



TRABAJO FIN DE GRADO

**Desarrollo de aplicación cliente-servidor para la
creación y realización de encuestas mediante
móvil**

Autor:

Alejandro Latorre Alberich

Director:

José Ramón Gállego Martínez

Escuela de Ingeniería y Arquitectura

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Julio 2014

Desarrollo de aplicación cliente-servidor para la creación y realización de encuestas mediante móvil

RESUMEN

Este TFG consiste en el desarrollo de una aplicación para Smartphone, EncuestAPP, cuyo propósito es el de facilitar la posibilidad al usuario de realizar encuestas para compartirlas con otros usuarios y poder consultar el estado de la misma en cualquier momento. También ofrece la posibilidad de responder a la encuesta que hayas recibido y seguir el resultado de ésta. Por otro lado, existe la posibilidad de borrar la encuesta una vez haya sido creada por si queremos que deje de estar disponible y nadie pueda verla de nuevo. Todo esto se realiza a través de una aplicación móvil, un servidor basado en PHP y una base de datos MySQL que gestiona todas las encuestas y los votos de las mismas. Si el usuario quiere que la encuesta sea privada, puede añadir una contraseña que será requerida para acceder al resultado de la encuesta y, si se desea, borrarla.

El objetivo es conseguir que la aplicación sea usable, de forma que se pueda utilizar de manera rápida y sencilla.

Índice general

1	Introducción	1
1.1	Motivación y objetivo final	2
1.2	Materiales y herramientas utilizadas.	2
1.3	Organización de la memoria	3
2	Análisis previo y requisitos	5
2.1	Objetivos de la aplicación	5
2.2	Público objetivo	6
2.3	Estado del arte	6
2.4	Especificación de requisitos	7
2.5	Plataforma de desarrollo	8
3	Desarrollo de la aplicación	11
3.1	Planteamiento general	11
3.2	Funcionamiento de la aplicación	13
3.2.1	Crear Encuesta.	14
3.2.2	Historial de Encuestas.	17
3.2.3	Borrar Encuesta.	18
3.2.4	Responder Encuesta.	19
3.2.5	Mostrar Resultado.	21
3.2.6	Añadir Encuesta.	22
3.2.7	Ayuda.	23
3.2.8	Clase Encuesta.	24

4 Desarrollo del servidor	27
4.1 Planteamiento general..	27
4.2 Funcionamiento del servidor..	27
4.2.1 Información del servidor.	28
4.2.2 Tratamiento de peticiones.	28
4.2.3 Comunicación con la base de datos	29
4.3 Base de datos.	31
4.4 Comunicación cliente-servidor.	32
5 Pruebas sobre la aplicación	33
5.1 Pruebas durante el desarrollo.	33
5.2 Pruebas tras el desarrollo.	34
6 Conclusiones y líneas futuras	37
6.1 Opinión personal.	37
6.2 Desarrollo futuro.	38
6.3 Timing.	39
6.4 Conclusión final.	39
Bibilografía	41
A Diagramas de navegación	43
B Bases de datos	51
B.1 Introducción.	51
B.2 Base de datos del móvil.	51
B.3 Base de datos del servidor.	52
C Elección de Plataforma	55
C.1 Plataformas existentes.	55
C.2 Plataforma escogida.	57

C.3 Versiones de Android a las que dar soporte.	58
---	----

D Manual de Usuario	61
----------------------------	-----------

D.1 Introducción.	61
---------------------------	----

D.2 Cómo utilizar la aplicación.	61
--	----

Índice de figuras

3.1 Diagrama de relación existente entre las clases más importantes.	11
3.2 Pestañas en la aplicación.	12
3.3 Apariencia de la clase principal.	14
3.4 Diagrama de la opción Crear Encuesta.	15
3.5 Ejemplo de envío de encuesta a través de WhatsApp.	16
3.6 Diseño de Spinner.	17
3.7 Toast al finalizar el borrado de la encuesta.	19
3.8 Apariencia del apartado Responder Encuesta.	19
3.9 Diagrama de estados de responder encuesta.	20
3.10 Gráfico de barras.	21
3.11 Gráfico de sectores.	22
3.12 Diagrama de estados de Añadir Encuesta.	23
3.13 Diseño de las pestañas del apartado Ayuda.	24
4.1 Diagrama de bloques del funcionamiento del servidor.	31
6.1 Diagrama de Gantt del proyecto.	39
A.1: Diagrama de navegación de la pantalla principal.	43
A.2: Diagrama de navegación del apartado Crear Encuesta.	44
A.3: Diagrama de navegación del apartado Historial de Encuestas.	45
A.4: Diagrama de navegación del apartado Borrar Encuesta.	46
A.5: Diagrama de navegación del apartado Añadir Encuesta.	47
A.6: Diagrama de navegación del apartado Ayuda.	48

A.7 Diagrama de funcionamiento del servidor.	49
C.1: Reparto de versiones de Android a 1 de Noviembre de 2013	58
C.2: Reparto de versiones de Android a 1 de Mayo de 2014.	59
D.1 Pantalla principal de EncuestaPP.	62
D.2 Pantalla Crear Encuesta.	62
D.3 Apariencia del apartado Responder Encuesta.	63

Índice de tablas

T.1: Ejemplo del contenido de las tablas.	51
T.2: Ejemplo del contenido de la tabla <i>Encuesta</i>	52

Capítulo 1

Introducción

En primer lugar, se puede ver a diario como el uso de teléfonos móviles en la población está aumentando de manera considerable. Desde los más pequeños hasta la gente mayor disponen de un Smartphone. Gracias a esta nueva tecnología, se ha hecho posible que la comunicación sea más sencilla y pueda realizarse desde cualquier lugar. Podemos acceder a contenidos en internet a través de nuestros móviles en cualquier momento gracias a la tecnología 2G, 3G y ahora 4G.

Por tanto, consideramos que se puede utilizar esta tecnología actual para crear una aplicación para móviles que sirva a los usuarios para realizar encuestas de manera rápida y sencilla para poder concretar una fecha para una cita, decidir qué película quieren ir a ver al cine, o cualquier pregunta que se les ocurra. Gracias a esta herramienta, los usuarios podrán responder en el momento que quieran y no tendrán que estar junto con los demás compañeros o amigos para poder participar en la elección. Actualmente la gente no dispone de tiempo suficiente para reunirse y tomar alguna decisión, o, simplemente, de estar pendiente de su teléfono móvil para ver en todo momento si ha recibido alguna llamada o algún mensaje que tenga que ver con alguna decisión en la que tenga que participar.

1.1 Motivación y objetivo final

El desarrollo de este Proyecto de Fin de Grado nace a partir de una idea personal que tenía de hace tiempo. Más de una vez me había ocurrido en mi vida cotidiana que en un grupo no conseguíamos ponernos de acuerdo en qué día quedar para hacer un trabajo o si queríamos ir a un sitio o a otro. Esto se debía la mayor parte de las veces porque no todos los integrantes del grupo contestaban y al final siempre había problemas. Por eso se me ocurrió que podría existir alguna aplicación para móviles en la que de manera rápida y sencilla para el usuario se pudieran crear encuestas para que la gente pudiera votar simplemente haciendo un par de “tics” en la pantalla del móvil. Le propuse la idea a mi director de proyecto y le pareció correcta para un Trabajo de Fin de Grado por lo que seguimos adelante con la idea.

El objetivo final del proyecto sería por tanto, realizar una aplicación para dispositivos móviles que fuera capaz de crear, responder y eliminar encuestas de manera rápida y sencilla. Para ello debíamos crear también un servidor que fuera capaz de gestionar la creación de las encuestas, los votos que realicen los usuarios en ellas y las posibles peticiones de eliminar las encuestas que han sido creadas anteriormente.

1.2 Materiales y herramientas utilizadas

Para el desarrollo de la aplicación, se utilizó el lenguaje de programación *Java* [1] y el entorno de desarrollo *SDK Eclipse* [2]. Para la creación del servidor se usó lenguaje *PHP* [3] y el entorno *PHPDesigner*. Para la base de datos del servidor PHP se utilizó una base de datos de tipo *MySQL* [4] ubicada en una Raspberry pi [5]. Para la gestión de las encuestas dentro de cada terminal móvil se utilizó la base de datos que aporta Android, *SQLite*. Para la interfaz gráfica se empleó *Adobe Photoshop CS6*. Para las pruebas se ha utilizado *Genymotion* que permite tener dispositivos virtuales de móviles diferentes y mi propio móvil, *Google Nexus 5* (versión 4.4.2).

1.3 Organización de la memoria

El contenido de la memoria se divide de la siguiente manera:

- En el **Capítulo 1** se ha realizado una breve introducción del TFG y el objetivo final que se quiere alcanzar.
- En el **Capítulo 2** se sitúa la aplicación en el contexto actual y se explica el análisis que se ha realizado previo al desarrollo. Se incluyen objetivos, estado del arte, público objetivo, especificación de requisitos y plataforma de desarrollo.
- En el **Capítulo 3** se explica el funcionamiento de la aplicación desarrollada y la relación entre las clases programadas.
- En el **Capítulo 4** se explica el funcionamiento del servidor y la relación entre las clases y funciones.
- En el **Capítulo 5** se desarrollan las pruebas que se han ido realizando para el correcto funcionamiento de la aplicación.
- En el **Capítulo 6** se realiza una pequeña conclusión final y algunos detalles para mejorar la aplicación en un futuro.

En la parte final se pueden encontrar diferentes Anexos que se indican a continuación donde aparecerá la información complementaria a la memoria principal:

- En el **Anexo A** aparecen unos diagramas de navegación para comprender de manera más sencilla el funcionamiento de la aplicación.
- En el **Anexo B** se explica más detalladamente cómo funcionan las bases de datos tanto del móvil como del servidor.
- En el **Anexo C** se encuentra el proceso de elección de la plataforma final.

- En el **Anexo D** se detalla el manual de usuario para el correcto uso de la aplicación.

Capítulo 2

Análisis previo y requisitos

2.1 Objetivos de la aplicación

Con esta aplicación se busca facilitar la creación de encuestas con múltiples opciones al usuario final. Se debe conseguir que el usuario prefiera utilizar nuestra aplicación y, para ello, debe ser fácil y rápida de utilizar. De esta forma se conseguirá que la gente se anime a utilizar nuestra aplicación.

Tras analizar los requisitos de la aplicación se optó porque la aplicación se dividiera en 4 partes diferenciadas:

- Una parte de creación de encuestas, en la cual el usuario será capaz de crear la encuesta que quiera añadiendo un máximo de 10 respuestas posibles.
- Una parte de historial de encuestas, donde el usuario podrá acceder al resultado de todas las encuestas que haya creado y respondido.
- Una parte de borrar encuestas, para que el usuario pueda borrar alguna encuesta que haya creado y por alguna razón desee que deje de existir.
- Una parte de responder encuesta, la cual no será visible directamente al abrir la aplicación, sino que aparecerá cuando se acceda a la encuesta a través del enlace de la misma.

2.2 Público objetivo

La idea principal surgió en un principio para un enfoque más relacionado con temas laborales (fechas, horarios para quedar, etc.), pero luego se vio que podría enfocarse a cualquier tipo de persona. Tanto mayores como pequeños disponen en la actualidad de un Smartphone por lo que eso no nos restringe la edad objetivo. Además, aunque las necesidades sean diferentes, todos en algún momento dado necesitamos tomar una decisión con varios compañeros o amigos, desde unos chicos de 14 años que quieran ir a ver una película al cine, hasta un grupo de profesores que tenga que concertar una reunión.

Por tanto, a la hora de realizar tanto el diseño gráfico como elegir la ubicación de las diferentes secciones, se debía tener en cuenta las capacidades y necesidades de cada sector.

Se decidió utilizar un diseño que fuera colorido para atraer la atención de los más jóvenes y que no pase desapercibido. Por otro lado se intentó que fuera lo más claro posible para que tanto los más mayores, que no están todavía muy familiarizados con las nuevas tecnologías, y los más jóvenes, puedan utilizarla sin complicaciones.

2.3 Estado del arte

La idea de la aplicación aparece cuando me encuentro con la necesidad de concretar una fecha con unos compañeros y éramos incapaces de ponernos de acuerdo. Para solucionarlo pensé en la posibilidad de encontrar una aplicación para móviles en la cual poder realizar una encuesta y compartirla con mis compañeros para que respondieran directamente. Se estuvo buscando alternativas en *Play Store* pero ninguna tenía las características idóneas. Por tanto se decidió realizar una aplicación con las características antes mencionadas. A continuación se detallan todas las aplicaciones que se encontraron y sus principales inconvenientes:

- **AskMe Encuestas:** Diseño muy pobre con mucha publicidad intrusiva para el usuario. Dispone de versión web para responder encuestas y admite hasta 20 respuestas.

<https://play.google.com/store/apps/details?id=net.softnow.askme>

- **Loop:** Diseño cuidado y enfocado a establecimientos que desean conocer la opinión de sus clientes. Mayor dificultad a la hora de crear una encuesta sencilla y compartirla con tus amigos.

<https://play.google.com/store/apps/details?id=com.loopsurvey.loop>

- **Quick Poll:** Limitada en su versión de prueba a 3 opciones, diseño poco cuidado y difícil de utilizar.

<https://play.google.com/store/apps/details?id=com.greggreno.quickpollfree20>

- **PicknPoll:** Esta aplicación es la que encontré que se parecía más a la que queremos hacer pero a la hora de crear la encuesta no se crea debido a algún fallo por lo que no funciona.

<https://play.google.com/store/apps/details?id=com.picknpoll.android>

Una vez ya había comenzado el desarrollo del proyecto, Google publicó una versión móvil de *Doodle* (principal aplicación para fijar horarios). Se detalla a continuación:

- **Doodle:** Su aplicación móvil cuesta 2,69€ a diferencia de la nuestra que sería gratis. De todas formas, dispone de versión web para móviles pero dependiendo del terminal del que disponga el usuario puede que le vaya muy lento o no se le muestre correctamente. Además, la creación de encuestas en *Doodle* desde el móvil puede llegar a ser tedioso para un usuario que no esté familiarizado con ello.

<https://play.google.com/store/apps/details?id=ch.neoos.doodle>

En conclusión, a pesar de que existen aplicaciones que están enfocadas en el mismo campo, en general presentan varias carencias, bien sea en el diseño, en la facilidad de uso, o en el público objetivo. Por ello, creemos que nuestra aplicación puede aportar lo necesario en el mercado para llegar a todos los usuarios y que sus necesidades queden resueltas.

2.4 Especificación de requisitos

En resumen de todo lo anterior, se detalla a continuación la lista de requisitos finales que debe tener la aplicación:

- **RF1:** La aplicación debe permitir al usuario la creación de encuestas de manera fácil e intuitiva, así como acceder al resultado de las mismas y permitir su borrado si así lo desea el usuario.
- **RF2:** La aplicación debe ser capaz de guardar internamente un historial de todas las encuestas creadas y respondidas.
- **RF3:** La aplicación permitirá al usuario responder a encuestas que hayan sido creadas a través de un enlace directo.
- **RF4:** La aplicación será capaz de comunicarse con el servidor PHP alojado en un dominio web y diferenciar el tipo de petición.
- **RF5:** El servidor será capaz de recibir las peticiones de la aplicación y guardar en una base de datos todas las encuestas y su resultado.
- **RF6:** El servidor devolverá a la aplicación el resultado de las peticiones que se hayan producido.
- **RF7:** La aplicación contará con una versión en inglés para el público de habla no española.
- **RF8:** La aplicación mostrará el resultado de las encuestas con dos tipos de gráficos, uno de barras horizontales y uno circular con sectores diferenciados.
- **RF9:** La aplicación será alojada en el *Play Store* para que pueda ser descargada por los usuarios

2.5 Plataforma de desarrollo

Para elegir la plataforma en la que se iba a desarrollar la aplicación se hizo un estudio de las distintas opciones. Finalmente se optó por implementarlo en dispositivos *Android*. Las principales razones por la que se tomó esta decisión fueron la facilidad que aporta Google para el desarrollo de aplicaciones y la gran cuota de mercado que tiene *Android*. La aplicación será compatible con todos los dispositivos *Android* con versión 2.2 y superiores. De esta forma se conseguirá abarcar la totalidad del mercado. En el **Anexo C** aparece la descripción detallada de la elección de la plataforma.

Capítulo 3

Desarrollo de la aplicación

3.1 Planteamiento general

Al desarrollar la aplicación en el entorno de *Eclipse* con lenguaje *Java*, la distribución de la aplicación se ha implementado mediante clases. Todas ellas se han agrupado en un paquete llamado *encuestapp* ya que resultó la forma más sencilla de ordenarlo. En la Figura 3.1 se puede ver cómo se relacionan unas clases con otras. Los diagramas de navegación en los que aparecen todas las clases que se utilizan se encuentran en el **Anexo A**.

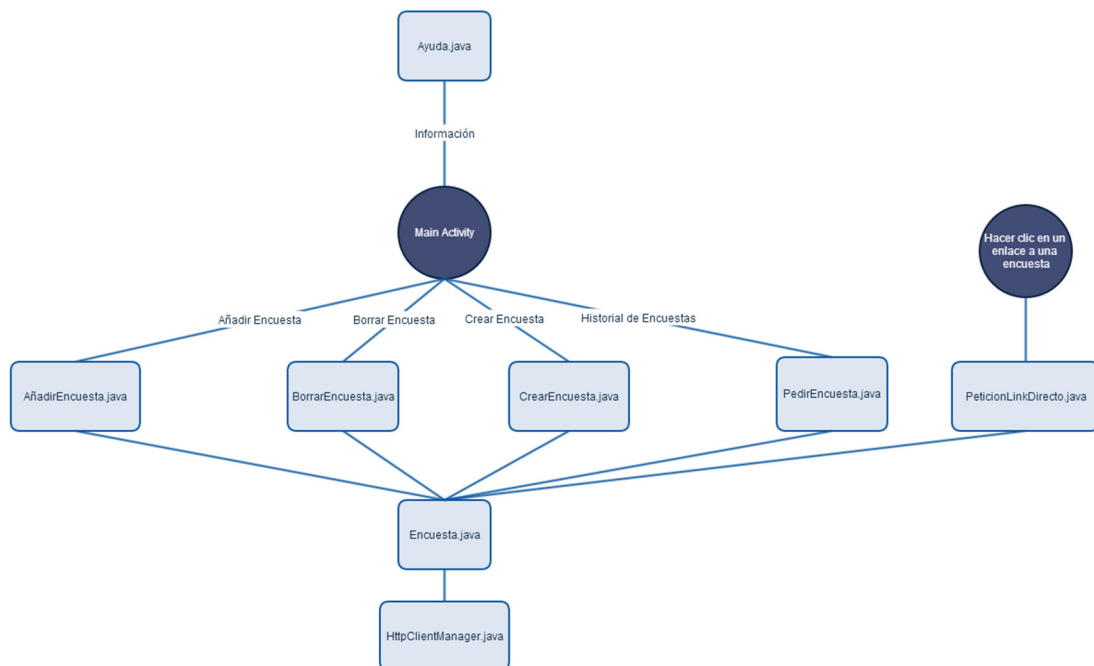


Figura 3.1 Diagrama de relación existente entre las clases más importantes.

En los siguientes capítulos se hará referencia a todas las clases que se han ido creando para desarrollar la aplicación. El código se puede encontrar en el CD adjunto a la memoria. Además se han añadido algunas capturas de las distintas pantallas de la aplicación. Todas ellas se muestran en castellano pero la aplicación está traducida también a inglés para que pueda ser utilizada en cualquier lugar del mundo.

La mayoría de las clases que se han utilizado para la creación de la aplicación derivan de la clase *Activity*. Todas estas clases representan la imagen que aparece en la pantalla del dispositivo. Por tanto se tendrá una clase de tipo *Activity* por cada una de las pantallas por las que podrá navegar el usuario.

Para la transición entre clases se recurrirá al método *Intent* a través del cual se puede decidir a qué clase queremos acceder una vez haya ocurrido el evento correspondiente. En algunos casos, las diferentes clases se deben comunicar entre sí para transmitirse información. Para esos casos utilizaremos un tipo de dato llamado *Bundle*. Este tipo de variable se añadirá al método *Intent* y se recibirá en la clase a la que hayamos accedido.

En algunas clases como en el *Historial de Encuestas*, se ha incluido una variante para poder tener en la misma *Activity* dos pantallas diferentes que se seleccionan a través de pestañas. Para conseguirlo se ha utilizado *TabHost*. Este recurso permite asignar a cada una de las pestañas el código *XML* [15] de lo que se muestra por pantalla y tratar en la misma clase los diferentes eventos de cada pestaña.



Figura 3.2 Pestañas en la aplicación.

Como se ha comentado anteriormente, la aplicación deberá guardar las distintas encuestas que se han ido respondiendo y creando. Para ello se utilizará una base de datos SQLite que proporciona directamente la plataforma de *Android*. Esta base de datos contendrá 3 tablas diferenciadas:

- Encuestas Creadas: En esta tabla se guardará la información *Pregunta de la Encuesta* y *Código de Encuesta* de todas las encuestas que vaya creando el usuario. Podrá eliminarlas de la lista cuando quiera.
- Encuestas Respondidas: En esta tabla se guardará la información *Pregunta de la Encuesta* y *Código de Encuesta* de todas las encuestas que vaya respondiendo el usuario. Podrá eliminarlas de la lista cuando quiera.
- Encuestas Respondidas Totales: En esta tabla se guardará la información *Pregunta de la Encuesta* y *Código de Encuesta* de todas las encuestas que vaya respondiendo el usuario. El usuario no podrá acceder a ellas.

En el **Anexo B** se explicará detalladamente cómo funciona la base de datos y cómo es gestionada.

Además, la aplicación debe comunicarse con un servidor ubicado en Internet. Para ello se dispone de dos clases que son las encargadas de realizar la comunicación. En una asignamos todas las variables que deben ser enviadas (clase *Encuesta*) y, esta misma, es la encargada de ejecutar la llamada a la otra clase (clase *HttpClientManager*) que mandará finalmente toda la información al servidor.

La aplicación se encuentra subida a la tienda de Android *Play Store* [11] para permitir su descarga. El enlace a la aplicación es el siguiente:

<https://play.google.com/store/apps/details?id=com.encuestapp>

3.2 Funcionamiento de la aplicación

Cuando el usuario ejecuta la aplicación, le aparece una pantalla básica con varias opciones a elegir: *Crear Encuesta*, *Historial de Encuestas* y *Borrar Encuesta*. Además, en la barra superior le aparecen dos iconos para pulsar: *Información* y *Añadir Encuesta*, como se puede comprobar en la Figura 3.3.

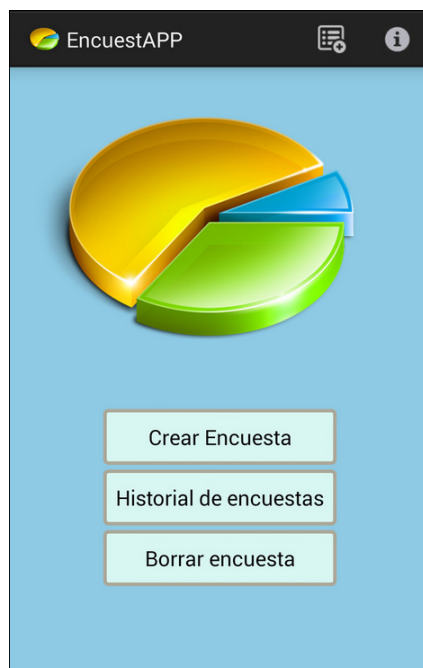


Figura 3.3 Apariencia de la clase principal.

La primera vez que se ejecuta la aplicación se generará la base de datos interna en el móvil para poder almacenar la distinta información. A continuación se explica cada una de las distintas opciones.

3.2.1 Crear Encuesta.

Esta es la opción principal de nuestra aplicación. Al pulsar en *Crear Encuesta* aparece una pantalla en la que el usuario introducirá su pregunta y sus opciones. Para ello se le ofrece en la parte superior un tipo de variable *EditText* en la que insertar la pregunta. Por defecto aparece una opción para introducir y, a través de los botones de *Añadir Opción* y *Borrar Opción* el usuario será capaz de añadir hasta un total de 10 opciones para su encuesta.

Adicionalmente se le ofrece la posibilidad de pulsar en *Encuesta Privada*. De esta forma aparecerá un cuadro de texto donde podrá insertar la contraseña que utilizará para ver el resultado de la encuesta o para borrarla. A continuación se indica un pequeño esquema de la funcionalidad de la misma.

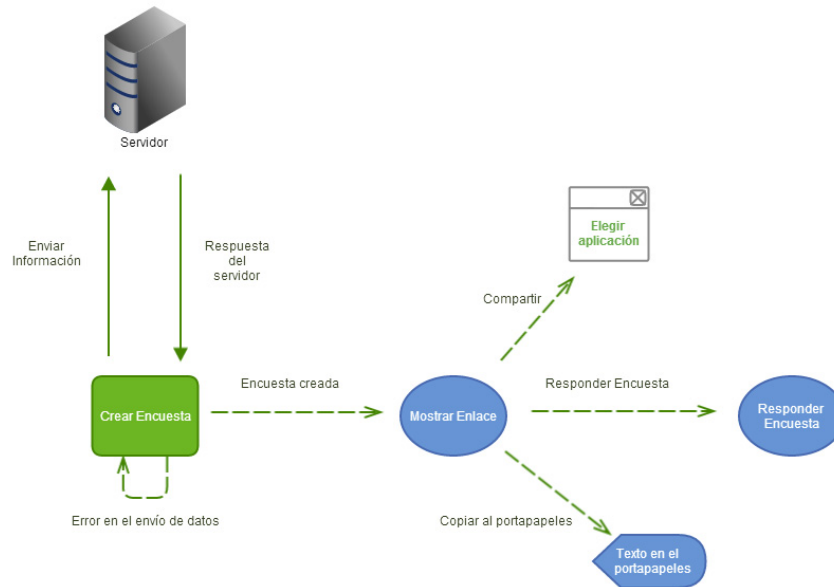


Figura 3.4 Diagrama de la opción Crear Encuesta.

Una vez el usuario pulsa en *Crear Encuesta*, se recupera toda la información que ha introducido (tanto la pregunta como las distintas opciones y la contraseña) y se añaden a las variables que contiene la clase *Encuesta*. En el momento en el que se han recogido, se crea un *Listener* (código que se ejecuta cuando ocurre un evento determinado) y se ejecuta el método de *Encuesta* encargado de enviar la información. Posteriormente se explica cómo funciona la clase *Encuesta* ya que es común a todas las opciones de las que dispone la aplicación.

En el momento en el que acaba el envío de la información y se ha recibido una respuesta del servidor puede haber dos situaciones:

- La llamada se ha realizado correctamente: Se recibe la confirmación por parte del servidor de que la encuesta ha sido guardada correctamente. Por tanto se procede a realizar un *Intent* sobre la siguiente clase llamada *MostrarEnlace*.
- Ha habido un error en la comunicación con el servidor: Si ha ocurrido un error con el servidor, se volverá a hacer la petición sobre la otra dirección que tenemos para el servidor. Esta asignación de direcciones se explicará posteriormente en la sección correspondiente al servidor. Si vuelve a fallar la comunicación, se mostrará un mensaje por pantalla indicando el tipo de error que se ha producido y se mantendrá al usuario en la misma

pantalla para que vuelva a pulsar en *Crear Encuesta*. Los tipos de errores los gestiona la clase *Encuesta* que se explicará posteriormente.

Si la comunicación ha sido exitosa, el servidor habrá devuelto un código con el que se identifica a cada encuesta. Además, se añadirá este código de la encuesta junto con la pregunta a la base de datos. Gracias a la clase *HandlerSQLite* podremos gestionar la base de datos. En el Anexo B se detalla el funcionamiento de esta clase.

Una vez finalizado, se realizará un *Intent* sobre la clase *MostrarEnlace* antes mencionada y se mostrará al usuario el enlace que se le ha asignado a su encuesta.

A través de los botones *Copiar Enlace* y *Compartir* se ofrece la posibilidad de compartir la encuesta con los destinatarios. Al pulsar *Copiar Enlace*, se copiará la dirección de la encuesta al portapapeles. De esta forma, podrá copiarlo en la aplicación que quiera. Al pulsar *Compartir* se abre un cuadro de diálogo en el que se puede seleccionar la aplicación en la que compartir el enlace. Al seleccionar una de las aplicaciones, le aparecerá directamente en el cuadro de texto de esa aplicación la dirección de la encuesta con la pregunta que ha hecho y un enlace a la descarga de la aplicación por si los destinatarios no la tienen descargada. En la siguiente figura se puede comprobar la apariencia del enlace.

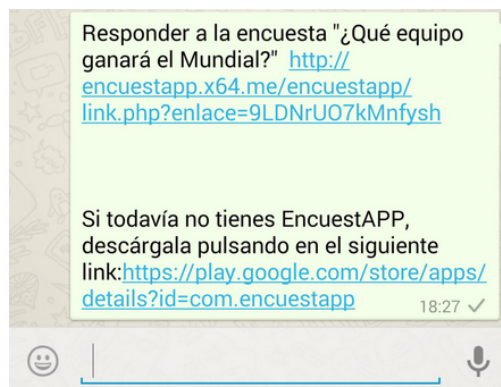


Figura 3.5 Ejemplo de envío de encuesta a través de WhatsApp.

Para concluir, aparece el botón *Responder Encuesta*. A través del mismo el creador de la encuesta podrá acceder directamente a responderla. *Responder Encuesta* se explica detalladamente más adelante.

3.2.2 Historial de Encuestas.

En este apartado se encuentran las encuestas que ha ido creando y respondiendo el usuario para acceder a su resultado. Se divide en dos pestañas como se ha comentado anteriormente. Estas pestañas son:

- Encuestas Creadas: En esta pestaña aparece un *Spinner* (lista en la que se puede seleccionar una opción determinada) en el que el usuario podrá seleccionar la encuesta que haya creado y quiera comprobar el resultado.
- Encuestas Respondidas: Al igual que en la pestaña anterior, aparece un *Spinner*, pero en este caso se incluyen todas las encuestas que se han respondido.

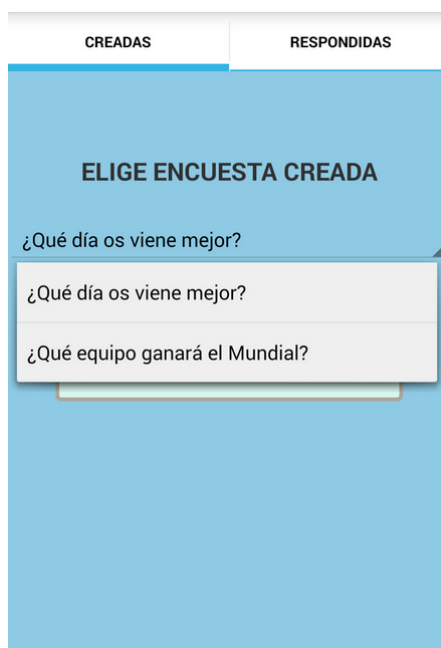


Figura 3.6 Diseño de Spinner.

Una vez se entra a este apartado, la aplicación accede a la base de datos SQLite en la que se encuentran todas las encuestas que ha respondido y creado el usuario para mostrárselas en los diferentes *Spinner*. Para ello crea una nueva instancia de la clase *HandlerSQLite* y se ejecutan los métodos correspondientes.

Para acceder al resultado de la encuesta, el usuario dispone del botón *Mostrar Resultado*. Al pulsar el botón, la aplicación recoge el enlace correspondiente a esa encuesta y añade en la clase *Encuesta* las variables necesarias para realizar la petición al servidor. El funcionamiento, si se ha producido un error, es equivalente al que se ha explicado en *Crear Encuesta*.

Si la comunicación ha sido correcta, pueden ocurrir dos eventos:

- El servidor nos ha devuelto la encuesta en la que hay contraseña: Si el usuario había puesto contraseña, se le mostrará una nueva *Activity* en la cual deberá introducir la contraseña para acceder al resultado.
- El servidor nos ha devuelto la encuesta y no hay contraseña: Si la encuesta no lleva contraseña, se mostrará directamente el resultado de la encuesta. Posteriormente se detalla el funcionamiento de la clase encargada de mostrar el resultado.

3.2.3 Borrar Encuesta.

En este apartado, al igual que ocurre en *Historial de Encuestas*, aparecen dos pestañas. Estas pestañas son:

- Encuestas Creadas: En esta pestaña aparece un *Spinner* en el que el usuario podrá seleccionar la encuesta que haya creado.
- Código: En este caso al usuario se le ofrece la posibilidad de introducir el código de la encuesta que quiera borrar.

Además de esta segunda pestaña, la otra diferencia es que en ambos casos existe un cuadro de texto en el que el usuario debe introducir la contraseña de la encuesta (en caso de que exista).

Al elegir la encuesta o haber introducido el código, el usuario debe pulsar en *Borrar Encuesta*. Como en los casos anteriores, se asignarán los valores adecuados a la clase *Encuesta* para realizar el envío de la petición.

Una vez terminada la comunicación pueden ocurrir dos eventos:

- Ha ocurrido un error: Se mostrará por pantalla que ha ocurrido un error dependiendo del error que nos haya devuelto la clase *Encuesta*.

- Ha finalizado correctamente: Se mostrará por pantalla un *Toast* (mensaje que aparece durante unos segundos en pantalla) indicando que se ha borrado correctamente. Análogamente, a través de una instancia de *HandlerSQLite*, se borrará la entrada referenciada a esa encuesta de todas las tablas en las que pudiera encontrarse.

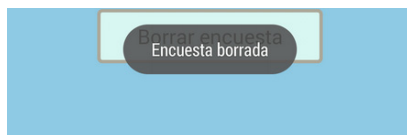


Figura 3.7 Toast al finalizar el borrado de la encuesta.

3.2.4 Responder Encuesta.

Para acceder a responder una encuesta, es necesario haber recibido un enlace previo que haga referencia a la encuesta. Por ello, no aparece en la clase principal ningún enlace para acceder a responder encuesta directamente.

Una vez el usuario haya pulsado el enlace correspondiente, se comprobará si la encuesta ha sido ya respondida. En caso contrario aparecerá una pantalla como la que se observa en la Figura 3.6, donde se puede comprobar la apariencia final del apartado *Responder Encuesta*.

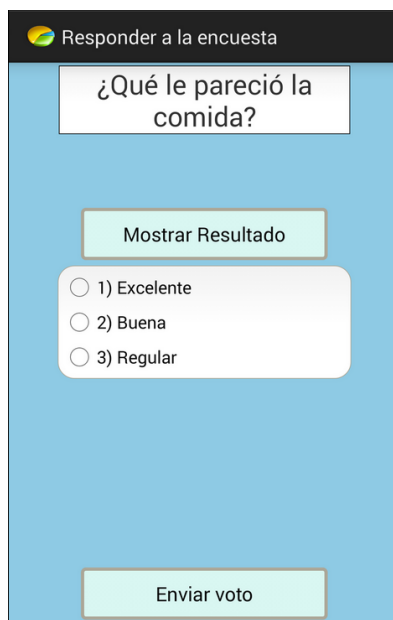


Figura 3.8 Apariencia del apartado Responder Encuesta.

Para mostrar la encuesta, la aplicación debe ser capaz de desglosar la información que recibe del servidor. Para ello, en un primer lugar, se recupera el

número de opciones que tiene la pregunta. Posteriormente recorre en un bucle el vector de opciones tantas veces como opciones hay. Cada vez que entra en ese bucle, crea un nuevo *RadioButton* (cada una de las opciones a seleccionar). Una vez creado el *RadioButton*, se le asigna como texto la opción correspondiente que hemos recibido del servidor. Finalmente se añade el *RadioButton* al grupo de *RadioButtons* (la lista que contiene todas las opciones).

Una vez que el usuario elige una opción, ya puede enviar el voto. Si no ha seleccionado ninguna opción y pulsa el botón *Enviar Voto*, le aparecerá un *Toast* como aviso indicándole que debe seleccionar alguna opción para continuar.

Si ha seleccionado una opción y pulsa *Enviar Voto*, se le asignarán los valores correspondientes a la clase *Encuesta* como en las secciones anteriores.

Adicionalmente puede acceder al resultado de la votación antes de mandar su voto por si quiere comprobar el resultado de ésta y que posteriormente su voto ha sido añadido.

Finalmente, tras enviar el voto, si no ha ocurrido ningún fallo con el servidor, se le mostrará una pantalla final en la que aparecerá un mensaje agradeciendo el uso de nuestra aplicación y otro botón para mostrar el resultado. En la Figura 3.9 se puede comprobar el funcionamiento del apartado *Responder Encuesta*.

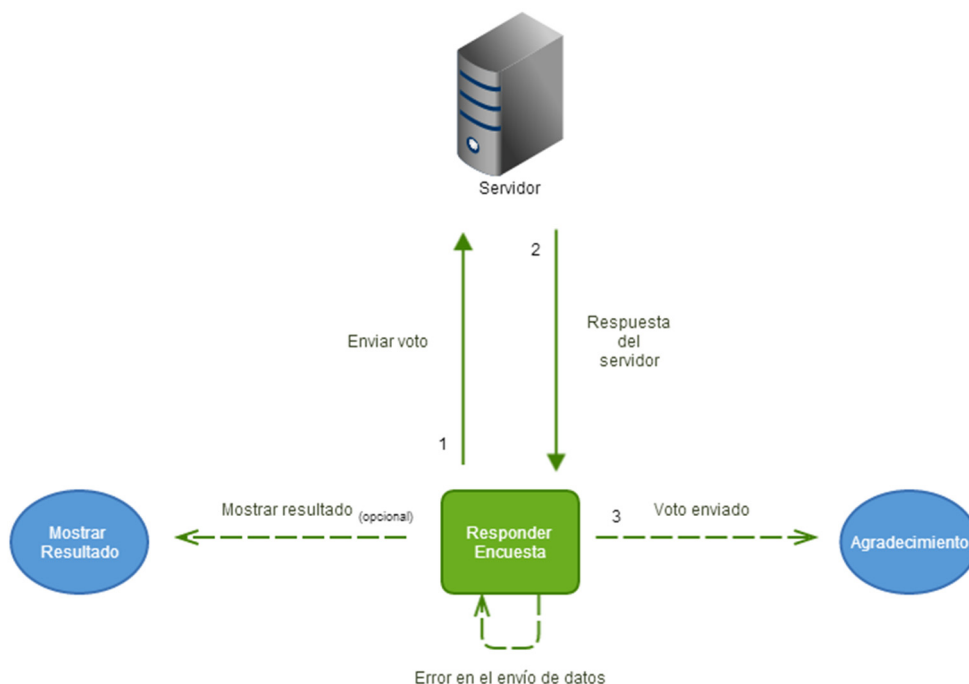


Figura 3.9 Diagrama de estados de responder encuesta.

3.2.5 Mostrar Resultado.

En este apartado, se mostrará al usuario el resultado de la encuesta. Para ello dispondrá de dos tipos de gráficos:

- Gráfico de barras: En primer lugar aparecerá el resultado con un gráfico de barras horizontales. Al lado de cada una de las barras horizontales aparecerá el número de votos para que quede más claro y, encima de cada barra, aparecerá el nombre de la opción. Para conseguir este gráfico, se han utilizado *ProgressBar* (barras que están pensadas para simular el progreso al realizar un envío de datos etc.). Aprovechando este recurso, se recuperan el número total de votos y se guarda el número de votos mayor. Finalmente se normalizan todos los votos con referencia al número mayor para establecer el tamaño de la barra.



Figura 3.10 Gráfico de barras.

- Gráfico de sectores: En el momento en el que el usuario pase a la pestaña de *Sectores*, le aparecerá un gráfico visual de tipo sectores. Para la

realización de este gráfico se ha utilizado una librería de código libre llamada *AChartEngine*. Esta librería nos permitía un gran abanico de gráficos. Se ha creído conveniente utilizar este gráfico ya que es el más representativo de cara a una encuesta. Los pasos a seguir son muy similares a los del caso anterior. Se recogen las opciones y el número de votos de cada una y se van asignando a las distintas variables de la librería.

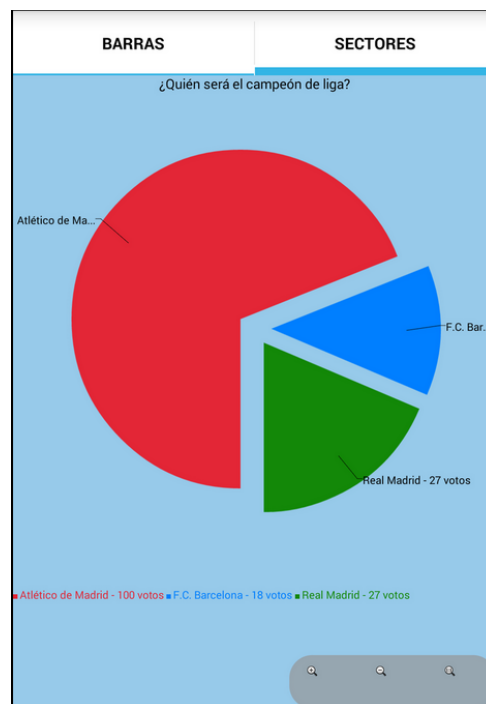


Figura 3.11 Gráfico de sectores.

3.2.6 Añadir Encuesta.

En la barra superior de la clase principal aparecen 2 botones. El primero de ellos es Añadir Encuesta.

Se ha creado esta opción para añadir una encuesta directamente a través del código asignado a la misma. De esta forma si por alguna razón el usuario no puede acceder a través del link directo, podrá hacerlo introduciendo el código.

Una vez pulsa en el botón *Ir a Encuesta*, asignará el valor del código correspondiente dentro de la clase *Encuesta*, y se realizará la consulta.

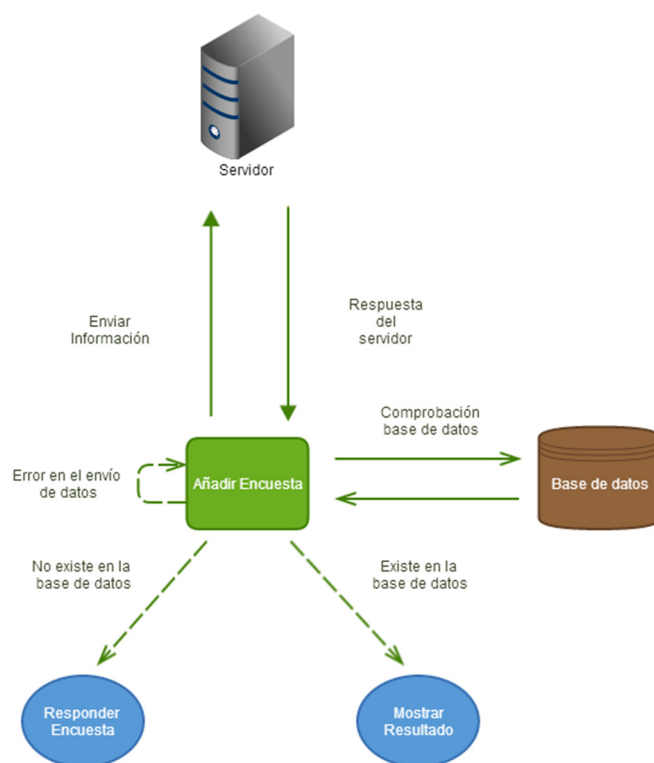


Figura 3.12 Diagrama de estados de Añadir Encuesta.

En la Figura 3.12 se puede ver el funcionamiento general de esta *Activity*. En un principio se comunica con el servidor para realizar la petición de la encuesta. Una vez recibe la encuesta con todos sus valores, se comprueba en la base de datos interna del móvil si esa encuesta ha sido respondida por el usuario. En caso de que haya sido respondida, se realiza un *Intent* hacia la clase de *MostrarResultado*. Si no ha sido respondida, se realiza un *Intent* sobre *ResponderEncuesta* para que proceda a responderla. En el caso de que tenga contraseña, antes de mostrar el resultado, se le pedirá que introduzca la contraseña como se ha comentado anteriormente.

3.2.7 Ayuda.

Como último apartado, aparece la ayuda. En ella encontraremos tres pestañas. En la primera, llamada *Ayuda*, aparece una pequeña explicación de

cada una de las partes principales de la aplicación. De esta forma el usuario será capaz de comprender correctamente la aplicación si tiene alguna duda.

En la segunda, llamada *Contacto*, aparece un pequeño cuadro de texto en el que el usuario podrá escribir sus dudas y enviarla. Al pulsar *Enviar*, aparece un cuadro de diálogo en el que el usuario seleccionará el cliente de correo que tenga instalado. De esta manera, el usuario mandará por correo su duda y podrá ser respondida.

Por último aparece *Acerca de*. En él se muestra la versión de la aplicación, el nombre del autor de la aplicación y la dirección de Twitter.



Figura 3.13 Diseño de las pestañas del apartado Ayuda.

3.2.8 Clase Encuesta.

Esta clase es básica para el funcionamiento de la comunicación cliente – servidor. En ella se encuentran los diferentes parámetros a modificar para poder enviar y recibir la información adecuadamente. Dispone de diferentes métodos para que las demás clases puedan modificar estos parámetros en función del tipo de petición. Una vez se ejecuta en la clase principal la llamada a la clase *Encuesta*, ésta realiza

una llamada a su vez a la clase *HttpClientManager*. Esta clase, que ha sido programada para utilizarla en esta aplicación, será la encargada de realizar la comunicación directa con el servidor. Sobre esta clase se añaden todas las variables para ejecutar la petición HTTP. Se crean distintos *Listeners* para poder actuar en función de si la respuesta del servidor es correcta o no.

De esta forma, una vez se haya recibido la información del servidor, la clase *Encuesta* devolverá a la clase principal la información recibida del servidor para que pueda ser tratada como corresponda.

Dentro de la clase *HttpClientManager* se ejecuta un hilo paralelo para poder realizar la comunicación a través de internet ya que lo exige *Android* a partir de las últimas versiones. En un primer momento al desarrollar la aplicación se utilizó un método en el cual no era necesario crear un hilo que realizara la petición en paralelo, pero una vez se avanzó se comprobó que estaba desfasado y se modificó la aplicación introduciendo esta clase.

Capítulo 4

Desarrollo del servidor

4.1 Planteamiento general

La segunda pieza clave para el desarrollo de nuestra aplicación es el servidor. Se debía conseguir crear un servidor que fuera capaz de comunicarse directamente con la aplicación del Smartphone a través de internet y además gestionase las distintas encuestas.

Para ello se decidió por el desarrollo de un servidor en lenguaje *PHP* ya que era el más conveniente por su facilidad para trabajar con bases de datos. Como base de datos se ha utilizado una base de datos MySQL.

Tanto la base de datos como el servidor se han alojado en una Raspberry Pi, la cual se encuentra encendida las 24 horas del día.

El servidor se ha dividido en 4 archivos ya que se ha considerado que era suficiente para lo que requería la aplicación. A continuación se explicarán los diferentes archivos y su comunicación con la base de datos.

4.2 Funcionamiento del servidor

En los siguientes puntos se desarrolla el funcionamiento de nuestro servidor. Como se ha comentado anteriormente, debe ser capaz de comunicarse con la aplicación y la base de datos de manera óptima.

A la hora de la programación se ha buscado que sea de la manera más sencilla y óptima posible.

Para el envío de datos tanto de cliente - servidor como de servidor - cliente se ha utilizado un tipo de dato llamado *Json*. Gracias a este tipo de dato se pueden asignar diferentes variables a diferentes nombres teniendo una estructura parecida al lenguaje *XML*.

4.2.1 Información del servidor

Toda la información para la comunicación con la base de datos se ubica en un fichero llamado 'globals.php'. En él se puede encontrar la dirección IP en la que se encuentra la base de datos, la base de datos a la que queremos acceder, el usuario y la contraseña.

En nuestro caso como servidor y base de datos están alojadas en el mismo dispositivo, la dirección IP será 'localhost'.

La base de datos a la que queremos acceder es 'encuestapp' y el usuario y contraseña son los que se hayan definido.

4.2.2 Tratamiento de peticiones

Para gestionar las distintas peticiones utilizamos un archivo que se ha denominado 'service.encuesta.add.php'.

Lo primero que realiza el archivo cuando le llega una nueva petición desde la aplicación móvil es comprobar si el campo correspondiente al tipo de petición está incluido en el *Json* recibido. Una vez comprueba que existe ese campo, decide de qué tipo es.

Para ello extrae la información que tiene ese campo. Puede ser de tipo 'crear', 'petición', 'borrar' o 'votar'.

Para cada uno de los distintos tipos deberá extraer la información que sea necesaria:

- Crear: En el caso del tipo crear, debe extraer la pregunta que ha introducido el usuario así como el número de opciones que hay, las propias opciones y, en caso de haberlo, el *password*.

- Petición: En el caso del tipo petición, tan sólo deberá extraer el código asociado con la encuesta ya que la finalidad de este tipo es la de devolver la encuesta con toda su información.
- Borrar: En el caso del tipo borrar, deberá extraer tanto el código asociado a la encuesta, al igual que en el caso anterior, como la contraseña en caso de que sea necesaria.
- Votar: En el caso del tipo votar, deberá extraer el código de la encuesta y el voto que ha realizado el usuario.

Una vez realizado lo anterior, se ejecutará la comunicación con la base de datos a través del fichero ‘encuesta.php’.

Al finalizar la comunicación con la base de datos, recogerá el resultado de la comunicación y la devolverá a la aplicación.

En cuanto al fichero al que se accede al pulsar el enlace de la encuesta, ‘link.php’ actúa como pasarela hacia otra dirección. Simplemente redirecciona a otro tipo de URL que la aplicación es capaz de comprender y, sobre ella, se abre la aplicación directamente y realiza la petición de la encuesta. Si el enlace a la encuesta es:

<http://encuestapp.x64.me/encuestapp/link.php?enlace=2070bDEeI62fZ1>

El archivo redirecciona a [encuestapp://encuestapp.com/2070bDEeI62fZ1](http://encuestapp.com/2070bDEeI62fZ1).

De esta forma, la aplicación se ejecuta cada vez que aparece un link con el protocolo encuestapp://. Recupera de ese enlace el código de la encuesta y finalmente construye la petición y realiza la consulta al servidor.

4.2.3 Comunicación con la base de datos

Para realizar la petición sobre la base de datos, utilizamos el último de los ficheros llamado ‘encuesta.php’. En este archivo se incluyen todas aquellas funciones que son necesarias para poder comunicarse con la base de datos.

Para poder explicar las funciones que existen en este fichero, dividiremos en tipos de peticiones como en el apartado anterior:

- **Crear:** Para la creación de la encuesta, en ‘service.encuesta.add’ se ejecuta la función *getJSONInsertEncuesta* que se encuentra en el archivo ‘encuesta.php’. Esta función recoge los parámetros que se han indicado anteriormente y llama a otra función interna denominada *insertencuesta*. En esta función se llama a la función *RandomString* que se encarga de generar el código de la encuesta, se realiza la conexión con la base de datos, se construye la petición SQL y se realiza la petición. Para evitar posibles ataques SQL, se han escapado los caracteres especiales. Finalmente se comprueba mediante otra petición SQL que se ha añadido correctamente y se devuelve el resultado de esta petición. En *getJSONInsertEncuesta* se recupera toda la encuesta y se construye un *Json* con toda la información de la encuesta. Se devuelve el *Json* al archivo ‘service.encuesta.add’ y éste finalmente lo envía a la aplicación.
- **Petición:** A la hora de realizar la petición de la encuesta, la función que se ejecuta se denomina *getJSONPedirEncuesta*. En ella se llama a la función *getEncuesta* que será la encargada de conectarse con la base de datos, construir la petición SQL mediante el código que haya recibido de la petición y finalmente recuperar la encuesta que le devuelva la base de datos. Este resultado lo devuelve a la función *getJSONPedirEncuesta*. En esta última, como en el caso anterior, se construye el *Json* con todas las características de la encuesta y se devuelve a ‘service.encuesta.add’. Finalmente se envía a la aplicación.
- **Votar:** Para acumular el voto, se utiliza la función *getJSONInsertVoto*. Funciona de manera análoga a las anteriores. Se diferencia en la llamada que realiza sobre la función *getVoto*. En esta función se realiza la conexión con la base de datos y se realiza la petición SQL. A diferencia de las anteriores, en este caso se realiza una actualización sobre una encuesta ya creada en la base de datos. Con esta petición lo que se indica a la base de datos es que actualice dentro de la tabla Encuesta y sume 1 a la columna votos correspondiente. Finalmente como en los casos anteriores se recupera la encuesta completa, se devuelve a *getJSONInsertVoto*, se construye el *Json*, se devuelve este *Json* a ‘service.encuesta.add’ y se envía a la aplicación móvil.

- **Borrar:** Para borrar una encuesta, se utiliza la función *getJSONBorrarEncuesta*. Como ocurre en los casos anteriores, esta función deriva en la función *getBorrar*. Ésta es la encargada de comunicarse con la base de datos y, a través del código y el *password*, borrar la encuesta de la base de datos. A diferencia de los casos anteriores, tan solo se devuelve el número de filas que se han visto afectadas. De esta forma *getJSONBorrarEncuesta* comprobará si es 1 ó 0. Si el resultado es 1, significa que se ha eliminado la fila, por lo tanto todo ha ido correcto. Construirá el *Json* correspondiente indicando que se ha borrado correctamente y, a través de ‘service.encuesta.add’, se devolverá a la aplicación.

A continuación se incluye un diagrama aclarativo del funcionamiento.

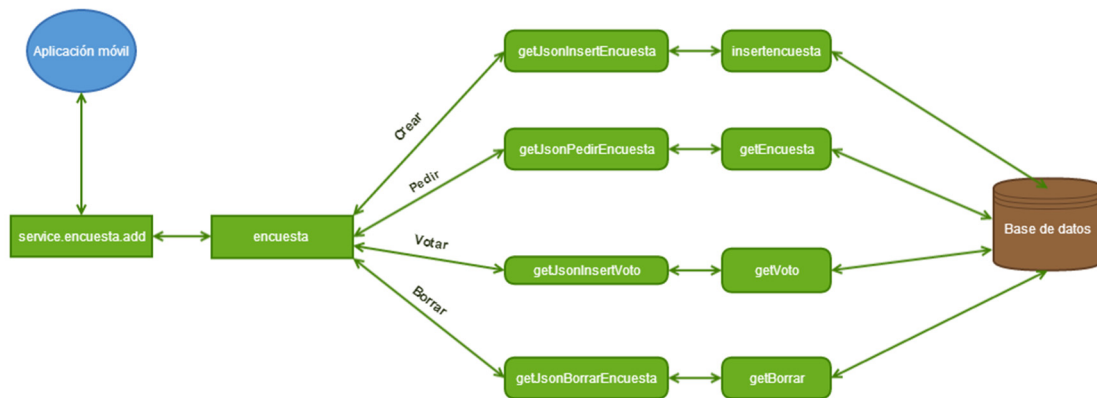


Figura 4.1 Diagrama de bloques del funcionamiento del servidor.

4.3 Base de datos

Una parte básica para el funcionamiento de la aplicación es la base de datos en la que se guardan las distintas encuestas. Tras valorar distintas opciones, se optó por utilizar una tabla en la cual se recogería toda la información.

Por tanto, tendremos una base de datos con una tabla principal que estará formada por 25 columnas.

Vienen ordenadas por un índice que se incrementa en 1 cada vez que se añade alguna encuesta nueva.

Todas las columnas por tanto son: índice, enlace, opciones, pregunta, opcion1, opcion2, opcion3, opcion4, opcion5, opcion6, opcion7, opcion8, opcion9, opcion10, votos1, votos2, votos3, votos4, votos5, votos6, votos7, votos8, votos9, votos10 y password.

De esta forma podremos almacenar hasta un total de 10 opciones y sus respectivos votos. En un futuro se podría estudiar la posibilidad de aumentar el número de opciones por si se necesitan más de 10. En el **Anexo B** se puede encontrar una representación gráfica de la base de datos.

4.4 Comunicación cliente – servidor

Como se ha comentado a lo largo de la memoria, el servidor se alojará en una Raspberry PI y el cliente se encontrará en el teléfono móvil del usuario.

Para la comunicación entre ambos se va a utilizar el protocolo *HTTP* [16]. Gracias a este protocolo, podemos realizar peticiones desde el teléfono móvil al servidor de manera sencilla y cómoda.

En un principio se planteó la posibilidad de utilizar un servidor programado en *Java* pero luego se comprobó que por problemas con el *Firewall* era mucho más cómodo realizarlo sobre PHP y que la comunicación fuera con *HTTP*.

La otra duda era si realizar la conexión a través de una dirección IP o a través de un nombre de dominio (DNS [17]). Se decidió al final por asignarle un nombre para que el enlace a las encuestas fuera más visual que una dirección IP.

Finalmente se ha optado por dotar al servidor de redundancia de nombres. Se le han asignado dos nombres diferentes ya que durante estos meses, el nombre principal estuvo caído durante varios días. Por tanto, se contará con un nombre predeterminado y uno auxiliar para añadir redundancia. La manera en la que el cliente se aprovecha de la redundancia ha sido explicada en el apartado 3.2.1.

Capítulo 5

Pruebas sobre la aplicación

5.1 Pruebas durante el desarrollo

En el siguiente apartado se van a explicar todas las pruebas que se han ido realizando y la evaluación de los usuarios.

Desde el comienzo del desarrollo de la aplicación se han ido realizando pruebas. Al ir programando, en todo momento se ha comprobado si los avances funcionaban ya que se ha creído beneficioso ir avanzando poco a poco, pero que todo fuese funcionando correctamente.

Por tanto las primeras pruebas se basaron en comprobar que las transiciones entre las distintas *Activities* se realizaban correctamente y que todos los botones funcionaban como nosotros deseábamos.

Una vez se consiguió que la parte de la aplicación que no requería la conexión a internet funcionara, se empezaron las pruebas con el servidor. En un primer momento se utilizó un servidor muy sencillo que simplemente devolviera lo que se le mandaba desde el móvil. De esta forma podíamos comprobar si estábamos mandando correctamente la información y, además, si en el servidor la recibíamos correctamente.

Más tarde se fueron incluyendo poco a poco las nuevas características que queríamos darle a nuestra aplicación. Se comprobó si conseguimos mandar una petición al servidor y que éste crease una misma entrada en la base de datos tantas veces como se accediera al servidor. De esta forma podíamos probar si las peticiones a la base de datos desde el servidor se hacían correctamente.

Tras comprobar lo anterior, el siguiente paso era que introdujera en la base de datos la información que nosotros le mandásemos ya que es la finalidad de nuestra aplicación. Se realizó de manera satisfactoria.

Lo último que quedaba por comprobar era si el servidor era capaz de devolvernos información de la base de datos ya que toda la comunicación consiste en enviar y recibir información hacia y desde la base de datos.

Una vez conseguido, solamente faltaba reordenarlo todo en la aplicación y en el servidor para que funcionase como se deseaba.

5.2 Pruebas tras el desarrollo

Al finalizar el desarrollo de la aplicación, el siguiente paso era realizar una batería de pruebas para comprobar que todo funcionaba correctamente. Durante varios días, se facilitó la aplicación a varios usuarios para que probaran la aplicación y pudiesen probar si todo funcionaba correctamente.

Gracias a estos comentarios se pudo ir mejorando la aplicación ya que sobretodo el apartado gráfico variaba mucho entre unos móviles y otros debido a la resolución de cada pantalla. Por tanto se tuvo que hacer mucho hincapié en el trabajo visual y trabajar con máquinas virtuales de móviles de diferentes resoluciones para comprobar su funcionamiento.

A lo largo de ese tiempo se fueron creando encuestas diarias, compartiéndolas entre los distintos usuarios y comprobando si se podían responder correctamente, además de borrarlas.

Atendiendo a las distintas opiniones de los usuarios se realizaron varios cambios, entre ellos:

- Añadir el gráfico de barras: En un principio solamente estaba disponible el gráfico de sectores y la mayoría de los usuarios consideraban que hacía falta un gráfico que fuera más descriptivo y directo. Se decidió por tanto por añadir un gráfico de barras.
- Limitar las votaciones de los usuarios: En un primer lugar, la idea planteada permitía a los usuarios votar tantas veces como quisieran. Finalmente tras hablar con los distintos usuarios, se eligió añadir la restricción de que tan sólo se pudiese votar una vez.

- Cambiar el modo de acceso a la encuesta: Cuando se terminó de desarrollar por primera vez la aplicación, para acceder a la encuesta una vez pulsabas el enlace se debía elegir EncuestAPP como método para abrir el enlace. Hubo problemas, ya que si elegías abrirlo con algún navegador, aparecía un error y era muy incómodo. Por tanto se modificó para que se abriera la encuesta al elegir cualquier navegador instalado en el móvil.

El resto de cambios fueron principalmente visuales y errores de programación que se fueron solventando.

Al finalizar este proceso de pruebas se subió la aplicación al *Play Store* para poder recibir mayor información acerca de posibles fallos que pudiesen existir en la aplicación.

Durante este periodo de tiempo tan sólo se ha recibido un error en la aplicación. El problema era que la aplicación crea la base de datos interna del móvil una vez se abría la aplicación desde el icono principal. Por tanto, cuando se accedía desde un enlace directamente, la aplicación intentaba comprobar si esa encuesta se había respondido ya, accediendo a la base de datos, y como no existía, se cerraba la aplicación. Se incluyó la posible creación de la base de datos si se accedía por primera vez a través de un enlace y se solventó.

Capítulo 6

Conclusiones y líneas futuras

6.1 Opinión personal

A nivel personal, la realización de este trabajo ha supuesto todo un reto ya que además de poner en práctica muchos conocimientos que he ido adquiriendo en la carrera, he tenido que realizar mucho trabajo de autoaprendizaje. Al no haber realizado ninguna aplicación para *Android*, ha sido muy complicado comprender qué pasos se tenían que ir siguiendo en cada momento. De todas formas, al no seguir unas pautas como en una asignatura normal, me ha dado libertad para decidir cómo trabajar y cómo organizarme.

El principal conocimiento que he adquirido es la programación en *Android*. A pesar de haber cursado asignaturas en las que se estudiaba *Java*, la diferencia a la hora de enfocarlo a una aplicación era bastante grande. Por tanto, un cierto tiempo lo dediqué para ir adquiriendo el conocimiento necesario respecto a *Android* para poder afrontar el comienzo del desarrollo de la aplicación. Gracias a este proyecto también he sabido cómo afrontar los problemas que iban surgiendo y encontrar las soluciones necesarias, bien acudiendo a libros de *Android*, a tutoriales, ejemplos de otras aplicaciones, foros de ayuda etc. Finalmente considero que la realización de este trabajo me ha ayudado a darme cuenta de que podía afrontar y conseguir terminar un proyecto tan importante como supone una aplicación para móviles de manera exitosa. Considero que a la hora de buscar trabajo, este proyecto me puede ayudar a estar más seguro de mí mismo y que también amplíe mi abanico de posibilidades ya que el desarrollo de aplicaciones me ha parecido muy interesante y, por qué no, podría acabar dedicándome a ello profesionalmente.

Además de la aplicación, he aprendido a programar desde cero un servidor, adecuándolo a mis necesidades. También he aprendido a gestionar una base de datos ya que en la carrera no se hace mucho hincapié en estos dos temas. Por tanto también dediqué un tiempo al autoaprendizaje de PHP y a cómo gestionar la base de datos, ampliando los conocimientos que había adquirido en alguna de las asignaturas de la carrera.

También al haber subido la aplicación al *Play Store*, he podido comprobar la aceptación que ha tenido la aplicación y poco a poco ver qué es lo que el usuario final quiere a través de sugerencias que voy recibiendo. Todo ello creo que es bastante importante ya que aunque podamos considerar que todo está programado de manera que para el usuario final sea muy sencillo, hasta que no lo prueba no podemos estar seguros de si lo hemos conseguido o no.

6.2 Desarrollo futuro

El futuro de la aplicación dependerá de la aceptación que consiga tener la misma en el mercado de *Android*. Si la acogida es bastante multitudinaria, podríamos plantearnos la posibilidad de seguir trabajando en ella y conseguir mejorar algunos puntos que podrían hacer de EncuestAPP una aplicación mejor.

Principalmente, antes de mejorar la aplicación, lo idóneo sería poder portar la aplicación a *iOS* y *Windows Phone* para poder alcanzar un mercado mayor y que todo el mundo pueda responder a las encuestas.

En cuanto a la aplicación, podríamos mejorar los siguientes puntos:

- Mejorar el diseño de la aplicación: A pesar de que estoy bastante satisfecho con el resultado final de la aplicación en cuanto al tema de diseño, creo que se podría mejorar y conseguir que tuviera un diseño más profesional. El desarrollo visual de la aplicación es bastante complicado por lo que habría que dedicar bastante tiempo en volver a reorganizar la aplicación y hacer un diseño nuevo.
- Aumentar el número de preguntas y opciones: Para que pueda ser útil de cara a empresas, se podría aumentar el número de preguntas para poder recibir mayor información del encuestado sin que tenga que acceder a más de una encuesta y sea más costoso. El número de opciones se podría aumentar por si se considera que con 10 no es suficiente. Aumentar el

número de preguntas por encuesta nos haría tener que reestructurar la base de datos ya que actualmente solamente está pensada para una pregunta.

- Activar notificaciones: Otra mejora importante que se podría realizar es el envío de notificaciones al creador de la encuesta cuando ésta sea contestada. De esta forma no tendría que ir accediendo al resultado de la encuesta sino que directamente le avisaría cuando hubiese algún cambio. El principal problema de añadir notificaciones es que se debería realizar una gestión de usuarios para que fuésemos capaces de identificar al creador y enviarle la notificación.
- Cambiar la ubicación del servidor: Al estar alojado en una Raspberry Pi, las posibilidades de que ésta falle son mucho mayores que si tenemos el alojamiento de nuestro servidor en un centro de servidores. Por tanto podríamos comprar un alojamiento para nuestro servidor y, además, un dominio web que nos asegurase mayor continuidad y menor tasa de fallos.

6.3 Diagrama temporal

A continuación se muestra el diagrama de Gantt con las diferentes tareas que se han ido realizando y la duración de las mismas.

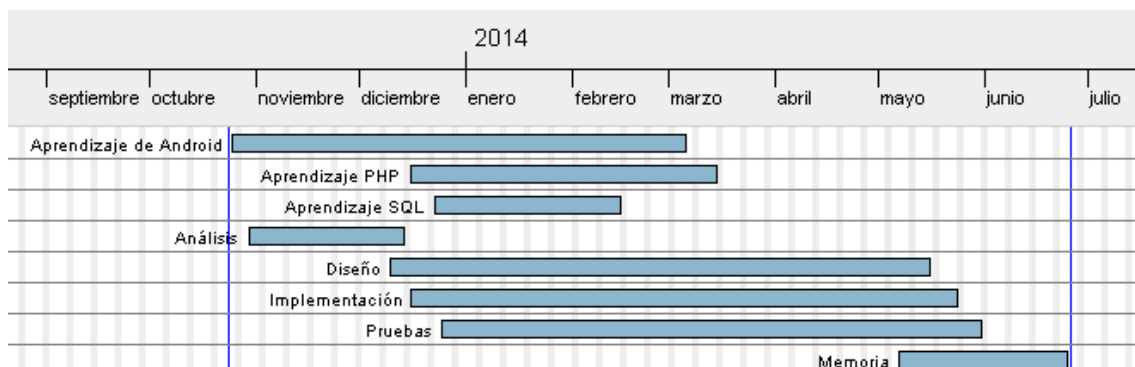


Figura 6.1 Diagrama de Gantt del proyecto.

En la Figura 6.1 se puede ver como una parte considerable del tiempo dedicado al proyecto se ha utilizado para el aprendizaje de *Android*, *PHP* y *SQL* ya que eran conocimientos básicos para poder avanzar con el proyecto. Por otro lado el trabajo de diseño, implementación y pruebas ocupan el mismo tiempo aproximadamente ya que íbamos poco a poco probando que funcionase toda la implementación.

6.4 Conclusión final

Como conclusión final, puedo indicar que estoy muy satisfecho con el trabajo realizado durante todos estos meses. Es muy satisfactorio comprobar como poco a poco va tomando forma el proyecto y, finalmente, palpar como ha finalizado con éxito el proyecto.

Hemos sido capaces de conseguir crear una aplicación capaz de crear encuestas y responderlas y, además, comprobar el apoyo que está teniendo a través de la plataforma *Play Store* como muestra el número de descargas, que confiamos en que siga creciendo durante los próximos meses.

Bibliografía

- [1] Java. Disponible en http://www.java.com/es/download/faq/whatis_java.xml, fecha de consulta: 15 de junio de 2014.
- [2] Entorno de desarrollo Eclipse. Disponible en <http://www.eclipse.org/eclipse/>, fecha de consulta: 15 de junio de 2014.
- [3] Lenguaje PHP. Disponible en <http://www.php.net//manual/es/langref.php>, fecha de consulta: 17 de junio de 2014.
- [4] Base de datos MySQL: Disponible en <http://www.mysql.com/>, fecha de consulta: 17 de junio de 2014.
- [5] Raspberry Pi: Disponible en <http://www.raspberrypi.org/>, fecha de consulta: 17 de junio de 2014.
- [6] Tienda de Aplicaciones de Google. Disponible en <https://play.google.com/store>, fecha de consulta: 20 de junio de 2014.
- [7] Satya Komatineni y Dave MacLean. Pro Android 4. Apress, 2012.
- [8] Página de desarrollo de Android. Disponible en <http://developer.android.com/index.html>, fecha de consulta: 20 de junio de 2014.
- [9] Página oficial de AChartEngine. Disponible en <https://code.google.com/p/achartengine/>, fecha de consulta: 10 de junio de 2014.
- [10] Página oficial de estadísticas de Android. Disponible en <http://developer.android.com/about/dashboards/index.html>, fecha de consulta: 15 de mayo de 2014.
- [11] Foro de desarrollo Stackoverflow. Disponible en <http://stackoverflow.com/>, fecha de consulta: 20 de junio de 2014.
- [12] Web de programación nosinmiubuntu. Disponible en www.nosinmiubuntu.com fecha de consulta: 10 de junio de 2014.
- [13] Curso de programación en Android de Salvador Gómez Oliver. Disponible en <http://www.sgoliver.net/blog/>, fecha de consulta: 10 de junio de 2014.

- [14] Tutoriales de Edu4java. Disponible en <https://www.youtube.com/user/edu4java/>, fecha de consulta: 10 de junio de 2014.
- [15] Extensible Markup Language (XML). RFC 3275, Marzo 2002.
- [16] Domain Name Server. RFC 1034 y RFC 1035, Noviembre 1987
<http://tools.ietf.org/html/rfc1034> y <http://tools.ietf.org/html/rfc1035>
- [17] Hypertext Transfer Protocol. RFC 2616 Junio 1999.
<http://www.ietf.org/rfc/rfc2616.txt>