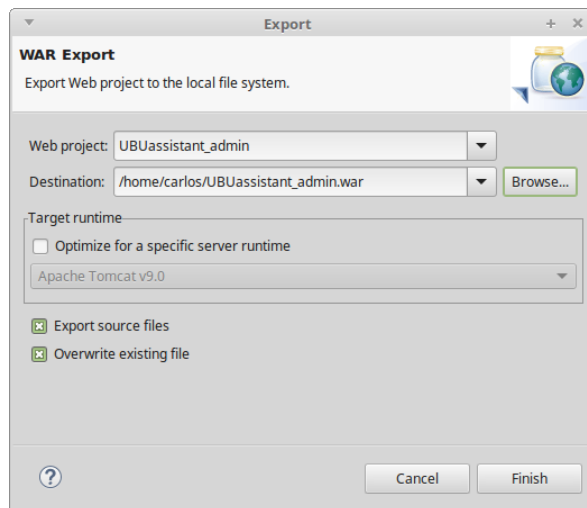


Figura D.14: Exportar servidor a fichero WAR

## D.5. Pruebas del sistema

Para comprobar las funcionalidades del proyecto se han realizado una serie de tests y pruebas. Las pruebas realizadas en el servidor son diferentes a las realizadas en la aplicación Android.

### Pruebas aplicación Android

Para la aplicación Android, se han realizado una serie de pruebas de forma manual para comprobar las diferentes funcionalidades que lleva implementada. Las diferentes pruebas que se han realizado, junto al resultado esperado, así como el objetivo del mismo se pueden observar en la siguiente tabla.

Entrada	Respuesta	Finalidad
esternocleidomastoideo (introducido por voz)	esternocleidomastoideo	Comprobar la funcionalidad de reconocimiento de voz.
informática máster online	informatica(master, online)	Comprobar la funcionalidad de obtención de única respuesta.
informática	informatica(master) informatica(online) informatica (master, online) informatica	Comprobar la funcionalidad de obtener múltiples respuestas.
google	lengua (online) abogacia mecanica y electronica	Comprobar la funcionalidad de búsqueda sin resultados y ofrece recomendaciones.
buenos días	Buenos días, ponme a prueba con tus preguntas.	Comprobar la funcionalidad de uso de palabras reservadas, como las de saludo.

Tabla D.1: Pruebas manuales

Para saber la respuesta que nos tenía que dar estas pruebas (exceptuando la prueba de comando de voz), se realizaron primero diferentes pruebas con la aplicación web original desarrollada en el anterior TFG. Por cada prueba se realizaron diferentes iteraciones por cada prueba, asegurándose así que siempre se obtiene el mismo resultado.

## Pruebas servidor Tomcat

Para probar los cambios en el servidor web Tomcat, se han aprovechado algunos de los test JUnit que se realizó Daniel Santidrián Alonso en la versión previa del TFG, mientras que otras pruebas se han realizado de forma manual.

### Test JUnit 4

Se han aprovechado los tests existentes de JUnit para comprobar que la funcionalidad de la base de datos ampliada a siete vocablos funciona correctamente. Para ello, se han hecho unos pequeños cambios en el test para que este se ajuste a este requisito del proyecto. Esta prueba consta en un total de 12 test unitarios alojados en 3 clases.

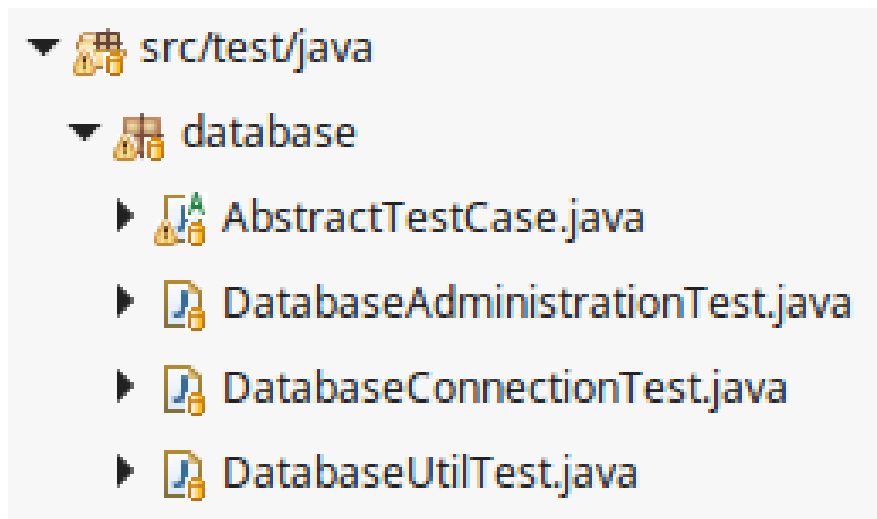


Figura D.15: Test unitarios de la base de datos

Para la ejecución de cada una de las clases, solamente es necesario abrirlas y ejecutarlas.



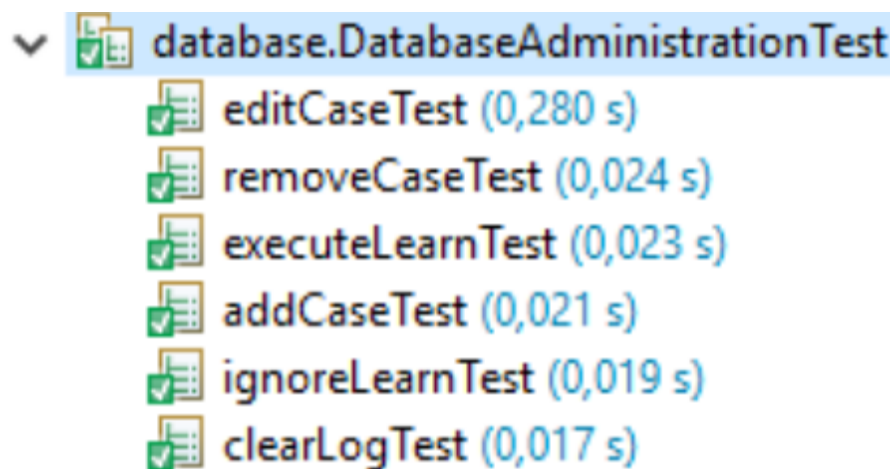


Figura D.16: Ejecución de una clase con test unitarios

### Test manuales

Para comprobar que los cambios en la interfaz de la web de administración se realizaron de forma manual, ya que los test que se encontraban preparados en la versión anterior del TFG solamente son válidos para Windows. Las pruebas que se realizaron fueron:

Entrada	Respuesta	Finalidad
Añadir nuevo caso	Caso añadido a la base de datos	Comprobar que la funcionalidad de adición de nuevos casos con siete vocablos funciona correctamente.
Editar caso existente	Caso modificado en la base de datos	Comprobar que la funcionalidad de modificación de casos con siete vocablos funciona correctamente.
Eliminar caso existente	Caso eliminado de la base de datos	Comprobar que a pesar de haber ampliado a siete vocablos, el borrado de casos sigue funcionando correctamente.
Ninguna	Visualización del log	Comprobar que el log se muestra correctamente a pesar de haber ampliado a siete vocablos.

continúa en la página siguiente

continúa desde la página anterior

Entrada	Respuesta	Finalidad
<i>Ninguna</i>	Fichero exporta- do	Comprobar que el log se exporta correctamente con siete vocablos.
Aceptar aprendi- zaje	Recomendación añadida como caso	Comprobar que las recomendaciones se añaden correctamente con siete vocablos al asistente virtual.

Tabla D.2: Pruebas manuales web de administración

---

## Documentación de usuario

---

### E.1. Introducción

En este apartado se van a detallar los requisitos necesarios para poder ejecutar la aplicación Android, así como la web de administración. A parte de esto, se especifica los pasos necesarios para la instalación y el funcionamiento de los mismos.

### E.2. Requisitos de usuarios

#### Aplicación Android

Los requisitos para poder ejecutar la aplicación en Android son:

- Dispositivo Android con versión 4.4 o superior.
- Acceso a Internet (WiFi o datos móviles).

#### Web de administración

Los requisitos necesarios para poder utilizar la web de administración son:

- Dispositivo con posibilidad de utilizar un navegador web.
- Navegador web con JavaScript compatible y activado.
- Acceso a Internet.

### E.3. Instalación

#### Aplicación Android

Para poder instalar la aplicación APK, primero tenemos que activar la opción *Orígenes desconocidos* en el dispositivo. Para ello accedemos a los

ajustes del dispositivo, accedemos a la opción *Seguridad* y activamos la opción *Orígenes desconocidos*.

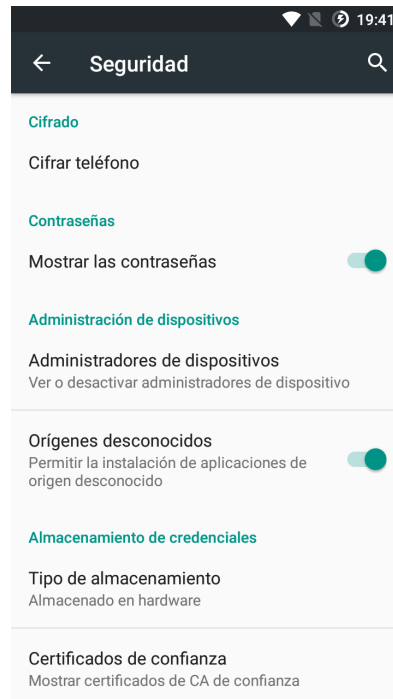


Figura E.1: Opciones de seguridad Android

A continuación procedemos a la instalación de la aplicación, para ello primero tenemos que copiar el fichero APK a nuestro dispositivo. Una vez que termine la copia del fichero, desde nuestro dispositivo abrimos el fichero con un explorador de archivos. En la ventana que nos aparece, le damos a la opción *Instalar*.

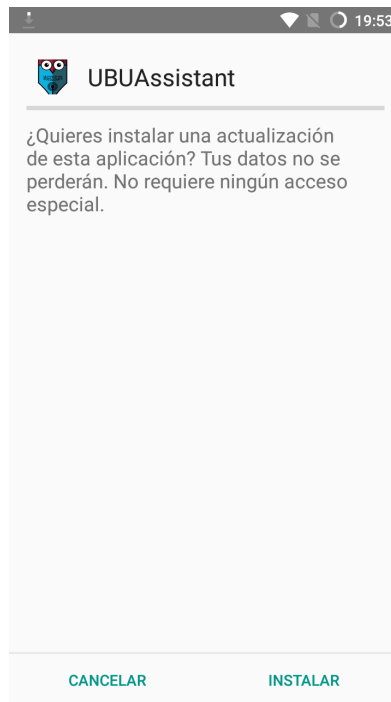


Figura E.2: Instalación UBUassistant

Con estos pasos, la aplicación ya está instalada en el dispositivo Android.

### Web de administración

Para poder utilizar la web de administración solamente es necesario tener instalado en el dispositivo en el que se vaya a utilizar un navegador web, ya que la aplicación de administración está alojado en un servidor.

Para acceder a la página de administración en el navegador hay que introducir la siguiente URL:

***`http://IP_SERVIDOR:8080/UBUassistant_admin`***

## E.4. Manual del usuario

En este manual se trata de guiar al usuario en el funcionamiento del asistente virtual con la aplicación Android, así como en la aplicación de administración web.

## Aplicación Android

Lo primero que vemos al iniciar la aplicación es una ventana en la que aparecen el logo de la aplicación, junto al de la Universidad de Burgos. Lo siguiente que aparece es la interfaz del asistente virtual, la cual se asemeja a diferentes aplicaciones de mensajería para dispositivos móviles.

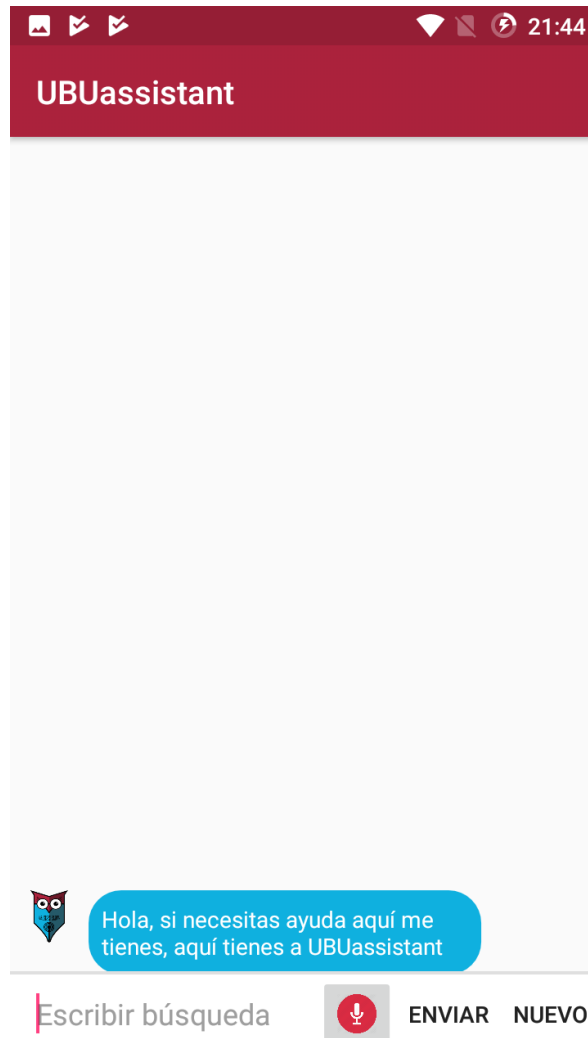


Figura E.3: Interfaz aplicación Android

En la parte inferior, encontramos una barra, la cual encontramos un cuadro donde podemos introducir el texto para interactuar con el asistente virtual. También nos encontramos con un botón para poder introducir el texto mediante voz y dos botones los cuales envían el texto al servidor.

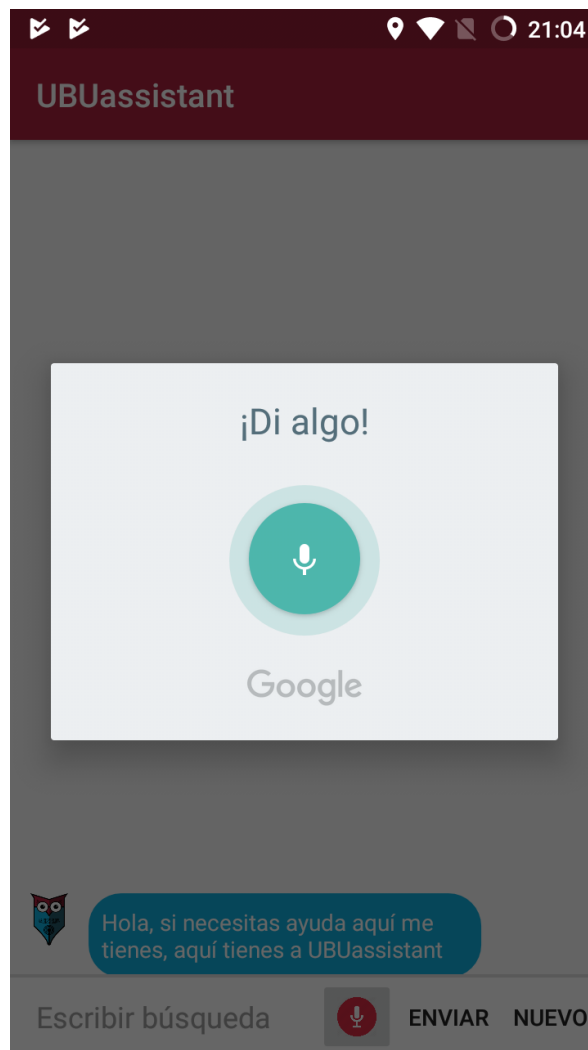


Figura E.4: Reconocimiento de voz

Dependiendo de la búsqueda que hagamos, podemos encontrar diferentes situaciones de respuesta.

### Interacción con el asistente virtual

Se puede interactuar con el asistente virtual con mensajes como *hola*, *buenos días*, *adiós*, *etc.*

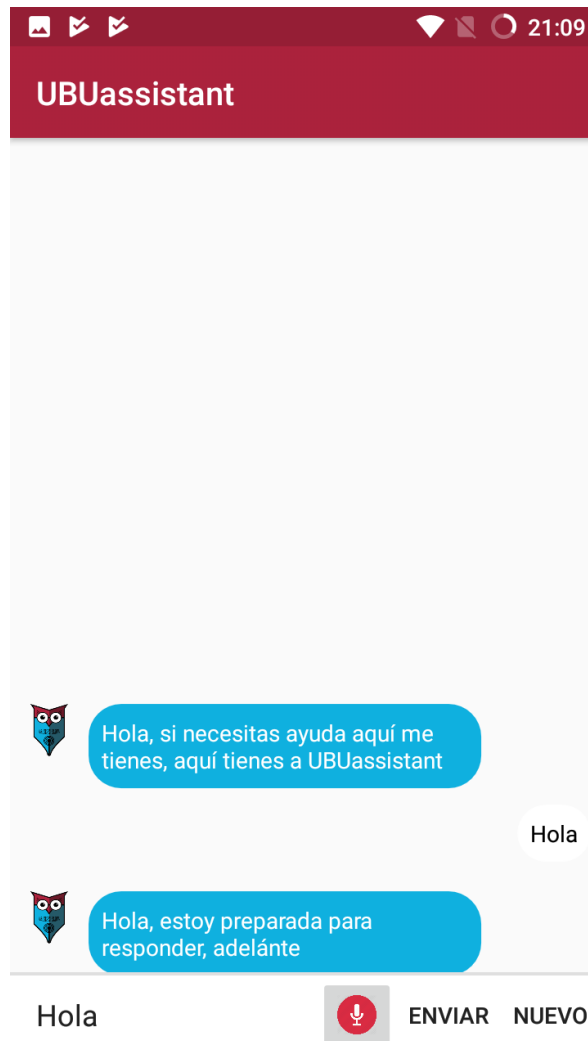


Figura E.5: Interacción con el asistente virtual



### Respuesta única

Tras realizar una consulta con el asistente virtual, se pueden obtener diferentes situaciones de respuesta. Una de ellas, es que la búsqueda tenga una única respuesta. Un ejemplo de esto, puede ser la búsqueda del término *correo*.

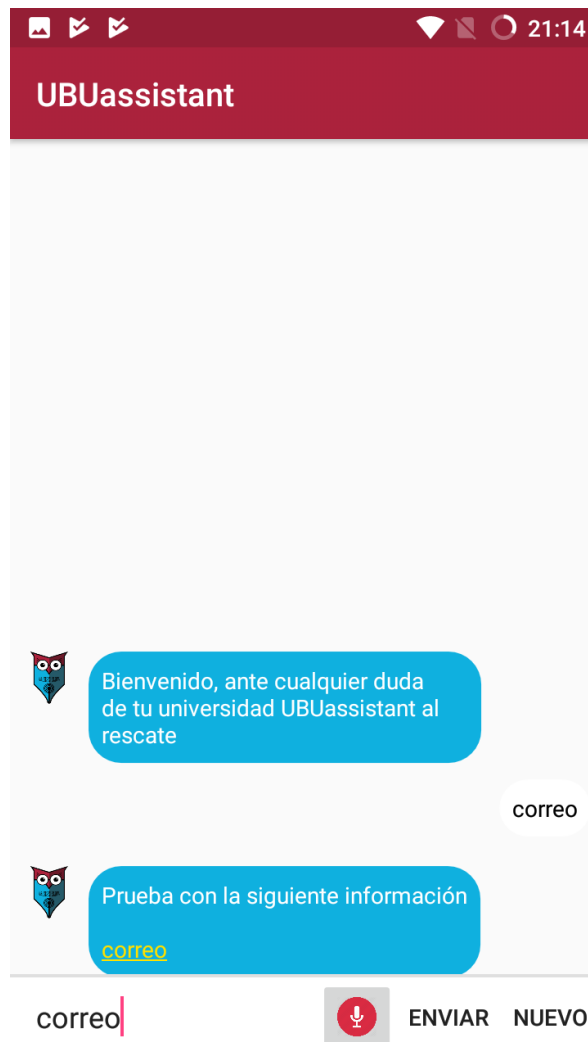


Figura E.6: Respuesta única

### Múltiples respuestas

Otra de las posibles situaciones, es que la búsqueda que hemos realizado tenga múltiples respuestas. En esta situación, el asistente nos enseña todas las posibles respuestas que ha encontrado, para que se elija la que mas se ajusta al objetivo del usuario. Un ejemplo de este tipo de búsqueda es utilizando el término *informática*.

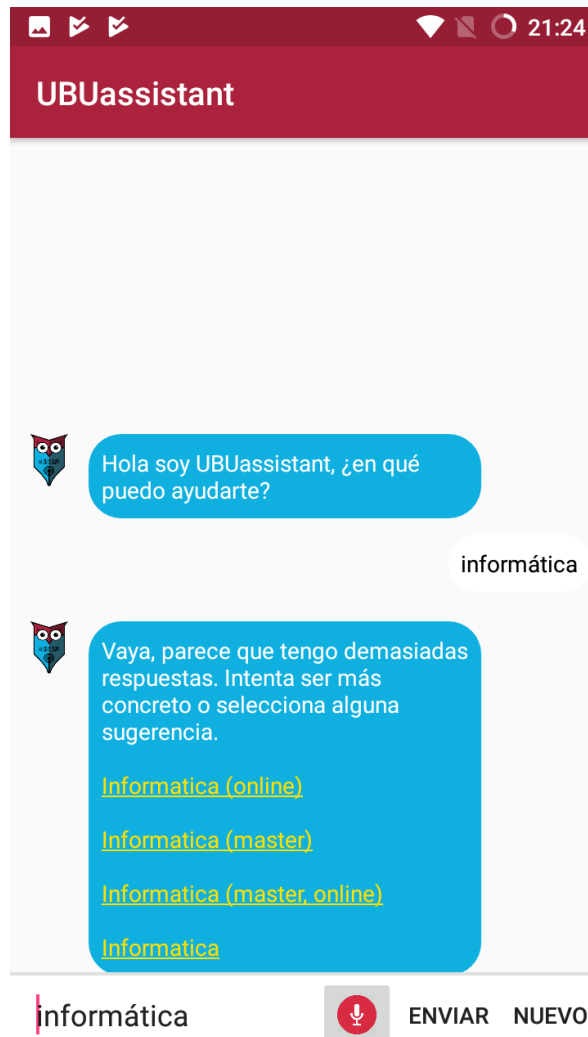


Figura E.7: Múltiples respuestas

### Sin respuesta

La última posible situación que se puede llegar a dar, es que la búsqueda que se ha realizado no posee ninguna respuesta. En este caso, el asistente mostrará tres recomendaciones para ver si se ajustan al objetivo de búsqueda realizada por el usuario. Para obtener un ejemplo de este tipo de búsqueda, se puede utilizar el término *Google*.

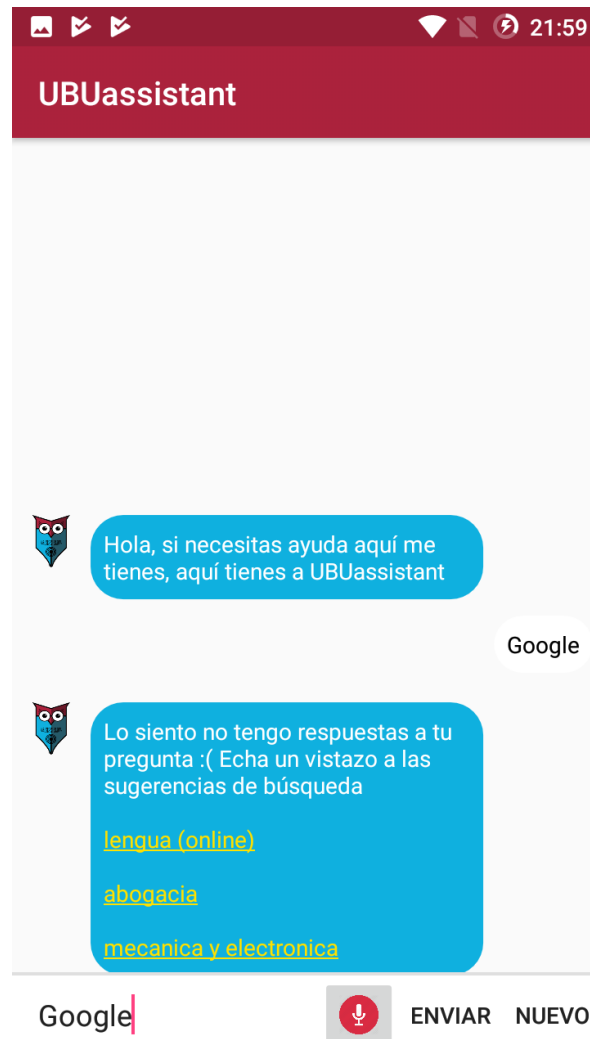


Figura E.8: Pregunta sin respuesta

### Enviar búsqueda y nueva búsqueda

En la interfaz, podemos encontrar dos botones, uno con el texto *ENVIAR* y otro con el texto *NUEVO*.

El botón *ENVIAR* sirve para enviar la respuesta siguiendo con las búsquedas que se han hecho con anterioridad, de esta manera, el usuario puede ir refinando la búsqueda hasta obtener la respuesta deseada.

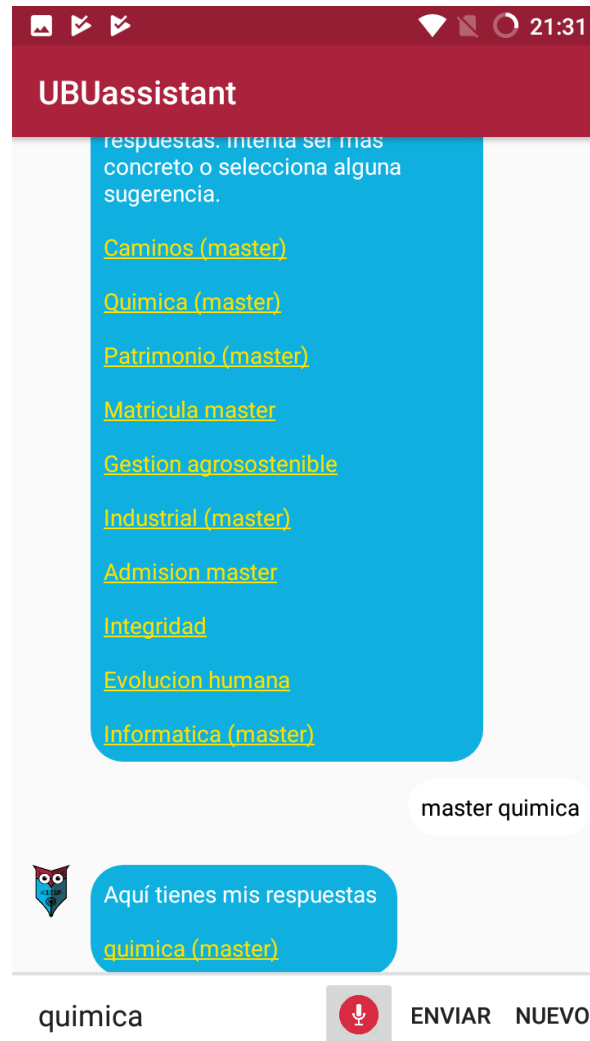


Figura E.9: Continuar búsqueda

El botón *NUEVO* sirve para reiniciar los términos de la búsqueda, de forma que el usuario puede empezar a realizar una búsqueda desde cero, no teniendo en cuenta el asistente los términos utilizados anteriormente.

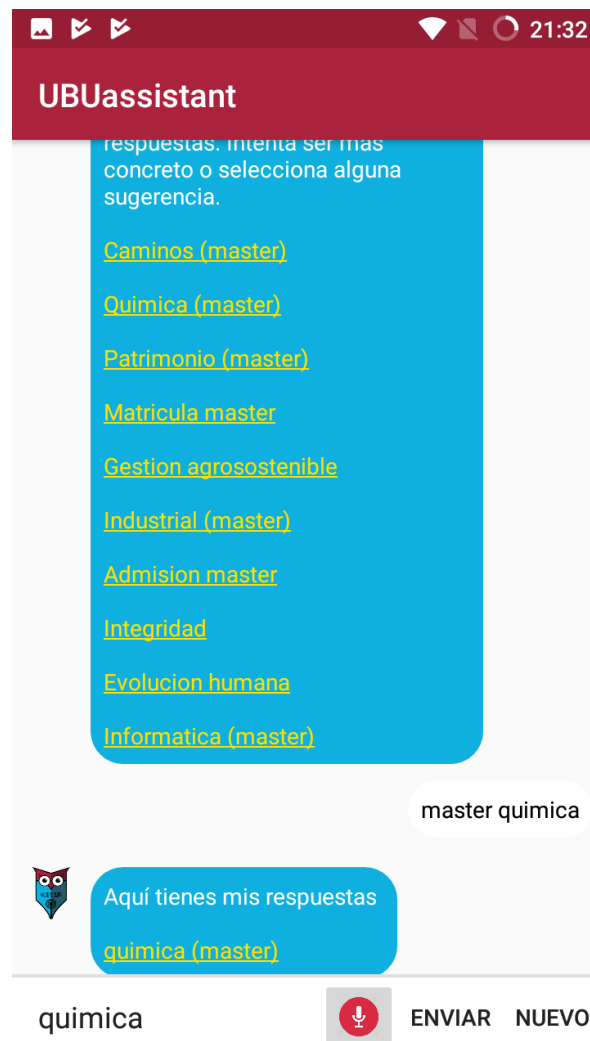


Figura E.10: Reiniciar búsqueda

### Página web

Cuando el usuario selecciona una respuesta otorgada por el asistente virtual, se nos abre la página web de la respuesta desde la propia aplicación, pudiendo interactuar con ella sin tener que salir de la aplicación.



Figura E.11: Respuesta consultada

### Valoración

Cuando el usuario termina de consultar la página web y vuelve al asistente, una ventana emergente, en la cual se le da la opción de valorar la respuesta obtenida.

IMAGEN-VALOR

### Aplicación web de administración

---

## Bibliografía

---

- [1] Wikipedia, “Scrum (desarrollo de software),” 2018, [Internet; Accedido 1-junio-2018]. [Online]. Available: [https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))
- [2] PageGroup, “Tendencias del mercado laboral,” 2017, [Internet; Accedido 2-junio-2018]. [Online]. Available: [https://www.michaelpage.es/sites/michaelpage.es/files/PG\\_ER\\_IT.pdf](https://www.michaelpage.es/sites/michaelpage.es/files/PG_ER_IT.pdf)
- [3] M. de Empleo y Seguridad Social, “Bases y tipos de cotización 2018,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: [http://www.seg-social.es/Internet\\_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm](http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm)
- [4] “Ordenador portátil hp,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://store.hp.com/SpainStore/Merch/Product.aspx?id=3BZ62EA&opt=ABE&sel=NTB>
- [5] “Pantalla hp,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://store.hp.com/SpainStore/Merch/Product.aspx?id=T3M80AA&opt=ABB&sel=MTO>
- [6] “Ratón hp,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://store.hp.com/SpainStore/Merch/Product.aspx?id=H4B81AA&opt=&sel=ACC>
- [7] “Teclado hp,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://www.amazon.es/HP-Classic-Teclado-al%C3%A1mbrico-Espa%C3%B1ol/dp/B005E03D7I>
- [8] “Smartphone android para hacer pruebas,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://www.pccomponentes.com/xiaomi-redmi-5a-4g-16gb-dual-sim-dorado-libre>

- [9] “Costo servidor microsoft azure,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://azure.com/e/298478ff72f247119527c820f7bf18dd>
- [10] “Costo cuenta desarrollador de google,” 2018, [Internet; Accedido 23-junio-2018]. [Online]. Available: <https://play.google.com/apps/publish/signup/>
- [11] “Creative cloud para equipos,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://www.adobe.com/es/creativecloud/plans.html>
- [12] “Fibra Óptica empresas - movistar,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://www.movistar.es/empresas/banda-ancha/oferta-fibra-empresas/?pid=PY-home-ql1b>
- [13] “jcolibri 2,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://gaia.fdi.ucm.es/research/colibri/jcolibri>
- [14] “Licencia eclipse ide,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://www.eclipse.org/legal/eplfaq.php>
- [15] “Licencia android studio,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://developer.android.com/license>
- [16] “Licencia tomcat,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://tomcat.apache.org/legal.html>
- [17] “Licencia mysql,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://www.mysql.com/about/legal/>
- [18] “Licencia java,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- [19] “Licencia apache commons,” 2004, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://commons.apache.org/proper/commons-bsf/license.html>
- [20] “Licencia json,” 2002, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://www.json.org/license.html>
- [21] “Licencia jersey,” 2013, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://jersey.github.io/license.html>
- [22] “Licencia junit,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://junit.org/junit4/license.html>



- [23] “Licencia hibernate,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://hibernate.org/community/license/>
- [24] “Licencia log4j,” 2004, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://logging.apache.org/log4j/1.2/license.html>
- [25] “Licencia selenium,” 2018, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://www.seleniumhq.org/about/license.jsp>
- [26] “Licencia guava,” 2004, [Internet; Accedido 2-junio-2018]. [Online]. Available: <https://github.com/google/guava/blob/master/COPYING>
- [27] “Licencia maven,” 2004, [Internet; Accedido 2-junio-2018]. [Online]. Available: <http://maven.apache.org/ref/3.0/license.html>
- [28] R. M. G. Labrador, “Tipos de licencia software,” 2005, [Internet; Accedido 3-junio-2018]. [Online]. Available: <http://www.informatica.us.es/~ramon/articulos/LicenciasSoftware.pdf>
- [29] “Lista de licencias con comentarios,” 2003, [Internet; Accedido 3-junio-2018]. [Online]. Available: <https://www.gnu.org/licenses/license-list.es.html>
- [30] “Reconocimiento-nocomercial-compartirigual 3.0 españa (cc by-nc-sa 3.0 es),” 2018, [Internet; Accedido 3-junio-2018]. [Online]. Available: <https://creativecommons.org/licenses/by-nc-sa/3.0/es/>
- [31] “Modelo vista controlador,” 2018, [Internet; Accedido 23-junio-2018]. [Online]. Available: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- [32] “Instalación jdk 8,” 2018, [Internet; Accedido 24-junio-2018]. [Online]. Available: <https://linuxconfig.org/how-to-install-java-on-ubuntu-18-04-bionic-beaver-linux>
- [33] “Descarga eclipse jee,” 2018, [Internet; Accedido 24-junio-2018]. [Online]. Available: <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/oxygen3a>
- [34] “Descarga android studio,” 2018, [Internet; Accedido 24-junio-2018]. [Online]. Available: <https://developer.android.com/studio/>
- [35] “Descarga tomcat 9,” 2018, [Internet; Accedido 24-junio-2018]. [Online]. Available: <https://tomcat.apache.org/download-90.cgi>
- [36] “Instalar tomcat 9,” 2018, [Internet; Accedido 24-junio-2018]. [Online]. Available: <https://www.howtoforge.com/tutorial/ubuntu-apache-tomcat/>

- [37] “Instalar mysql,” 2018, [Internet; Accedido 24-junio-2018]. [Online]. Available: <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04>
- [38] “Repositorio github de ubuassistant,” 2018, [Internet; Accedido 24-junio-2018]. [Online]. Available: <https://github.com/cgc0045/TFG-UBUassistant>