

Ticky: Gestión de garantías a través del ticket de compra

Miguel del Amor Herrera

Máster universitario de Desarrollo de aplicaciones para dispositivos móviles
M0.659 - Trabajo final de máster DADM aula 1

Eduard Martín Lineros
Carles Garrigues Olivella

10/2020



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](https://creativecommons.org/licenses/by/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Ticky: Gestión de garantías a través del ticket de compra</i>
Nombre del autor:	<i>Miguel del Amor Herrera</i>
Nombre del consultor/a:	<i>Eduard Martín Lineros</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	10/2020
Titulación:	<i>Máster universitario de Desarrollo de aplicaciones para dispositivos móviles</i>
Área del Trabajo Final:	<i>M0.659 - Trabajo final de máster DADM aula 1</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Ticket, garantía, compras</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Con el fin de ofrecer una solución al usuario para la gestión de las garantías sobre los productos que compra con ticket de caja, se realiza una aplicación de móvil para la plataforma Android; la cual enriquecerá la información del ticket fotografiado con la cámara del dispositivo, con capacidades de indexado para su posterior búsqueda y posibilidad de ser notificado por eventos que le afecten.</p>	
<p>Abstract (in English, 250 words or less):</p>	
<p>To offer a solution for product guarantees dependant on ticket notes, we create an Android application; that will enrich the ticket captured through the devices webcam, with index capabilities for search and being able to get notified about events that affect that ticket.</p>	

Índice

1. Introducción.....	2
1.1 Contexto y justificación del Trabajo.....	2
1.1.1 Competidores, soluciones actuales en el mercado.....	2
1.2 Objetivos del Trabajo.....	3
1.2.2 Objetivos no funcionales.....	4
1.3 Enfoque y método seguido.....	4
1.4 Planificación del Trabajo.....	5
1.5 Breve sumario de productos obtenidos.....	6
1.6 Breve descripción de los otros capítulos de la memoria.....	7
2. Usuarios y contexto de uso.....	8
2.1 Tipos de usuario.....	8
2.2 User persona.....	8
2.3 Requisitos de funcionalidad.....	9
3. Diseño Conceptual.....	10
3.1 Escenarios de uso.....	10
3.1.1 Antonio.....	10
3.1.2 Elena María.....	10
3.1.3 Juan.....	11
3.2 Flujos de interacción.....	11
3.3 Diagramas de navegación.....	13
4. Prototipado.....	14
4.1 Prototipo de baja calidad.....	15
4.2 Prototipo.....	16
4.2.1 Bienvenida.....	16
4.2.2 Login.....	17
4.2.3 Listado de tickets.....	18
4.2.4 Añadir ticket.....	18
4.2.5 Filtrado de tickets.....	20
5. Casos de uso.....	21
5.1 Onboarding.....	21
5.2 Login.....	22
5.3 Uso anonimo.....	23
5.4 Primera demo.....	23
5.5 Demo.....	24
5.6 Añadir ticket.....	24
5.7 Editar ticket.....	25
5.8 Eliminar ticket.....	26
5.9 Exportar tickets.....	26
5.10 Filtrar tickets.....	27
5.11 Eliminar publicidad.....	28
6. Arquitectura.....	29
6.1 Arquitectura del sistema.....	29
6.2 Modelo de datos.....	31
6.2.1 Ticket.....	31
6.2.2 Usuario.....	32

6.2.2 Categoría	32
7. Implementación	33
8. Pruebas	35
8.1 Plan de pruebas	35
8.2 Resultados de los tests	37
9. Aspecto final de la aplicación	39
10. Conclusiones	42
10.1 Lecciones aprendidas.....	42
10.2 Logro de los objetivos.....	42
10.3 Seguimiento de la planificación y metodología.....	42
10.4 Líneas de trabajo futuro.....	43
11. Glosario	44
12. Bibliografía	46
13. Anexos	47
13.1 Gantt planificación	47
13.2 Instrucciones para la compilación.	48
13.3 Manual de usuario	48
13.3.1 Requisitos de la aplicación.....	48
13.3.2 Login	48
13.3.3 Creación de un ticket	49
13.3.4 Modificación y borrado de un ticket.....	50
13.3.6 Eliminar publicidad y aumentar la cantidad de tickets posibles.....	52
13.3.7 Filtrado de tickets en el listado.....	53
13.3.8 Notificaciones.....	54
13.3.9 Cloud storage.....	55

Lista de figuras

Ilustración 1. Gantt planificación, miniatura	5
Ilustración 2. Flujo de interacción	12
Ilustración 3. Navegación inicial	14
Ilustración 4. Navegación principal	14
Ilustración 5. Prototipo de baja calidad	15
Ilustración 6. Prototipo, pantalla de bienvenida	16
Ilustración 7. Prototipo, pantalla de login	17
Ilustración 8. Prototipo, listado de tickets	18
Ilustración 9. Prototipo, recortar ticket	19
Ilustración 10. Prototipo, añadir ticket	19
Ilustración 11. Prototipo, seleccionar alarma	20
Ilustración 12. Prototipo, filtrar lista	21
Ilustración 13. Arquitectura del sistema	29
Ilustración 14. Arquitectura de la aplicación	30
Ilustración 15. Ticket, ejemplo de modelo	32
Ilustración 16. Usuario, ejemplo de modelo	32
Ilustración 17. Categoría, ejemplo de modelo	33
Ilustración 18. Paquetes de la aplicación	33
Ilustración 19. Estructura del proyecto	34
Ilustración 20. Gantt planificación, tamaño completo	47

1. Introducción

1.1 Contexto y justificación del Trabajo

En el día a día acumulamos montones de justificantes de compra que son imprescindibles para poder ejercer el derecho de garantía o devolución del mismo.

Estos justificantes a menudo acaban perdidos, en un cajón o deteriorados por el tiempo a consecuencia del propio material con el que están fabricados (por ejemplo, los tickets con impresión térmica).

Los compradores más cuidadosos realizan una fotocopia y guardan el justificante junto a su fotocopia en el propio embalaje del artículo, o en un cajón.

A la hora de necesitar ejercer el derecho de garantía o devolución, es muy común que el comprador no recuerde cuándo cumplía el plazo o donde tenía el ticket; incluso es habitual que pasado cierto tiempo no sepa dónde compró el artículo.

El trabajo propone dar solución a estos problemas con una aplicación para móvil que sea capaz de guardar una foto del ticket y ofrezca funcionalidades de indexado, búsqueda, recordatorio de expiración de garantías e impresión de los tickets; además sería ideal si esta información queda registrada en un servidor para dar una garantía extra al comprador.

1.1.1 Competidores, soluciones actuales en el mercado

Se ha realizado una investigación sobre las posibles opciones que habría en el mercado para cubrir esta necesidad, no encontrando ninguna que ofrezca funcionalidad específica para manejar la gestión de garantías de los artículos. Las aplicaciones analizadas enfocaban su uso a la gestión de los gastos y facturación ligadas a los tickets como justificantes de pago.

Podemos citar los casos de las dos que me han parecido más conocidas:

- *Okticket*
 - Ofrece llevar la gestión de gastos acumulados en tickets en tu móvil para prescindir del papel y el riesgo que conlleva.
 - Se enfoca en trabajadores autónomos y de pyme que necesitan recoger estos gastos.
 - Es gratuita, hasta un cupo de usuarios.
 - Permite capturar tickets y facturas.
 - Permite generar informes de gasto.
 - Tiene una web para administrar los datos además de en el móvil.

- *Tiketi*

- Ofrece llevar la gestión de gatos acumulados en tickets en tu móvil, sin importar la categoría.
- Se enfoca, aunque no en exclusiva, en usuarios particulares.
- Realiza una clasificación automática de los gatos y genera informes para tener control sobre en qué estamos gastando el dinero.
- Permite la creación de alertas para gastos, como fijar límites.
- Lleva un sistema de promociones y comendaciones de compra basadas en los datos que registra de los tickets que añades.

1.2 Objetivos del Trabajo

El objetivo del trabajo es la realización de una aplicación para móvil, para la plataforma Android que posibilite una gestión sencilla de las garantías asociadas a los mismos, sirviendo además esta como archivo permanente de estos tickets de compra.

1.2.1 Objetivos funcionales

- La aplicación permitirá su funcionamiento de forma anónima, sin registro de usuario o utilizando las credenciales para guardar la información en la nube.
- La aplicación mostrará publicidad para su monetización. Dicha publicidad podrá ser desactivada previo pago.
- Permitirá capturar un ticket con la cámara del dispositivo o introducirlo como una imagen.
- El ticket será post procesado para obtener los bordes del mismo y hacer recorte, además de limpiar la imagen de brillos y mejorarla en medida de lo posible.
- Se recogerá la geolocalización del momento de la creación. Además, se permitirá modificar esta posición a posteriori.
- Se permitirá almacenar una cantidad limitada de tickets, pudiendo aumentar esta cantidad previo pago de un modo premium.
- Se configurará la fecha de caducidad/alerta para la expiración de la garantía, la cual se basará en una fecha prefijada por defecto y permitirá especificar otra distinta en el mismo momento.
- Se dispondrá de un sistema de categorías sobre el ticket para proceder a su filtrado, dichas categorías serán personalizables incluirán un icono y podrán ser modificadas sobre un ticket concreto.

- Los tickets se ordenarán en un listado que permitirá su filtrado por las propiedades del ticket.
- Un ticket o una lista de tickets serán exportables a PDF para su posterior impresión, envío o compartición con otras aplicaciones del sistema.
- La aplicación tendrá una funcionalidad para recomendar la aplicación a través de un mensaje en otras aplicaciones del sistema.
- La aplicación será multi idioma y seleccionará este del propio idioma del dispositivo.
- La aplicación mostrará un tutorial de primer uso.

1.2.2 Objetivos no funcionales

- La aplicación permitirá su funcionamiento con y sin internet, necesitando internet para la sincronización con la nube.
- La solución está orientada, aunque no en exclusiva, a un usuario no profesional.
- Será una aplicación compatible con dispositivos móviles y tabletas Android.
- El desarrollo será *open source*.

1.3 Enfoque y método seguido

El enfoque elegido es la elaboración de una aplicación nueva para móvil para la plataforma Android elaborada de forma nativa, utilizando como servidor para guardar los tickets Firebase.

Se han valorado otras opciones como un desarrollo in-house para el servidor, o el uso de la plataforma Azure; así como la elaboración de la aplicación en un framework multi plataforma como React native.

Se ha optado por aplicación nativa, porque el framework multi plataforma podría incurrir en problemas en el post procesado, ya que se utilizarán librerías que dependerán del hardware específico.

Sobre el servidor, creo que Firebase es la opción correcta para el servidor, ya que el soporte en plataformas móviles es nativo y además ofrece características de soporte para el software como la detección de errores en tiempo de ejecución, que puede ser de utilidad. Además, esta plataforma ha sido vista y utilizada en el transcurso del máster, por lo que se tienen los conocimientos necesarios para hacer uso de ella.

1.4 Planificación del Trabajo

El desarrollo del proyecto se divide en 4 bloques principales de los cuales se producen 4 entregas o resultados.

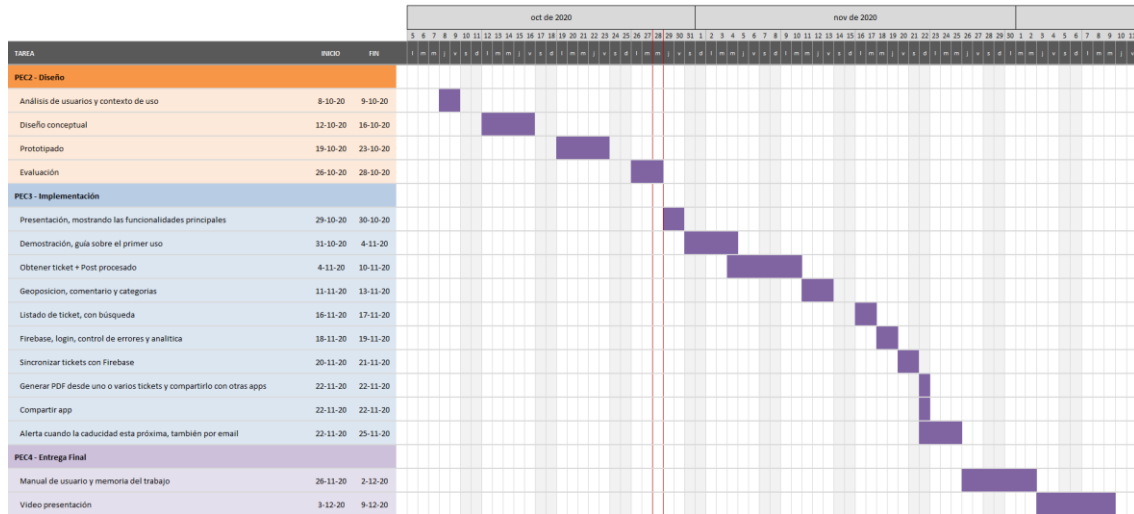


Ilustración 1. Gantt planificación, miniatura

Diseño

Recursos

- 10 horas a la semana.
- Ordenador con Android Studio.
- Posibles usuarios para toma de requisitos.

Tareas

- Análisis de usuarios y contexto de uso (4 horas).
- Diseño conceptual (10 horas)
- Prototipado (10 horas)
- Evaluación (6 horas)

Resultados

- Diseño completo de la aplicación tanto en navegabilidad como en arquitectura.

Total, horas: 30 horas

Horas por día: 2 horas por día laborable

Total, días: 15 días laborables de desarrollo.

Implementación

Recursos

- 10 horas a la semana.
- Cuenta en Firebase.
- Cuenta en Google Play.

- Ordenador con Android Studio.
- Móvil Android con cámara.

Tareas

- Presentación, mostrando las funcionalidades principales (6 horas)
- Firebase, login, reporte de errores y analítica. (6 horas)
- Demostración, guía sobre el primer uso (12 horas).
- Obtener ticket + Post procesado del ticket. (14 horas).
- Sistema de publicidad, con capacidad de eliminarla (12 horas).
- Geoposicion, comentario y categorías (8 horas).
- Listado de tickets, con búsqueda (6 horas).
- Sincronizar tickets con Firebase (6 horas).
- Generar PDF desde uno o varios tickets y compartirlo con otras apps. (4 horas)
- Alerta cuando la caducidad de un ticket esta próxima, también por email (12 horas).
- Compartir app (2 horas).

Resultados

- Aplicación móvil para la plataforma Android

Total, horas: 88 horas

Horas por día: 3 horas por día laborable, 1 horas en día festivo.

Total, días: 29 días, 27 laborables y 3 festivos.

Entrega Final

Recursos

- 10 horas a la semana.
- Ordenador con programa de grabación y maquetado de video.

Tareas

- Manual de usuario y memoria del trabajo (12 horas).
- Video presentación (12 horas)

Resultados

- Video presentación del trabajo.
- Memoria del trabajo.
- Manual de usuario.
- Ejecutable de la aplicación.
- Código fuente.

Total, horas: 24 horas

Horas por día: 2 horas por día laborable, 1 hora en día festivo.

Total, días: 14 días, 12 laborables y 2 festivos.

1.5 Breve sumario de productos obtenidos

El entregable consiste en una aplicación de móvil para la plataforma Android con la funcionalidad descrita implementada, así como la documentación asociada a su realización y una video presentación describiendo el proyecto.

Dicha aplicación conectará con un servidor de Firebase donde hará la sincronización de la base de datos de los usuarios, además en el propio Firebase será donde se lleve la gestión de los usuarios.

Se hará uso de una cuenta de Google Admob que proveerá de los anuncios que aparecerán en la aplicación para su monetización. Y de una cuenta de Google Play para llevar la gestión de los pagos inapp para eliminar anuncios y activar el modo premium sin restricciones de cantidad de tickets.

1.6 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos de la memoria se definirá el proyecto a nivel de usabilidad y arquitectura, se implementará el mismo y se producirá la documentación de uso, así como las conclusiones que se han extraído durante la realización del mismo. Además, se desarrollará un video explicativo sobre el trabajo realizado.

2. Usuarios y contexto de uso

2.1 Tipos de usuario

El sistema está diseñado para el público general, poniendo el foco, aunque no en exclusiva, sobre el usuario no profesional.

Puesto que es un sistema que sirve de apoyo para algo que deriva de la compra de productos de consumo podemos decir que está enfocado a consumidores.

Y estando las funcionalidades principales enfocadas en la gestión de garantías de productos de consumo, podemos acotar que el usuario será un consumidor de un tipo de producto que pueda ser objeto de una gestión de garantías. Dejando a un lado bienes como alimentos u otros productos sobre los que no puede ejecutarse una garantía o sobre los que no es interesante su gestión.

2.2 User persona

Nombre: Antonio **Edad:** 23 años **Ocupación:** Estudiante

Antonio es estudiante en la Universidad de Murcia, por las tardes trabaja de repartidor para conseguir algunos ingresos extra.

Fruto de esos ingresos extra, compró hace una cantidad de tiempo indeterminado unas zapatillas de deporte, las cuales han tenido un desgaste a su juicio inusual quedando inservibles.

Cree casi seguro que las adquirió en el Corte Inglés, pero no encuentra el ticket de compra; ha ido varias veces a preguntar y le dicen que es imprescindible el ticket o haber comprado con la tarjeta del Corte Inglés. Puesto que el suele operar en efectivo no tiene modo de reclamar la garantía de sus zapatillas.

Es un usuario muy activo de redes sociales y usa su teléfono móvil en el día a día para multitud de tareas; piensa que ojalá hubiese fotografiado el ticket y ahora habría evitado este problema.

Nombre: Elena María **Edad:** 42 años **Ocupación:** Enfermera

Elena María es enfermera en el Hospital Morales Meseguer, vive aquí, pero es de Sevilla.

Elena, es una gran consumidora de música, en su último viaje a Sevilla su hermana le compró un reproductor de mp3 para poder llevarlo en el trabajo y que se le haga un poco más ameno en las guardias nocturnas. Hace poco al reproductor le ha empezado a fallar la batería, llegando a durarle sólo unos minutos. Ha preguntado a su hermana, si se puede devolver pues

recuerda lo compró en un Media Markt y en Murcia también hay. Lamentablemente su hermana no conserva el ticket.

Elena utiliza el teléfono móvil y la Tablet a diario para comunicarse con sus amigos y familiares y piensa que quizá habría sido tan sencillo como fotografiar el ticket al comprar el producto y haberlo enviado en caso de necesitarlo.

Nombre: Juan **Edad:** 38 años **Ocupación:** Programador

Juan trabaja de programador en una empresa de informática de Murcia. Es muy aficionado a la tecnología y los videojuegos, aunque comparte esas aficiones con el deporte.

Hace tiempo compro un Wall ball de 6 kg, con el que estaba muy contento hasta la fecha, pero se le ha descosido la costura central. Recuerda que no fue barato, pero hace ya un tiempo de su compra y no recuerda ni donde ni cuando lo compró; pues estuvo mirando precios y modelos en varios sitios guiado por foros de aficionados al cros-fit.

No es la primera vez que le ocurre algo así y mientras decide buscar una solución la garantía ya le ha expirado.

Siendo como es bastante techie, cae en la cuenta de que si tuviese en alguna parte imágenes de los tickets con algún indexado para buscarlos podría haberse evitado este problema. De forma similar a lo que hace con su correo cuando no recuerda exactamente el asunto o el emisor.

2.3 Requisitos de funcionalidad

Tras analizar los usuarios potenciales de la aplicación y el contexto en el que se emplaza la solución se identifican los siguientes requisitos:

- Debe permitir capturar un ticket con la cámara y guardarlo en el sistema.
- Debe tener sistemas de indexado para la búsqueda sencilla del ticket.
- Debe permitir enviar un ticket a otra persona, vía los canales de comunicación que ya utiliza el usuario en su dispositivo.
- Debe permitir imprimir el ticket para poder entregarlo físicamente.
- Listado de tickets con búsqueda actual.
- Post procesado del ticket para guardarlo de una forma que sea perfectamente legible.
- Recibir notificaciones cuando va a expirar la garantía de un producto de un ticket.
- La aplicación debe estar disponible para dispositivos móviles Android.

3. Diseño Conceptual

3.1 Escenarios de uso

Pasamos a mostrar a los usuarios anteriores interactuando con la solución propuesta como solución a los problemas que han tenido.

3.1.1 Antonio

Antonio conoció Ticky por recomendación de un amigo, el cual le envió un enlace para que se la descargase en su móvil y la empezase a usar.

Hoy ha ido al centro comercial, esta vez para comprar un ratón nuevo para el ordenador, ha recordado la aplicación que le han recomendado y decide probarla.

Al principio desconfiaba de que realmente fuese a guardar los tickets de una forma legible, pues alguna vez había hecho fotografías de tickets en el pasado y por el papel satinado y el carácter endeble del soporte no le había dado buen resultado.

Gracias al post procesamiento del ticket queda una imagen de forma aceptable, la cual categoriza bajo “informática” y deja la fecha de garantía por defecto que es de 2 años pues es la más adecuada para él.

Después de unos meses, el ratón dejó de funcionar correctamente, se apreciaba que la luz del emisor estaba apagada de forma permanente. Antonio pensó que ya había perdido otra vez tiempo y dinero, no obstante, recordó la aplicación que le recomendaron hacía un tiempo.

La abrió y ahí seguía el ticket de compra de su ratón, respiró aliviado y comprobó que la garantía no había caducado por lo que sólo tenía que ir al centro comercial de nuevo y reclamar su derecho a que le diesen una solución.

3.1.2 Elena María

Elena ha comprado un regalo para su sobrina Mara, ya que va a hacer la comunión la próxima semana; el regalo en cuestión es un ordenador portátil, que han pagado entre varios hermanos.

Lo ha comprado en un centro Worten y lo han enviado desde Murcia, ya que el ordenador salía más barato sin sistema operativo y su marido le dijo que podía configurárselo.

Al comprarlo ha hecho una captura con Ticky y ha guardado el ticket de compra de forma segura en el sistema, dejando el periodo de garantía de 2 años por defecto. Y categorizándolo bajo “regalos” e “informática”.

Pasado un tiempo el portátil de Mara parece calentarse de más, tras estar un tiempo encendido este avisa de que va a apagarse. El hermano de Elena le comenta el problema que tiene el portátil.

Elena le comenta que se compró en un centro Worten y que al haber en Sevilla le deben dar soporte sin problema; le avisa de que le va a enviar el Ticket por Whatsapp.

Abre la aplicación Ticky busca el ticket dentro de la categoría “regalos” en la fecha aproximada de la comunión, lo encuentra y lo comparte directamente en Whatsapp con la conversación de su hermano.

3.1.3 Juan

Tras su problema Juan ha estado buscando una solución, encontró Ticky en Google Play y la instaló para probarla; estuvo comprando varias cosas y añadiéndolas a la aplicación para probar su funcionalidad.

Pasado un tiempo, un ejercitador de mano para guitarra que había comprado en una tienda de su barrio, tuvo un fallo y se le salió uno de los muelles que lo sustentan; pensó en ir el jueves próximo que era cuando tenía libre a cambiar el aparato.

Se le pasó el tiempo y se olvidó un poco del aparato, pues lo tenía en un cajón y al estar inservible no lo tenía en cuenta. Un día, el teléfono le muestra una notificación avisándole de que la garantía de su ejercitador de mano está próxima a espirar, ¡le quedaban 15 días! Ahora sí, se apresuró a llevarlo a la tienda antes de que expirase el plazo.

3.2 Flujos de interacción

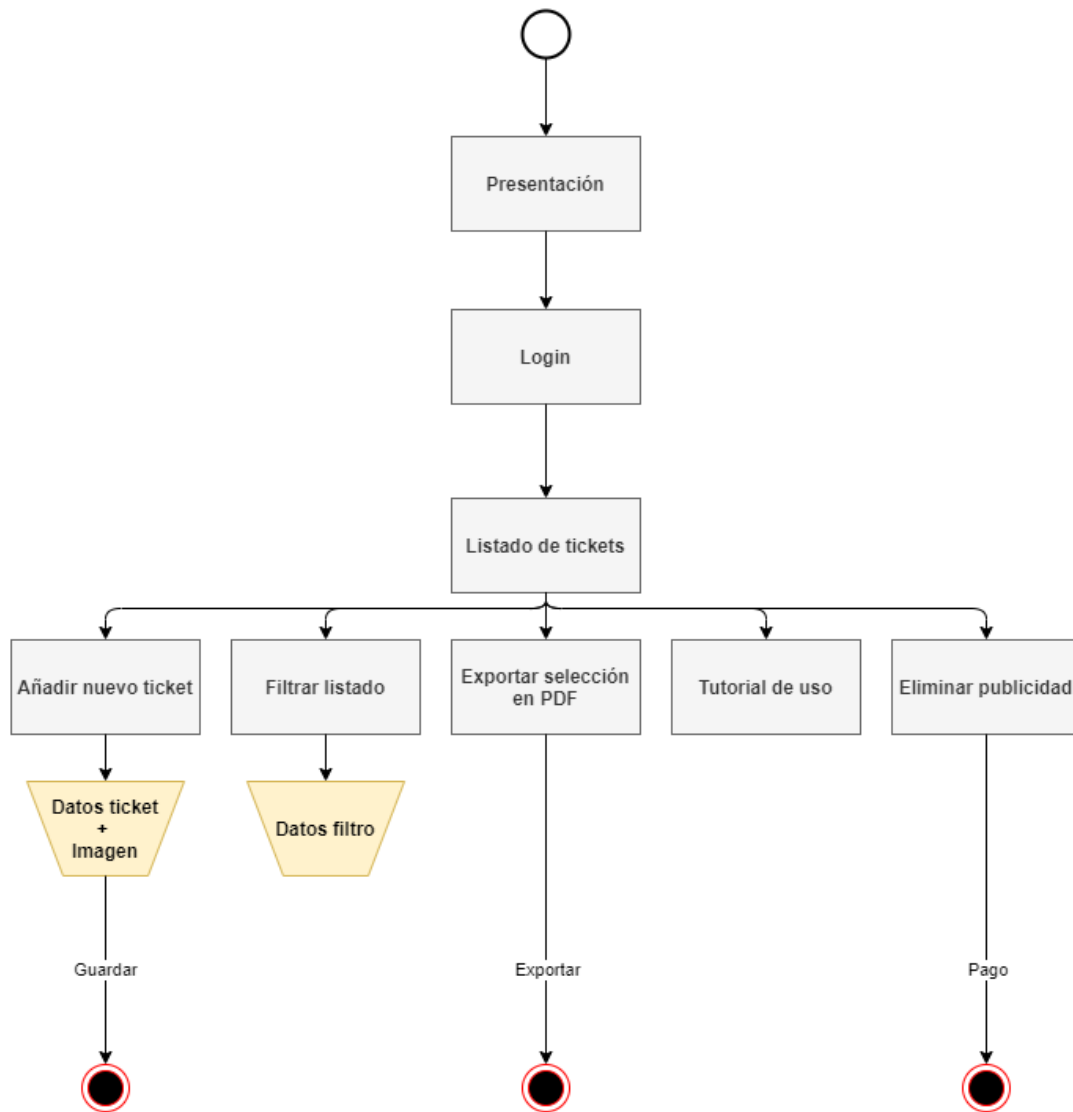


Ilustración 2. Flujo de interacción

En el anterior diagrama se muestra el flujo principal de interacción de la aplicación, el cual pasará por la presentación realizará el login el cual es opcional y en el listado de tickets tendrá las interacciones posibles de usuario.

Las interacciones posibles no guardan un orden necesario, a excepción del filtrado y exportación de tickets que lógicamente tienen como requisito el haber añadido algún ticket en el pasado (o haberlos obtenidos de la sincronización de datos).

3.3 Diagramas de navegación

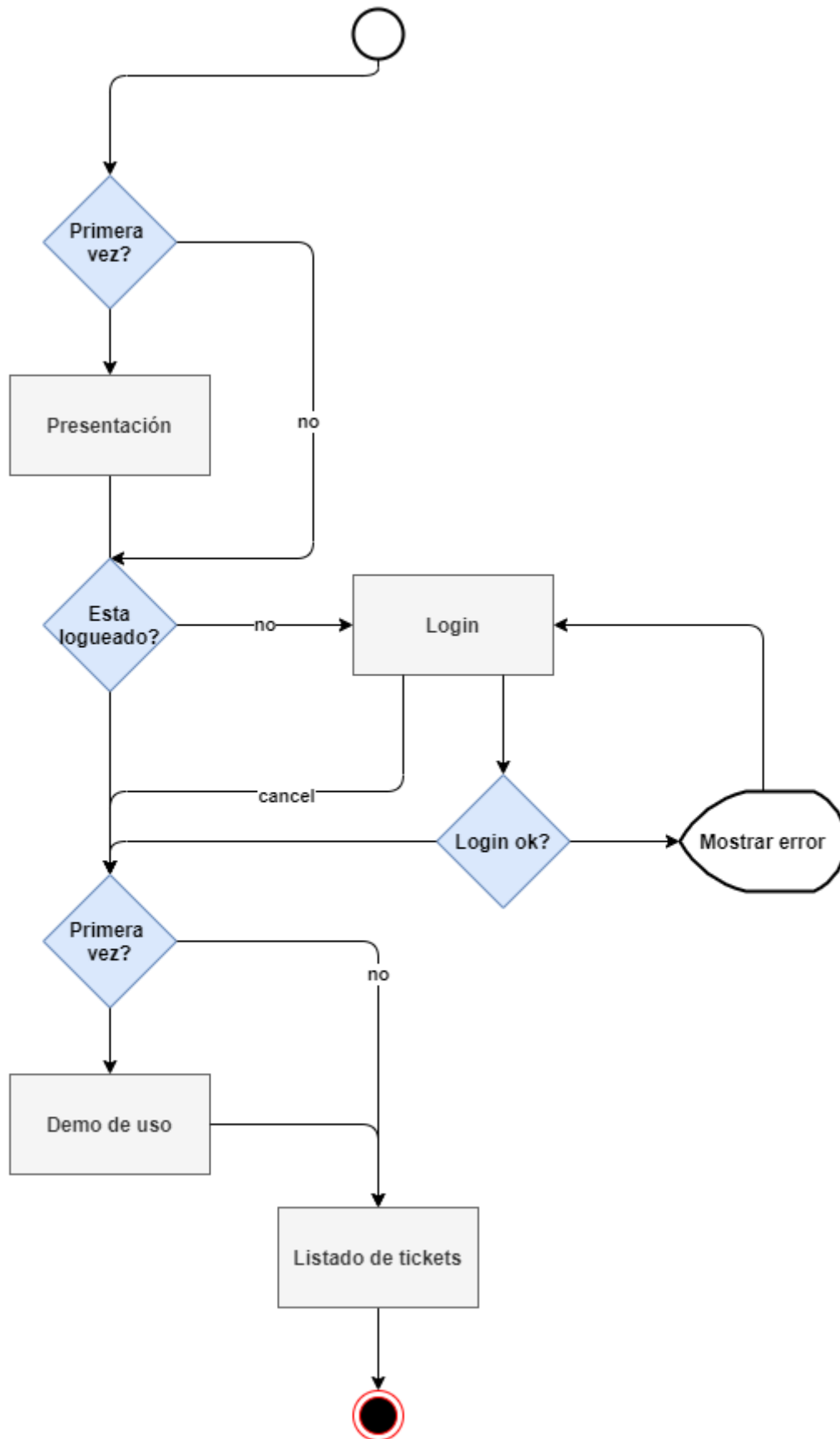


Ilustración 3. Navegación inicial

Aquí se define la navegación desde la apertura de la aplicación, cuando es la primera vez que se abre la aplicación se mostrará una presentación exponiendo las principales ventajas al usuario; así mismo la primera vez que lleguemos al listado de tickets, se mostrará una demo de interacción sobre el producto para guiar en el primer uso.

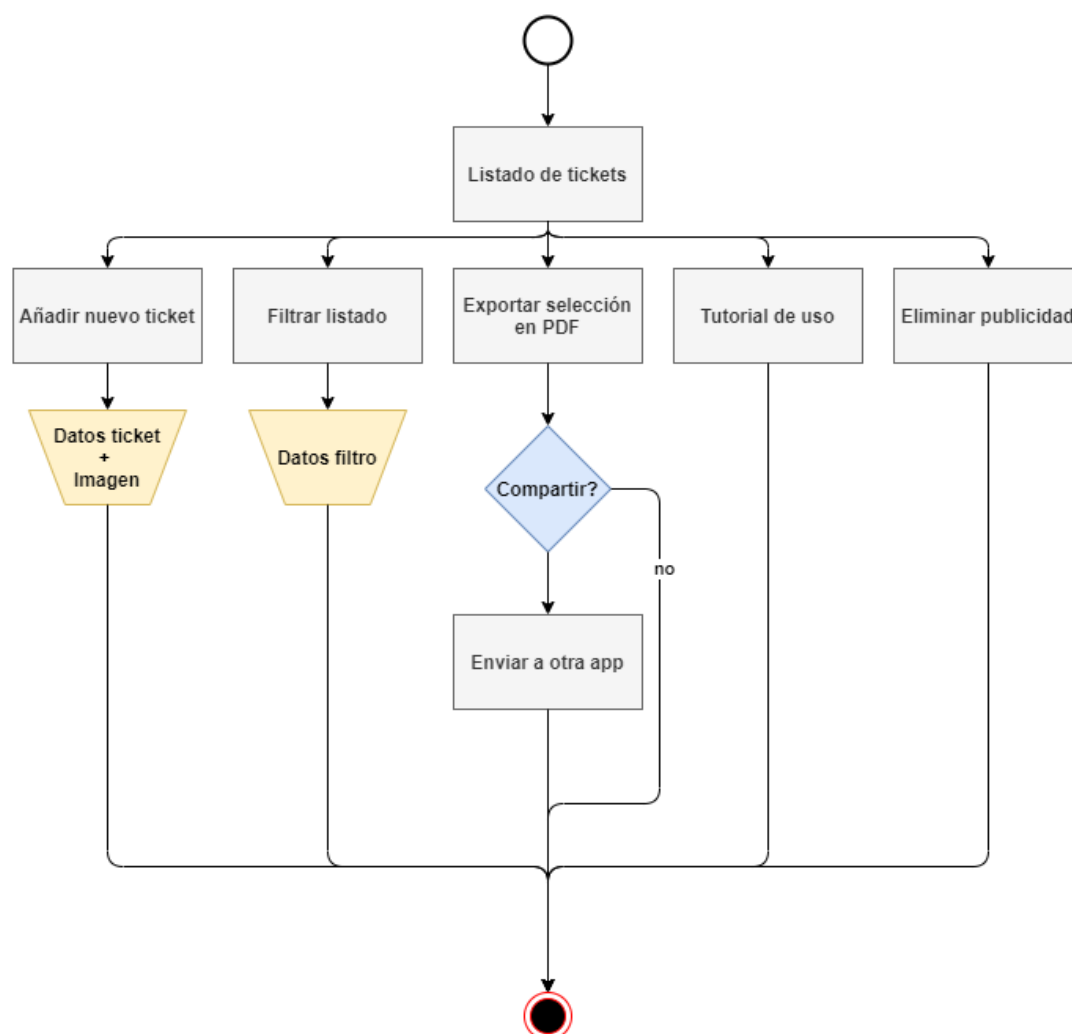


Ilustración 4. Navegación principal

En este otro diagrama se define la navegación por las diferentes pantallas y secciones de la aplicación.

4. Prototipado

Teniendo como base las definiciones de los diagramas anteriores, se realiza un prototipado para definir gráficamente como quedarán las interfaces de usuario de la aplicación.

La aplicación se centrará en el listado de ticket pues es el nexo de unión de todas las acciones y donde reside el valor de su utilidad.

En la primera ejecución mostraré una presentación a modo de onboarding para mostrar al usuario los principales beneficios del uso de la aplicación los cuales quizá no sean percibidos con un vistazo a la descripción de la misma. Puesto que este proceso es algo bloqueante y costoso en tiempo, se diseñará de forma que sea fácil pasarlo y que sólo aparezca la primera vez.

Al principio opté por incluir un botón para pasar completamente la presentación, pero creo que es más interesante reducir las partes de la presentación al mínimo añadiendo algunas ilustraciones que llamen la atención del usuario y permitiéndole pasar por todas estas pantallas arrastrando simplemente de izquierda a derecha.

En la pantalla principal de listado de ticket se mostrará un vistazo rápido de cómo añadir tu primer ticket, para guiar al usuario sobre el uso inicial práctico de la aplicación. Dicho vistazo rápido o demostración de uso enseñará la propia interfaz de la aplicación señalando al usuario dónde debe interactuar y para qué.

Las interacciones más comunes que se identifican que son añadir un ticket y filtrar el listado se han añadido como botones directamente accesible en la pantalla principal; dejando el resto de opciones que no son tan habituales dentro de un submenú.

Todos los campos de texto llevarán una sugerencia de contenido para hacer más intuitivo su uso, así como los emplazamientos para la imagen del ticket.

4.1 Prototipo de baja calidad

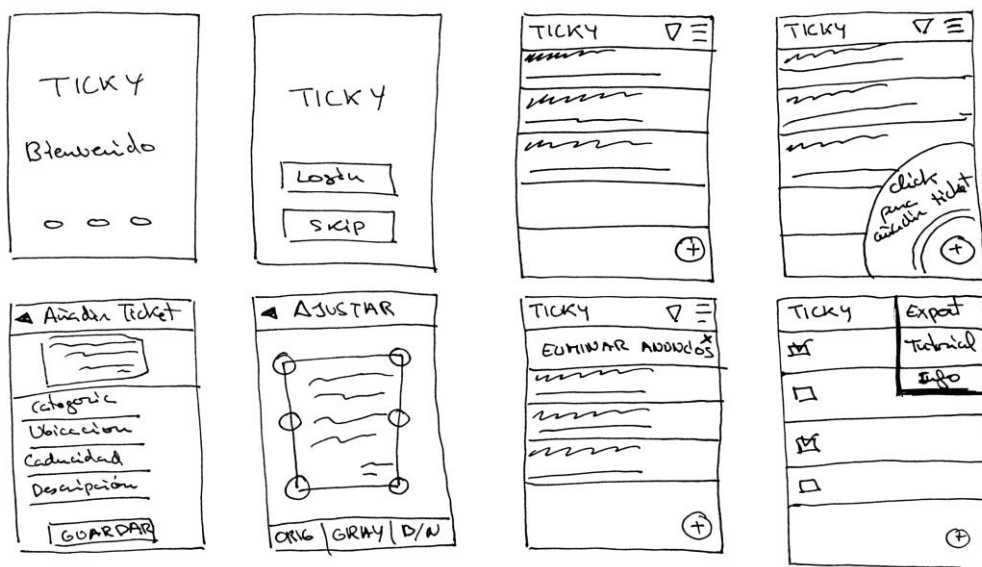


Ilustración 5. Prototipo de baja calidad

En la figura anterior se muestran las principales pantallas de la aplicación mostrando la funcionalidad, hechas a modo de boceto a mano.

4.2 Prototipo

4.2.1 Bienvenida.

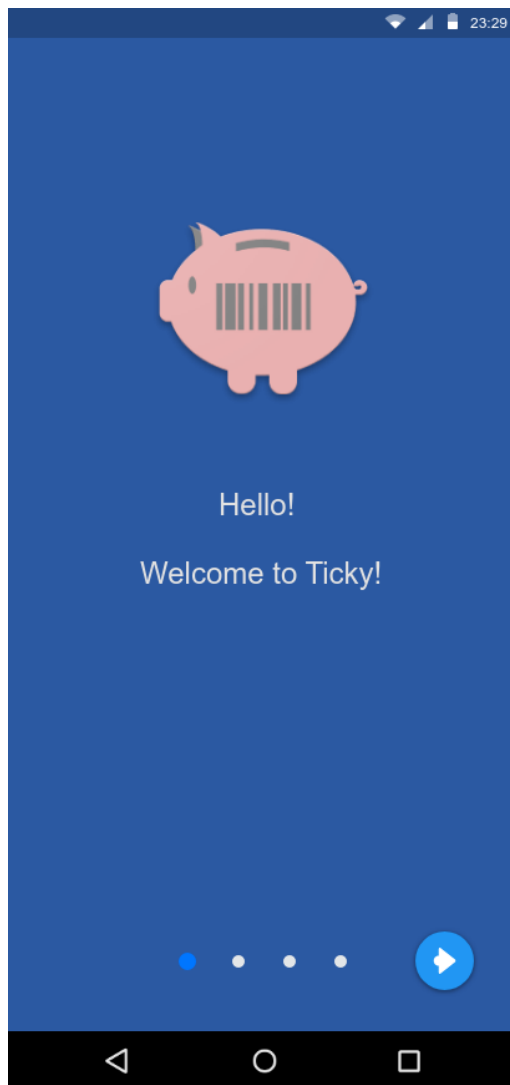


Ilustración 6. Prototipo, pantalla de bienvenida

En esta pantalla se dará un repaso por las funcionalidades más importantes a modo de onboarding para que el usuario sepa qué podrá realizar con la aplicación. Se plantea que sea poco texto e ilustraciones vistosas para llamar la atención del usuario y que preste atención a lo que se le cuenta.

No obstante, se podrá salir de esta pantalla rápidamente arrastrando de izquierda a derecha; se plantea añadir además un botón de continuar por si el hecho de tener que arrastrar no fuese lo suficiente obvio.

Esta pantalla sólo se presentará la primera vez que se utilice la aplicación.

4.2.2 Login

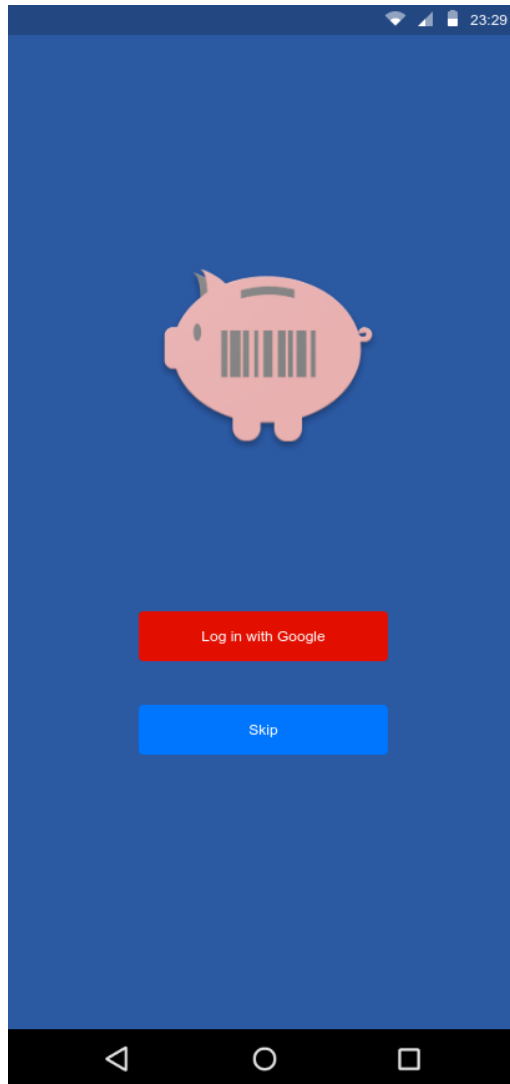


Ilustración 7. Prototipo, pantalla de login

Si el usuario no ha hecho login, al abrir la aplicación se le presentará esta pantalla para que haga login utilizando su cuenta de Gmail. En este paso el sistema realiza un registro del nuevo usuario en el caso de que no hubiese realizado este paso anteriormente. Además, se ofrece la opción de entrar sin hacer login para trabajar anónimamente con la aplicación.

El que el usuario haga login le permitirá sincronizar el contenido de su aplicación con la nube.

4.2.3 Listado de tickets.

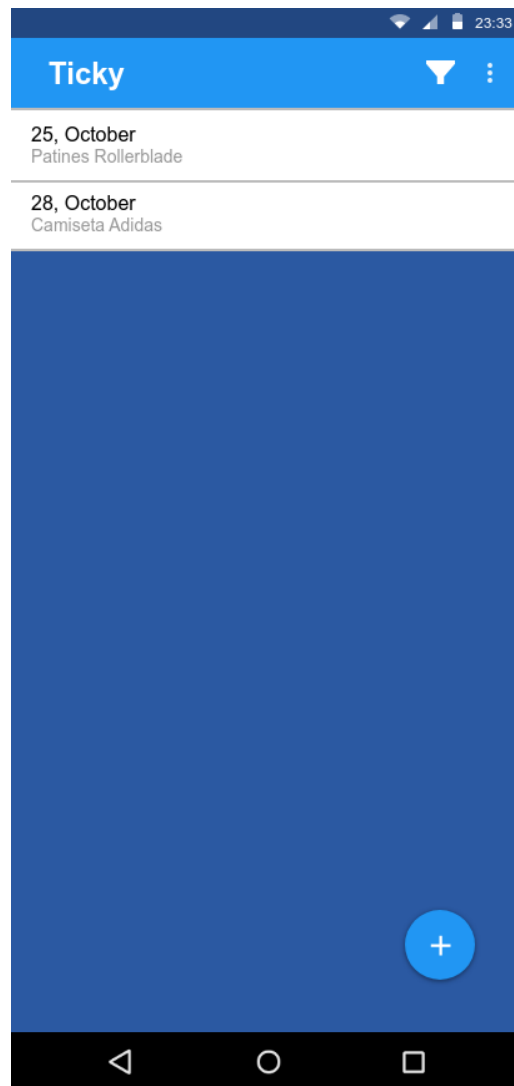


Ilustración 8. Prototipo, listado de tickets

Es la pantalla principal en la que se centra la aplicación, sirve de nexo central para realizar todas las acciones sobre los datos. La primera vez que se entre a esta pantalla se mostrará paso a paso con un efecto de spot-on las acciones que debe realizar para añadir su primer ticket, a modo de guía o demo.

En esta misma pantalla aparecerá la interacción para las compras inapp que permitirán eliminar los anuncios de AdMob que irán apareciendo a modo de banner abajo y aumentar el límite de tickets que puede el usuario guardar en el sistema de forma permanente.

4.2.4 Añadir ticket

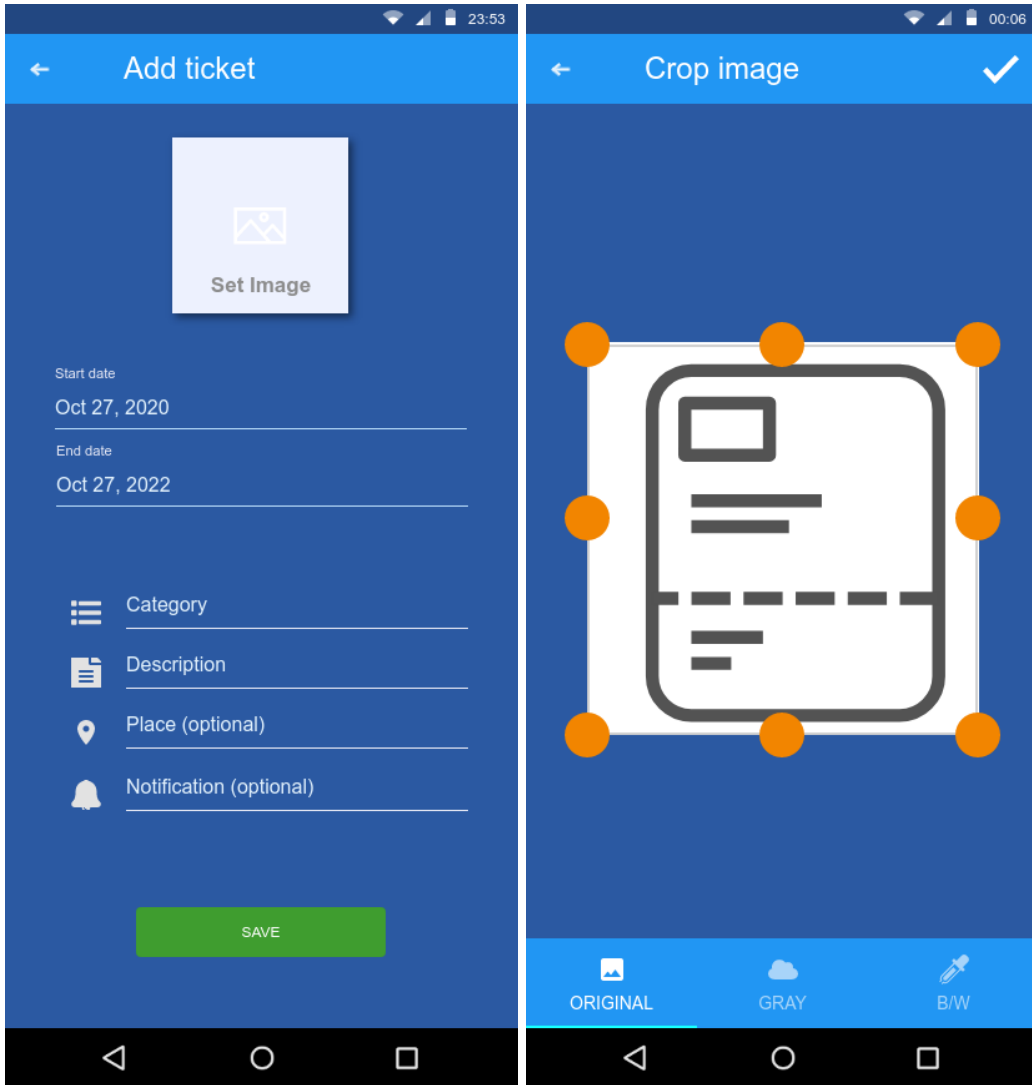


Ilustración 9. Prototipo, recortar ticket
Ilustración 10. Prototipo, añadir ticket

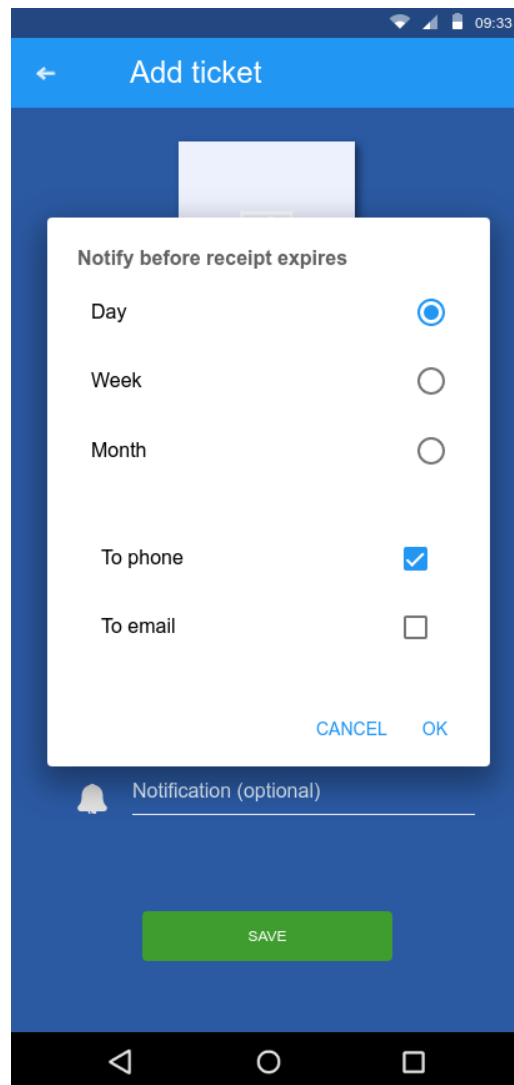


Ilustración 11. Prototipo, seleccionar alarma

En esta pantalla se nos pide a modo de formulario que añadamos los detalles de un nuevo ticket.

El primer paso, la captura de una imagen del ticket, nos llevará a otra pantalla en la que se nos permitirá recortar la imagen cambiando el ángulo y aplicar efectos para eliminar el fondo de los tickets a modo de post procesado.

Nos permitirá realizar una categorización del ticket y nos prefijará las alertas según la caducidad estándar de los productos en España, pudiendo modificarse este dato por el usuario.

Tras introducir todos los detalles del ticket nos dejará guardarlo, validando que no sea posible guardar un ticket que no tenga la información mínima necesaria.

4.2.5 Filtrado de tickets

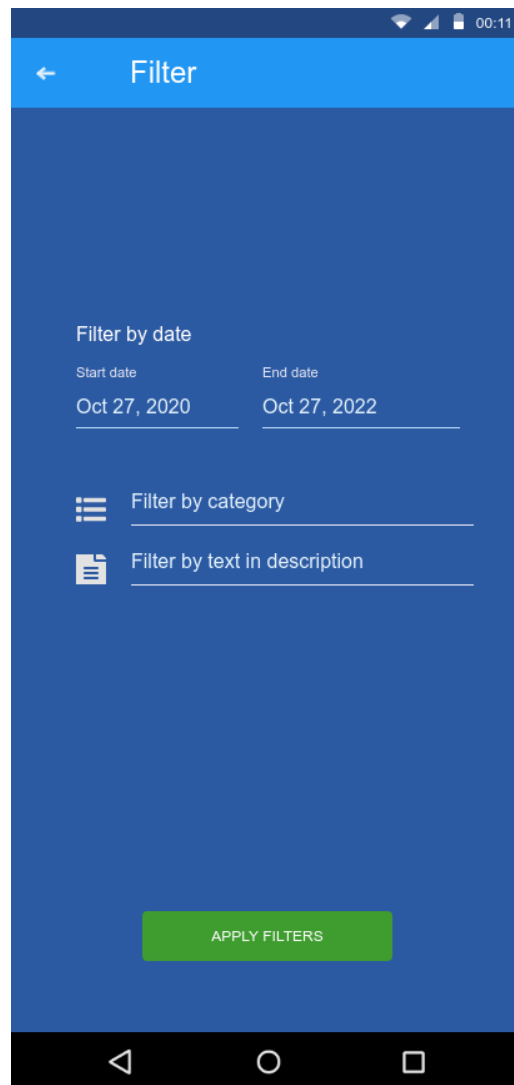


Ilustración 12. Prototipo, filtrar lista

Desde la lista de tickets se nos permite llegar a esta que muestra una serie de filtros que son aplicables al listado de ticket para la búsqueda de uno concreto. Según su categoría, texto contenido en la descripción o la fecha en la que fue tomado.

5. Casos de uso

5.1 Onboarding

Nombre	Onboarding
Actores	Usuario no logueado.
Precondiciones	El usuario ha instalado la aplicación y la abre por primera vez.

Postcondiciones	Este caso de uso no se volverá a mostrar para este usuario.
Flujo	<ol style="list-style-type: none"> 1. El usuario abre la aplicación por primera vez. 2. Desliza las pantallas hacia la izquierda hasta llegar a la última. 3. Presiona el botón de completado. 4. En la base de datos local se marca que este usuario ya ha completado el onboarding.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario abre la aplicación por primera vez. 2. Pulsa el botón de la derecha para cambiar de pantalla hasta llegar a la última. 3. Presiona el botón de completado. 4. En la base de datos local se marca que este usuario ya ha completado el onboarding.

5.2 Login

Nombre	Login
Actores	Usuario no logueado.
Precondiciones	<ul style="list-style-type: none"> • El usuario no está logueado • El usuario ha completado el onboarding.
Postcondiciones	El usuario queda registrado en el sistema
Flujo	<ol style="list-style-type: none"> 1. Al terminar el onboarding el usuario presiona en el botón de login con Google. 2. El usuario finaliza el login con las credenciales de Google. 3. En la base de datos local se marca que el usuario esta registrado en el sistema y se guarda su usuario.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario no se ha logueado en el sistema y abre la aplicación. 2. El usuario finaliza el login con las credenciales de Google.

	3. En la base de datos local se marca que el usuario esta registrado en el sistema y se guarda su usuario.
--	------------------------------------------------------------------------------------------------------------

5.3 Uso anonimo

Nombre	Uso anonimo
Actores	Usuario no logueado.
Precondiciones	<ul style="list-style-type: none"> • El usuario no está logueado • El usuario ha completado el onboarding.
Postcondiciones	La próxima vez que abra la aplicación se le pedirá si quiere hacer login.
Flujo	<ol style="list-style-type: none"> 1. Al terminar el onboarding el usuario presiona en el botón para entrar en la aplicación sin hacer login. 2. El usuario entra a utilizar la aplicación en modo desconectado

5.4 Primera demo

Nombre	Primera demo
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none"> • El usuario ha completado el onboarding. • El usuario está logueado, o ha pasado decidido hacer uso anónimo. • El usuario llega a la pantalla principal de la aplicación por primera vez.
Postcondiciones	Este caso de uso no se volverá a mostrar para este usuario.
Flujo	<ol style="list-style-type: none"> 1. El usuario llega a la pantalla principal de la aplicación 2. La aplicación le guía señalando dónde debe tocar para hacer el primer uso.

	3. En la base de datos local se marca que el usuario ya ha realizado la primera demo.
--	---------------------------------------------------------------------------------------

5.5 Demo

Nombre	Demo
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none"> • El usuario ha completado el onboarding. • El usuario está logueado, o ha pasado decidido hacer uso anónimo.
Flujo	<ol style="list-style-type: none"> 1. El usuario llega a la pantalla principal de la aplicación 2. El usuario pulsa el botón demo en el menú. 3. La aplicación le guía señalando dónde debe tocar para hacer el primer uso.

5.6 Añadir ticket

Nombre	Añadir ticket
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none"> • El usuario ha completado el onboarding. • El usuario está logueado, o ha pasado decidido hacer uso anónimo.
Postcondiciones	El ticket es guardado en la base de datos de la aplicación y sincronizado con la nube.
Flujo	<ol style="list-style-type: none"> 1. El usuario llega a la pantalla principal de la aplicación 2. Mediante el botón de acción abre la pantalla para añadir un nuevo ticket. 3. Mediante el botón de selección de imagen el usuario abre la cámara y realiza una foto. 4. Le aparece la vista previa con unos controles de recorte y efectos sobre la imagen.

	<ol style="list-style-type: none"> 5. Selecciona el recorte y efecto deseado para mejorar la calidad de la imagen. 6. Selecciona una categoría y los demás detalles necesarios del ticket en el formulario. 7. Presiona en guardar. 8. El ticket pasa a guardarse en la base de datos y a sincronizarse en la nube. 9. La pantalla se cierra y el ticket esta visible en la lista.
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.7 Editar ticket

Nombre	Editar ticket
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none"> • El usuario ha completado el onboarding. • El usuario está logueado, o ha pasado decidido hacer uso anónimo.
Postcondiciones	Las modificaciones en el ticket se guardan en la base de datos y se sincronizan con la nube.
Flujo	<ol style="list-style-type: none"> 1. El usuario llega a la pantalla principal de la aplicación 2. Hace clic en un ticket. 3. Se abre una pantalla similar a la de creación de ticket que le permite modificar los detalles del mismo. 4. Modifica lo que cree conveniente. 5. Presiona en guardar. 6. El ticket se actualiza en la base de datos y se sincroniza con la nube. 7. La pantalla se cierra y el ticket esta visible en la lista con los cambios aplicados.

5.8 Eliminar ticket

Nombre	Eliminar ticket
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none">• El usuario ha completado el onboarding.• El usuario está logueado, o ha pasado decidido hacer uso anónimo.
Postcondiciones	El ticket se elimina de la base de datos y de la nube.
Flujo	<ol style="list-style-type: none">1. El usuario llega a la pantalla principal de la aplicación2. Hace clic en un ticket.3. Se abre una pantalla similar a la de creación de ticket que le permite modificar los detalles del mismo.4. Hace click en el botón de eliminar ticket.5. Le aparece un diálogo de confirmación para comprobar que esta seguro de la acción.6. El usuario confirma la acción.7. La pantalla se cierra y el ticket desaparece de la lista de tickets.

5.9 Exportar tickets

Nombre	Exportar tickets
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none">• El usuario ha completado el onboarding.• El usuario está logueado, o ha pasado decidido hacer uso anónimo.
Postcondiciones	Se genera un PDF con el ticket y se comparte con otra aplicación.
Flujo	<ol style="list-style-type: none">1. El usuario llega a la pantalla principal de la aplicación2. Hace clic prolongado sobre un ticket.

	<ol style="list-style-type: none"> 3. La lista cambia al modo selección, el botón añadir tickets cambia a exportar. 4. El usuario selecciona el ticket o tickets que desea exportar. 5. El usuario presiona en el botón de acción que ahora tiene un icono de exportar. 6. Se crea un PDF con la información de los tickets seleccionados añadida y se abre el menú del sistema para compartir con otra aplicación. 7. El usuario comparte con otra aplicación y vuelve a la lista de tickets.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario llega a la pantalla principal de la aplicación. 2. Ejecuta la acción del menú identificada como “Exportar a PDF”. 3. La lista cambia al modo selección, el botón añadir tickets cambia a exportar. 4. El usuario selecciona el ticket o tickets que desea exportar. 5. El usuario presiona en el botón de acción que ahora tiene un icono de exportar. 6. Se crea un PDF con la información de los tickets seleccionados añadida y se abre el menú del sistema para compartir con otra aplicación. 7. El usuario comparte con otra aplicación y vuelve a la lista de tickets.

5.10 Filtrar tickets

Nombre	Filtrar tickets
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none"> • El usuario ha completado el onboarding. • El usuario está logueado, o ha pasado decidido

	hacer uso anónimo.
Postcondiciones	La lista de tickets es filtrada según los criterios.
Flujo	<ol style="list-style-type: none"> 1. El usuario llega a la pantalla principal de la aplicación 2. Presiona en el botón de filtro. 3. Selecciona los criterios por los que desea filtra la lista de tickets. 4. Presiona en aplicar filtro. 5. El usuario vuelve a la lista de tickets, la cual se muestra con el filtrado ya aplicado.

5.11 Eliminar publicidad

Nombre	Eliminar publicidad
Actores	Cualquier usuario de la aplicación
Precondiciones	<ul style="list-style-type: none"> • El usuario ha completado el onboarding. • El usuario está logueado, o ha pasado decidido hacer uso anónimo.
Postcondiciones	Si el usuario compra, se elimina el banner de publicidad de la aplicación.
Flujo	<ol style="list-style-type: none"> 6. El usuario llega a la pantalla principal de la aplicación 7. Como primera fila ve un mensaje que le avisa de que puede eliminar la publicidad, y que le quedan X tickets de espacio en su almacenamiento. 8. Hace clic en el enlace y le aparece un mensaje explicándole algún detalle más de las ventajas, con un enlace para hacer compra inapp. 9. El usuario realiza una compra inapp y la publicidad desaparece de su aplicación.

6. Arquitectura

6.1 Arquitectura del sistema

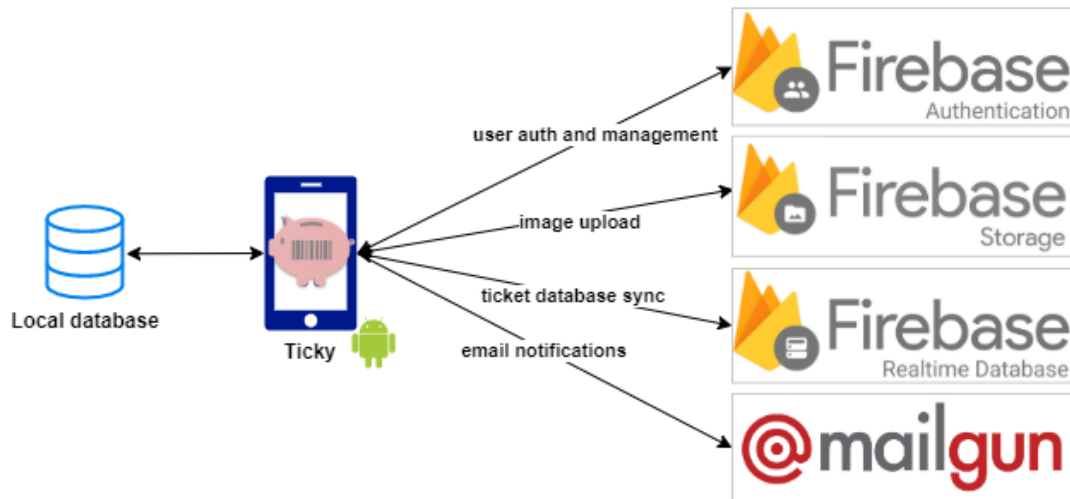


Ilustración 13. Arquitectura del sistema

El sistema se centra en una arquitectura serverless, para evitar tener que mantener un sistema central que pudiera tener problemas de escalabilidad así mismo como evitar tener que hacer desarrollo del lado de servidor.

De este modo, el sistema se nutre de Firebase Authentication para el login, registro y gestión de los usuarios, de Firebase Storage para subir las imágenes de los tickets bajo un espacio seguro asignado al usuario que los sube y de Firebase Realtime Database para persistir el estado de los tickets con sus propiedades.

Por último, utiliza un servicio de MailGun para realizar las notificaciones sobre el correo del usuario directamente, a través el API segura de MailGun.

6.2 Arquitectura de la aplicación

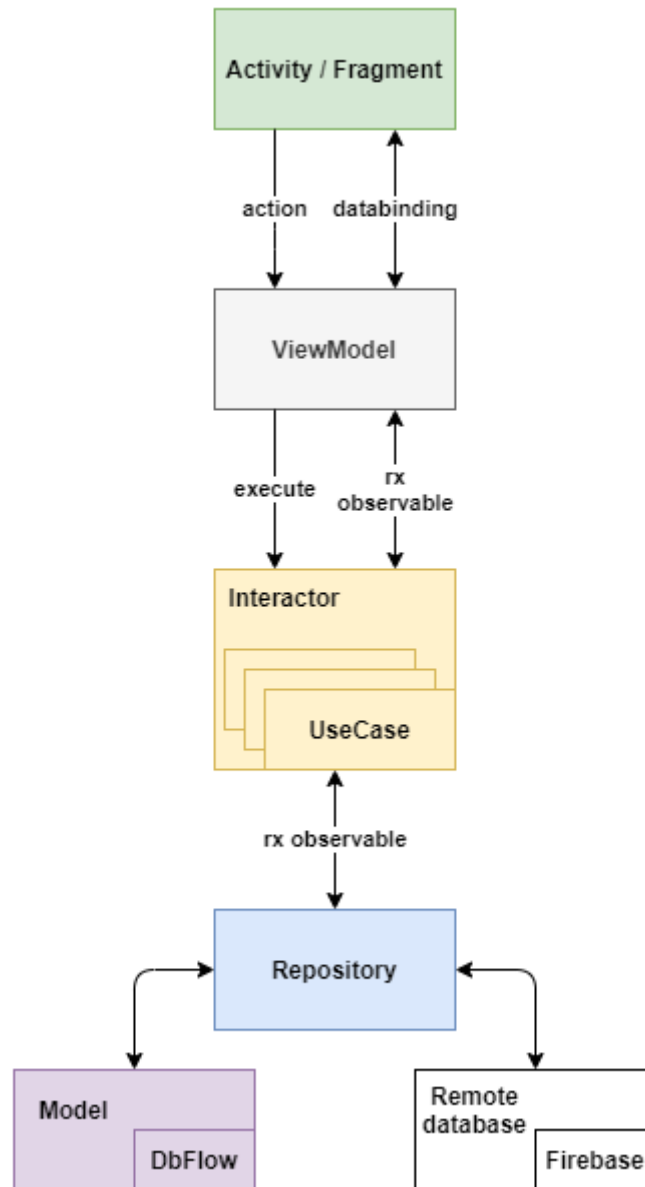


Ilustración 14. Arquitectura de la aplicación

La arquitectura que se plantea para la aplicación es MVVMi, como puede verse en la figura anterior en detalle.

Me he decantado por MVVMi porque hace énfasis en que en el ViewModel no debe tener ninguna referencia directa a ningún elemento de vista; lo cual garantiza la separación de roles y su posterior testabilidad. Con el añadido de los Interactors que nos permiten combinar con reglas de negocio sobre los datos.

Se valoraron MVP, MVVM y VIPER con los siguientes resultados:

- MVP: fue descartado porque, aun siendo similar a MVVM en su base, no maneja correctamente el ciclo de vida de la interfaz y habría que ir

teniéndolo en cuenta con añadidos para liberar memoria del Presenter y no realizar ningún Context leak.

- VIPER: fue descartado porque, aunque se focaliza en la separación de responsabilidad única lo cual es muy interesante si se estima que va a crecer la aplicación, no justifica la carga de trabajo necesario para su aplicación. Sería sobre ingeniería para nuestro propósito.

6.2 Modelo de datos.

Tanto en la base de datos local como en la remota en Firebase Realtime Database, utilizamos el siguiente modelo de datos. Estos datos se intercambian con el servidor como JSON por lo que están optimizados para ser serializados a este formato.

6.2.1 Ticket

Fields

- id: String
- photo: String
- description: String
- start_date: Long (millis from epoch)
- end_date: Long (millis from epoch)
- category_id: Int
- place_name: String
- place_id: String
- lat: Double
- lng: Double
- notif_time: Long
- notif_phone: Boolean
- notif_email: Boolean
- notified: Boolean

Example

```
{
  "category_id": "10",
  "description": "Comida",
  "end_date": 1506290400000,
  "id": "3a5beb87-42ae-41b9-840d-832d984ec56e",
  "lat": 37.9860967,
  "lng": -1.127894,
  "notif_email": false,
  "notif_phone": false,
  "notif_time": 0,
  "notified": false,
  "photo": "https://picture.url.com/fake/photo",
  "place_id": "ChIJxYac-xqCYw0RfaAHYvFcKdM",
  "place_name": "Huertamar",
  "start_date": 1506290400000
},
```

Ilustración 15. Ticket, ejemplo de modelo

6.2.2 Usuario

Fields

- id: String
- mail: String
- name: String
- avatar: String

Example

```
{
  "id": "3a5beb87-42ae-41b9-840d-832d984ec56e",
  "avatar": "https://fake.avatar.com/userfakeId",
  "mail": "jhondoole@fake.com",
  "name": "John Doole"
},
```

Ilustración 16. Usuario, ejemplo de modelo

6.2.2 Categoría

Fields

- id: String
- name: String
- image: String
- color: Int

Example

```
{
  "id": "3a5beb87-42ae-41b9-840d-832d984ec56e",
  "name": "Electrodomésticos",
  "image": "https://fake.image.url/imageFakeId",
  "color": 1506290
},
```

Ilustración 17. Categoría, ejemplo de modelo

7. Implementación

La aplicación se ha implementado siguiendo el patrón descrito en el punto 6.2 de este documento, siendo este MVVMi.

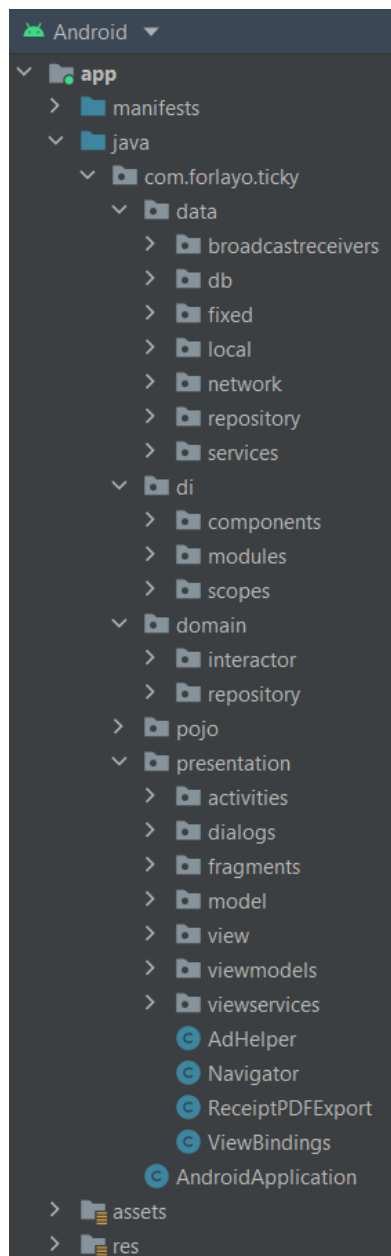


Ilustración 18. Paquetes de la aplicación

El proyecto de la aplicación no obstante se compone de dos partes, la aplicación propiamente dicha y una envoltura sobre una librería en C++ que hace uso de OpenCV (OpenCV, 2020) para realizar el post procesamiento de las imágenes que se incluyen como tickets.

Esta librería está basada en la implementación de AndroidScannerDemo (Karee, 2015), la cual tiene licencia MIT “AS IS” por lo que no entra en conflicto con la licencia de este proyecto.

Dicha librería provee a la aplicación de una actividad especial llamada ScannActivity que es la responsable de toda la parte de tomar la imagen recortarla y aplicar filtros con OpenCV (OpenCV, 2020). La librería nativa que consume ha sido compilada para todas las arquitecturas de procesador que soporta Android actualmente.

Quedando un proyecto compuesto de una librería y una aplicación que consume a esta.

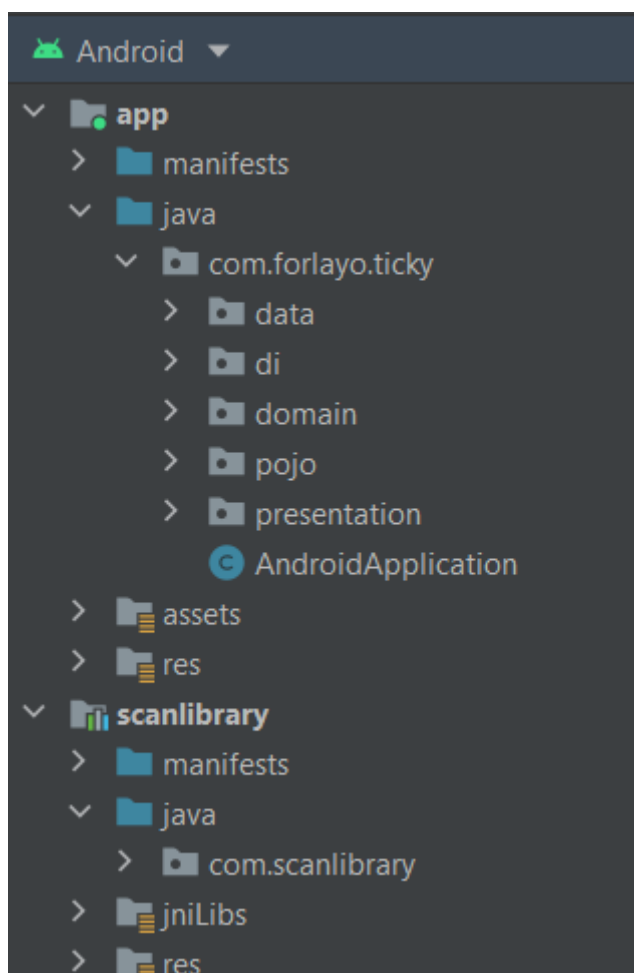


Ilustración 19. Estructura del proyecto

8. Pruebas

A fin de garantizar que el funcionamiento de la aplicación se corresponde con el esperado se diseña un plan de pruebas funcionales que cubran los casos de uso de la aplicación.

Idealmente este plan de pruebas debería ejecutarse de forma automatizada por algún sistema de integración continua como validación final en posteriores versiones de la aplicación.

Para la elaboración de este proyecto los tests han sido ejecutados de forma manual.

8.1 Plan de pruebas

Id	Comportamiento esperado	Área
1	En primera ejecución, el usuario ve una pantalla de presentación.	Presentación
2	Después de haber pasado la pantalla de presentación, esta no vuelve a aparecer al abrir la aplicación.	Presentación
3	El usuario puede hacer login con su cuenta de Google.	Login
4	El usuario puede evitar hacer login con su cuenta de Google haciendo clic en el botón atrás.	Login
5	Si el usuario no hizo login, se le volverá a ofrecer la opción al entrar en la aplicación.	Login
6	Si el usuario hizo login, no se le volverá a ofrecer la opción al entrar en la aplicación.	Login
7	La primera vez que se llega a la página principal, aparece una visita guiada por la aplicación, la cual muestra al usuario las características principales.	Onboarding
8	Después de haber visto la visita guiada esta no vuelve a aparecer al entrar a la aplicación.	Onboarding
9	Desde el menú de los tres puntos, se puede volver a abrir la visita guiada.	Onboarding
10	Al final de la visita guiada es posible generar el primer ticket.	Onboarding
11	En la pantalla principal de la aplicación se observa un	Publicidad

	banner de publicidad en la zona inferior	
12	Se puede hacer clic en el banner de publicidad.	Publicidad
13	Desde la pantalla principal existe un botón que nos permite crear un ticket.	Crear Ticket
14	En la creación del ticket podemos seleccionar una imagen, desde la cámara o la galería.	Crear Ticket
15	La imagen seleccionada de la cámara o la galería ofrece recorte de bordes y cambio de color.	Crear Ticket
16	En la creación del ticket podemos seleccionar fecha de expiración (automática a 2 años) categoría, localización, descripción y notificación.	Crear Ticket
17	La notificación configurable permite que sea por correo o en el propio teléfono como una notificación local.	Crear Ticket
18	La notificación configurable permite seleccionar un día, una semana, dos semanas o un mes; antes del periodo de expiración del ticket.	Crear Ticket
19	Cuando guardamos un ticket, este aparece en la pantalla principal.	Listado Tickets
23	Cuando en la pantalla principal existe al menos un ticket, aparece un header ofreciendo al usuario quitar la publicidad y aumentar la cantidad de ticket que puede guardar en su nube.	Inapp purchase
24	El header de compras in app se puede ocultar.	Inapp purchase
25	Al hacer click en el header de compras in app, aparece la oferta para que haga la compra directamente en la app.	Inapp purchase
26	El header de compras se puede mostrar a través del menú de los tres puntos.	Inapp purchase
27	Si se compra eliminar la publicidad, el banner deja de aparecer.	Inapp purchase
28	Si se compra aumentar el limite de tickets de la nube, puede añadir tickets sobre dicho limite.	Inapp purchase
29	La lista de tickets puede ser filtrada por categoría, rango	Filtrar ticket

	de fechas o texto incluido.	
30	Un ticket puede abrirse haciendo clic en el para ver sus detalles.	Editar ticket
31	Al abrir un ticket existe la posibilidad de eliminarlo.	Editar ticket
32	Al abrir un ticket existe la posibilidad de modificarlo.	Editar ticket
33	Si el usuario hizo login con su cuenta de Google. Los tickets que guarde aparecerán si reinstala la aplicación.	Cloud
34	Existe una opción para seleccionar tickets, generar un PDF con su detalle y enviar a una tercera aplicación.	Exportar PDF

8.2 Resultados de los tests

Los tests han sido realizados sobre 3 emuladores distintos, para Pixel C, Pixel 2 y Pixel 3a. Adicionalmente, se han reproducido los mismos tests sobre un Samsung Galaxy S8.

Teniendo en todos los mismos resultados los cuales detallo en la siguiente tabla.

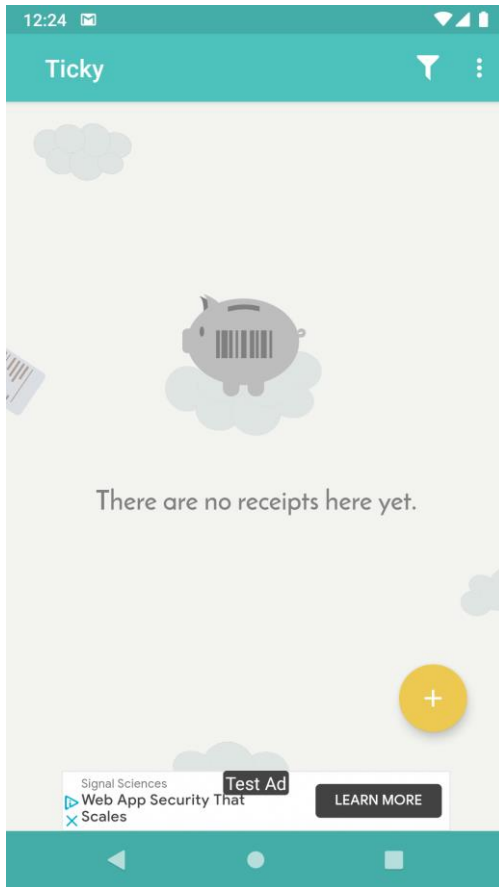
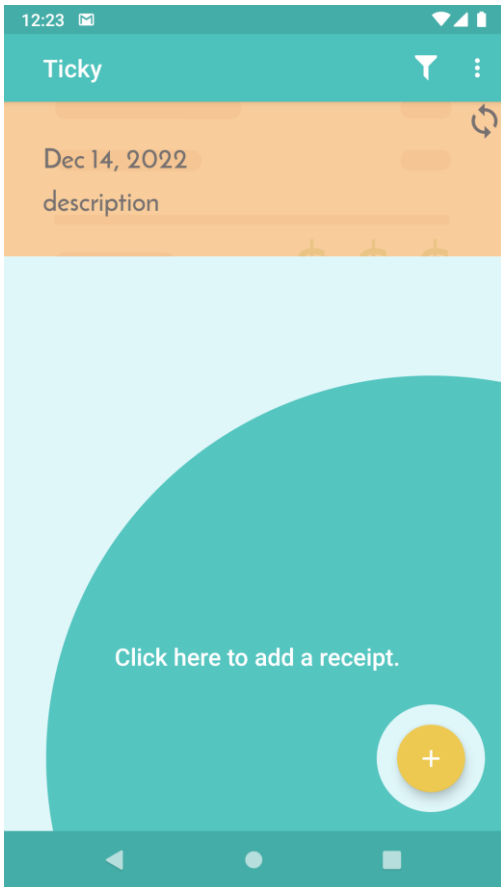
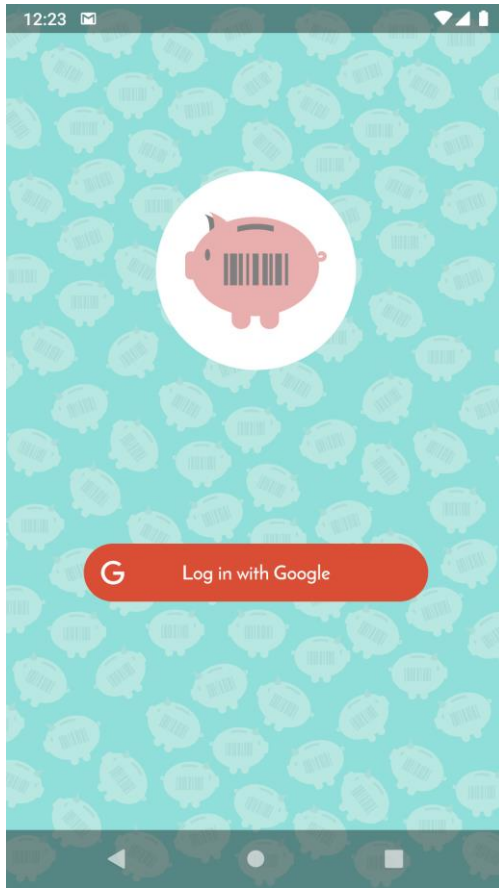
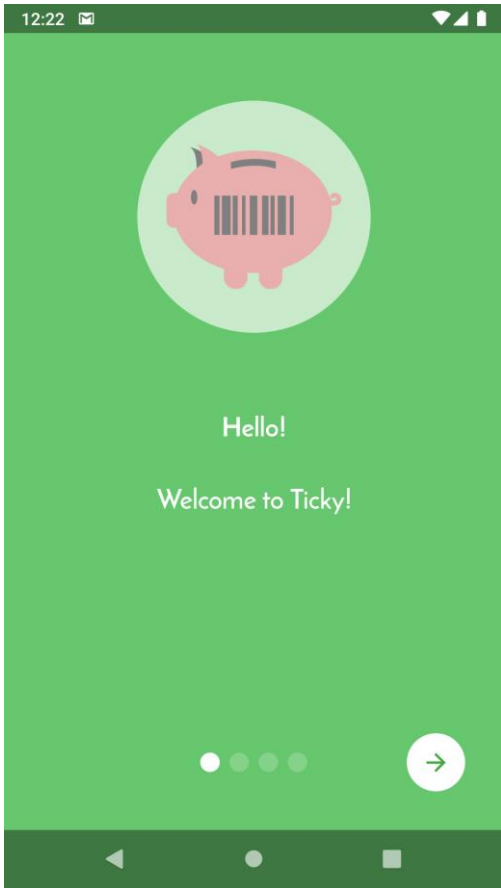
Id	Comportamiento esperado	Resultado
1	En primera ejecución, el usuario ve una pantalla de presentación.	OK
2	Después de haber pasado la pantalla de presentación, esta no vuelve a aparecer al abrir la aplicación.	OK
3	El usuario puede hacer login con su cuenta de Google.	OK
4	El usuario puede evitar hacer login con su cuenta de Google haciendo clic en el botón atrás.	OK
5	Si el usuario no hizo login, se le volverá a ofrecer la opción al entrar en la aplicación.	OK
6	Si el usuario hizo login, no se le volverá a ofrecer la opción al entrar en la aplicación.	OK
7	La primera vez que se llega a la página principal, aparece una visita guiada por la aplicación, la cual muestra al usuario las características principales.	OK
8	Después de haber visto la visita guiada esta no vuelve a	OK

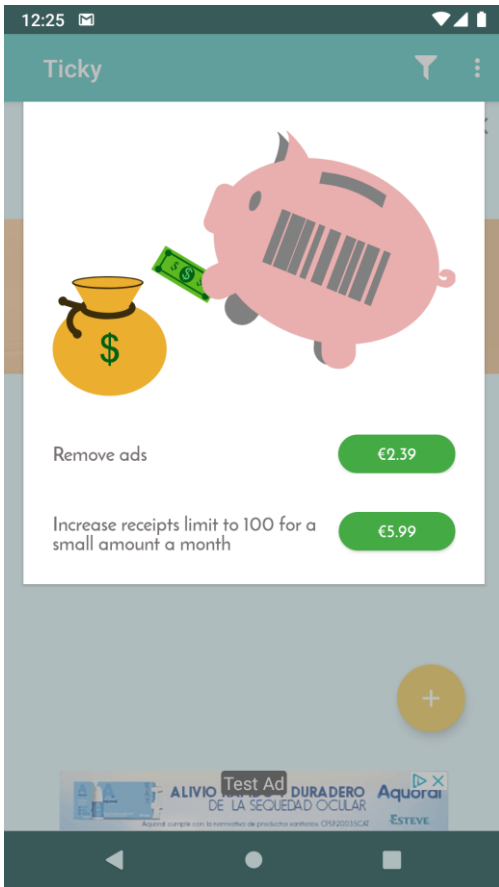
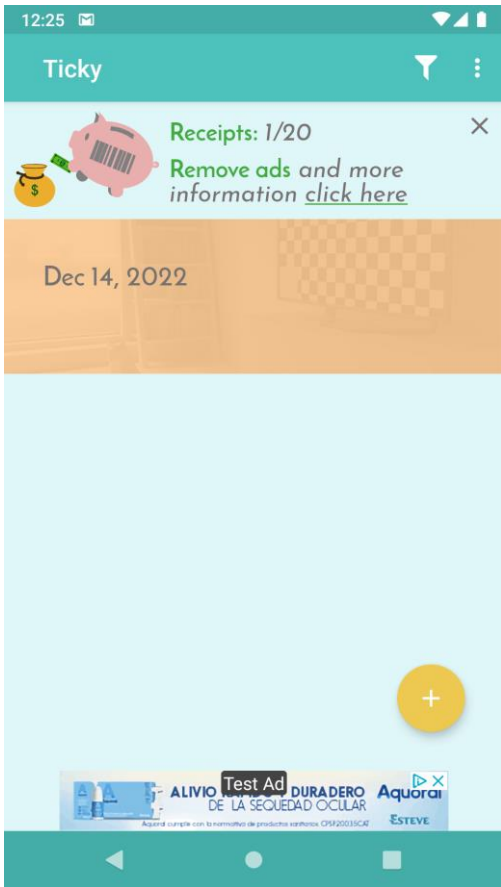
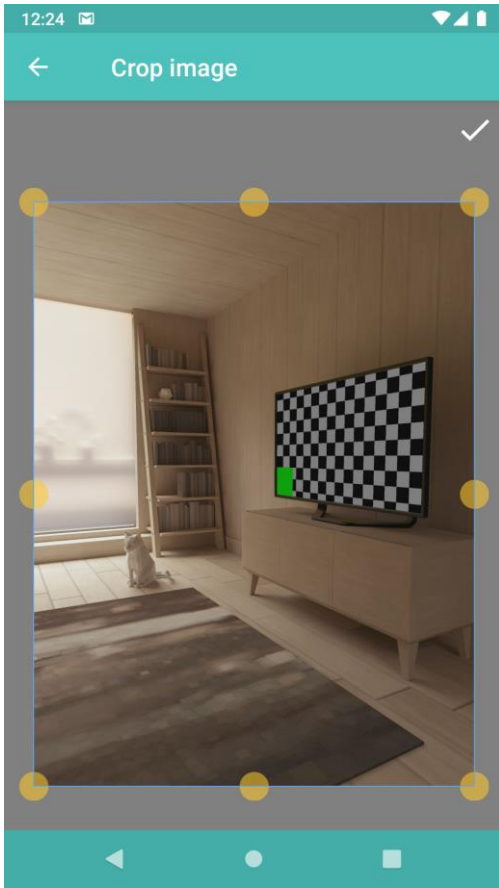
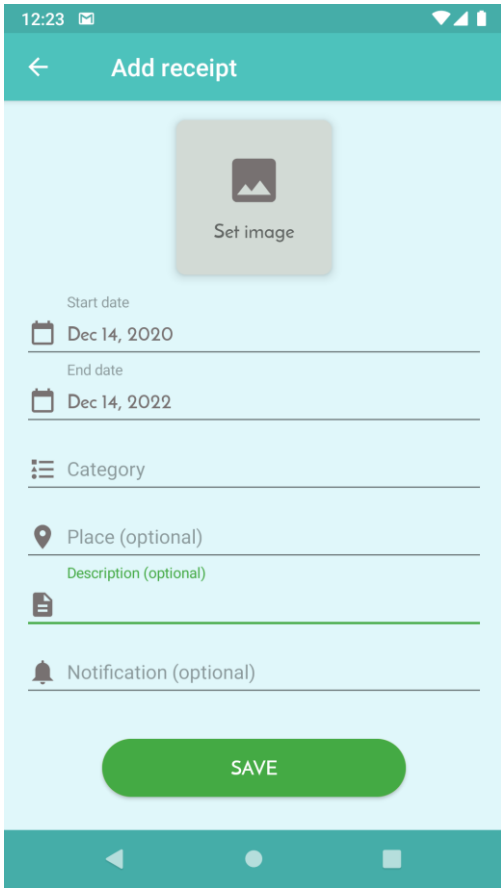
	aparecer al entrar a la aplicación.	
9	Desde el menú de los tres puntos, se puede volver a abrir la visita guiada.	OK
10	Al final de la visita guiada es posible generar el primer ticket.	OK
11	En la pantalla principal de la aplicación se observa un banner de publicidad en la zona inferior	OK
12	Se puede hacer clic en el banner de publicidad.	OK
13	Desde la pantalla principal existe un botón que nos permite crear un ticket.	OK
14	En la creación del ticket podemos seleccionar una imagen, desde la cámara o la galería.	OK
15	La imagen seleccionada de la cámara o la galería ofrece recorte de bordes y cambio de color.	OK
16	En la creación del ticket podemos seleccionar fecha de expiración (automática a 2 años) categoría, localización, descripción y notificación.	OK
17	La notificación configurable permite que sea por correo o en el propio teléfono como una notificación local.	OK
18	La notificación configurable permite seleccionar un día, una semana, dos semanas o un mes; antes del periodo de expiración del ticket.	OK
19	Cuando guardamos un ticket, este aparece en la pantalla principal.	OK
23	Cuando en la pantalla principal existe al menos un ticket, aparece un header ofreciendo al usuario quitar la publicidad y aumentar la cantidad de ticket que puede guardar en su nube.	OK
24	El header de compras in app se puede ocultar.	OK
25	Al hacer click en el header de compras in app, aparece la oferta para que haga la compra directamente en la app.	OK
26	El header de compras se puede mostrar a través del	OK

	menú de los tres puntos.	
27	Si se compra eliminar la publicidad, el banner deja de aparecer.	OK
28	Si se compra aumentar el límite de tickets de la nube, puede añadir tickets sobre dicho limite.	OK
29	La lista de tickets puede ser filtrada por categoría, rango de fechas o texto incluido.	OK
30	Un ticket puede abrirse haciendo clic en el para ver sus detalles.	OK
31	Al abrir un ticket existe la posibilidad de eliminarlo.	OK
32	Al abrir un ticket existe la posibilidad de modificarlo.	OK
33	Si el usuario hizo login con su cuenta de Google. Los tickets que guarde aparecerán si reinstala la aplicación.	OK
34	Existe una opción para seleccionar tickets, generar un PDF con su detalle y enviar a una tercera aplicación.	OK

9. Aspecto final de la aplicación

En este apartado se muestran algunas capturas de la versión final de la aplicación que se ha implementado, intentando cubrir un vistazo rápido sobre la aplicación y su funcionalidad.





10. Conclusiones

10.1 Lecciones aprendidas

Se ha aprendido que antes de comenzar con el diseño de una aplicación o servicio es primordial realizar un estudio del mercado actual para determinar qué competidores tenemos y qué funcionalidad están ofreciendo. Con el fin de determinar si nuestra idea es realmente innovadora o cómo va a mejorar lo que está actualmente disponible. Es decir, en definitiva, determinar la viabilidad de la aplicación o servicio.

Además, esta información ha sido muy relevante para recopilar casuísticas de uso y tipos de interfaz más adecuadas para la aplicación.

Otra lección importante que cabe destacar es que la estimación de tiempo de las tareas que se van a llevar a cabo en un proyecto es vital para el desarrollo del mismo; sobre todo aquellas que no siendo técnicas son necesarias y también reducen la cantidad de tiempo disponible. Pues en este proyecto se ha podido hacer *crunch* pero en un proyecto real podría haber incurrido en algún retraso.

10.2 Logro de los objetivos

Los objetivos que se proponían en los requisitos definidos por el proyecto se han cumplido todos, tanto los funcionales como los no funcionales por lo que se ha conseguido un producto final que ha podido validarse manualmente en distintos dispositivos compatibles.

Se han realizado diseños de logotipos e imágenes para completar el aspecto visual de la aplicación, así como algunas pequeñas animaciones para darle un efecto atractivo de cara al usuario.

Si bien es cierto que, aunque los objetivos se han logrado, me habría gustado mejorar alguna parte de la interfaz de usuario; como el formulario de introducción del ticket. Dejo esto reflejado también como punto de trabajo a futuro.

10.3 Seguimiento de la planificación y metodología

La planificación definida se ha podido seguir en su totalidad, no obstante, algunas de las tareas fueron estimadas erróneamente se han tenido que invertir muchas más horas de trabajo de las que la planificación planteaba; para poder cumplir con el objetivo en la fecha de entrega definida.

Afortunadamente se cumplió finalmente con las fechas propuestas, aunque de forma algo ajustada y con una inversión inesperada de horas.

10.4 Líneas de trabajo futuro

En un planteamiento a futuro, se podrían realizar investigaciones para implementar una funcionalidad que permita recopilar directamente la información contenida en los tickets evitando así que el usuario tenga que hacer una introducción manual de el detalle de los mismos.

Desde el punto de vista de la evaluación del impacto, se podrían realizar varias iteraciones para evaluar la usabilidad del diseño final y las opiniones que pudiera tener los usuarios tras el uso prolongado de la aplicación.

Así mismo, sería interesante añadir un plan de test automático para cada complicación.

11. Glosario

Admob

Servicio de banners de publicidad proporcionado por Google.

Crunch

Término en la industria del software que se emplea para cuando hay que cumplir una fecha de entrega inamovible a costa de trabajar más horas de las esperadas.

Firebase

Conjunto de servicios en la nube de Google, especialmente orientado para aplicaciones móviles.

Framework

Conjunto de herramientas reutilizables en diferentes proyectos.

Inapp

Algo que sucede sin abandonar una aplicación.

In-house

Algo que realizamos por nosotros mismos.

JSON

JavaScript Object Notation, una forma de definición de objetos mediante texto.

Login

Iniciar sesión.

Logueado

Hecho de haber iniciado sesión actualmente.

MIT

Massachusetts Institute of Technology, es una Universidad privada estadounidense.

MVP

Modelo-Vista-Presentador, es un patrón de diseño de aplicaciones a nivel de código.

MVVM

Modelo-Vista-VistaModelo, es un patrón de diseño de aplicaciones, a nivel de código.

Onboarding

Proceso de transmisión de conocimiento y habilidades necesarias para el uso de un sistema o herramienta.

Open source

Software en el que el código fuente es de dominio público, con posibilidad de modificación.

PDF

Portable Document Format, es un popular formato de archivo.

Prototipo

Modelo de aplicación o servicio en el que se refleja la definición funcional esperada, a modo de ensayo o primer molde.

Spot-on

Efecto a modo de lupa que enseña una parte concreta de una pantalla.

VIPER

Vista-Interactor-Presenter-Entity-Router, es un patrón de diseño de aplicaciones a nivel de código.

12. Bibliografía

- Google. (2020). *Android Developers Documentation*. Obtenido de <https://developer.android.com/docs>
- Google. (2020). *Firebase Documentation*. Obtenido de <https://firebase.google.com/docs>
- Karee, J. (2015). *AndroidScannerDemo*. Obtenido de GitHub Jhansi Karee: <https://github.com/jhansireddy/AndroidScannerDemo>
- Mailgun. (2020). *Mailgun Documentation*. Obtenido de <https://documentation.mailgun.com/en/latest/>
- okticket. (2020). *okticket*. Obtenido de <https://www.okticket.es/>
- OpenCV. (2020). *OpenCV Library*. Obtenido de <https://opencv.org/>
- UNUNUZI Consulting S.L. (2020). *Tiketi*. Obtenido de <https://tiketi.es/>
- Yun Cheng, A. O. (2019). *Advanced Android App Architecture*. McGaheysville, Virginia, USA: Razeware LLC.

13. Anexos

13.1 Gantt planificación

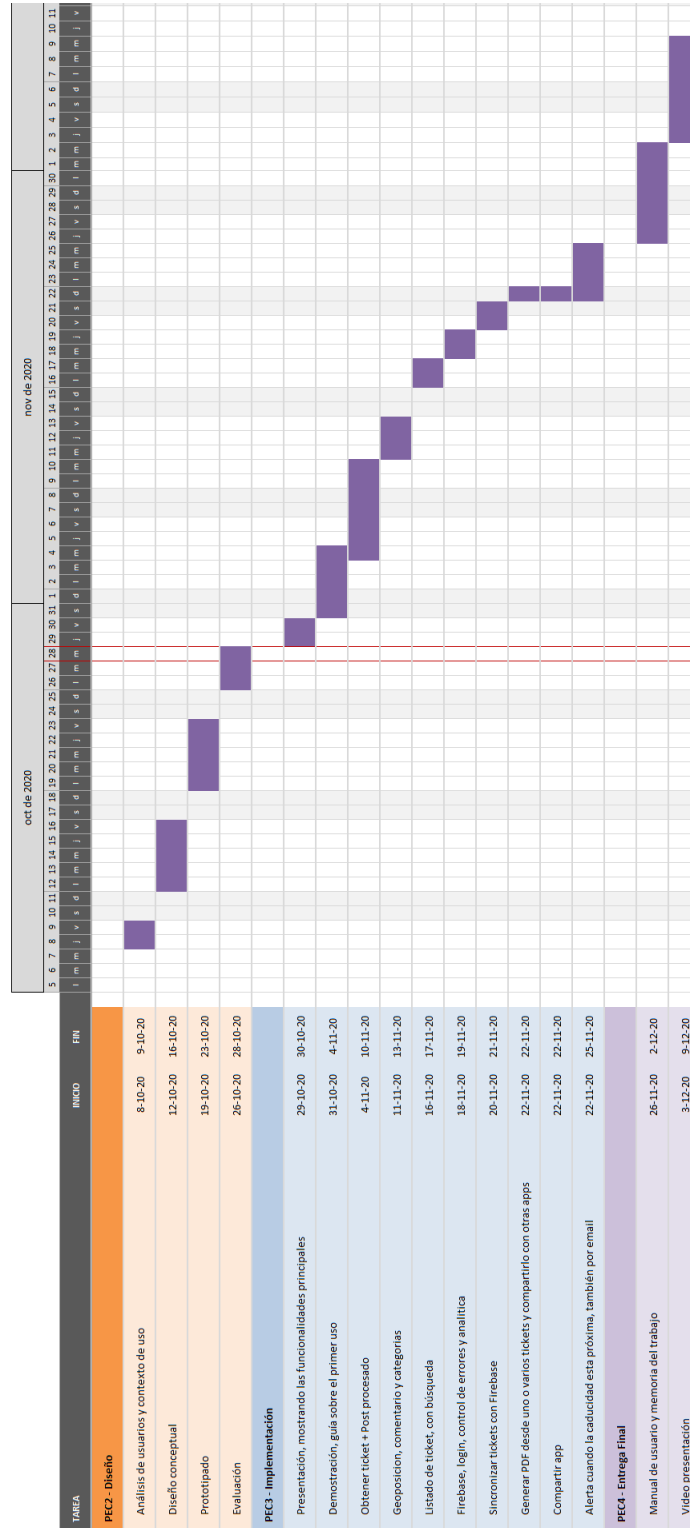


Ilustración 20. Gantt planificación, tamaño completo

13.2 Instrucciones para la compilación.

Prerrequisitos:

- Android Studio con la versión 30 de Android SDK instalada.
- Es importante usar el keystore con nombre debug.keystore para compilar, ya que utiliza varios servicios de Google que van a comprobar la firma del APK.

Los pasos para compilar la aplicación son los siguientes:

1. Importar el proyecto Ticky con Android Studio.
2. Compilar el módulo app, el cual consumirá scanlibrary y ejecutarlo en un emulador o dispositivo real.

También puede ser compilada desde consola, ejecutando el siguiente comando en el directorio raíz del proyecto:

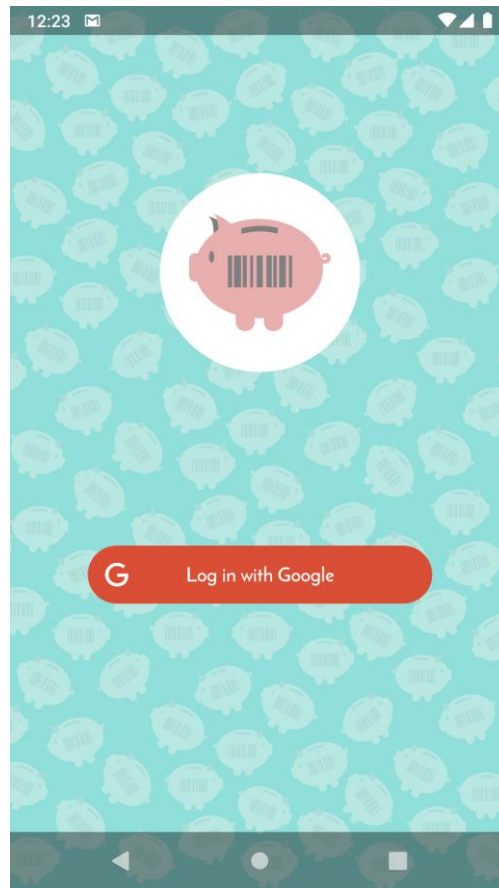
```
gradlew.bat aR
```

13.3 Manual de usuario

13.3.1 Requisitos de la aplicación

- Dispositivo Android, los emuladores están soportados también.
- Android JellyBean 4.1 o superior

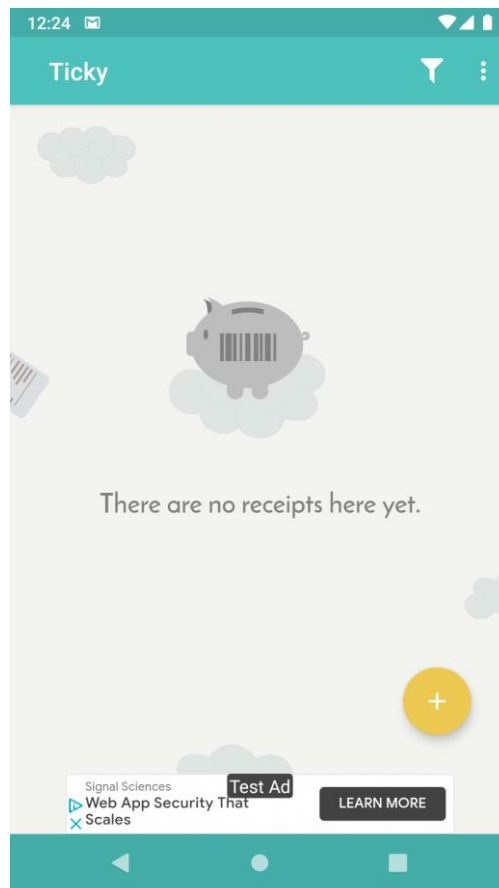
13.3.2 Login



Tras visualizar el onboarding, llegamos a la pantalla de login, en la cual tenemos dos opciones:

- Login con cuenta de Google: Se accede haciendo clic en el botón que así lo indica
- Login anónimo: Simplemente hay que hacer clic en el botón atrás. Tenga en cuenta que en este caso los tickets se guardarán sólo de manera local.

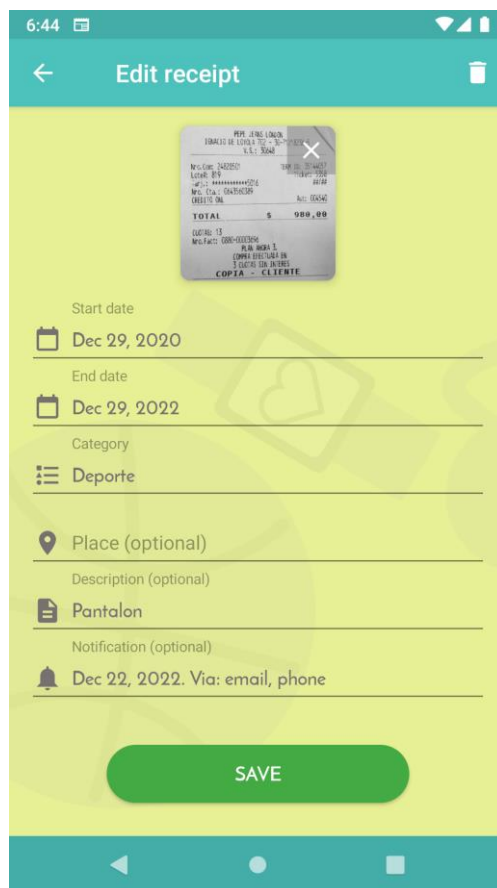
13.3.3 Creación de un ticket



La primera vez que se entra en la aplicación se le guiará por la misma y acabará justo en el punto de creación de un nuevo ticket.

Para crear más en el futuro, sólo necesita hacer clic en el botón con el símbolo 'x' que encontrará en la pantalla principal.

13.3.4 Modificación y borrado de un ticket.



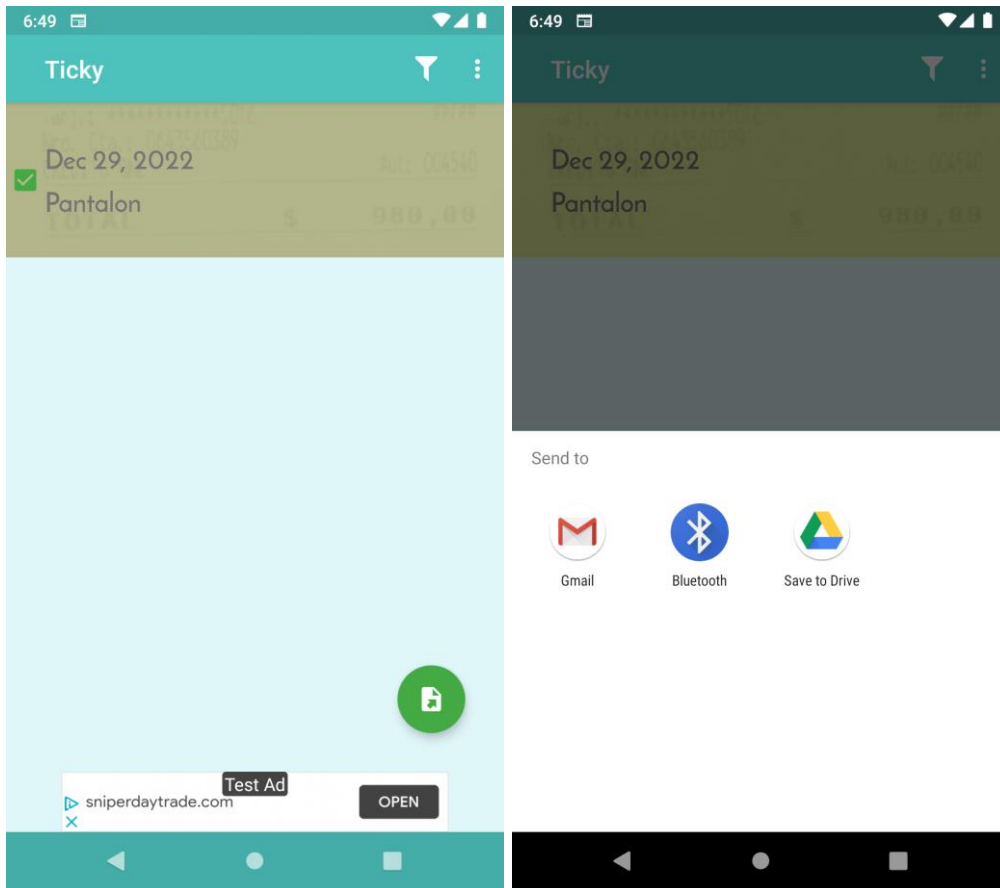
Pulsando sobre cualquiera de los tickets de la lista accederá a sus detalles es en esta pantalla donde puede, además de modificar los detalles del ticket, borrarlo si así lo desea.

13.3.5 Exportación de tickets en PDF.

Tiene dos métodos para realizar esta acción:

- Realizando una pulsación prolongada sobre cualquier ticket en el listado.
- Seleccionando "Export as PDF..." del menú de tres puntos que encontrará en la barra superior a la derecha

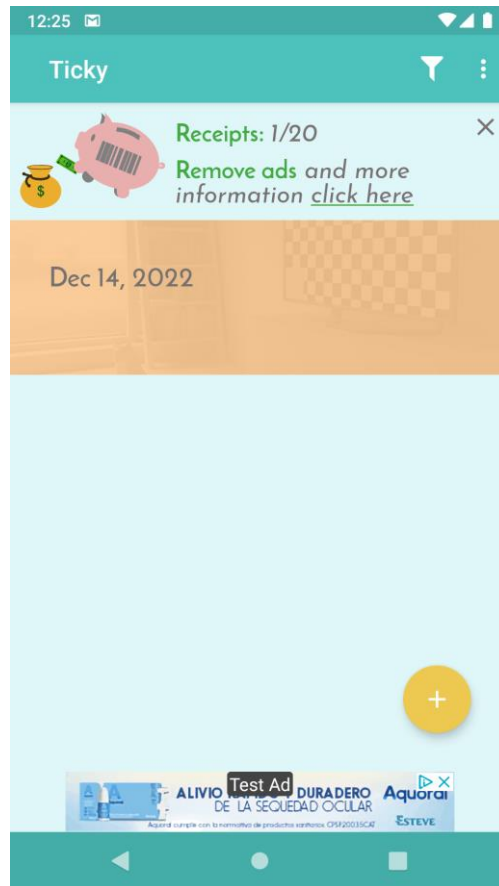
Tras realizar cualquiera de las dos opciones la lista cambiará para que pueda seleccionar los tickets que quiere exportar a PDF, y el botón que antes era para crear tickets ahora pasará a tener un icono indicando que va a exportar lo que seleccione.



Una vez seleccionado lo que queremos pulsamos en el botón para exportar, antes comentado y se abrirá el selector de aplicaciones para que indiquemos a dónde queremos enviar el PDF con los tickets seleccionados.

Para cancelar esta acción, simplemente haga clic en el botón atrás en su dispositivo.

13.3.6 Eliminar publicidad y aumentar la cantidad de tickets posibles.



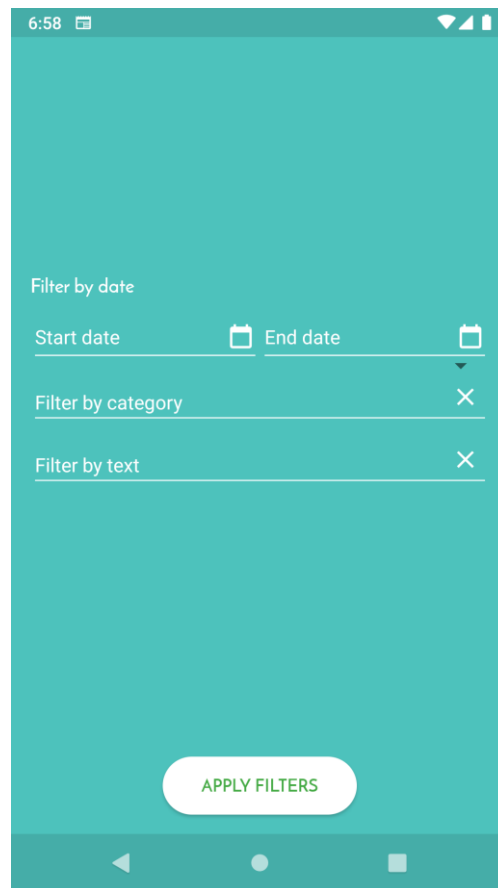
Cuando la lista contiene algún ticket, aparecerá un encabezado ofreciendo poder ampliar el límite de tickets o eliminar la publicidad.

Haciendo clic en él podremos acceder a las compras inapp y, mediante un pago a través de nuestra cuenta de Google Play, podremos realizar cualquiera de las dos acciones.

Si ha ocultado dicho encabezado, puede volver a hacer aparecer haciendo clic en la opción “Show receipts info” del menú de los tres puntos que tiene a la derecha de la barra superior.

13.3.7 Filtrado de tickets en el listado.

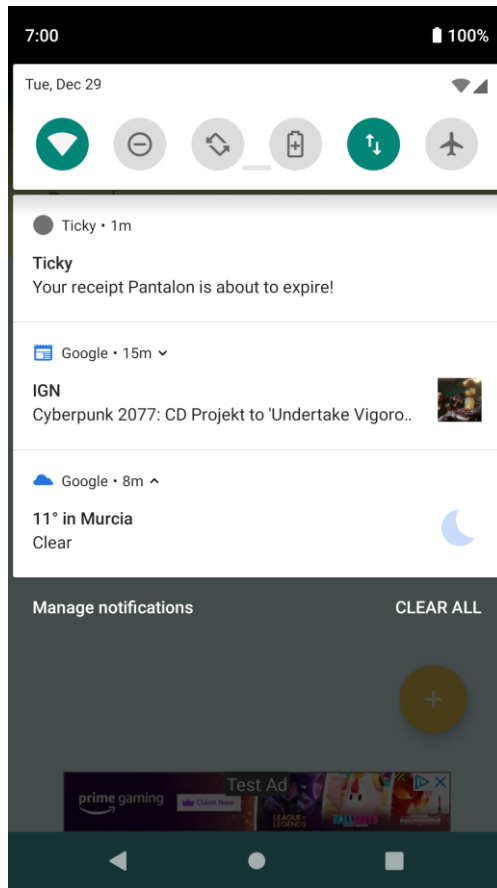
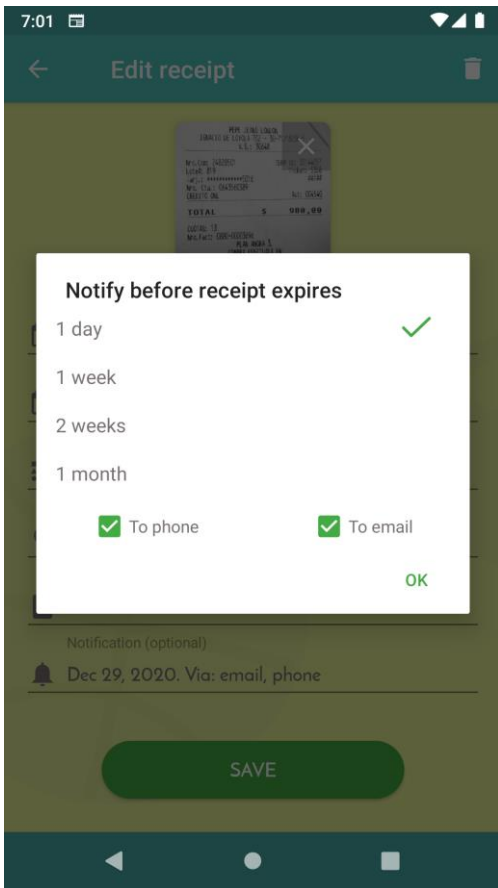
En la pantalla principal pulsando en el icono de la barra superior con forma de embudo, accederá a la configuración para filtrar la lista de tickets.



Al aplicar el filtro, los criterios seleccionados se aplicarán sobre la lista de tickets. Cuando se sale y se vuelve a entrar a la aplicación el filtrado se reinicia.

13.3.8 Notificaciones.

La aplicación le notificará cuando la caducidad de uno de sus tickets esté próxima, según las opciones que seleccione en el propio ticket.



Dependiendo de lo que seleccione, recibirá un correo o una notificación en el propio teléfono cuando el ticket esté próximo a expirar.

13.3.9 Cloud storage.

Si hizo login con su cuenta de Google, todos los tickets y sus detalles serán guardados en nuestro Cloud; si perdiese el teléfono o accediese desde otro distinto con la misma cuenta recuperaría los tickets sin problemas.