



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Sistema de reseñas y taquilla inversa en Android

Review and Inverse Ticket system for Android OS

Eugenio José González Luis

La Laguna, 6 de septiembre de 2019

D. **Alejandro Pérez Nava**, con N.I.F. 43821179-S profesor asociado de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como tutor

D. **Fernando Pérez Nava**, con N.I.F. 42091420-V profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

“Sistema de reseñas y taquilla inversa en Android”

ha sido realizada bajo su dirección por D. **Eugenio José González Luis**,
con N.I.F. 43378765Y.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 6 de septiembre de 2019

Agradecimientos

A Tatiana por hacer de tester de la aplicación en múltiples ocasiones.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

El objetivo de este trabajo ha sido la realización de un sistema de reseñas para la plataforma Android. El funcionamiento del mismo se ha simplificado de manera que solamente haga falta escanear un código QR que identifique el servicio a valorar y se nos permita posteriormente realizar dicha valoración. Adicionalmente se han añadido funcionalidades de taquilla inversa para poder dejar propinas a aquellos servicios que nos hayan parecido especialmente satisfactorios.

Aunque existan ya múltiples plataformas y productos similares el nuestro se destaca en que la comunicación es directa con el proveedor de servicios y en que la comodidad de uso es mayor al poderse realizar desde nuestro propio dispositivo. Esto último además solucionará un problema higiénico que surge del uso de este tipo de productos cuando son en un dispositivo suministrado por el proveedor en lugares en los que pueda haber contaminación como baños u hospitales.

Para ello es necesario la codificación de una arquitectura cliente-servidor (Ambos se harán en Java), acceso a los distintos servicios del móvil como el GPS, el internet o la cámara, uso de bases de datos relacionales (Se usará PostgreSQL), la generación automática de códigos QR y una pequeña página web que sirva para ingresar nuevos servicios en la base de datos, así como para ver las estadísticas de dichos servicios.

Palabras clave: Reseñas, Aplicación, Android, Taquilla Inversa, Java

Abstract

The goal of this project was to develop a review system for Android OS. Using this system has been simplified in such a way that you'll just need to scan a QR Code identifying a service and it'll let you review that service. Additionally inverse ticket capabilities have been implemented to let the user tip those services he found specially satisfying.

Even though many platforms and similar products already exist ours stand out because communication will be direct to the service provider and it will be more convenient to use as you can emit your reviews from your own device. Lastly using your own device will solve a hygienic problem which happens when you have to make reviews on service provider devices in highly contaminated places such as toilets and hospitals.

To make this work we will need to code a client-server architecture (Both will be coded in Java), access to the mobile phone services such as the GPS, internet or the camera, the use of relational database (PostgreSQL will be used), automatic QR Code Generation, and a small website which will be used to sign new services up into the database, as well as watching each service's statistics.

Keywords: Reviews, App, Android, Inverse ticket, Java

Índice general

Capítulo 1	Introducción.....	1
-------------------	--------------------------	----------

1.1 Sección Uno.....	1
1.1.1 Tecnologías utilizadas.....	1
1.2 Aplicación.....	2
1.3 Servidor principal.....	2
1.4 Servidor de la página web.....	2
Capítulo 2 Antecedentes y Estado del Arte.....	3
2.1 Sistemas de Review On Site.....	3
2.1.1 Online.....	3
2.1.2 Offline.....	4
2.2 Sistemas de Review Off Site.....	5
Capítulo 3 Problemática.....	6
3.1 Feedback.....	6
3.2 Imagen.....	7
3.3 Marketing.....	7
Capítulo 4 Objetivos y Planificación.....	8
4.1 Objetivos.....	8
4.2 Planificación.....	9
4.2.1 Metodología de trabajo.....	9
4.2.2 Plazos de entrega.....	9
Capítulo 5 Funcionamiento del proyecto.....	11
Capítulo 6 Conclusiones y líneas futuras.....	15
Capítulo 7 Summary and Conclusions.....	17
7.1 Summary.....	17
7.2 Conclusions.....	17
Capítulo 8 Presupuesto.....	19
8.1 Sección Uno.....	19

Índice de figuras

Ilustración 1: Pagina de registro.....	14
--	----

Ilustración 2: Pagina de añadir servicios y confirmación.....	15
Ilustración 3: Lector QR.....	15
Ilustración 4: Valoraciones y Donaciones.....	16
Ilustración 5: Agradecimientos.....	16
Ilustración 6: Pagina de cuenta y de servicio.....	17

Índice de tablas

Tabla 7.1: Resumen de tipos.....	8
----------------------------------	---

Capítulo 1

Introducción

1.1 Sección uno

El producto final de este proyecto está dividido principalmente en 3 partes, a saber:

Una aplicación para el sistema operativo Android que hará las veces de cliente ligero, es decir solamente recibirá y enviará datos al servidor y en ningún momento hará cálculos propios o almacenará información sensible. De este modo se evitará que sea fácilmente *hackeable* y se disminuirá la carga en el teléfono móvil haciendo la aplicación más accesible.

Un servidor codificado en Java que hará la mayoría de los cálculos. Este servidor será, además, el que se encargue de comunicarse con la base de datos en PostgreSQL la cual almacenará toda la información de reseñas emitidas, métodos de pago, cuentas de usuario, servicios registrados, etc...

Un segundo servidor Tomcat, alojado en la misma máquina que el primero, que se encargará de controlar la página web, en la que se registran los nuevos servicios, se consultaran las estadísticas de dichos servicios, se obtienen los códigos QR que son generados previamente por el servidor principal, etc...

1.1.1 Tecnologías utilizadas

Para completar este TFG se usarán las siguientes tecnologías, herramientas o lenguajes de programación:

Java: El servidor principal, los servlets y la aplicación estarán todos codificados en Java 11, teniendo como único inconveniente que no sé podrá usar la aplicación para fines comerciales debido a que con la nueva licencia de Oracle si se usase para fines comerciales tendríamos que pagar.

XML: Los recursos, el *Manifest* y las layouts de la aplicación estarán codificadas en XML, del mismo modo los ficheros de configuración del servidor encargado de la página web estarán también escritos en XML.

Gradle: Tanto en la aplicación de Android como en el servidor principal se usará Gradle como herramienta de la automatización de la construcción.

Maven: En el servidor encargado de gestionar la pagina web se usará como herramienta de la automatización de la construcción Maven debido a problemas e incompatibilidades surgidas del uso de Gradle.

Java Server Pages: Se usará JSP para el *front end* de la pagina web que usarán los clientes para realizar los inicios de sesión, registros de servicios, etc...

Bootstrap: Se usará Bootstrap para los CSS de la página web y la decoración de la misma.

Google Pay: Se usará este servicio de Google para realizar los cobros y los pagos de las taquillas inversas de los distintos servicios registrados en la aplicación.

Google Play Services: Se usarán los servicios de Google Play para varios apartados de la aplicación como por ejemplo la obtención de la posición GPS.

Zxing: Se usarán las librerías del proyecto Zxing para el escaneo de los codigos QR

JDBC: Se usarán las bibliotecas de JDBC para la conexión del servidor con la base de datos.

JSTL: Se usará la biblioteca JSTL para poder ejecutar código dentro de los JSP como por ejemplo el uso de bucles para mostrar el listado de los servicios registrados por un usuario concreto.

Tomcat: Se usará Tomcat para la gestión de los servlets relacionados con la página web y como servidor de la misma.

PostgreSQL: Como lenguaje de gestión de la base de datos se usará PostgreSQL.

PgAdmin: Se usará PgAdmin para la gestión de la base de datos de manera directa.

Eclipse: Eclipse será el IDE utilizado para la codificación de ambos servidores.

Android Studio: Para la codificación de la aplicación se usará Android Studio, que es un *fork* de IntelliJ Idea.

1.2 Aplicación

La aplicación es bastante sencilla, limpia y transparente, en cuanto la abres sin preocuparse por menús engorrosos, registros u otras cosas innecesarias te muestra un lector de QR con el que puedes escanear un servicio registrado en la base de datos.

La aplicación solo acepta códigos QR registrados en la base de datos y si se escanease cualquier otro QR simplemente no pasaría a la siguiente pantalla y se quedaría esperando un QR válido. En el caso de que se escanee un QR válido pasará a la pantalla de valoraciones. En el caso de aceptar por error un QR no valido te pasará directamente a la pantalla final avisándote de que el QR escaneado no es válido.

La pantalla de valoraciones será también sencilla, ya que tan solo mostrará 5 caras con distintos niveles de satisfacción. Una vez el usuario elija el nivel de satisfacción se comprobará si el servicio acepta o no taquilla inversa. En el caso de no aceptarla se pasará a una pantalla de agradecimiento que permitirá salir de la aplicación o volver al escáner de QR.

Si aceptase taquilla inversa se pasará a una última pantalla en la que el usuario podrá elegir cuanto dinero donar al servicio, si eligiese 0 pasará directamente a la pantalla de agradecimiento ya comentada, en cambio si eligiese cualquier otra cantidad se abrirá la interfaz de Google Pay para realizar el pago y una vez realizado pasará a la pantalla de agradecimientos.

1.3 Servidor principal

El servidor principal tendrá un objetivo muy sencillo que es el de comunicar los distintos clientes con la BD, en una aplicación con un modelo MVC (Modelo-Vista-Controlador) el servidor correspondería al controlador, aunque sin embargo en nuestro caso concreto algunos cálculos mínimos se realizan en la aplicación o en los propios servlets para aliviar la carga del servidor.

Está codificado en Java y es un servidor multihilo, es decir por cada conexión que recibe crea un nuevo hilo para gestionarla y posteriormente este hilo es el encargado de hacer las operaciones necesarias según la llamada recibida.

La conexión se hace mediante *sockets* de Java y la comunicación con la base de datos con JDBC y las principales operaciones que realiza son las de extraer datos de la BD y registrar nuevos datos en la misma. También tiene una pequeña cache de los datos solicitados para evitar sobrecargar la base de datos principal con búsquedas y que de este modo si algún servicio ya está cargado en la cache lo extraiga de ahí en lugar de la base de datos principal.

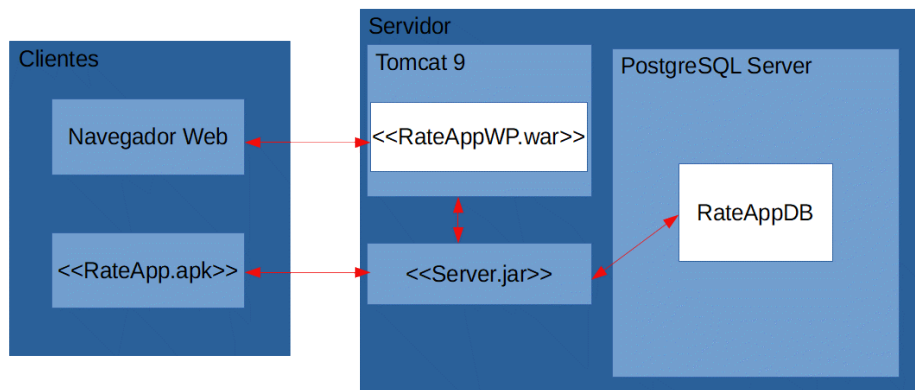
1.4 Servidor de la página web

El servidor de la página web tendrá como objetivo ser la interfaz de los proveedores de servicios que quieran ofrecer sus servicios en nuestra aplicación, contará con funciones de registro, inicio de sesión, eliminación, modificación y adición de servicios, seguimiento de los pagos (Aunque no cobros dado que los pagos se harán directamente a través de Google Pay de Cliente a Cliente)

Del mismo modo la pagina servirá para generar los códigos QR que los servicios deberán imprimir para disfrutar de la aplicación y para algunas otras funciones básicas como edición de los datos de pago, cambio de datos del perfil, etc...

1.5 Arquitectura

Las 3 partes que se han expuesto a lo largo de este capítulo se comunicarán entre ellas atendiendo al siguiente esquema de funcionamiento:



Es decir los clientes del proyecto serán tanto la Aplicación de Android como cualquier navegador web, pero cada cliente tendrá funcionalidades distintas e interactuará con el servidor de una manera distinta.

En el caso de los navegadores web, estarán destinados a que proveedores de servicios registren sus servicios, así como sus cuentas, y además a que puedan revisar estadísticas de estos servicios, obtener los QR asociados a los mismos, eliminarlos, etc. Y serán atendidos por una aplicación web gestionada por un servicio Tomcat 9, este a su vez se comunicará con la aplicación Java que hará las veces de controlador del servidor.

En el caso de la app, estará destinada a la valoración de servicios y a la realización de pagos de taquilla inversa a estos servicios y se comunicará directamente con el controlador del servidor.

Por último este controlador del servidor será el único capaz de comunicarse con la base de datos alojada en un servidor PostgreSQL y será el que obtenga e inserte los datos en la misma, así como el que realice el tratamiento de dichos datos antes de servirlos a los demás servicios.

Capítulo 2

Antecedentes y Estado del Arte

A lo largo de este capítulo se analizarán sistemas previos que ya estén en el mercado de reseñas de servicios, como nombrábamos en la introducción la gama de estos sistemas es casi infinita y la mayoría de servicios en línea los incorporan de uno u otro modo.

Por este motivo vamos a dividir en tres los sistemas que vamos a evaluar:

- Sistemas de reseñas *onsite*: Estos son sistemas de reseña que se hacen en el propio sitio donde se está realizando la transacción o se disfruta el servicio.
 - Sistemas de reseña *onsite online*: Esto se refiere a sistemas en los que se valora servicios recibidos en línea en la propia página web en la que se reciben los servicios o en su defecto en la del distribuidor de los mismos.
 - Sistemas de reseña *onsite offline*: Esto se refiere a los sistemas de reseñas instalados en, por ejemplo, las cajas de algunas tiendas, en especial de grandes superficies como aeropuertos, Corte Inglés o Decathlon.
- Sistemas de reseña remotos: Estos son sistemas de reseña en los que las valoraciones se ponen en páginas externas a las que proveen el servicio, aunque estén relacionadas con el mismo, por ejemplo, TripAdvisor previo a la inclusión de un sistema de reservas o las reseñas de Google Maps.

2.1 Sistemas de Reseña *On Site*

Los sistemas de reseña *onsite* no son los más populares, sin embargo, son los más extendidos, pues presentan una serie de ventajas mayor para los proveedores de servicio. Es importante notar como, en muchos casos por los mismos motivos por los que son favorables para los proveedores no son tan populares entre el público, junto al hecho de ser menos cómodos.

Estas ventajas son:

- Responsabilidad y veracidad: El proveedor de servicios puede verificar que la persona que deja los comentarios ha hecho uso de sus servicios y además al estar la persona presente de una forma u otra el nivel de responsabilidad que siente es mayor lo cual evita comportamientos irrespetuosos o comentarios inventados.
- Filtrado de la información: Tener la información de manera más directa hace más fácil filtrarla lo que a la larga hace que el nivel de interés del proveedor de servicios por la opinión de sus clientes incremente.

Aunque como comentábamos antes, esta responsabilidad puede cohibir a los clientes, los cuales pueden sentir que se juzga sus comentarios y por lo tanto sentirse más incómodos y ser menos honestos que lo serían con una capa de anonimidad.

2.1.1 Online

Entre los sistemas de reseña *On Site* podemos fijarnos en que son distintos si son *online* u *offline*, en el caso de los *online* tenemos diversas variaciones de estos sistemas, aunque probablemente la forma más común sean los comentarios.

Debemos entender como sistema de reseña cualquier sistema que te permita dar una opinión acerca de un servicio. Al ser esta una definición tan amplia el abanico de posibilidades es igualmente amplio y la línea que separa un sistema de reseña de un sistema de reacciones o uno conversacional es muy fina en muchos

casos.

Por poner un ejemplo, en un video de YouTube tenemos el pulgar arriba y el pulgar abajo como sistema de reseña. Sin embargo, también habla sobre la opinión que tiene la gente del video el número de visualizaciones, el tiempo que se quedan viéndolo, cuanta gente lo comparte o lo añade a listas, el tipo de comentarios que hay, el tono medio de los mismos y muchas otras métricas.

Así que para definir un sistema de reseña hará falta a su vez saber la intención que tienen las herramientas que se nos presentan en cada plataforma y como se usan. Volviendo al ejemplo de YouTube sabemos de antemano que todas estas cosas las usa la propia plataforma en su algoritmo para saber como monetizar el video, que clase de anuncios ponerle, cuando se los pone, etc... Además, estos mismos datos les son dados a los creadores de contenido para evaluar como están yendo sus videos. Así que en mayor o menor medida todo forma parte del sistema de reseñas. Por otro lado, en plataformas como un periódico digital o un *blog* muy poca gente usa la sección de comentarios para evaluar la calidad de la escritura, la veracidad de la noticia u otras cosas que nos puedan servir como reseña y lo usan más para comentar el contenido de la noticia o dialogar.

También puede darse el caso de sistemas que no fuesen pensados como sistemas de reseñas pero que indirectamente se usen como tales. Volviendo al periódico, el número de veces que tu noticia sea compartida, visitada o enlazada en especial frente a la cobertura de la misma noticia realizada por otro diario puede dar un indicativo de la calidad de tu noticia, incluso aunque originalmente fuese pensado como un método de comunicación. De hecho, PageRank, uno de los algoritmos que decide que cosas aparecen primero en Google basa la calidad y el valor que le da a cada página web en el número de enlaces que van hacia ella y la calidad de las paginas de las que vienen esos enlaces.

Por todos estos motivos que se han dado, los sistemas de reseñas online son tan extendidos y se encuentran en la mayoría de servicios. Hasta tal nivel que incluso en servicios que no disponen de ellos de manera indirecta o pasiva hay formas de saber la opinión de la gente acerca de los mismos.

El principal aspecto negativo de estos sistemas es la falta de estandarización, las dos formas más prevalentes son el conocido sistema de estrellas que podemos ver en la Play Store, o el sistema de *upvote* y *downvote* que podemos ver en Reddit, Steam o YouTube. Sin embargo, hay muchos otros o variaciones de los mismos en los que cambian la cantidad de estrellas, hay *upvote* pero no *downvotes*, hay distintos niveles de *upvote* y *downvote* (muy recomendado, recomendado, poco recomendado, no recomendado), etc... Y a la larga esto puede resultar confuso sobre todo al desplazarse de un ambiente a otro en el que tienes que acostumbrarte al sistema que allí se use.

Otro aspecto negativo es que al ser *online* o ir normalmente las reseñas acompañadas de comentarios o otros sistemas complementarios, el caos que suele surgir y el volumen masivo de información pueden dificultar la tarea de verificar la información. En especial si el servicio es grande y empieza a ser objetivo de comunidades de *trolls*, *memers*, *haters* y otros grupos igualmente dañinos y tóxicos.

2.1.2 Offline

En el caso de las *offline* el número de herramientas específicamente diseñadas para ello y los tipos de herramientas son bastante más limitados. Si bien, se podrían considerar al igual que en las *online* el flujo de clientes, las reclamaciones u otros sistemas no son los que nos preocupan en este caso porque no se tratan de sistemas digitales en la mayoría de ocasiones.

Los sistemas digitales que se usan para saber el grado de satisfacción de un cliente con un servicio *offline* se limitan casi de manera exclusiva a encuestas, en algunas ocasiones estas encuestas son realizadas de forma aleatoria por miembros del personal, en otras hay un puesto en el que realizar la encuesta de forma libre y en la mayoría se hacen tras la compra en la propia caja, en *el pin-pad* o mediante algún código a través de una página web o aplicación.

La forma que toman estas encuestas varía, aunque hay tres tipos principales: pocas preguntas sencillas, un cuestionario largo y completo que suele venir acompañado de incentivos para que los clientes los completen (Como por ejemplo ofertas) o un sistema de caras que muestren distintas emociones o de estrellas en el que valores en una sola pregunta el grado de satisfacción general.

Los tres tipos de sistemas son robustos y provocan múltiples beneficios a los propietarios de empresas que los aplican, sin embargo, cuanto más compleja sea la encuesta más probabilidades existen de que el interés de los clientes por completarla se pierda y por lo tanto hay que buscar un balance correcto en el número de preguntas que permita conocer suficiente información sin aburrir al cliente. Porque incluso en el caso de recurrir a incentivos para mantener su interés, aunque consigas que acaben las encuestas, en el punto en el que se aburran empezarán a contestar de una manera poco útil.

Del mismo modo donde pongas la encuesta puede afectar a los resultados de la misma, que sea voluntaria puede afectar al número de respuestas, que se haga con la persona a la que estás evaluando cerca puede suponer que ocurra un sesgo en las respuestas, pero hacerlo en una aplicación o página web puede generar un efecto similar a la de los sistemas *online* en el que la información deje de ser honesta.

2.2 Sistemas de Reseña Off Site

Aunque estos sistemas no son los más extendidos sí que son los más populares entre los usuarios. Dicha popularidad es debida principalmente al hecho de que estos sistemas son más cómodos, por lo general son de más ayuda a los usuarios debido a que congregan múltiples servicios en una sola plataforma, lo cual permite hacer comparativas y porque mayoritariamente se centran en el servicio de reseñas en lugar de diversificar o de usar sistemas que no son diseñados para hacer reseñas de forma integral para ello.

Un ejemplo claro de este tipo de servicios puede ser Yelp o TripAdvisor, en los que el cliente puede poner una reseña tanto numérica como redactada y los demás clientes se encargan en gran parte de la moderación marcando estas reseñas como útiles o poco útiles.

Centrándonos en el caso de TripAdvisor dado su éxito podemos ver como este servicio partió de una funcionalidad central para realizar valoraciones y una vez tuvo una base de clientes fue diversificando en base a lo que sus propios consumidores le pedían, creando un servicio cada vez más atractivo y un diseño centrado en los usuarios que sigue girando en torno a la funcionalidad de las reseñas. Algunas de estas mejoras son comparativas de los distintos lugares divididas en comparativas por zona y comparativas por categoría, una *dashboard* para los propietarios de los lugares con opciones para modificar los perfiles, contestar a las reseñas, comunicarse con los clientes, etc..., posibilidad de subir fotos, entrega de premios, diplomas, certificados de excelencia y otra miscelánea a los servicios, entre muchas otras.

De hecho, es importante estudiar este caso, porque los propios servicios luchan por tener buenas valoraciones en TripAdvisor, y se publicitan con las mismas. Los premios que otorga esta página son motivo de orgullo de muchos lugares. Un ejemplo claro de ello es el Siam Park y sus publicidades en cuanto alcanzó el puesto número 1 en parques acuáticos en Europa y un ejemplo de que este tipo de publicidad y servicios funciona es como inmediatamente comenzó a crecer de manera más rápida hasta alcanzar el puesto número 1 en el mundo, momento en el que volvió a lanzar una campaña publicitaria.

De este tipo de éxito es del que hay que aprender cuanto construyamos nuestra plataforma, pues es el tipo de sistema que se relaciona de una manera más cercana con nuestro proyecto. Por lo tanto, la fórmula debería ser construir primero la funcionalidad principal, pulirla y tenerla en un estado comercial para posteriormente escuchar las peticiones de los clientes y crecer hasta irrumpir en el mercado de los sistemas de reseñas.

Capítulo 3

Problemática

Como se vio en el anterior capítulo los sistemas de reseñas son algo bastante popular y común y por consiguiente se asume que intentan resolver un problema igualmente común y crucial. A lo largo de este capítulo se evaluará las distintas problemáticas que se resuelven al implementar un sistema de reseñas y se intentará mediante este conocimiento aplicar cambios y mejoras a nuestro diseño que eleven la calidad del producto final y la exactitud con la que se resuelven dichos problemas.

Cabe destacar antes de entrar en más detalle a todos y cada uno de los problemas que el problema principal que se pretende resolver al implantar un sistema de reseñas es la comunicación con los consumidores, y en general la obtención de *feedback*. Mediante un sistema de reseñas podemos saber que opinan nuestros consumidores de nuestros servicios y tomar medidas acordes a sus críticas y opiniones.

Aunque claramente no es este el único factor, un sistema de reseñas nos puede servir para obtener datos estadísticos sobre nuestros clientes, a que horas hay más, cuando son más activos, de donde son, comparativas entre dinero gastado y reseña, entre muchos otros.

Pero además puede servir como herramienta para promocionar la imagen de la empresa dado que se crea una confianza con el cliente, el cual al ver que nos preocupamos por saber su opinión se siente mejor atendido y por lo tanto mejora su experiencia, incluso una mala experiencia puede ser subsanada por una buena respuesta al *feedback* de un cliente.

Por último, está el factor marketing, las opiniones de los usuarios se pueden usar como herramientas de marketing las mejores opiniones se pueden añadir en nuestra página web en una sección historias de los usuarios, una nota alta puede servirnos como distintivo de calidad, mantener dicha nota puede ser un reclamo, etc...

3.1 *Feedback*

Definimos el *Feedback* o la retroalimentación como el “Efecto retroactivo de un proceso sobre la fuente que lo origina” o también como “Capacidad de un emisor para recoger reacciones de los receptores y modificar su mensaje, de acuerdo con lo recogido.” Es decir, en el caso que nos atañe que es la valoración de servicios prestados el *feedback* es la información que tenemos de vuelta de las personas a las que estamos prestando ese servicio.

Este *feedback* puede presentarse de diversas maneras y puede indicar cualidades de distintos ámbitos del servicio que se está prestando. Que los platos en un restaurante vuelvan más o menos llenos es una forma de *feedback* y puede hablar de la calidad de las comidas o del tamaño de las raciones entre otros, que los clientes dejen o no propina es de nuevo una forma de *feedback* que puede hablar del ambiente del restaurante, de la calidad de la comida, de la atención del camarero o incluso del rato que pasaron los clientes en el restaurante, un comentario que le hagan al chef sobre la comida, una reseña en TripAdvisor o Google Maps, son todas formas de *feedback*.

Tener este *feedback* es importante para los servicios puesto que esta información que se obtiene puede servir para mejorar la calidad del servicio, minimizar costes y aumentar los beneficios. Volviendo al ejemplo del restaurante si muchos platos empiezan a llegar con comida, el chef puede disminuir las raciones y minimizar los costes y las pérdidas, si siguen llegando platos con comida puede intentar mejorar la calidad.

En cuanto al *feedback* se puede considerar que nunca hay suficiente, cuanta más información complementaria haya sobre una situación más sencilla puede ser encontrar una solución a la misma. Para

terminar con el ejemplo si el camarero pregunta el motivo de la comida en los platos de una forma cordial o si se les hace encuestas a los clientes, puede evitarse dar pasos en falso o poner falsas soluciones que sean perjudiciales para la salud del negocio. Como por ejemplo disminuir la cantidad cuando el problema es la calidad y perder los clientes que viniesen porque ahora aparte de servirles mala comida les sirves poca.

Esta importancia económica de la retroalimentación es la que hace que sea la problemática principal que se intenta solucionar y alrededor de la cual debemos centrar nuestro diseño y debemos entender que toda la información que pueda recoger nuestra aplicación o que se pueda extrapolar de esta información que se recoge es valiosa para los proveedores de servicios que usen nuestra aplicación para obtener reseñas de sus negocios. Por ejemplo, la hora en la que se hacen las reseñas o la cantidad de reseñas por periodo temporal pueden ayudarnos a determinar a que es debido la valoración, o cuando son nuestros momentos de mayor actividad y podremos tomar medidas al respecto como por ejemplo poner ofertas en nuestras horas de menor actividad para atraer clientes.

También debemos entender que los proveedores de servicios que usen nuestro sistema puedan querer modularidad y quieran usar la aplicación para valorar a los distintos empleados y posteriormente usar estas múltiples valoraciones para obtener una general o incluso tener una para cada empleado separada de la general y debemos en la medida de lo posible facilitar eso.

3.2 Imagen

Otro de los problemas que se solucionan con nuestro proyecto es el de mejorar la imagen de las empresas. Especialmente, cuando se trata de empresas grandes o en las que el trato directo con el cliente es reducido dado que en muchos de estos casos se da una impresión de que el trato es frio y poco personal.

Incluir un sistema de reseñas en un servicio puede erosionar esta mala impresión y hacer que los clientes sepan que la empresa se preocupa por su opinión y por entender sus necesidades. Sentirse escuchados puede ser reparador en casos de malas experiencias, pero sobre todo puede mejorar de sobremanera la forma en la que el cliente percibe la atención que recibe de la empresa.

En general la implementación de este tipo de sistemas es un valor positivo a la imagen de una empresa que hace que esta sea percibida como una marca de confianza e interesada por las inquietudes de sus clientes y usuarios.

3.3 Marketing

Como apuntábamos en el apartado 2.2 un servicio de reseñas puede ser una buena herramienta de mercadotecnia de costes tremendamente reducidos frente a una campaña publicitaria tanto tradicional como en las redes sociales.

Si la plataforma consigue posicionarse entre los líderes del mercado de reseñas, incluso aunque sea solamente a nivel local, significa que el flujo de visitantes a las estadísticas de un servicio publicado en la misma puede tomar cantidades orbitales. Por ejemplo, en el caso de TripAdvisor esta cifra asciende a unos 315 millones de usuarios al mes.

Este gran trafico de potenciales clientes sumado a un buen *rating*, unas estadísticas interesantes y servicios complementarios que llamen la atención como comparativos de servicios, publicación de reseñas, *rankings*, fotos y perfiles, entre otros. Puede suponer que esos clientes potenciales se transformen en clientes objetivos de una forma tremendamente efectiva.

Si a esto le sumamos futuras posibilidades de campañas de marketing, entregas de premios y certificados, promociones pagadas u otras estrategias de marketing similares, un servicio puede suplir gran parte de sus necesidades de marketing usando únicamente nuestra plataforma.

Capítulo 4

Objetivos y Planificación

A lo largo de este capítulo se describirán los objetivos que se pretenden conseguir con este TFG, y una vez establecidos se generará una planificación del trabajo, esta constará de un listado de distintas tareas concretas que se vayan a completar junto a fechas en las que se pretende que estén listas estas tareas.

Como resumen preliminar cabe destacar que el objetivo principal es el de la realización de los 3 apartados descritos en el Capítulo uno, a saber: La aplicación móvil para realizar las reseñas, el servidor que se encargue de la página web en la que se inscribirán los servicios, los métodos de pago y se podrán revisar los resultados de sus servicios y el servidor que conecte tanto aplicación como página web con la base de datos.

En cuanto a la planificación se pretende seguir una metodología de prototipado rápido por lo que cada vez que se vayan implementando nuevas funcionalidades se compilarán y se pondrán a prueba como pequeños productos finales y a partir de estos podremos ver los cambios que hacen falta de cara al nuevo prototipo.

Teniendo en cuenta siempre que se pretende tener el proyecto acabado para la fecha de entrega de la convocatoria de septiembre.

4.1 Objetivos

Aunque un sistema de reseñas puede ser una herramienta completa y compleja el objetivo de este TFG es el de realizar las funcionalidades básicas de un sistema de este tipo. Para ello se ha dividido en trabajo en distintos módulos:

En primer lugar, se debe diseñar y crear una base de datos que contendrá toda la información referida a las cuentas de usuario, reseñas, pagos, servicios y demás funcionalidades que se requieran en el TFG. Esta base de datos será desarrollada en PostgreSQL y será el único lugar donde habite el modelo de datos del proyecto.

En segundo lugar, se desarrollará un servidor que será el encargado de obtener, modificar e insertar datos de la base de datos, adicionalmente tratará y transformará estos datos según sea necesario y los servirá a las distintas interfaces a las cuales podrán acceder los usuarios.

En tercer lugar, se desarrollará una aplicación con la que los clientes de los servicios podrán hacer las reseñas de los mismos. Aquí es donde se empezarán a ver las limitaciones de las que hablábamos en el primer párrafo puesto que la aplicación estará limitada a la función de, tras escanear un código QR que identifique a un servicio, permitirte, si estás en el perímetro determinado por el proveedor de servicios como válido, dar una nota del 1 al 5 sobre tu satisfacción con el servicio y por último si el servicio ha activado el sistema de taquilla inversa hacer una donación voluntaria al mismo.

Y por último se desarrollará una página web, destinada en primera instancia únicamente a los proveedores de servicios, en la cual podrán registrar sus cuentas, añadir servicios al sistema para que comiencen a ser valorados, modificar sus datos referidos a la pasarela de pagos y ver las estadísticas de sus propios servicios. Es en este último punto en el más que se notan las limitaciones de alcance de este proyecto dado que es en la página web donde se podría hacer más hincapié en todos los servicios de mercadotecnia de

los que se hablan en el apartado 3.3.

4.2 Planificación

4.2.1 Metodología de trabajo

La metodología de trabajo utilizada fue similar a un modelo de prototipos o a uno incremental, aunque no casa completamente con la descripción de ninguno de los dos:

Inicialmente se hizo una elicitación de requisitos, durante la reunión correspondiente al primer seminario del TFG. Es decir, el tutor explicó la problemática que se requería resolver y las funcionalidades y mecanismos que iba a ser necesario implementar para ello.

Posteriormente, una vez obtenidos los requisitos se realizó un análisis, los conocimientos obtenidos del mismo fueron aprovechados para realizar el anteproyecto, tomar decisiones acerca de las metodologías y herramientas a utilizar e investigar sobre los temas que me eran desconocidos previamente a la realización de la práctica.

Tras ello se comenzó a el trabajo propiamente dicho, a partir de este momento la metodología fue la siguiente:

Se seleccionaba un requisito, *bug*, funcionalidad o cualquier otra tarea a desarrollar e implementar, se trabajaba en el hasta tenerlo desarrollado y completamente probado, se integraba con el resto del sistema y se hacían pruebas de integración y se entregaba el prototipo a algún *stakeholder* o *tester*, para que diese el visto bueno y volver a seleccionar otro requisito. O pidiese cambios respecto al análisis inicial.

Idealmente esta entrega de prototipos debería haber sido al tutor, pero problemas de compatibilidad de horarios y con el acceso a los servidores dificultaron en algunas ocasiones esta tarea. Sin embargo, siempre se mantuvo la comunicación, en especial cuando hubo dudas o dificultades o se necesito tomar decisiones que cambiasen el paradigma con el que se partía originalmente.

Se continuó usando esta metodología hasta tener un prototipo lo suficientemente completo como para considerarse un entregable.

A grosso modo la hoja de ruta que se siguió fue: trabajar primero en la base de datos, posteriormente en la aplicación, luego en el servidor y por último en la página web.

4.2.2 Plazos de entrega

La idea inicial cuando nos enfrentábamos a este proyecto era tenerlo acabado para la entrega de Julio o para el mes de Julio y debido a la incompatibilidad de horarios entre tutor y alumno se estructuraron las entregas para hacerse de manera mensual.

La planificación inicial como se comentó en el apartado anterior era, a lo largo del primer mes, es decir marzo, tener realizado el análisis, diseño y haber completado la base de datos y el servidor. Tras esto, el siguiente mes sería dedicado a la aplicación, mayo sería dedicado a la página web y por último en junio se cumplimentaría la memoria. Teniendo en cuenta que tras la realización del análisis y debido a la metodología que se opto por usar las prioridades podrían cambiar durante el desarrollo.

La primera entrega se atraso hasta el mes de abril, por lo que ya se habían hecho avances en todos los frentes del desarrollo, la aplicación, el servidor y la base de datos estaban prácticamente en su forma final y era la página web la que carecía de mayor número de funcionalidades puesto que faltaba aún definir algunos requisitos funcionales referidos a ella. Durante esta reunión se definieron estos requisitos y se solventaron dudas acerca del sistema de pagos, pero no se pudo probar los avances debido a problemas de conectividad con el servidor. También durante esta reunión la hoja de ruta cambió, la idea se mantuvo a una entrega mensual, pero debido al avanzado estado del desarrollo se decidió poner prioridad en solucionar los problemas de conectividad con el servidor y en avanzar la memoria.

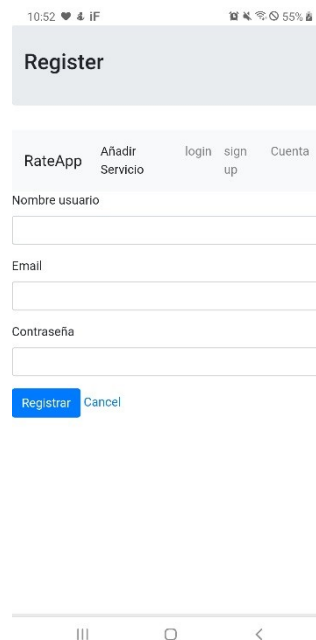
Tras ello siguió habiendo comunicación por vías electrónicas, pero sin modificaciones a la hoja de ruta, finalmente en junio todos los problemas de conectividad estaban resueltos y las funcionalidades de todas las distintas partes del proyecto estaban finalizadas por lo que se realizó una entrega en la que se pudo probar el aplicativo. Durante esta reunión se descubrieron algunos *bugs* menores que junto a la revisión final de la memoria deberían estar solucionados durante el mes de Julio.

Capítulo 5

Funcionamiento del proyecto

El funcionamiento del proyecto es sencillo:

Un proveedor de servicios accedería a la página web y se crearía una cuenta, introduciendo un nombre de usuario contraseña y correo electrónico:



The screenshot shows a mobile application interface for registration. At the top, the status bar displays the time 10:52, signal strength, Wi-Fi, and 55% battery. The app title 'RateApp' is visible in the top left. A navigation bar contains the text 'Añadir Servicio', 'login', 'sign up', and 'Cuenta'. The main form area is titled 'Register' and contains three input fields: 'Nombre usuario', 'Email', and 'Contraseña'. Below the fields are two buttons: 'Registrar' (highlighted in blue) and 'Cancel'. The bottom of the screen shows the standard Android navigation bar with three icons: a square, a circle, and a triangle.

Ilustración 1: Pagina de registro

Tras ello recibirá un correo de verificación y podrá iniciar sesión en la página web, este inicio de sesión le permitirá hacer dos cosas: añadir un servicio y revisar su cuenta. En el caso de crear un servicio obtendrá un código QR que será el que deberá imprimir para que los usuarios valoren su servicio.

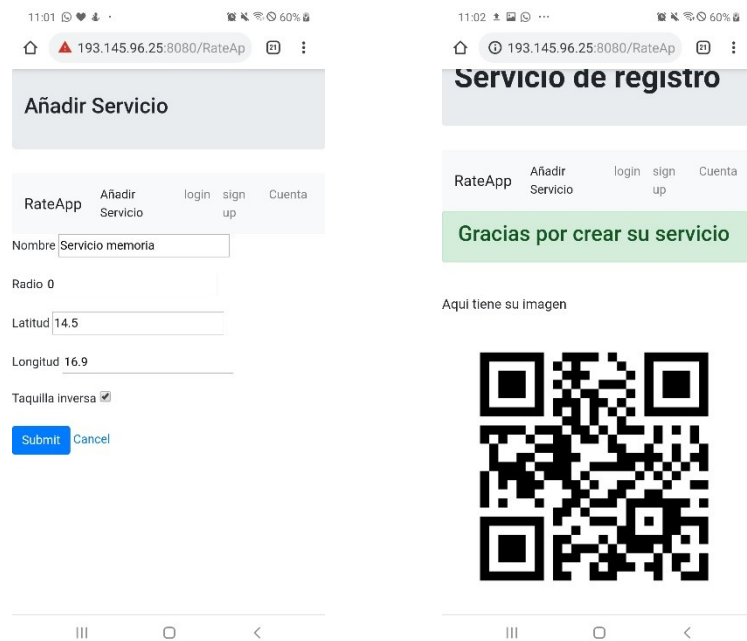


Ilustración 2: Pagina de añadir servicios y confirmación

Antes de ver la visualización de la cuenta pasaremos a revisar lo que haría un cliente que quisiera valorar este servicio, para así registrar una valoración y un pago y poder ver datos.

El cliente accedería a su aplicación desde el móvil, dicha aplicación muestra en su pantalla inicial un escáner de códigos QR con el que escanearíamos el código que el proveedor de servicios nos haya proporcionado.

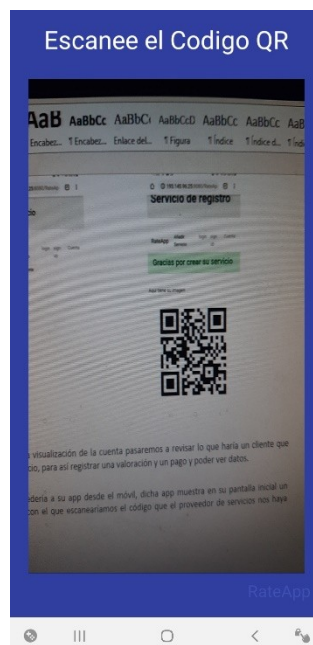


Ilustración 3: Lector QR

Una vez escaneado un QR valido, se mostrará la pantalla de valoración, que mostrará cinco caras con distintos grados de satisfacción sobre la que deberemos pulsar para dar nuestra opinión sobre el servicio. Una vez valorado, si el servicio lo tuviese activado se pasara a la pantalla de taquilla inversa.

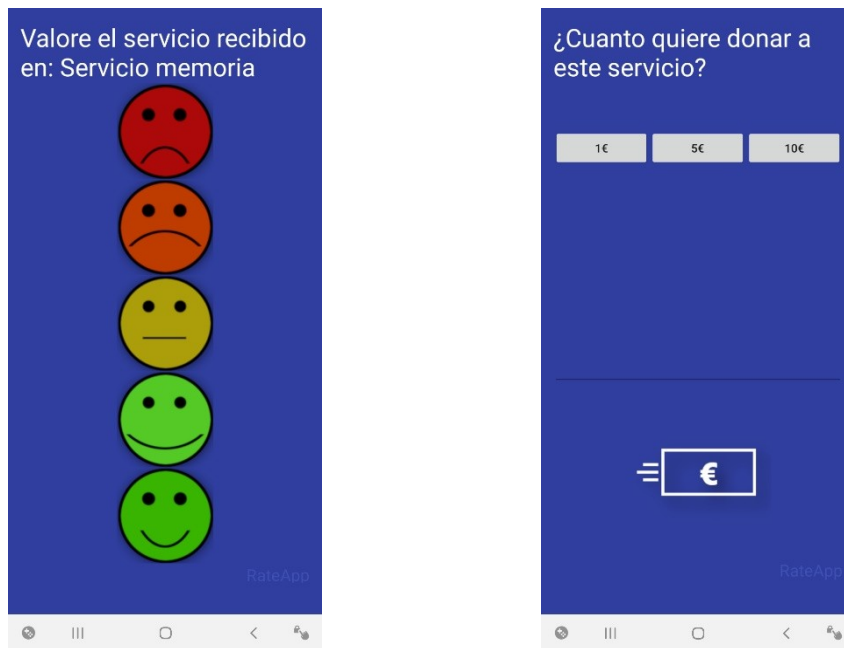


Ilustración 4: Valoraciones y Donaciones

La pantalla de donaciones usará Google Pay, pero por motivos de seguridad no se permiten sacar capturas de esta funcionalidad cuando está en marcha. Finalmente, tras realizar el pago, una pantalla de agradecimiento finalizará el proceso por parte del cliente.



Ilustración 5: Agradecimientos

Por último, y regresando al proveedor desde la página de su cuenta podrá acceder a su información de pago y al listado de servicios asociados a su cuenta y pinchando en cualquiera de estos servicios podrá ver estadísticas asociadas a los mismos.

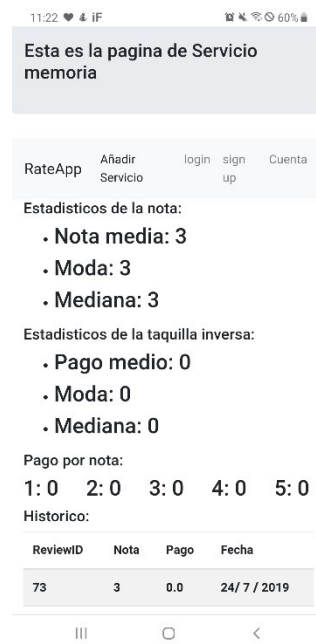
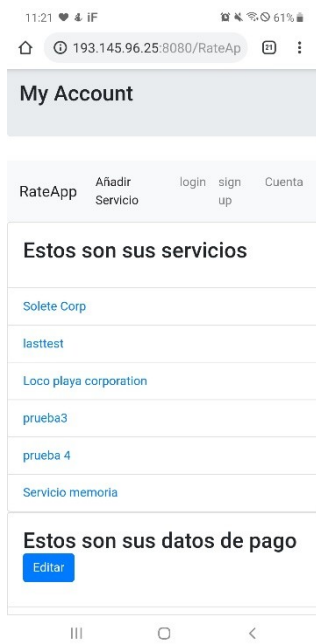


Ilustración 6: Pagina de cuenta y de servicio

Capítulo 6

Conclusiones y líneas futuras

Tras acabar de realizar el TFG y ver este tipo de herramientas cada vez implantadas en más y más sitios y ya no solo en ámbito de atención y valoración de productos, sino incluso en algunas empresas unos empleados valoran las “habilidades blandas” de otros mediante estos sistemas. Se considera que es un proyecto de alta importancia que puede tener grandes implicaciones económicas, no solamente en el sector servicios como se proyectaba inicialmente sino también, a nivel empresarial o incluso a nivel organizacional y económico.

Aún así se considera que el trabajo realizado en el TFG es un prototipo inicial del núcleo de un modelo de negocio extensivo. Tiene las funcionalidades justas para ser lanzado y explotado en una etapa temprana del ciclo de vida del producto, pero carece de muchas posibilidades que podrían aumentar el interés que generaría el proyecto.

Estas funcionalidades junto con una limpieza del apartado gráfico, especialmente en lo que a la web se refiere dan nuestras líneas de actuación futuras:

Sería interesante aumentar el número de estadísticas que se muestran en la página de cada servicio y buscar estadísticas más interesantes para los propietarios de cada servicio. Del mismo modo sería interesante separar la información de un servicio que debe ser privada de la que debe ser pública y usar la pública para llevar un repositorio de servicios que los agrupe por categorías, tenga distintos *rankings*, como por ejemplo satisfacción, donación media, etc..., tenga un sistema de valoraciones *online* e incluso perfiles públicos en los que la gente pueda comentar o subir fotos.

También aumentaría la tracción que provocaría la aplicación si se aumentasen las posibilidades de customización, como por ejemplo añadiendo preguntas, eligiendo tus propios iconos de satisfacción personalizados o modificando los precios básicos de donaciones, entre otras posibilidades.

Es importante, también, trabajar más de lleno en el apartado de seguridad, encriptar toda la información sensible, evitar que se muestre en ningún lado, volver más robusto el protocolo de comunicación con el servidor para evitar inyecciones de instrucciones y muchas otras cosas que no son tan difíciles de modificar o programar, pero requieren su tiempo para realizar una correcta planificación.

Y por último mejorar el apartado gráfico, aunque se mantuviese el estilo limpio y minimalista que tiene actualmente, la página web no está maquetada correctamente y en la aplicación muchos de los iconos parecen placeholders en lugar de iconos definitivos, y se podría hacer un mejor trabajo con sombras, luces, tamaños, posicionamiento, colores, que harían la aplicación más llamativa y por lo tanto más dada a ser usada. Todo esto son cambios que con el tiempo deberían hacerse para mejorar la apariencia de la aplicación.

Así que, aunque en general se puede ver la importancia del proyecto, muestra solamente un estado larvario de lo que, con la dirección adecuada, un buen marketing y trabajo podría llegar a volverse una herramienta imprescindible en el panorama comercial y de las RRPP y los RRHH.

Capítulo 7

Summary and Conclusions

7.1 Summary

This Project is made of three different parts each one with its own functions: A server made in Java which is in charge of communicating with the database and sending and receiving data from the other two parts, an Android app which is in charge of scanning the QR codes, and allowing the users to review and donate to the services, and finally a Tomcat server which hosts the website in which service providers can register their services to start offering them as reviewable in the app.

The Project has been entirely made in Java or Java related technologies, and the majority of tools chosen has been selected because of that. Therefore, the database used is PostgreSQL, IDEs are eclipse and Android Studio, build automatization tools are Gradle and Maven, etc...

As stated in the abstract the objective of this “TFG” is that of building an application that allow users to review and donate money to any kind of service they are using via the scanning of a QR code that identifies the service. This is not a revolutionary idea as review services have been around since the beginning of times and have grown in importance with the internet as it has made them easier to use and access. Examples of this would be TripAdvisor, comment sections on online shops or the poll after shopping at Corte Ingles.

All this service offers the same, and they solve the same problem, feedback and communication from customer to service provider. But our solution takes that old paradigm and make it better by making it more accessible, cheaper, easier to use and comfortable as it all is done in the same app for every kind of service, and the way is used is simple and pleasing.

The planning is lax as for the methodology we have used a prototypes model. Which means once a functionality is added, it is tested, and once it's fully working it's then implemented in a prototype which gets reviewed in order to search for new requisites or requisites that are missing. This model has been chosen as it's the easiest way of doing things in a solo Project and makes advances more susceptible to be seen and felt.

7.2 Conclusions

We believe that this Project is of big importance, and that it can be implemented in many places to increment the way those services are connected with their audiences and the possibilities those places have to get better in what they're doing.

Also, as we have observed an increment in the use of this kind of services over time, we believe it can have a big economic impact in many companies making the Project something interesting for the global market.

But we understand this is just a core functionality and not a finished product and so we propose that the Project keeps being built mainly increasing in these 4 aspects:

Better and more relevant statistics: the statistics being shown now are fairly simple and they show close to no relevant information. Revamping the statistic system, showing more information, and even creating

public profile pages for services to allow users to see some of the statistics, review online, comment, posts pictures, etc... could be a big upgrade to the current app.

More customization: custom satisfaction icons, the possibility to add questions to your poll, customizing the by default quantities of money for donations, choosing the colours, fonts, etc... being showed during your poll, and many other customizing options would make the experience unique in each service and will make for a more interesting user experience.

More security: this speaks for itself but encryption, anonymization, privatization of information, more robust protocols, etc... Will make the app more competent and attractive in the market.

Better UI: the actual UI seems rushed polishing it, making better icons, better use of colours, lights, and shadows, resizing and remapping everything would be eye candy which will make the Project much more susceptible of being likeable by non-tech audiences.

Capítulo 8

Presupuesto

8.1 Sección Uno

Para el cálculo del presupuesto del proyecto se va a suponer que la empresa emplea la misma metodología que se ha utilizado en el desarrollo del TFG, es decir se trabaja con un modelo de prototipos, sin un analista o diseñador designado sino los propios desarrolladores son los encargados de la extracción de requisitos, de los diseños, de la implementación, de las pruebas y de la verificación y validación.

Sin embargo, a vistas de abaratar costes y de hacer el proyecto más realista se emplearán 40 horas semanales en lugar de las que se empleaban en el TFG. Se añadirá a estos costes todos los costes físicos del proyecto como oficinas, equipos en los que se desarrolle, servidor, licencias para Java, etc...

Se asume además que este es el único proyecto en el que está trabajando la compañía actualmente para facilitar los cálculos.

Recurso	Coste unitario	Número unidades	Coste Total
Analista Junior	30 €/hr	40 hr	1200€
Desarrollador Junior	25 €/hr	233 hr	5825€
Renting: Ordenador desarrollo	99 €/d	34 d	3366€
Renting: Local	24€/ d	47 d	1128 €
Facturas varias	3€/d	47 d	141 €
Licencia Programador Java:	2,5 €/m	2 m	5€
Licencia Java: Servidor	25€ /m	12 m	300 €
Servidor Virtual	10,5 €/m	12 m	126€
Total:			12091€

Tabla 1: Resumen de tipos

Por lo tanto, el coste final del proyecto sería de unos 12091€ aunque este podría aumentarse si se contratase a más programadores, si surgieran problemas en las entregas, se alargase el tiempo, se usase un servidor dedicado en lugar de uno virtual y muchas otras variables.