



Inventario de colecciones privadas para iOS

Oscar Fernández Ardanuy

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: Eduard Martin Lineros

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

06/2016



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Inventario de colecciones privadas para iOS</i>
Nombre del autor:	<i>Oscar Fernández Ardanuy</i>
Nombre del consultor/a:	<i>Eduard Martin Lineros</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega:	06/2016
Titulación:	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave:	<i>Desarrollo, iOS, Swift</i>
Resumen del Trabajo:	
<p>Este documento describe el desarrollo y creación de una aplicación para iPhone, que permite a sus usuarios la creación de un inventario organizado para sus colecciones privadas.</p> <p>Esta aplicación centra su funcionamiento en el plano offline. El perfil de usuario ideal es el que pretende tener un inventario privado, donde consultar que piezas tiene con facilidad. La aplicación pretende substituir las listas manuales, u hojas de cálculo, que utilizan estos usuarios para tal fin.</p> <p>Antes de realizar la aplicación, se ha analizado el mercado en busca de alternativas, con la intención de encontrar similitudes y carencias, con las que buscar perfiles de usuarios no representados. También se han realizado estudios de usuarios, para conocer sus necesidades y poder adaptar mejor la aplicación.</p> <p>Buscando la mejor experiencia de usuario, durante el proceso de diseño del prototipo, se ha tenido en cuenta la valoración de usuarios que cumplen el perfil deseado, hasta dar con el prototipo más adecuado.</p> <p>Se ha diseñado una programación, para repartir los recursos de la forma más adecuada, y tener un mayor control del estado del proyecto. La aplicación se ha desarrollado para iOS, en nativo, para aprovechar las capacidades del sistema y las herramientas oficiales.</p> <p>Como resultado, se ha desarrollado una aplicación estable y funcional. Se han añadido todas las funcionalidades programadas, y algunas mejoras, aprovechando ciertas características de la plataforma.</p> <p>Durante el proyecto se ha aprendido a valorar la necesidad de una planificación y de lo importante que es tener en cuenta a los usuarios.</p>	

Abstract:

This document describes the development and creation of an application for iPhone, which will allow users the creation of an organized inventory for private collections.

This application focuses its operation in the offline scenario. The ideal user profile is intended to have a private inventory, where user can check the owned pieces. The application aims to replace manual lists or spreadsheets used traditionally for that purpose.

Before making the application, we have analyzed the market searching for alternatives with the intention of finding similarities and shortcomings to identify non-represented users. Users' studies have also been done to identify their needs and adapt the application accordingly.

Looking for the best user experience during the design process of the prototype, we have taken into account the users evaluations that meet the desired profile until we found the most suitable prototype.

We have designed a system to allocate resources in the most appropriate manner as well as ensuring the greatest project status control. The application has been developed for iOS, native, to leverage the capabilities of the system and the official tools.

As a result, we have developed a stable and functional application. All programmed functions and some improvements were added in order to take advantage of certain features of the platform.

Along the project we have learnt about the need for planning and the importance of taking into account users input.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo	5
1.3 Enfoque y método seguido.....	6
1.4 Planificación del Trabajo	7
1.5 Breve resumen de productos obtenidos.....	8
1.6 Breve descripción de los otros capítulos de la memoria	9
2. Usuarios y contexto de uso	10
2.1 Toma de datos cuantitativa de los usuarios	10
2.2 Contexto de uso	13
2.3 Características de la aplicación.....	14
3. Diseño conceptual	16
3.1 Perfil de usuario	16
3.2 Personas modelo	17
3.3 Escenarios de uso.....	19
4. Diseño del prototipo	21
4.1 Árbol de navegación de la aplicación.....	21
4.2 Prototipo de bajo nivel.....	23
4.3 Prototipo de alto nivel.....	26
4.4 Evaluación del prototipo	28
5. Diseño técnico.....	29
5.1 Casos de uso	29
5.2 Modelo de datos.....	31
5.4 Modelo de clases	32
5.5 Navegación y flujo de pantallas.....	33
6. Desarrollo	34
6.1 Arquitectura	34
6.2 Diseño de pantallas.....	38
6.3 Puntos clave.....	39
6.4 Librerías de terceros	40
6.5 Funcionalidades añadidas.....	41
7. Pruebas	44
8. Conclusiones.....	45
9. Glosario	47
10. Bibliografía	48

Lista de figuras

Figura 1 - Planificación del proyecto	8
Figura 2 - ¿Cuántas colecciones tienen?	10
Figura 3 - ¿Cómo organizan sus colecciones?	11
Figura 4 - ¿Qué organizarías?	11
Figura 5 - ¿Dónde usarías principalmente la aplicación?	12
Figura 6 - ¿Qué te gustaría que tuviera?	12
Figura 7 - Fotografía de perfil coleccionista casual [4]	17
Figura 8 - Fotografía de perfil coleccionista experimentado [5]	18
Figura 9 - Árbol de navegación	21
Figura 10 - Prototipo de bajo nivel, menú inicial + lista de categorías	23
Figura 11 - Prototipo de bajo nivel, añadir categoría	24
Figura 12 - Prototipo de bajo nivel, lista de piezas + futuras compras	24
Figura 13 - Prototipo de bajo nivel, añadir pieza	25
Figura 14 - Prototipo de bajo nivel, ver pieza	25
Figura 15 - Prototipo de bajo nivel, buscador	26
Figura 16 - Prototipo de alto nivel, buscador	26
Figura 17 - Prototipos bajo y alto nivel, añadir pieza	27
Figura 18 - Prototipos bajo y alto nivel, lista de piezas y futuras compras	27
Figura 19 - Prototipos bajo y alto nivel, añadir colección	27
Figura 20 - Prototipos bajo y alto nivel, lista de colecciones	27
Figura 21 - Diagrama UML de casos de uso	29
Figura 22 - Modelo entidad-relación	31
Figura 23 - Modelo-vista-controlador	32
Figura 24 - Navegación de pantallas	33
Figura 25 - Diagrama de clases, servicios y modelo	34
Figura 26 - Diagrama de clases, controladores de vistas y navegación	35
Figura 27 - Modelo de datos en Core Data	36
Figura 28 - Storyboard + Controller	37
Figura 29 - Evolución de la pantalla de lista de colecciones	38
Figura 30 - Evolución de la pantalla de lista de piezas	38
Figura 31 - Demostración de la librería MGSwipeTableCell	40
Figura 32 - Captura de código de barras	41
Figura 33 - Internacionalización de la aplicación	42
Figura 34 - Accesos rápidos desde el icono	43

1. Introducción

1.1 Contexto y justificación del Trabajo

Coleccionar es “*reunir un conjunto de cosas de la misma clase o que tienen algo en común y, generalmente, clasificarlas, por afición*” [1]. Muchas personas coleccionan objetos, e incluso tienen más de una colección diferente. Las opciones van desde monedas a folletos de publicidad, cuadros, libros, cartas, figuras, etc. Dentro de cada una de estas categorías se podrían encontrar innumerables variantes. [2]

Un buen coleccionista va ampliando su colección a medida que encuentra piezas que le faltan. No suele ser una afición económica, debido a que se centra en la adquisición de objetos, a menudo difíciles de conseguir. Los coleccionistas suelen ser cuidadosos con sus adquisiciones. También les importa saber que han adquirido con anterioridad, de otro modo pueden acabar pagando, sin pretenderlo, por una pieza que ya poseen.

Existen innumerables métodos para gestionar un inventario. Dependiendo de los requisitos y preferencias de cada coleccionista, y de su colección, podrán optar por un sistema u otro. Habrá el que prefiera tener el inventario reducido a la mínima expresión, tan solo con una breve nota de cada pieza. Por otro lado, habrá el que no le importe ocupar una mayor cantidad de espacio, si así puede documentarse mejor, añadir fotografías, etc.

Teniendo en cuenta el formato para gestionar un inventario, se pueden hacer 3 grupos elementales:

- Lista manual. Es fácil de generar pero dificulta la gestión posterior. Está limitada a la hora de querer aportar detalles, fotos o un orden específico. Su tamaño permite transportarla y consultarla en cualquier lugar.
- Fichas y ficheros. Es fácil de generar, permite añadir datos específicos, fotos y llevar un orden para facilitar las consultas. El tamaño impide que se transporte con facilidad.
- Formato digital. Es fácil de generar y permite añadir todo tipo de características. Facilita la consulta y se puede gestionar desde un *smartphone*. Sus limitaciones solo dependen de la tecnología actual y del equipo del usuario.

Siendo la opción que más se adapta a los tiempos actuales, está claro que el formato digital es el que tiene todas las de ganar en un futuro. Con un potencial solo limitado por la capacidades técnicas del momento, el formato digital será el encargado de gestionar las colecciones de todos los coleccionistas amantes del orden.

Si algo tiene el mundo digital es la libertad que otorga a los creadores de realizar “su versión de todo”. Este concepto no iba a ser menos cierto en este caso. Analizando el mercado se han encontrado diferentes opciones que se analizan a continuación [3]:

Hojas de cálculo

Las hojas de cálculo, o más comúnmente conocidas por el producto Microsoft Excel, son la versión moderna de las listas manuales de toda la vida. El hecho de que hoy en día todo el mundo tenga un ordenador en casa, y que posiblemente ese ordenador tenga un programa de hojas de cálculo, las convierte en un recurso muy utilizado por los coleccionistas, para realizar sus inventarios personales.

Ofrecen las ventajas de las listas de papel, con el añadido de poderlas modificar sin problemas, e incluso añadir multitud de información tanto escrita como en formato gráfico o de imágenes. La única desventaja frente a las lista de papel es la necesidad de un dispositivo digital para consultarlas o editarlas, aunque eso hoy en día no sea un problema debido a la popularidad de los *smartphone*.

Analizando el uso que hacen de ella los coleccionistas se extrae que, aprovechando la capacidad para tener varias pestañas dentro del mismo documento, utilizan estas pestañas para diferenciar entre colecciones de un mismo tipo. Cuando la colección es diferente prefieren crear un documento distinto, así, por ejemplo, tendrían un documento para los cómics, con una pestañas por cada serie o género, y otro para las figuras. Gracias a esto, pueden configurar las columnas de cada pestaña dependiendo de las necesidades de cada colección.

Sobre el papel, su capacidad para introducir cualquier tipo de dato, y en cantidades casi ilimitadas, las presenta como una opción ideal. A la práctica, esa misma flexibilidad las convierte en un formato demasiado genérico, poco atractivo para el usuario que prefiere una herramienta más preparada.

Hobbydb

Hobbydb es una comunidad web de coleccionistas de todo tipo. Con un total de 130 secciones, distribuidas entre 11 categorías como “Automoción y otros vehículos”, “Juegos y puzles”, “Muñecas y figuras” o “Material impreso”, permite a sus usuarios introducir los datos de sus elementos coleccionables, de forma bien organizada.

En el momento que un usuario añade un elemento coleccionable nuevo, pasa a estar disponible para los demás usuarios. Estos podrán consultarlo y añadirlo a su colección, especificando una serie de datos de su pieza en concreto, como es la condición en la que se encuentra, si ha sido restaurado, el precio pagado o donde se adquirió.

Aprovechando que es una comunidad, los usuarios pueden curiosear que piezas tienen los demás usuarios y, en caso de que les interese, marcarlas como piezas a conseguir. Otra de las opciones que añade es un sistema de compra-venta, donde cada usuario puede decidir si ofrece la posibilidad de vender su producto.

Como se ha dicho al principio de la descripción, es una comunidad web, lo que significa que solo está disponible online. No tiene aplicación propia para dispositivos móviles, aunque está adaptada para poder ser visualizada desde ellos.

La cantidad de opciones a la hora de escoger a que grupo pertenece un ítem es abrumadora, aunque siempre puede ocurrir que alguien quiera introducir una pieza, cuya categoría no está planteada, y no pueda hacerlo. Por otro lado, es posible que usuarios prefieran una comunidad más centrada en su tipo de colección, y no una en la que un coleccionista LEGO entre al portal y vea aviones de juguete, botellas de cerveza o pistolas de agua.

Brickset

Brickset es un portal web para aficionados de los sets de la empresa LEGO. A diferencia de hobbydb, el estar centrado en un solo tipo de producto permite tener un listado completo previo de todos los elementos coleccionables. Tampoco tiene opción de compra-venta entre usuarios, ni se puede personalizar el perfil de cada set adquirido.

Entre sus posibilidades está la de marcar si se posee un set, en que cantidad, escribir una valoración personal y marcar si se quiere pero aun no se posee. También ofrece en la descripción de los sets enlaces directos a tres tiendas online, amazon, ebay y bricklink, para poder comprarlos.

En el portal se vanaglorian de tener listados 13263 sets (en el momento de redactar este informe) lo cual se supone que incluye todas las posibilidades, dentro del pequeño mercado en el que se sitúan. Pero hay que tener en cuenta que LEGO es una empresa que sigue sacando novedades, por lo que el portal deberá incluir cada una de estas, para que sus usuarios las puedan seleccionar. Esto podría suponer un problema en el momento que el portal deje de estar actualizado.

Una carencia que puede encontrar un aficionado a LEGO es que le será imposible registrar los sets que monte por si mismo. En caso de que construya algo que no ha salido oficialmente de la casa LEGO, aunque lo haya hecho con las mismas piezas. Tampoco podrá registrar sus piezas en caso de que no colecciona sets como tal. Por ejemplo, un usuario podría coleccionar solo las figuras de personajes de LEGO, o los vehículos que aparecen en los sets, y no podría añadirlos a su colección.

En el apartado técnico hay que destacar que es un portal web, lo que significa que solo está disponible online. No tiene aplicación propia para dispositivos móviles, aunque está adaptada para poder ser visualizada desde ellos.

Boxes

Boxes es una comunidad de coleccionistas con un concepto similar a hobbydb. En este caso, permite a los usuarios subir sus piezas, siguiendo una clasificación que se puede personalizar, y ponerlas a la venta.

La primera vez que el usuario abre la app debe marcar, de una lista de 60 opciones, cuales son sus intereses. En esta lista aparecen conceptos como “cámaras”, “relojes”, “faldas”, “futbol”, “Star Trek” o “sellos”. Gracias a esta selección previa, la aplicación ofrece unos contenidos u otros en la pantalla de inicio. Igualmente, el usuario tiene la opción de buscar otros contenidos, filtrando por popularidad, más vendidos, categoría o directamente con un buscador.

Cuando un usuario está visualizando una pieza de otro, tiene las opciones de indicar “me gusta”, “lo quiero”, “mostrar a mis seguidores”, “mandar un mensaje” o “comprar”. Una opción que ofrece es la de marcar usuarios como “seguidos”, para poder consultar que publican o mandarse mensajes.

En el apartado técnico, solo está disponible en dispositivos móviles mediante una app para iOS y Android. Aun así, la aplicación tiene serias carencias al funcionar offline, como no permitir subir nuevas piezas o consultar otras subidas anteriormente.

Home Contests

Home Content es una aplicación para iOS diseñada para realizar el inventario de una casa. De una manera sencilla, permite añadir elementos y sus características. Por supuesto, también permite consultar los elementos añadidos.

En el momento de añadir un nuevo elemento pregunta en que parte de la casa se encuentra. Una vez seleccionada una opción, permite añadir fotos, categoría, valor, cantidad, modelo, número de serie, etc.

Debido a que está pensada para realizar reportes de daños, la aplicación no se adapta bien a las necesidades de un coleccionista. De todos modos, la interface, las opciones de registro de los elementos y la posibilidad de funcionar completamente offline hacen que sea un buen ejemplo a tener en cuenta.

Tras analizar el mercado, se echa en falta una app sencilla, sin la complejidad de incluir una red social, o un sistema de compras y ventas, con la que un coleccionista pueda gestionar sus piezas.

Se pretende realizar una app enfocada en el concepto de “inventario de colecciones personales”. Esto supone una app que permita al usuario introducir sus colecciones y piezas, añadiendo la información más relevante, para facilitar futuras consultas allá donde vaya. Sin necesidad de conexión a internet y con el potencial de realizar consultas rápidas y sencillas.

1.2 Objetivos del Trabajo

Con la realización de este trabajo se plantean varios retos y objetivos a superar, tanto personales como profesionales:

- Buscar una idea interesante de producto, que pueda ser plasmada en una aplicación móvil.
- Realizar una planificación adecuada para la creación del trabajo, respetando las fechas límite.
- Analizar el mercado en búsqueda de opciones que cubran esta idea o similar, recogiendo que ofrecen y detectando donde fallan o que perfil de usuario no cubren.
- Recoger las necesidades y preferencias de los usuarios.
- Crear un perfil de usuario ideal a cubrir.
- Elaborar un diseño con los requisitos más demandados, sin salirse del concepto propuesto para la aplicación y teniendo en cuenta que perfiles de usuario se quieren cubrir.
- Diseñar una aplicación que ofrezca una experiencia de usuario de calidad, siguiendo los patrones que marca Apple.
- Ampliar mis conocimientos en desarrollos con iOS, partiendo de lo aprendido en la asignatura del semestre anterior.
- Lograr un producto que cumpla con las expectativas, dentro de las fechas propuestas.

Con la aplicación en fase embrionaria, dentro de los objetivos funcionales está lograr que la aplicación permita realizar las siguientes acciones de manera adecuada:

- Crear colecciones nuevas.
- Añadir piezas a cada colección.

- Añadir imágenes de las piezas.
- Añadir información de las piezas.
- Realizar búsquedas de las piezas con palabras clave.
- Consultar las colecciones en cualquier momento.
- Editar los datos introducidos.
- Eliminar colecciones.
- Eliminar piezas.

1.3 Enfoque y método seguido

El trabajo se enfoca en ofrecer a los usuarios una aplicación para *smartphone*, para que puedan organizar sus colecciones de forma personal y privada. El mercado de las aplicaciones móviles tiene varias opciones, pero acostumbran a pecar por su rigidez, o por convertirlo todo en una red social. En esta aplicación, el foco se sitúa en usuarios que solo quieren una aplicación para guardar sus colecciones, sean cuales sean, sin pretensiones de realizar compra-venta o de ver que colecciones tienen otras personas.

Siguiendo el concepto de “inventario de colecciones personales”, la aplicación permitirá al usuario introducir sus piezas, organizadas por colecciones, añadiendo la información más relevante, para facilitar futuras consultas allá donde vayan. Con un funcionamiento totalmente offline y capacidades de búsqueda rápida.

Dentro del apartado técnico, el desarrollo web está descartado por las necesidades tecnológicas de la aplicación. Este tipo de desarrollo no permite funcionalidades offline, ni acceder a elementos del dispositivo como la cámara o el almacenamiento.

Respetando los requisitos, la app se puede desarrollar con tecnologías híbridas, para poder desplegarla sobre cualquier dispositivo. El resultado no sería tan bueno como utilizando las tecnologías del propio dispositivo, pero se ahorraría tener que realizar un desarrollo por cada plataforma que se quisiera soportar.

Desarrollar con el SDK oficial ofrece la mejor experiencia al usuario, y las mejores herramientas al desarrollador. Esto es debido a que ofrece el mayor rendimiento del dispositivo, con el uso de todas las tecnologías, y el *look & feel* esperado, al utilizar los mismos elementos que el sistema. A cambio, se obtiene una aplicación nativa, no exportable hacia otras plataformas.

La aplicación se realizará para iPhone, con el sistema operativo iOS, utilizando el SDK oficial. Se ha escogido esta plataforma por tener intereses personales en ella.

1.4 Planificación del Trabajo

Para la correcta ejecución de este trabajo se utilizarán las siguientes herramientas:

- *Framework Xcode 7.3.1*: Es el *framework* oficial de Apple para realizar aplicaciones nativas para iOS. Solo disponible para Mac OS X.
- Simulator 9.3: Es el emulador de dispositivos iOS oficial de Apple. Solo para Mac OS X.
- Ordenador MacBook Pro con Mac OS X 10.11.14: Necesario para ejecutar los programas indicados para Mac OS X.
- *Smartphone* iPhone 6s con iOS 9.2.1: Se utiliza para realizar pruebas fuera del emulador.
- SourceTree: Gestor de repositorios compatible con GitLab.
- GIMP 2.8: Editor gráfico para crear imágenes.

Las tareas a realizar, con su previsión de horas, son las siguientes:

- Definición de usuarios y requisitos – 24 horas
- Diseñar navegación – 7 horas
- Crear prototipo básico – 11 horas
- Evaluar prototipo básico – 5 horas
- Crear prototipo en alta resolución – 6 horas
- Diseñar la arquitectura de la aplicación – 40 horas
- Configurar el entorno de desarrollo – 2 horas
- Desarrollar:
 - Añadir colección – 12 horas
 - Editar colección – 11 horas
 - Añadir pieza – 13 horas
 - Editar pieza – 17 horas
 - Navegación – 10 horas
 - Mejoras – 10 horas
- Pruebas de la aplicación – 30 horas
- Documentación – 50 horas

A continuación se muestra un diagrama de Gantt con la planificación de las tareas:

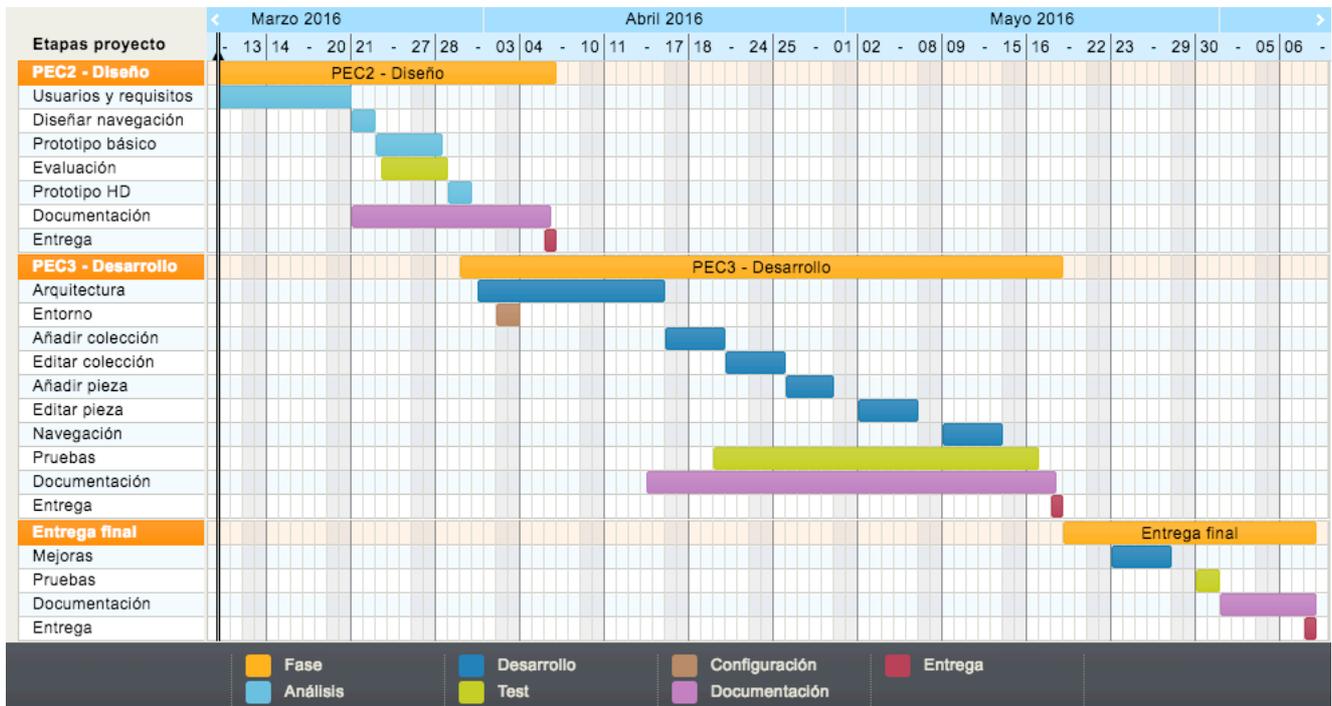


Figura 1 - Planificación del proyecto

La planificación se apoya en la siguiente distribución de horas:

- Fase Diseño y Desarrollo
 - 4 horas de lunes a viernes
 - 1 hora sábados y domingos
- Fase de entrega final
 - 2 horas de lunes a viernes
 - Libres sábados y domingos

1.5 Breve resumen de productos obtenidos

Al finalizar cada una de las 4 fases del trabajo se realizará una entrega acumulativa. Por su carácter evolutivo, es posible que una entrega rectifique algún concepto de una entrega anterior. A continuación se listan las principales características de cada entrega:

- Entrega 1: Siendo la primera entrega, se centra en la presentación, intenciones y la planificación de todo el proyecto.
- Entrega 2: Centrada en la fase de diseño, se centra en el análisis de los usuarios, requisitos, necesidades, diseño de pantallas y navegación. También se desarrolla un prototipo navegable y parte del diseño técnico.

- Entrega 3: Centrada en la fase de desarrollo, se centra en la creación de la aplicación y testeo, siguiendo el diseño de la 2ª entrega.
- Entrega final: Como última entrega, se centra en documentar todo lo que haya quedado fuera en entregas anteriores, junto a la realización de un video de presentación de la aplicación resultante. También se aprovecha para añadir algunas mejoras en el funcionamiento, que hayan quedado fuera de la 3ª entrega.

1.6 Breve descripción de los otros capítulos de la memoria

El capítulo 2, usuarios y contexto de uso, analiza los usuarios, sus necesidades y sus condiciones. En el se analiza una encuesta pública, realizada a usuarios potenciales. Con los datos obtenidos, se extraen los contextos de uso, definiendo las condiciones en las que los usuarios utilizarán la aplicación. Finalmente, se resumen las principales características que deberá tener la aplicación.

El capítulo 3, diseño conceptual, extrae las características de los usuarios, analizadas en el capítulo 2. Describe los perfiles de los usuarios, para los que la aplicación está destinada. Crea personas modelo, basadas en los perfiles, y escenarios de uso, para detectar funcionalidades y necesidades que deban tenerse en cuenta en la aplicación.

El capítulo 4, diseño del prototipo, crea un prototipo de la aplicación. El primer paso es definir la navegación de la aplicación. Después se realiza un prototipo de bajo nivel, teniendo en cuenta la navegación, donde se crean varias pantallas para cada opción. Tras analizar las pantallas y decidir cuales son las más adecuadas, se realiza el prototipo de alto nivel. Este prototipo de alto nivel tiene el aspecto de una aplicación para iOS nativa, y permite una navegación simulada entre las pantallas.

El capítulo 5, diseño técnico, organiza la estructura lógica del proyecto. Muestra los casos de uso de la aplicación. También presenta el modelo de datos, la arquitectura de clases y el árbol de navegación.

El capítulo 6, desarrollo, muestra la representación real de la arquitectura de la aplicación. Se presentan los modelos de clases, datos y navegación desarrollados. También muestra el resultado de las pantallas, puntos clave del desarrollo, librerías de terceros utilizadas y otras mejoras.

El capítulo 7, pruebas, muestra las pruebas que se han realizado para asegurar el correcto funcionamiento de la aplicación.

El capítulo 8, conclusiones, ofrece la visión del estudiante sobre el proyecto completado, valorando la metodología seguida, el proceso de realización y los resultados obtenidos.

2. Usuarios y contexto de uso

Los usuarios de una aplicación son el motivo de su existencia. Cualquier aplicación desarrollada, sin tener en cuenta los usuarios, está destinada al fracaso. Por este motivo, es importante tener en cuenta su punto de vista desde el principio del proyecto.

Tan importante es este punto de vista como usuario, como saber filtrar sus valoraciones como profesional en diseño de aplicaciones. Los usuarios tienen un punto de vista muy valioso hacia posibles mejoras de lo que ya conocen, pero en ocasiones, ni todo lo que piden les será útil, ni todo lo que descartan no les hará falta.

2.1 Toma de datos cuantitativa de los usuarios

Para la toma de decisiones inicial, respecto a que características son más importantes para los usuarios, se ha realizado una encuesta pública a coleccionistas donde se han conseguido 123 participantes [6]. A continuación se muestran una serie de gráficos y datos importantes, extraídos de los resultados:

¿Cuántas colecciones tienen?

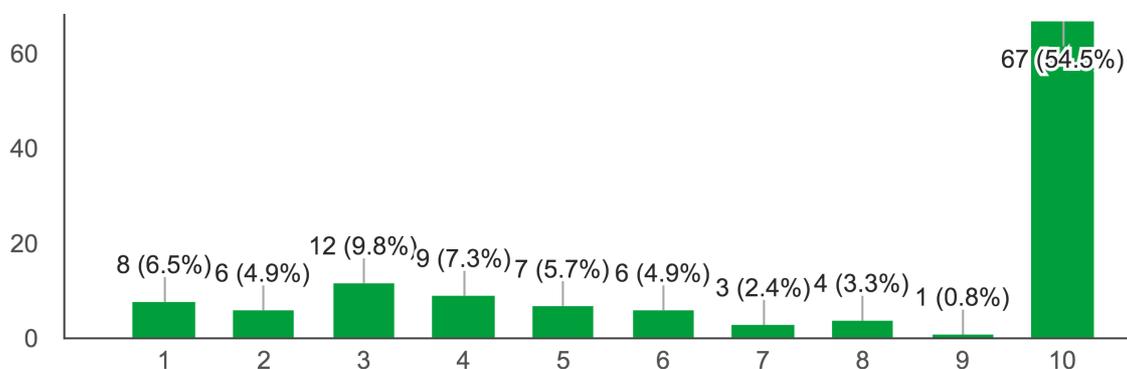


Figura 2 - ¿Cuántas colecciones tienen?

La figura 2 indica que el 71,5% de los usuarios, que han participado en la encuesta, tienen 5 o más colecciones. Siendo un número considerable, 54,5%, los que tienen 10 o más colecciones.

Estos datos dan un valor extra a la encuesta, puesto que indica que los usuarios que han participado son coleccionistas consumados.

¿Quieren organizar sus colecciones?

El 94% pretende organizar al menos el 80% de sus colecciones. Este dato es importante porque demuestra que la propuesta tiene mercado.

¿Como organizan sus colecciones?

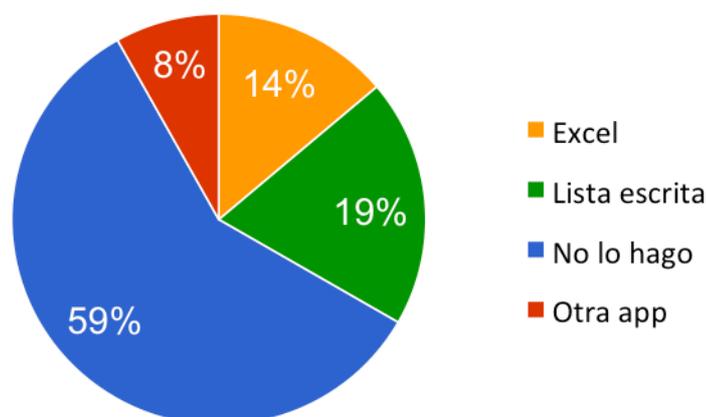


Figura 3 - ¿Como organizan sus colecciones?

Como se ve en la figura 3, solo el 8% utiliza métodos modernos, frente al 33% que aún utiliza métodos tradicionales. Por otro lado, el 59% afirma que no utiliza ninguno.

Este gráfico, junto al punto anterior con un 94% que pretende organizar sus colecciones, deja entrever que solo hay un 8% de usuarios que ha encontrado una aplicación que cumpla sus expectativas. Esto es un dato muy revelador, porque indica que hay un mercado muy grande por explotar, donde ninguna aplicación ha logrado aposentarse ante el gran público.

Filtrando los datos, teniendo en cuenta solo los coleccionistas con 10 o más colecciones, se contempla que la cantidad de usuarios, que utiliza alguna aplicación, prácticamente se dobla. Aún así, un 14% no deja de ser testimonial.

¿Qué organizarías?

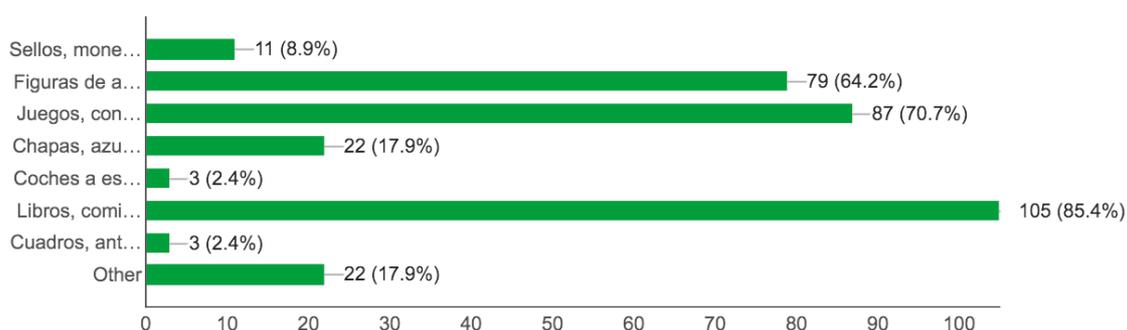


Figura 4 - ¿Qué organizarías?

A la pregunta de que colecciones organizarían, la figura 4 muestra los resultados sobre las opciones propuestas y sus resultados. Se puede ver que las opciones más demandadas son “libros, comics, ...”, “juegos, videoconsolas, ...” y “figuras de acción”, en este mismo orden de preferencia. Estos datos parecen tener sentido, en relación al tipo de contenido que se ha encontrado, subido por usuarios, en páginas y aplicaciones analizadas anteriormente.

¿Dónde usarías principalmente la aplicación?

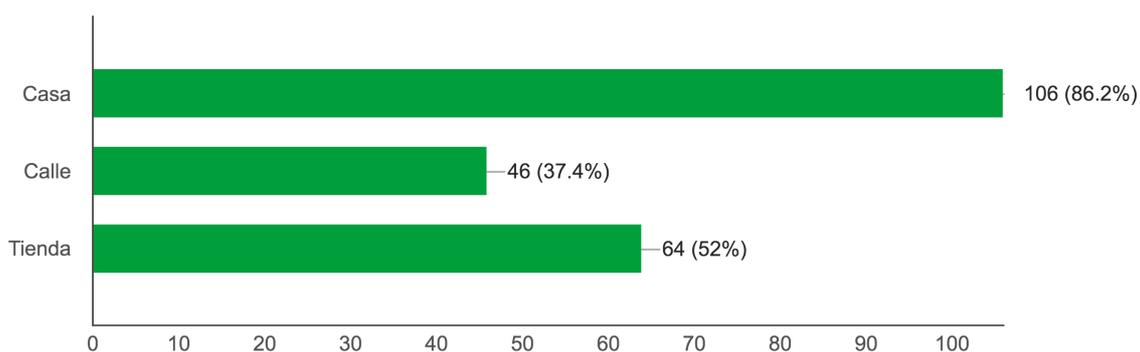


Figura 5 - ¿Dónde usarías principalmente la aplicación?

Buscando información para situar el contexto de uso de la aplicación, la figura 5 muestra que el 86% de los usuarios la utilizarían en casa, un 52% en tiendas y un 37% en la calle. Estos datos reflejan que la intención de los usuarios es, principalmente, consultar y registrar sus colecciones en casa con tranquilidad, y utilizarla de apoyo en las tiendas a la hora de comprar nuevas piezas.

¿Qué te gustaría que tuviera?

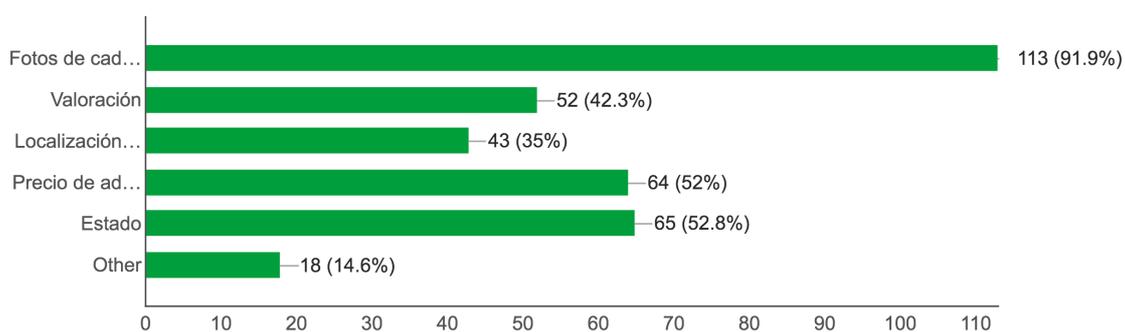


Figura 6 - ¿Qué te gustaría que tuviera?

La figura 6 muestra el interés de los usuarios frente a varias funcionalidades, planteadas para la aplicación. Como prioridad indiscutible, con un 92%, aparece el poder añadir imágenes. Seguido de esta, logrando el interés del 52%, están el precio de adquisición y el estado. Como 4ª opción, con un 42%, aparece poder añadir una valoración. Finalmente, con un discreto 35%, queda la localización de compra de la pieza.

¿Qué es lo más importante para ti?

En el apartado de respuestas libres, con la intención de recibir propuestas no contempladas en la figura 6, los usuarios coinciden mayormente en cinco conceptos. A parte de repetir la necesidad de añadir fotografías, aparecen la necesidad de un buscador, para encontrar rápido las piezas, que la aplicación permita una buena organización, sea ágil de utilizar y contenga una lista de piezas pendientes.

2.2 Contexto de uso

El contexto de uso define las condiciones en las que el usuario utilizará la aplicación. En una aplicación móvil es especialmente importante, debido a que las características de la plataforma permiten un sin fin de posibilidades. En caso de no tenerlo en cuenta, el diseño y la experiencia de usuario de la aplicación no se adaptarán a las necesidades del usuario. No es lo mismo plantear una aplicación que se va a utilizar en casa, de vacaciones en una ciudad desconocida o conduciendo por la carretera.

Un método para desarrollar el contexto de uso es tratar de responder tres preguntas básicas: ¿Dónde, cuándo y en que entorno?

¿Dónde la utilizará el usuario?

El lugar marca los recursos, comodidad, etc. En lo que a este proyecto respecta, siguiendo la gráfica de la figura 5, la aplicación se utilizará sobretodo en casa, pero también en tiendas. Esto proyecta varios conceptos importantes:

- El 86% de los usuarios pretenden utilizarla en casa. Sin lugar a dudas se plantea el mejor lugar para añadir piezas y consultarlas con tranquilidad.
- El 52% de usuarios pretende utilizarla en las tiendas y el 37% en la calle. Sin duda, como herramienta de consulta, parece lo más razonable a la hora de consultar y comparar con otras piezas. Para que sea posible, la plataforma ideal es un teléfono inteligente, en contraposición de una tablet o un ordenador.
- En una tienda se deben poder realizar consultas con rapidez. La aplicación ha de ser ágil y estar bien organizada.
- Tanto una casa como las tiendas son lugares cerrados, sin necesidad de estar en movimiento, por lo que no es necesario utilizar una interface excesivamente simplista.

¿Cuándo la utilizará el usuario?

El momento de uso influye en la dedicación del usuario con la aplicación. En este caso, supone varias cosas:

- Cada pieza nueva se registra de forma individual. Esto se hace con la intención de que la ficha quede lo más completa posible. Puede durar algo más de un minuto por pieza, dependiendo de cada usuario. Por esto, es de esperar que no se realice durante otra actividad.
- A la hora de hacer consultas, puede estar comparando productos, con los amigos mostrando sus adquisiciones, viendo la televisión, etc. En esta caso el tiempo de uso será reducido y con distracciones.

¿En que entorno la utilizará el usuario?

El entorno determina la interacción del usuario con el dispositivo, en cuanto a sus posibilidades tecnológicas, sociales y ambientales.

- Realizando una consulta en una tienda es probable que tenga alguna mano ocupada, por lo que la aplicación ha de ser funcional en vertical.
- Pueden ser lugares privados o públicos, por lo que hay que tener cuidado con los datos que se muestran por pantalla. También hay que tener en cuenta la posibilidad de pérdida del dispositivo, por lo que es recomendable no incluir datos sensibles que no sean necesarios.
- En casa se dispone de todos los recursos, y parece el lugar ideal donde dedicarse a añadir las colecciones a la aplicación, con comodidad. En este proyecto la aplicación no requiere conexión a internet, pero de ser así, en una posible ampliación, sería importante tenerlo en cuenta.
- En contraste con el punto anterior, en la calle no se dispone de todos los recursos. En este proyecto, al no requerir conexión a internet, no afecta al funcionamiento, pero en caso de que se guardasen las colecciones en un servidor, se podría ver afectado el funcionamiento. El usuario debe poder realizar consultas tenga o no conexión en ese momento, y eso debería ser prioritario. De requerir conexión a internet, se estaría entorpeciendo el uso a quienes no tuviesen.

2.3 Características de la aplicación

Teniendo en cuenta el análisis de los datos, aportados por un grupo de usuarios potenciales en el punto 2.1, y los contextos de uso, analizados en el punto 2.2, se ha creado la siguiente lista, donde se resumen las principales características que deberá tener la aplicación para satisfacer a los usuarios:

- Funcionamiento en teléfonos inteligentes.
- Distribución organizada.
- Navegación ágil con una mano.
- Permitir añadir imágenes, precio, estado y valoración de las piezas.
- Buscador rápido de piezas registradas.
- Diseño polivalente, no centrado en un tipo de colección.
- Permitir uso continuado e intermitente.
- Lista rápida de piezas deseadas.

También se ha realizado una selección de los datos que el usuario podrá incluir en la aplicación, respecto a las colecciones y piezas que añada. Esta selección se ha realizado teniendo en cuenta las peticiones, que los usuarios han hecho en las encuestas.

Por colección:

- Nombre (título)
- Serie (subtítulo)
- Si está completa

Por pieza:

- Nombre
- Lo tengo / no lo tengo
- Fotos
- Valoración personal
- Estado
- Si está completa
- Si tiene la caja
- Autor
- Donde se compró
- Fecha de salida
- Tipo
- Número en la serie
- Tamaño de la serie
- Precio
- Donde la guarda
- Notas

3. Diseño conceptual

El diseño conceptual pretende converger los datos obtenidos, mediante análisis de requisitos, usuarios y escenarios, y extraer los perfiles de usuario, personas modelo y escenarios de uso de la aplicación.

3.1 Perfil de usuario

Los perfiles de usuario son una generalización del tipo de usuario para los que la aplicación está destinada. Se crean buscando los rasgos comunes más importantes. Para esta aplicación se pueden agrupar en dos tipos: “Coleccionista casual” y “Coleccionista experimentado”.

Coleccionista casual

- Edad: media-baja .
- Posición social: estudiante / trabajador de perfil medio-alto, interesado en ocio cultural como lectura, cine, eventos, videojuegos, etc.
- Tamaño de la colección: mediana.
- Que colecciona: tiene alguna colección que va ampliando con piezas que encuentra de su interés. No suele hacerse con todas las piezas de una colección, prefiere tener las que más le interesan de las colecciones que descubre.
- Como colecciona: su círculo social y su ocio le llevan a visitar lugares donde encuentra piezas que le pueden interesar. En caso de desear una con mucho interés, puede dedicar tiempo libre a encontrarla en alguna tienda online.
- Conocimiento y uso de tecnologías móviles: medio-alto.

Coleccionista experimentado

- Edad: media-alta.
- Posición social: trabajador de perfil medio-alto, con familia, interesado en ocio cultural como lectura, cine, eventos, videojuegos, etc.
- Tamaño de la colección: grande.
- Que colecciona: tiene varias colecciones iniciadas que va ampliando con todo tipo de piezas que encuentra. Si encuentra otras colecciones interesantes pasará a seguirlas.
- Como colecciona: invierte parte de su tiempo libre buscando piezas nuevas para su colección. No pierde la ocasión de acercarse a una feria o un mercadillo en búsqueda de piezas que le faltan.
- Conocimiento y uso de tecnologías móviles: medio.

3.2 Personas modelo

Con los perfiles de usuario definidos, es interesante crear personas modelo, basadas en los rasgos descritos en los perfiles. Estas personas deben tener definidas características personales concretas, para hacerlas más reales. Esto ayuda a meterse en su piel y experimentar como la aplicación se puede adaptar a su realidad. Aprovechando que se han creado dos perfiles de usuario, a continuación se presenta una “persona modelo” siguiendo cada uno de ellos.

Andrea Marlet

Perfil: coleccionista casual

Edad: 23 años

Actividad laboral: estudiante de psicología

Residencia: Calella

Estado civil: soltera



“Me encantan los cómics y las figuras de mis personajes favoritos, pero odio perderles la pista y no saber ni que tengo”

Figura 7 - Fotografía de perfil coleccionista casual [4]

Objetivos

- Guardar un perfil de cada pieza con fotos
- Organizar las piezas a su manera
- Añadir comentarios a las piezas
- Valorar las piezas según le hayan gustado
- Incluir el precio que ha pagado

Comportamientos

- Lleva el teléfono inteligente a todas partes
- Comparte con sus amigos sus nuevas adquisiciones
- Como estudiante, no tiene mucho dinero y suele buscar ofertas
- Prefiere algo bonito y sencillo que algo complejo
- Acude a eventos relacionados con la cultura pop japonesa

Necesidades

- Una interface sencilla y amigable
- Consultar de forma rápida si ya tiene alguna pieza
- Consultar donde compró cada pieza
- Un campo de anotaciones en los perfiles de las piezas

Antonio Pueyo

Perfil: coleccionista experimentado

Edad: 36 años

Actividad laboral: editor

Residencia: Barcelona

Estado civil: mujer y una hija



“Desde pequeño me encantan los videojuegos, mi reto es lograr tener todas las videoconsolas, de todos los colores y versiones que existan”

Figura 8 - Fotografía de perfil coleccionista experimentado [5]

Objetivos

- Organizar las piezas por colecciones
- Controlar que piezas tiene
- Controlar que piezas está buscando
- Valorar de forma extensa el estado de sus piezas
- Marcar si una colección está completa

Comportamientos

- No pierde la oportunidad de entrar en una tienda a mirar si hay algo que le interese
- Clasifica cada pieza en un lugar concreto
- Es exigente con la documentación de sus piezas
- Utiliza su teléfono móvil lo justo y necesario fuera del trabajo
- No se va a dormir sin echar un vistazo en ebay
- Acostumbra a reemplazar piezas en mal estado
- Considera que una pieza no está completa si no tiene la caja

Necesidades

- Acceso ágil al perfil de las piezas
- Buscar una pieza por descripción, notas, autor, etc.
- Consultar el estado de una pieza en concreto
- Listas ordenadas
- Consultar donde compró y guardó cada pieza
- Marcar si tiene piezas repetidas

3.3 Escenarios de uso

Crear escenarios de uso, donde introducir a las personas modelo, es muy importante. Esto ayuda a ponerse en la piel del usuario a la hora de utilizar la aplicación. Creando una experiencia ficticia, donde su entorno y sus necesidades los hagan utilizar la aplicación, se pueden detectar carencias y posibilidades desconocidas. También es muy importante a la hora de realizar el diseño de la aplicación, porque ayuda a comprender mejor las necesidades del usuario.

Con cada persona creada, en el punto 3.2, se han planteado dos escenarios de uso.

Andrea Marlet

De compras, uno para hoy y otro para otro día

Llega el sábado y Andrea ha quedado con sus amigos, para ir a su tienda favorita. Al llegar ve dos figuras que le encantan. Todo y ser del mismo personaje, son bastante diferentes. Consulta los precios y se da cuenta de que tendrá que escoger una.

Para decidirse, prefiere comparar con las figuras que ya tiene y escoger la que menos se parezca. Para ello abre el buscador de la aplicación, introduce el nombre de la figura y consulta los perfiles creados anteriormente. Es un poco indecisa así que prefiere comentarlo con sus amigos, que se van pasando su teléfono para ver los perfiles y ayudarla a comparar.

Una vez se ha decidido, decide crear un perfil a la que no va a comprar, para marcarla como “futura compra” y recordar donde la vio, que precio tenía y como era.

En casa, novedades

Hoy ha sido un gran día, Andrea ha encontrado varias figuras de un nuevo personaje. Nunca antes había adquirido nada de el y está entusiasmada. Los observa con detenimiento mientras piensa donde las va a colocar.

Parece que ya lo tiene claro, pero antes quiere crearles un perfil en la aplicación, así luego ya no las tendrá que mover. Abre la aplicación pero se encuentra con que, al ser diferentes a las que ya tiene, no encajan en ninguna categoría. Por suerte no es ningún problema, la aplicación le permite crear una nueva categoría. Una vez creada, ya puede crear perfiles a sus nuevas figuras.

Sus padres la llaman para poner la mesa y aún le quedan perfiles por crear. Por suerte, no hay ningún problema, acaba el último que tenía a medias y ya hará los demás cuando pueda.

Antonio Pueyo

De compras, renovando y ampliando

Antonio ha decidido ir a una pequeña tienda que le han recomendado. Con una colección tan extensa como la suya, pone en duda que vaya a encontrar nada que le interese. Una vez allí ve que, en parte, tiene razón, pero algo le llama la atención. Encuentra una pieza antigua en muy buen estado, el precio es algo alto pero le sorprende su buen estado.

Al cogerla se da cuenta de que ya la tiene. Sin embargo, recuerda que el estado de la suya es cuestionable. Abre la aplicación, consulta sus fotografías, ve que en sus notas añadió que no encendía, y decide que es el momento para renovarla.

A la hora de pagar, aprovecha para preguntar al tendero si tiene alguna de las piezas que tiene en el apartado de "futuras compras". Para su sorpresa, hay una pieza que sí tiene. Parece que ha sido un día redondo, Antonio podrá sacarla de ese apartado y rellenar el perfil con los datos de su propia pieza.

En casa, la pieza perdida

Antonio tiene un problema, ha prometido que enseñará a un amigo una de sus pertenencias pero no recuerda donde la puso. Debido a que su colección es muy grande, tiene piezas en varios sitios de la casa, repartidas en estanterías y cajas.

La idea de empezar a abrir cajas no le convence. Por suerte suele apuntar donde las guarda, en el perfil creado en la aplicación. Antonio tarda unos segundos en encontrar su pieza, consultar donde la guardó e ir directo a por ella. Parece que no faltará a su palabra y se la podrá enseñar a su amigo.

Cuando su amigo ve la pieza se le abren los ojos. Parece que también quiere una. Antonio, como buen amigo, consulta el perfil y le comenta donde la compró y por que precio. Su amigo crea un perfil en su propia aplicación, en el apartado de "futuras compras".

4. Diseño del prototipo

Un buen prototipo tiene que cubrir las necesidades detectadas durante el proceso de diseño conceptual. Su diseño formará las bases sobre las que se sustentará la aplicación, por ello debe realizarse teniendo en cuenta los datos analizados de los usuarios. Durante este proceso es importante realizar evaluaciones, donde grupos de expertos y/o usuarios valorarán si las decisiones son correctas, o hay que cambiar algo. El proceso de diseño de un prototipo consta de 3 partes.

1. Árbol de navegación de la aplicación
2. Prototipo de bajo nivel
3. Prototipo de alto nivel

4.1 Árbol de navegación de la aplicación

El árbol de navegación es un esquema que representa, mediante nodos y líneas, las pantallas y sus principales funciones. Se utiliza para organizar, de forma sencilla, la distribución de funciones y pantallas de la aplicación. Gracias a su sencillez, es muy fácil recrear mentalmente la navegación en la aplicación, analizarla, evaluarla y encontrar mejoras.

El árbol de navegación, diseñado para este proyecto, se muestra en la figura 9.



Figura 9 - Árbol de navegación

La distribución de las funciones ha sido realizada, buscando la máxima agilidad en su uso. La intención es que la experiencia de usuario sea agradable y

completa. Durante el diseño, se ha tenido contacto con coleccionistas con bastante experiencia, para evaluar todas las opciones y conseguir el mejor resultado.

Teniendo en cuenta el árbol, es importante comprender la función de cada nodo, antes de pasar a diseñar las pantallas:

- Menú inicial (0)

Representa la pantalla inicial con las opciones de menú.

- Lista de categorías (1)

Muestra el listado de categorías, creadas por el usuario. Permite seleccionarlas, para explorarlas (1.1), y crear nuevas (1.2).

- Añadir categoría (1.1)

Permite crear una categoría, añadiendo nombre y descripción.

- Lista de piezas (1.2)

Muestra el listado de piezas, dentro de la categoría padre, creadas por el usuario. Permite seleccionar, para explorar (1.2.2), y crear nuevas (1.2.1).

Inicialmente se planteó un sistema de anidado de categorías, donde cada categoría pudiese tener varias categorías, a parte de piezas. Finalmente se descartó, por la confusión que se podría generar al permitir crear tantos sub-apartados. En ocasiones, dar demasiadas opciones a los usuarios juega en su contra, ya que les permite crear estructuras complejas que a la larga son complejas de utilizar.

- Añadir pieza (1.2.1)

Permite crear una pieza, añadiendo nombre, descripción, fotografía, etc. La propiedad “futura compra” estará desactivada por defecto.

- Ver pieza (1.2.2, 2.2, 3.1)

Muestra la información de una pieza y permite editarla.

- Lista de futuras compras (2)

Muestra el listado de piezas que tienen marcada la opción “futura compra”. Permite seleccionarlas, para explorarlas (2.1), y para crear nuevas (2.2).

- Añadir pieza (2.1)

Como el punto 1.2.1, permite crear una pieza. La principal diferencia es que la propiedad “futura compra” vendrá activada, por defecto.

- Buscador (3)

Permite realiza una búsqueda rápida, sobre todas las piezas almacenadas, y las muestra en una lista. Permite seleccionarlas para explorarlas (3.1).

- Información (4)

Muestra información diversa de la aplicación.

4.2 Prototipo de bajo nivel

Con el árbol de navegación definido ya se pueden diseñar las pantallas. El prototipo de bajo nivel presenta las pantallas en bocetos, fáciles de crear y modificar. Una buena práctica es hacerlos pasar por un proceso de “evaluación-rediseño” iterativo, hasta encontrar el diseño ideal.

Se ha diseñado una pantalla por cada nodo del árbol de navegación. Cada pantalla plasma la esencia de su funcionamiento, sin centrarse en los detalles, como iconos o texto final. Durante el proceso de diseño de las pantallas han surgido varias opciones, que se pueden ver en el anexo sobre el prototipo de bajo nivel [7]. Estas opciones han sido evaluadas y revisadas hasta dar con un diseño óptimo.

Por coherencia, las pantallas con funcionamiento similar comparten diseño. A continuación se muestran los diseños finales de las pantallas.

Menú inicial (0)

El diseño escogido para la pantalla de “menú inicial” es el que aparece en la figura 10. Frente a otras opciones, se ha optado por un menú de navegación inferior permanente, que agrupa las 4 ramas. Esta elección permite al usuario saltar entre categorías en cualquier momento. En el ejemplo, se puede ver como, en el espacio restante, aparece la pantalla correspondiente al menú seleccionado.

El motivo principal ha sido buscar armonía en el funcionamiento de la aplicación. Para facilitar la navegación al usuario, se deben situar las opciones de forma coherente. Es importante, para una buena experiencia de usuario, no confundir al usuario situando opciones similares en lugares diferentes.

Otra ventaja que ofrece este diseño es que, libera completamente la barra superior de navegación. De esta manera, se puede destinar a contener opciones propias de la pantalla, que se esté mostrando en ese momento.

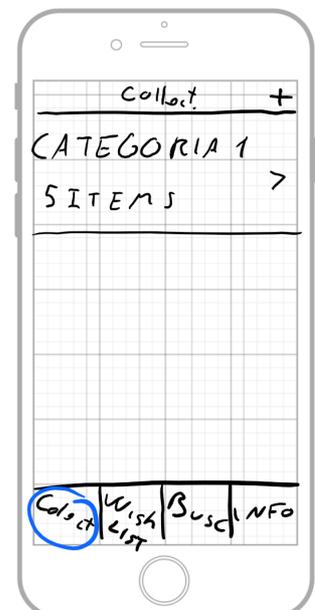


Figura 10 - Prototipo de bajo nivel, menú inicial + lista de categorías

Lista de categorías (1)

El diseño de la “lista de categorías” aparece en la figura 10. Esta pantalla muestra una lista, con las categorías creadas, que enlaza con la pantalla de “lista de piezas” (1.2). Cada fila muestra el título, un subtítulo opcional y la cantidad de piezas que contiene cada categoría.

El extremo derecho, de la barra de navegación, muestra la opción de crear nueva categoría. Esta acción enlaza con la pantalla de “añadir categoría” (1.1). Con la intención de homogeneizar y potenciar la experiencia de usuario, todas las pantallas, que lo requieran, utilizarán ese mismo espacio para las acciones de añadir o editar.

Inicialmente se planteó la opción de poder añadir piezas a la pantalla de categorías. También se planteó poder anidar categorías. Estas opciones se descartaron, en pro de simplificar la navegación de la aplicación. Durante el proceso de evaluación, se vio que un usuario, intentando organizar al extremo su colección, podría acabar formando una maraña de categorías que la harían difícil de utilizar y mantener. Como alternativa, se ha preferido ofrecer la posibilidad de añadir un subtítulo a la categoría.



Figura 11 - Prototipo de bajo nivel, añadir categoría

Añadir categoría (1.1)

El diseño de la pantalla de “añadir categoría” aparece en la figura 11. No adquiere ninguna complejidad, más que permitir completar los datos que se decidan para cada categoría. Su contenido final dependerá de los datos que tenga que incluir.

Lista de piezas (1.2) y futuras compras (2)

El diseño de la “lista de piezas” es compartido con la pantalla de “futuras compras”. Aunque con diferente contenido y origen, ambas pantallas muestran una lista de piezas, que permiten explorar, y añadir de nuevas.

El diseño compartido aparece en la figura 12. Muestra la lista de piezas creadas, y enlaza con la pantalla de ver pieza, (1.2.2) en el caso de la lista de piezas, o (2.2) en el caso de lista de futuras compras.

Cada fila muestra el título, una fotografía y algún dato más por determinar. El extremo derecho, de la barra de navegación, muestra la opción de crear nueva pieza. Por coherencia en el diseño, todas las pantallas sitúan en la misma posición esta opción.

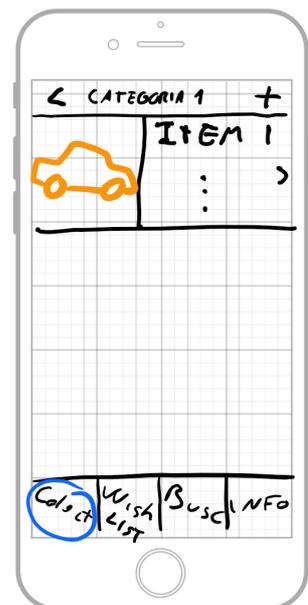


Figura 12 - Prototipo de bajo nivel, lista de piezas + futuras compras

Inicialmente, se valoró diseñar las filas de la lista con imágenes de mayor tamaño, para lograr un aspecto más atractivo. Durante la evaluación esta opción fue desestimada. De hacerse, generaría una lista muy extensa, haciendo más costoso desplazarla, en caso de tener muchas filas.

Añadir pieza (1.2.1)(2.1)

La pantalla de “añadir pieza” muestra los campos a completar, para crear un perfil de una pieza. Este diseño, que aparece en la figura 13, es compartido por todas las pantallas con la misma funcionalidad.

La pantalla muestra un selector de “lo tengo / quiero”. Esta opción será la responsable de que el perfil aparezca, en la lista de futuras compras, o no lo haga. Dependiendo de cómo se llegue a esta pantalla, esta opción estará marcada por defecto con uno u otro valor.



Figura 13 - Prototipo de bajo nivel, añadir pieza

Como se puede observar, la pantalla muestra un componente grande, de tipo carrusel, para añadir las imágenes. Este bloque se ha diseñado así para darle un aspecto atractivo y moderno a la pantalla.

Durante la evaluación se valoró un diseño más discreto y funcional. En el los campos eran más sencillos y ordenados, incluyendo el de las imágenes. Finalmente, ese diseño se descartó, a favor de uno más similar al diseño de la pantalla de “ver pieza”.

Ver pieza (1.2.2)(2.2)(3.1)

El diseño de la pantalla de “añadir pieza”, que aparece en la figura 14, es compartido por todas las pantalla destinadas al mismo hecho. Igual que en anteriores casos, el motivo es armonizar la experiencia de usuario.

Como se puede observar, la pantalla comparte la estética de la pantalla de “añadir pieza” (figura 13). En este caso, los campos no informados no aparecerían, para presentar una pantalla más limpia.

La pantalla permitirá la opción de editar, mediante un botón situado en el extremo derecho de la barra de navegación. De esta manera se mantiene la norma, que sitúa en este espacio las acciones de añadir y editar.

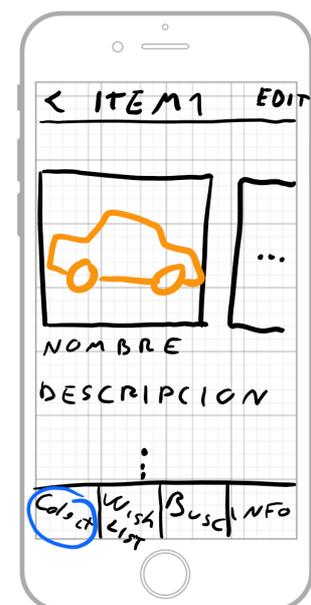


Figura 14 - Prototipo de bajo nivel, ver pieza

Buscador (3)

La pantalla de buscador hereda el diseño de las pantallas de “lista de piezas” y “lista de futuras compras”, con un añadido. Para permitir la búsqueda, se sitúa una barra de búsqueda debajo de la barra de navegación superior, como se puede ver en la figura 15.

La lista de piezas aparecerá como resultado de la búsqueda, teniendo el mismo aspecto que en la figura 12.

Información (4)

La pantalla de información no sigue ningún diseño concreto. Su función es mostrar algunos datos de la aplicación, como la cantidad de piezas o la licencia de uso.

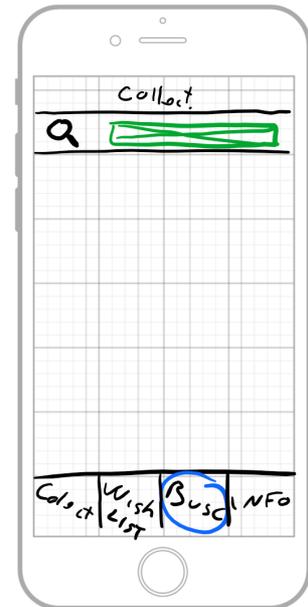


Figura 15 - Prototipo de bajo nivel, buscador

4.3 Prototipo de alto nivel

Con el prototipo de bajo nivel terminado, ya se puede realizar el prototipo de alto nivel. En comparación con el prototipo de bajo nivel, este representa una alta fidelidad en su diseño. Como ejemplo, se pueden comparar la figura 16 y 15, que representan la pantalla del buscador, en alto y bajo nivel respectivamente.

Los objetos de las pantallas están creados con elementos reales, iguales, o muy similares, a los que pone a disposición la arquitectura del dispositivo móvil. Al ser un desarrollo par iOS, se han utilizado librerías que simulan el aspecto de este entorno.

El diseño de las pantallas se ha realizado con la herramienta *justinmind*. Esta herramienta ofrece una gran cantidad de objetos, que simulan el aspecto y funcionamiento de objetos reales, y se adapta al entorno escogido. Otra de las virtudes de esta herramienta es que permite generar una navegación, entre pantallas, para simular el uso de la aplicación real. Esta navegación se puede compartir online o exportar, para abrir con un navegador.

Aprovechando las posibilidades de *justinmind*, se ha creado un proyecto que simula la navegación de la aplicación. Este proyecto se puede encontrar en el anexo 3 [8], junto a un documento que analiza cada una de las pantallas diseñadas.

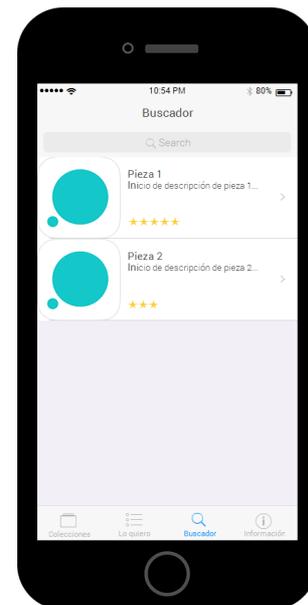


Figura 16 - Prototipo de alto nivel, buscador

Para poder comprobar la evolución entre el prototipo de bajo nivel y de alto nivel, a continuación, las figuras 17, 18, 19 y 20 muestran una comparación de las pantallas, anteriormente descritas en apartado 4.2 Prototipo de bajo nivel, con sus homólogas de alta resolución.



Figura 20 - Prototipos bajo y alto nivel, lista de colecciones



Figura 19 - Prototipos bajo y alto nivel, añadir colección

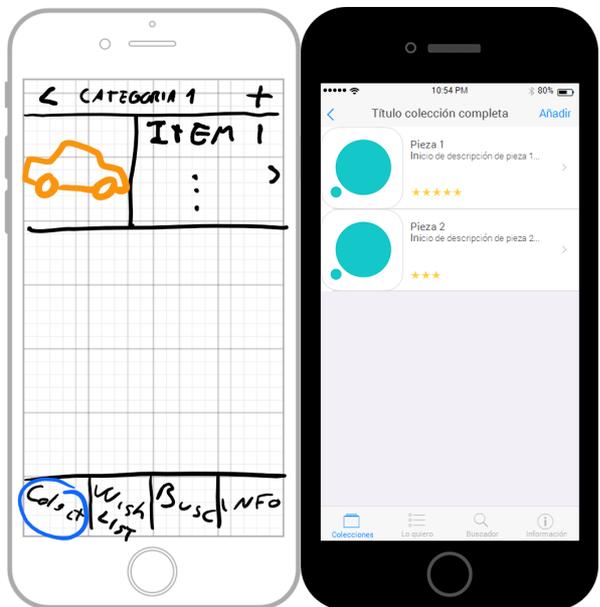


Figura 18 - Prototipos bajo y alto nivel, lista de piezas y futuras compras



Figura 17 - Prototipos bajo y alto nivel, añadir pieza

4.4 Evaluación del prototipo

Durante el proceso de diseño del prototipo, se han realizado iteraciones de “progreso – evaluación”. La navegación y el diseño de las pantallas han sido evaluados, a medida que se iban desarrollando, para lograr un mejor resultado.

Como se ha informado en varios apartados, las pantallas han ido sufriendo modificaciones, teniendo en cuenta la valoración de varios coleccionistas. Analizando de forma independiente sus aportaciones, se puede detectar el efecto que han causado, en el diseño de las pantallas y la navegación. A continuación se muestran varias de estas aportaciones, en un formato de “**valoración**”: resolución.

- **“No complicar la navegación” y “No crear un exceso de sub-apartados”**: estas valoraciones dieron pie a replantearse si era recomendable permitir anidar categorías. Como ya se ha visto en el apartado de navegación, el resultado fue limitar a un nivel de categorías.
- **“Mostrar listas pequeñas, sin imagen o con imagen pequeña para poder buscar mejor”**: el diseño inicial de las listas de piezas mostraban imágenes grandes, del mismo estilo de instagram. Esta valoración hizo replantearse esta opción. Es cierto que el diseño visual era más atractivo, pero no era una buena opción si iba a complicar la navegación. Finalmente se optó por añadir una imagen pequeña en cada fila, reduciendo sensiblemente el grosor de estas.
- **“Añadir un buscador independiente”**: inicialmente el buscador estaba en cada categoría. Esta valoración hizo replantearse la navegación, y darse cuenta de que un buscador alternativo y global es mucho más útil.
- **“Es más importante saber si una colección está completa que no cuantas piezas tiene”**: inicialmente, en la lista de colecciones se mostraba el número de piezas que incluía cada colección. Como información puede ser interesante, pero parece ser bastante más importante saber si está completa. De otra manera, si no se sabe cuantas piezas la componen, es difícil saber si está completa o no.

Otro punto importante, a la hora de evaluar el diseño, ha sido la posibilidad de crear una navegación entre pantallas en el prototipo de alto nivel. Llegados a este punto, con todas las pantallas diseñadas, detectar un problema con la navegación supondría rectificar el árbol de navegación y el prototipo de bajo nivel, pero siempre es preferible a encontrarlo en la fase de desarrollo.

En el caso de este proyecto, poder probar la navegación, sobre el prototipo de alta resolución [8], ha ayudado a comprobar que el planteamiento había sido el correcto, y que la aplicación es sencilla de manejar.

5. Diseño técnico

Una vez finalizado el prototipo ya se aprecia que aspecto, y funcionalidad, tendrá la aplicación, pero aún queda mucho por recorrer hasta obtener una versión funcional. Un prototipo no es más que una representación visual, carente de funcionamiento interior. El diseño técnico organiza la estructura lógica del proyecto, para sentar las bases en las que se erigirán sus capacidades reales.

5.1 Casos de uso

Los casos de uso son las funcionalidades que la aplicación ofrece, ejecutables por uno o varios actores. En este caso, el actor es el usuario de la aplicación, pero sería cualquier ente externo, que tuviese acceso a estas funcionalidades. Son importantes tenerlos presentes para comprobar, de manera visual y sencilla, las relaciones entre las funcionalidades del sistema y el usuario.

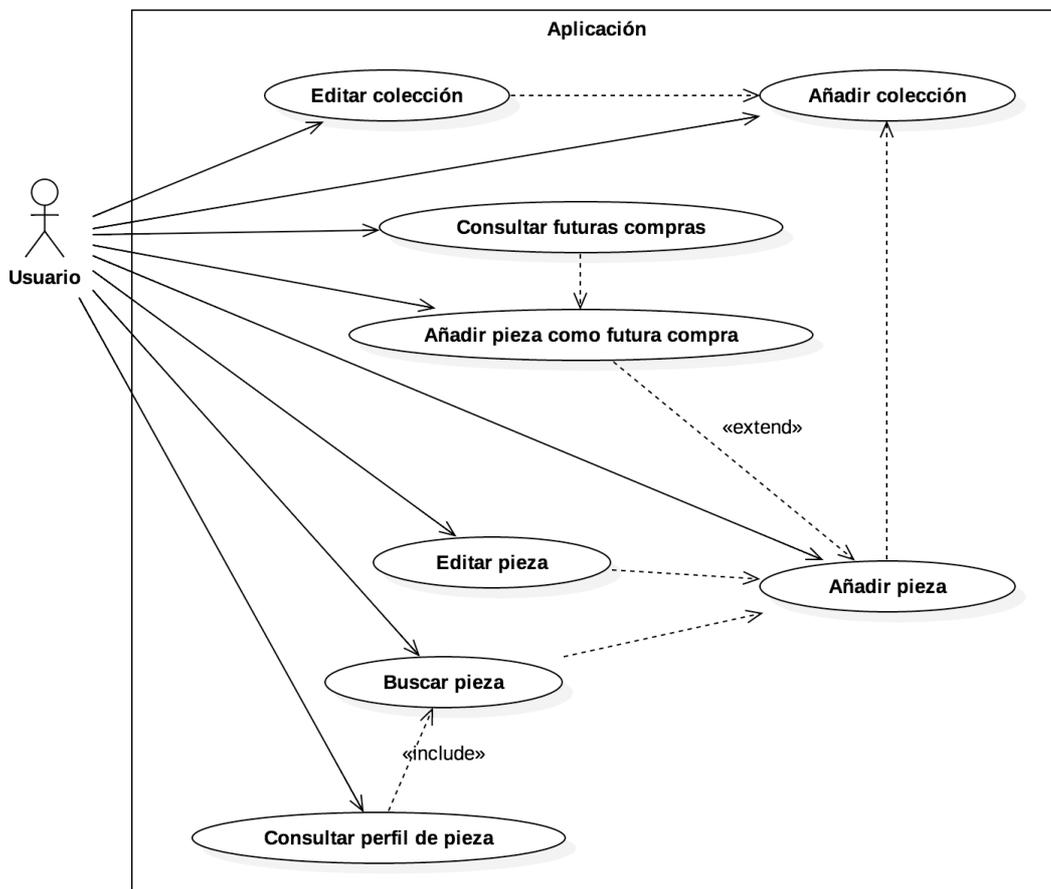


Figura 21 - Diagrama UML de casos de uso

La figura 21 muestra el diagrama de casos de uso de la aplicación, donde se ven los casos de uso y sus relaciones. Estos son descritos brevemente en las siguientes líneas.

Añadir colección

Crear una colección nueva.

- Precondiciones: Que no exista una colección con los mismos datos.
- Postcondiciones: Hay una colección nueva en la que añadir piezas.

Editar colección

Cambiar los atributos de una colección existente.

- Precondiciones: Que se haya creado alguna colección previamente.
- Postcondiciones: La colección editada tiene otros atributos.

Añadir pieza

Crear un perfil de pieza nueva, dentro de una colección existente.

- Precondiciones: Que se haya creado alguna colección previamente.
- Postcondiciones: Hay un perfil de pieza nuevo.

Añadir pieza como futura compra

Crear un perfil de pieza nueva, dentro de una colección existente, y marcarla como futura compra.

- Precondiciones: Que se haya creado alguna colección previamente.
- Postcondiciones: Hay un perfil de pieza nuevo marcado como futura compra.

Editar pieza

Cambiar los atributos del perfil de una pieza existente.

- Precondiciones: Que se haya creado alguna pieza previamente.
- Postcondiciones: El perfil de la pieza editada tiene otros atributos.

Buscar pieza

Buscar el perfil existente de una pieza.

- Precondiciones: Que se haya creado alguna pieza previamente.
- Postcondiciones: Se muestra un listado con las piezas que coinciden con la búsqueda

Consultar perfil de pieza

Abrir el perfil existente de una pieza.

- Precondiciones: Que se haya creado alguna pieza previamente
- Postcondiciones: Se muestra el perfil de la pieza seleccionada.

Consultar futuras compras

Mostrar un listado de perfiles de piezas existentes, marcadas como “futuras compras”.

- Precondiciones: Que se haya creado alguna pieza, marcada como futura compra, previamente.
- Postcondiciones: Se muestra un listado de piezas por comprar.

5.2 Modelo de datos

Un modelo de datos describe, de forma abstracta, la estructura de datos de un proyecto. Una manera de representarlo es mediante el “modelo entidad-relación”. Como se muestra en la figura 22, con los datos correspondientes a este proyecto, el modelo entidad-relación representa en un diagrama las relaciones de las entidades, que formarán parte de la aplicación. Se ha creado teniendo en cuenta los campos que, en el punto 2.3 (características de la aplicación), se han definido para formar parte de la aplicación.

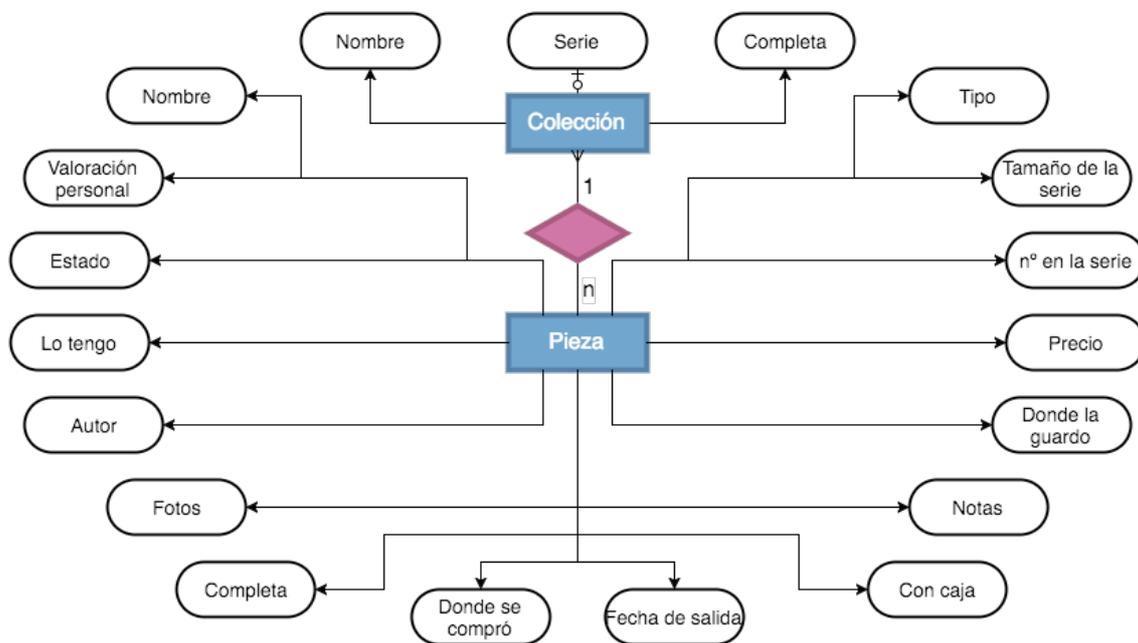


Figura 22 - Modelo entidad-relación

En el caso de este proyecto, la estructura de datos es íntegramente interna a la aplicación, debido a que no hay ningún tipo de comunicación exterior, salvo la introducción manual de datos por parte del usuario. Los datos de la aplicación se guardarán utilizando las librerías Core Data, que pone iOS a disposición.

5.4 Modelo de clases

Con el modelo de datos definido, es importante realizar un modelo de clases. Siguiendo la arquitectura MVC (modelo-vista-controlador), se ha realizado el diagrama de la figura 23, que muestra la estructura de clases básica de la aplicación.

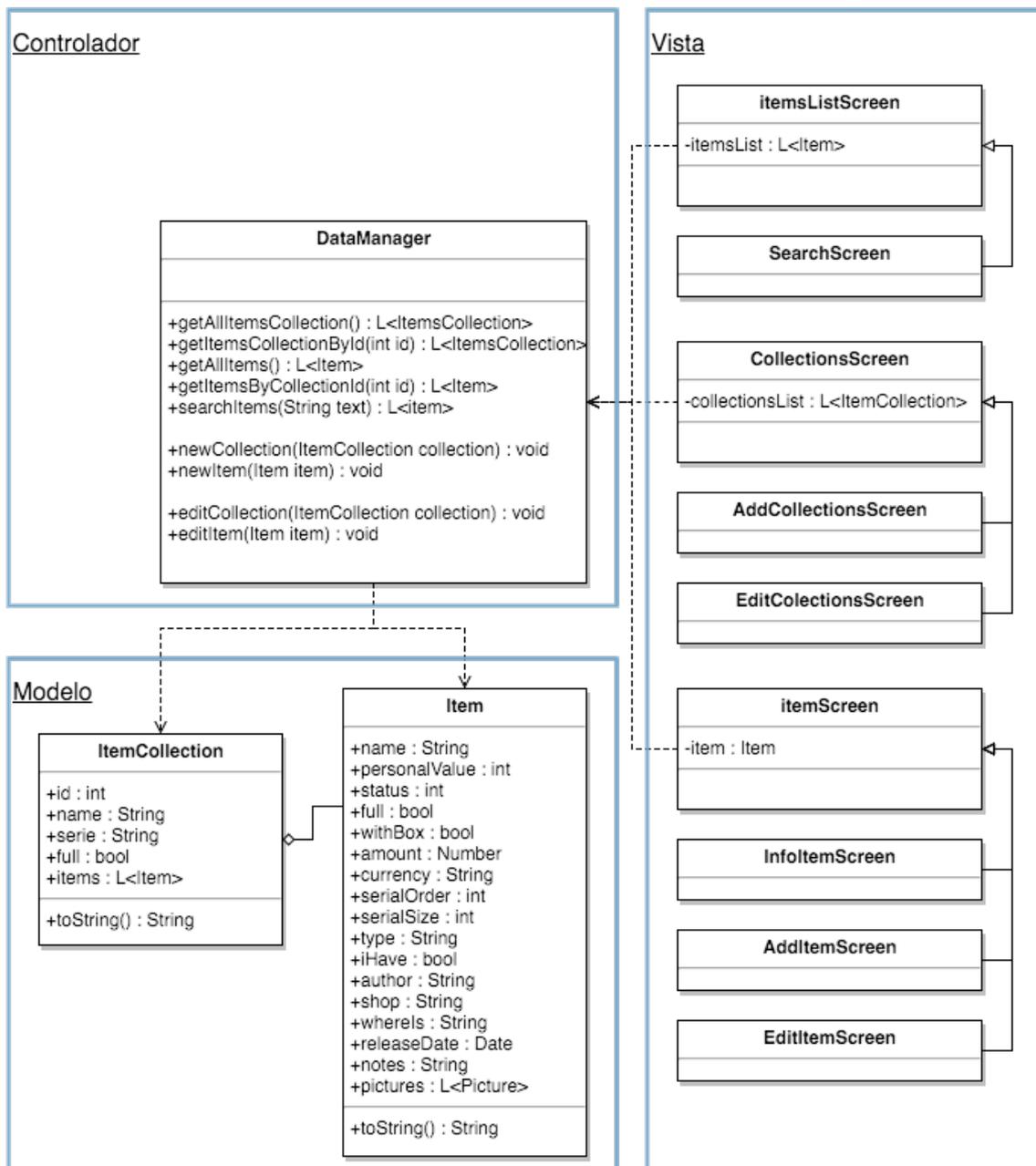


Figura 23 - Modelo-vista-controlador

El modelo-vista-controlador considera una arquitectura donde las clases se organizan en capas, separando los datos, de la lógica de negocio y la presentación. Su correcta aplicación consigue una estructura más limpia y sin dependencias entre capas, facilitando el mantenimiento y la comprensión del código. Su aplicación en una aplicación para un dispositivo móvil, sin conexión a internet, es limitada, pero sigue siendo recomendable.

5.5 Navegación y flujo de pantallas

El árbol de navegación diseñado durante el proceso de creación del prototipo, que se puede consultar en la figura 9, se amplía hasta tener el aspecto del que aparece en la figura 24. De esta manera se organiza el flujo entre todas las acciones que se pretenden incorporar.

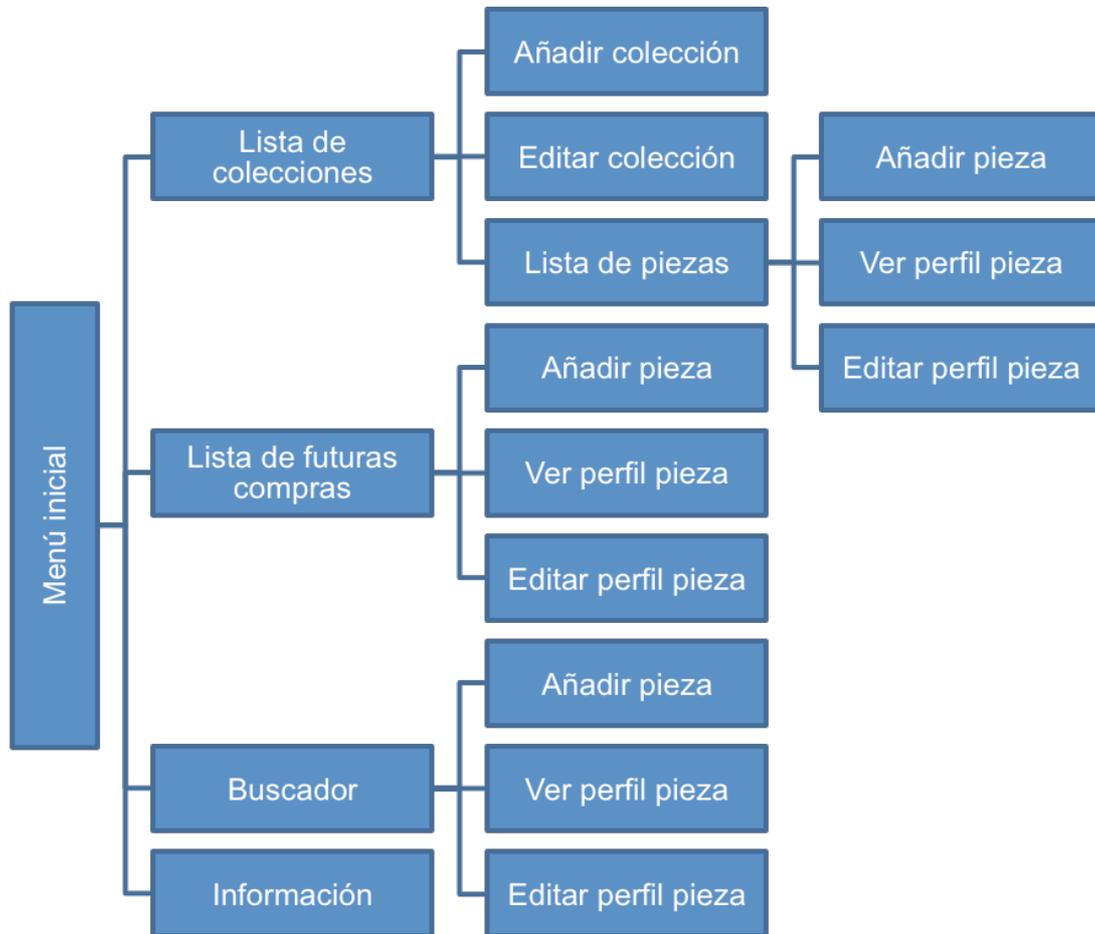


Figura 24 - Navegación de pantallas

La principal diferencia es la inclusión de las pantallas de edición, y el cambio de nomenclatura de “categoría” a “colección”.

6. Desarrollo

Con todo el diseño sobre la mesa, ya puede empezar el desarrollo. Al ser una aplicación nativa para iOS, se requiere el uso de un ordenador Apple, que ya viene equipado con las herramientas necesarias. Como destacadas está Xcode, como *framework* de desarrollo, y Simulator, como simulador de cualquier tipo de dispositivo con iOS, perfecto para probar la aplicación.

6.1 Arquitectura

Modelo de clases

En la figura 23 se ha mostrado una idea conceptual del modelo de clases a seguir. En la práctica, y teniendo en cuenta las características del entorno, el diagrama de clases queda como se muestra en las figuras 25 y 26.

Puede parecer que se pierde el modelo-vista-controlador, pero en realidad no es así. El modelo se ha adaptado a las características de iOS, donde cada vista tiene su propio controlador *ViewController*, y cada rama de navegación se organiza mediante su propio controlador de navegación *NavigationController*. Por su parte, la gestión de los datos se cede a un módulo de servicios. Por último, el modelo de datos incluye estructuras auto-generadas, a través del sistema de datos *Core Data*, opción nativa de Apple para aplicaciones iOS.

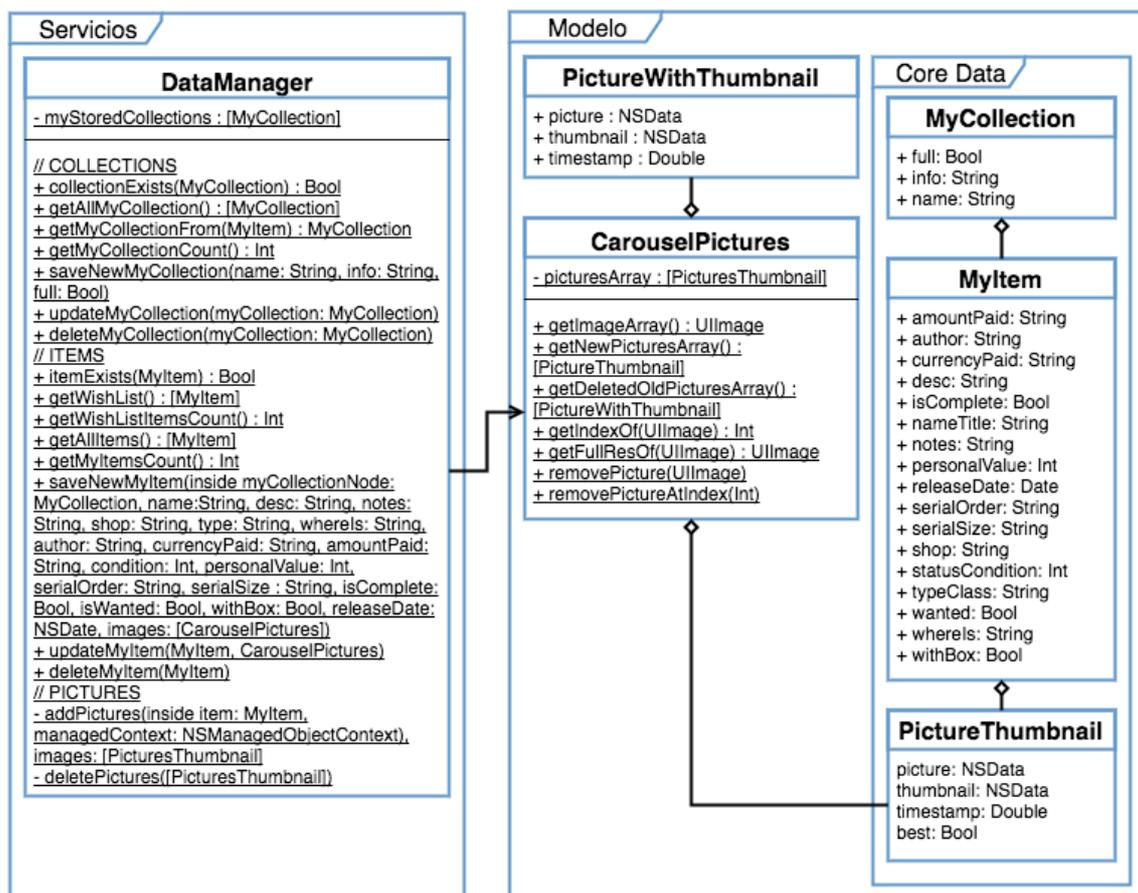


Figura 25 - Diagrama de clases, servicios y modelo

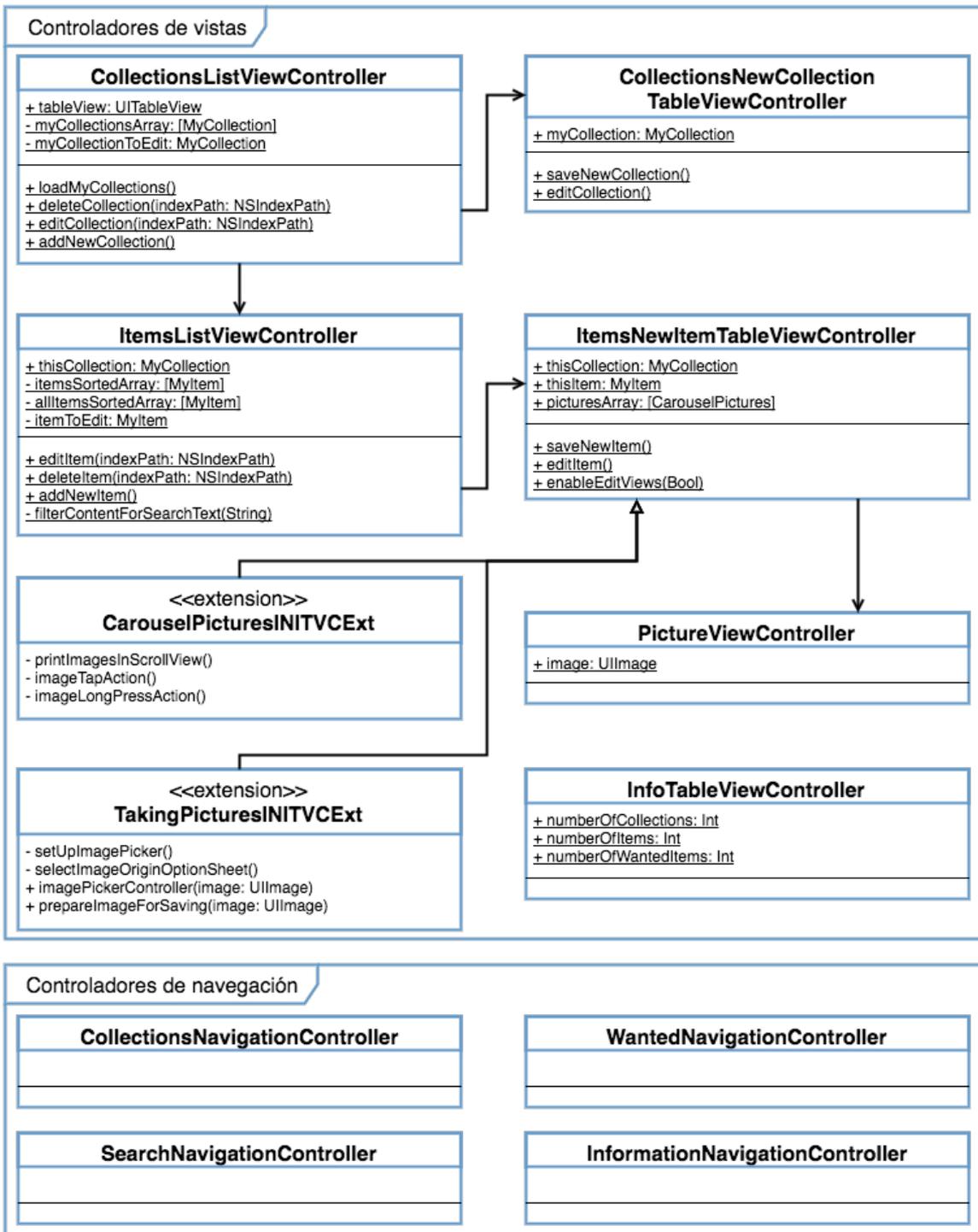


Figura 26 - Diagrama de clases, controladores de vistas y navegación

Modelo de datos

En la figura 22 se ha mostrado el modelo entidad-relación, que es una representación de los datos que forman parte de la aplicación. En la práctica, el modelo de datos se ha creado aprovechando el sistema Core Data de Apple, para facilitar la integración con el resto de la arquitectura. En la figura 27 se puede ver una representación de las tablas, generadas desde el propio Xcode.

Core Data ofrece un potente motor de almacenamiento permanente de datos, optimizado para la plataforma. Este motor, gestionado desde la capa de servicios, permite abstraer las tablas de contenido, a través de métodos específicos y clases auto-generadas. Si se compara la figura 27 con el apartado Core Data de la figura 25, se observa que los datos concuerdan.

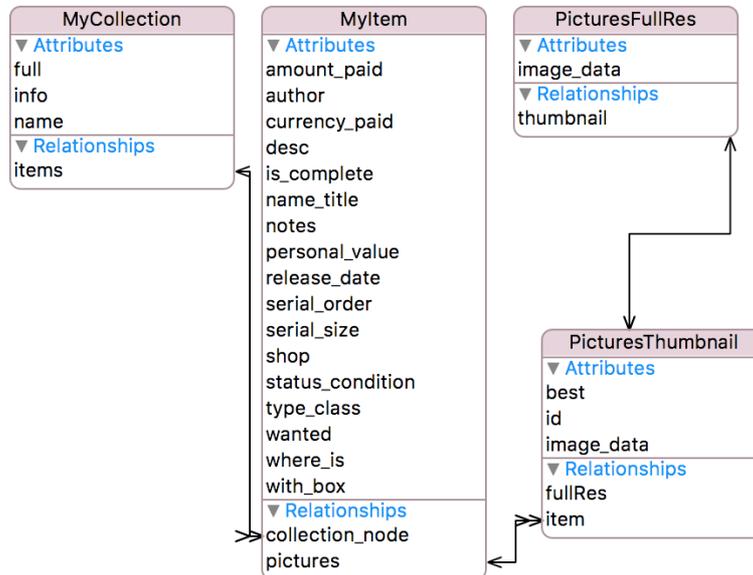


Figura 27 - Modelo de datos en Core Data

Estructura y navegación

La figura 28 muestra el *storyboard* de la aplicación. Esta es la representación gráfica de las vistas, dentro del *framework* Xcode, que permite editarlas y enlazarlas de manera sencilla. Sobre cada vista se ha añadido el controlador asociado, representados previamente en la figura 26.

La vista, marcada con una etiqueta roja que indica “Inicio”, representa la vista del menú de pestañas. Este menú, que encabeza la navegación por la aplicación, enlaza con las vistas de navegación de cada una de las 4 secciones, representadas con etiquetas verdes. A su vez, cada vista de navegación inicia una “rama” independiente, donde se podrá navegar siguiendo el flujo previsto, pudiendo volver siempre hasta el origen.

Observando el *storyboard*, se detecta que, a partir de las vistas de navegación, varias vistas se comparten entre diferentes ramas. Esto no influye en ningún momento con lo descrito anteriormente, y permite la reutilización de código y la homogeneización de pantallas con el mismo fin. Aunque una vista se comparta entre varias ramas, su comportamiento es el de elementos independientes de cada rama. Por otro lado, programáticamente se ha de tener en cuenta a la hora de componer su controlador, ya que, todo y ser independientes, al ser el mismo código hay que tener presente de que vista de navegación se parte.

El menú de pestañas solo ocupa la base de la pantalla, de manera que el resto se aprovecha para representar las otras vistas, permitiendo cambiar de pestaña en cualquier momento, sin perder el progreso en ninguna rama de navegación.



Figura 28 - Storyboard + Controller

Observando la estructura del *storyboard* se puede extraer la estructura de pantallas, y comprobar que sigue el planteamiento del árbol de navegación previsto de la figura 24. En el anexo 4 [9] hay un análisis en profundidad, sobre las pantallas finales y la navegación entre estas.

6.2 Diseño de pantallas

Las pantallas se han creado con las herramientas que Xcode, el *framework* de Apple, ofrece para tal fin. El aspecto se ha basado en las diseñadas para el prototipo de alto nivel [8], aprovechando las ventajas que Xcode ofrece para darles un aspecto más acorde con la plataforma. En las figuras 29 y 30 se puede observar una muestra de la evolución que han sufrido, desde el prototipo de bajo nivel (izquierda) hasta el aspecto final (derecha), pasando por el prototipo de alto nivel (centro).

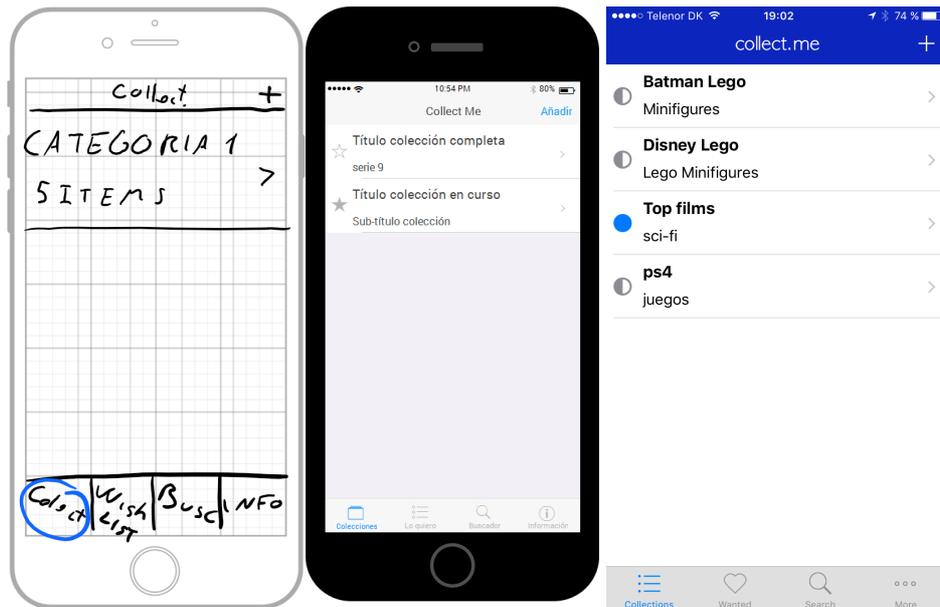


Figura 29 - Evolución de la pantalla de lista de colecciones

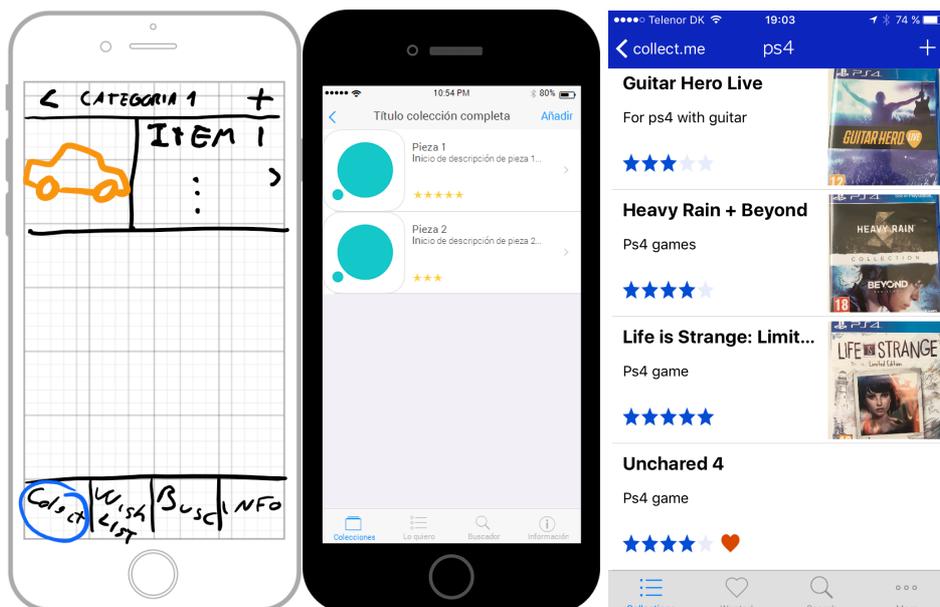


Figura 30 - Evolución de la pantalla de lista de piezas

El resto de pantallas finales se pueden ver en el anexo 4 [9].

6.3 Puntos clave

A la hora de realizar un desarrollo, es importante no malgastar los recursos del sistema, como son la memoria y la capacidad de proceso, ni realizar complejas estructuras de código que complican su comprensión y futura modificación. También es importante pensar en el usuario, buscando ofrecer la mejor experiencia posible. Por estos motivos, el desarrollo se ha apoyado en los siguientes puntos clave:

Centralizar la gestión de datos

La gestión de datos se ha centralizado en la capa de servicios, concretamente en una clase llamada *DataManager*. Esta clase tiene la capacidad de extraer, actualizar, eliminar y guardar los datos en un sistema de memoria permanente. A demás, incorpora una caché para poder ofrecer estos datos de forma rápida, sin tener que ir a consultar siempre a la fuente.

En la aplicación, existe la posibilidad de cargar los mismos datos desde varias pantallas. Por este motivo, es importante asegurar la integridad de los datos. Desde cualquier punto del código, se puede acceder a este servicio para gestionar los datos de la aplicación, con total seguridad de que los datos recibidos están actualizados, y no hay varias fuentes con estados diferentes.

Estabilidad

No hay experiencia más negativa, para un usuario, que sufrir sucesos inesperados. El cierre súbito o los resultados incongruentes, son muestras de inestabilidad por parte de la aplicación. En pro de lograr una aplicación estable, se han añadido controles de seguridad, para evitar que se realicen procesos que cuelguen la aplicación, o desestabilicen los datos. Ejemplos de estos controles son:

- Comprobar que una colección existe antes de mover o crear una pieza en ella.
- Comprobar que una pieza existe antes de editarla.
- Comprobar que una imagen existe antes de acceder a ella.
- Comprobar que el dispositivo tiene cámara antes de ofrecerla como opción.

Dispositivo sin bloqueos

En la aplicación, todo el consumo de datos es interno al dispositivo, por lo que los tiempos de carga no son apreciables al no requerir llamadas externas. De todos modos hay que evitar que los procesos pesados, como las cargas de datos, puedan bloquear el dispositivo de forma temporal. Para ello, todos los métodos de gestión de datos se realizan de forma asíncrona, mostrando un aviso de proceso en marcha.

6.4 Librerías de terceros

En programación existe el concepto “reutiliza, no reinventes”. Durante el desarrollo se han utilizado tres librerías de terceros, con licencias libres del MIT, para evitar tener que desarrollar funcionalidades que ya han desarrollado otras personas. Las librerías, y sus funcionalidades, son las siguientes:

ImageScrollView

Esta librería agiliza la creación de una vista, donde se pueda mostrar una imagen, con posibilidades de zoom y desplazamiento. El creador se llama Nguyen Cong Huy y la ofrece desde el portal GitHub [10].

En el proyecto se ha utilizado en la pantalla que muestra las imágenes a tamaño completo, tras seleccionarlas en el carrusel.

MGSwipeTableCell

Esta librería permite añadir botones de acción a las filas de una tabla, quedando ocultos por defecto, y apareciendo al realizar desplazamientos horizontales sobre ellas. El creador se llama Imanol Fernández y la ofrece desde el portal GitHub [11].

En el proyecto se ha utilizado en las listas de colecciones y de piezas, para añadir las opciones de edición y eliminar. La figura 31 muestra el resultado en ambas pantallas, tras desplazar una fila hacia la derecha.

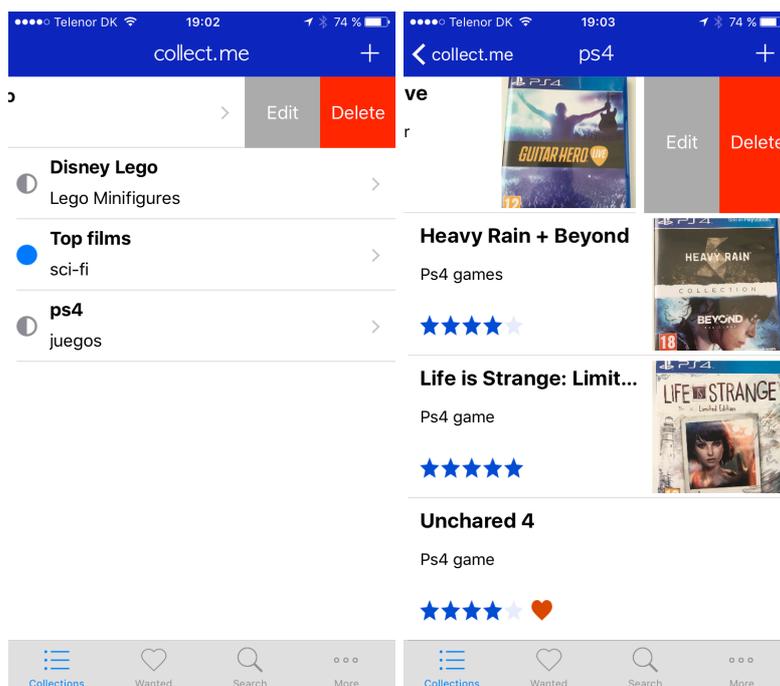


Figura 31 - Demostración de la librería MGSwipeTableCell

RSBarcodes_Swift

Esta librería facilita la labor de capturar un código de barras, con la utilización de la cámara del dispositivo. El creador la ofrece desde el portal GitHub [12].

En el proyecto se ha utilizado en la pantalla de perfil de pieza. Gracias a la captura mediante la cámara, el usuario puede decidir entre introducir manualmente el código de barras, o mediante este sistema. La figura 32 muestra una captura de la acción.

6.5 Funcionalidades añadidas

Durante la última fase del proyecto se han añadido varias funcionalidades, no programadas en un principio. Estas mejoras surgen de recuperar ideas descartadas, durante la fase de toma de datos de los usuarios, y de un mayor conocimiento de la plataforma. Las funcionalidades añadidas son las siguientes:

Lectura de código de barras para las piezas

Se ha añadido un campo nuevo en el perfil de las piezas, para poder incluir un código de barras. Este campo se puede completar manualmente, mediante el teclado, o de forma automática, utilizando la cámara del dispositivo para detectar el código de barras de la propia pieza. La figura 32 muestra esta funcionalidad en acción.

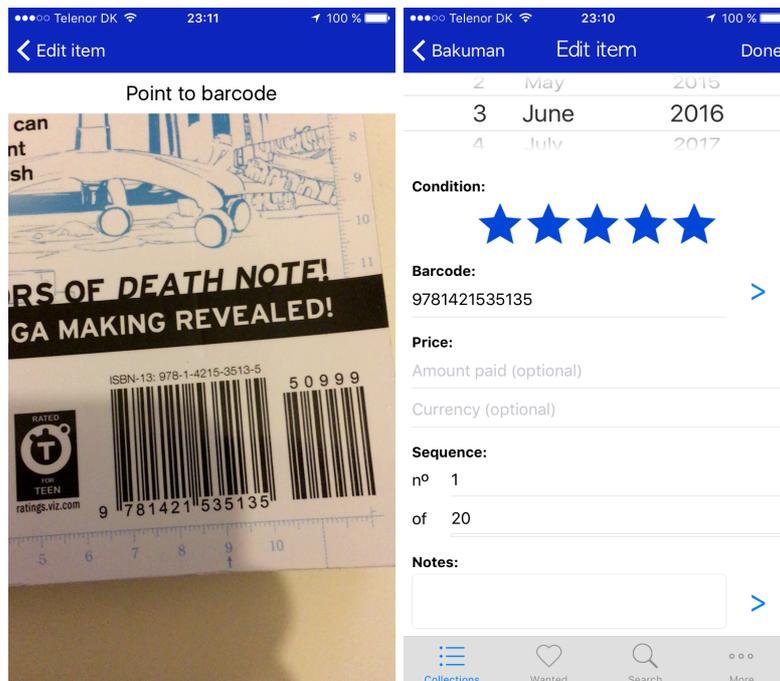


Figura 32 - Captura de código de barras

Internacionalización de la aplicación

Se han añadido los idiomas castellano, catalán e inglés, como idiomas disponibles en la aplicación. Dependiendo de la configuración del dispositivo, la aplicación se mostrará en uno de estos tres idiomas, siendo el inglés el idioma por defecto en caso de no coincidir ninguno.

En la figura 33 se puede ver, como muestra, la pantalla de “añadir pieza” en todas las versiones.

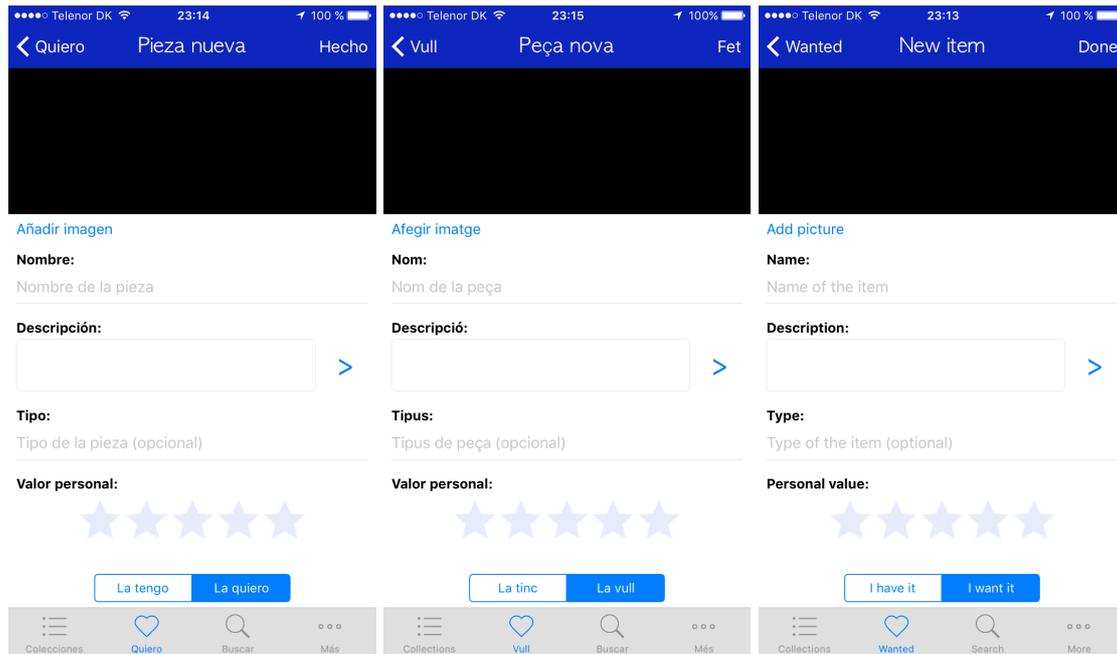


Figura 33 - Internacionalización de la aplicación

Como se puede apreciar, todos los textos están traducidos, sin influir en la estética de la aplicación. Esto es especialmente importante porque la experiencia de usuario debe ser la misma, sin depender del idioma de su dispositivo.

Para lograr que la apariencia de la aplicación se adapte sin problemas a los diferentes textos, se ha tenido que diseñar cada elemento teniendo en cuenta que su tamaño podía variar. A la hora de traducir los textos, también se ha tenido en cuenta lograr longitudes similares.

Marcar colecciones con un color

Se ha añadido un campo nuevo en el perfil de las colecciones, para poder indicar un color, dentro de una lista prefijada. Este color se presentará como color de fondo de la celda, para cada colección, en la pantalla de lista de colecciones. La intención es permitir al usuario categorizar de forma sencilla las colecciones.

Accesos rápidos desde el icono

Aprovechando la capacidad 3D Touch de las nuevas pantallas que montan los iPhone 6s, se han añadido accesos directos hacia las pantallas de “añadir pieza” y “añadir pieza deseada”, desde el mismo icono de la aplicación. La figura 34 muestra el menú emergente, en el cual se ofrecen ambas opciones, al pulsar con intensidad el icono de la aplicación.

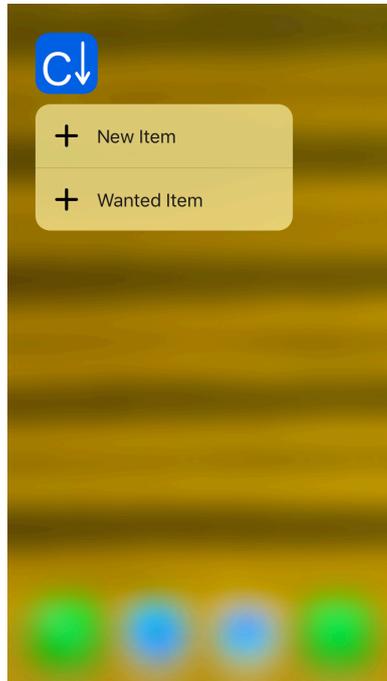


Figura 34 - Accesos rápidos desde el icono

Esta funcionalidad permite al usuario acceder, con una simple acción, a cualquiera de las pantallas nombradas. Esto es especialmente útil cuando el usuario quiere crear un perfil con rapidez.

7. Pruebas

Durante el desarrollo es importante ir comprobando que, las funciones que se desarrollan, funcionan correctamente. Esto entra dentro del juego de pruebas unitarias, que el desarrollador va realizando sobre la marcha. Este tipo de pruebas se realizan teniendo el código en mente, deteniendo el flujo de la aplicación y comprobando que las variables contienen los datos correctos.

Las pruebas unitarias también se pueden programar y automatizar. Si el *framework* lo permite, como es el caso de Xcode, se pueden crear baterías de pruebas automáticas, que se ejecuten antes de empaquetar la aplicación, o bajo petición.

Con el desarrollo más avanzado, existen otro tipo de pruebas, de carácter más funcional, que se utilizan para garantizar el correcto funcionamiento de la aplicación. Estas pruebas no requieren de los conocimientos de un desarrollador, por lo que, en proyectos de cierta envergadura, suelen ser realizadas por especialistas en este campo.

Los juegos de pruebas funcionales acostumbran a ordenar, de forma secuencial, una serie de acciones sobre la aplicación, e indicar un resultado concreto. Estas acciones se deben ejecutar del mismo modo que lo haría un usuario. Quién realiza cada prueba, ha de comprobar que el resultado obtenido es el esperado.

Para este proyecto se ha creado un juego de pruebas funcionales, con la intención de comprobar el buen funcionamiento de la aplicación. Se han añadido desde pruebas sencillas, como crear una colección, a más rebuscadas, que suponen crear, editar y eliminar, desde varias ramas, una misma pieza. Estas pruebas más complejas y rebuscadas se diseñan para intentar encontrar fallos en el sistema, que puedan producir resultados inesperados.

El proceso de creación de este juego de pruebas ha ayudado a encontrar errores en la aplicación, lo que demuestra la necesidad de realizar este tipo de práctica. El anexo 5 [13] contiene la lista de pruebas diseñada para la aplicación.

8. Conclusiones

El mercado de las aplicaciones para *smartphones*, desde su nacimiento en 2008 hasta ahora, no ha dejado de crecer, superando los 100 mil millones de descargas acumuladas en el caso de Apple [14]. Tamaño negocio no ha podido pasar desapercibido y cada vez son más las personas que se suben al tren, aunque este no es el único motivo para hacerlo.

Hasta ese momento un desarrollador de aplicaciones, como máximo, podía aspirar a realizar un trabajo que se quedase en los despachos de los usuarios, o que llegase a un entorno profesional concreto. Con la llegada de los *smartphones*, se abrió la posibilidad de desarrollar aplicaciones que los usuarios llevarían siempre encima, de forma tan inseparable como las llaves de casa o la cartera.

La intención de este proyecto ha sido adentrarse en el desarrollo de una aplicación para *smartphone*, concretamente iPhone, siguiendo un proceso organizado en fases. Para ello ha sido importante concretar una idea de producto, un perfil de usuario ideal, y definir las características, a través a los datos obtenidos a través de encuestas a potenciales usuarios.

La definición de funcionalidades, a través de los intereses de los usuarios, ha sido un proceso muy interesante. La realización de una encuesta pública ha permitido comprobar los intereses sobre las funcionalidades propuestas. También se ha podido comprobar, mediante campos con respuestas libres, la variedad de propuestas alternativas, con mayor o menor popularidad.

Funcionalidades que se han acabado incorporando, con gran peso dentro de la aplicación, como la opción de marcar “lo tengo / lo quiero”, una organización sencilla, o el buscador, surgieron gracias a las ideas introducidas por los encuestados. Esto demuestra lo importante que es, dentro de la toma de decisiones, tener en cuenta la opinión de usuarios potenciales y no quedarse solo con una idea preconcebida.

La puesta en marcha de un proyecto no es una labor sencilla. Se debe planificar cada etapa para distribuir los recursos y tener conocimiento, durante el proyecto, del progreso real respecto al esperado. Esta planificación puede sufrir modificaciones durante el proyecto, debido a cambios inesperados o a desajustes con la distribución del tiempo.

Durante el proyecto la planificación ha sufrido varios reajustes. Uno de las más importantes fue en la etapa de diseño, referente a la toma de requisitos. La toma de datos, a través de encuestas y su posterior análisis, requirió más tiempo del esperado. Esto obligó a modificar la planificación para absorber el retraso. Por suerte, se pudo compensar en la creación del prototipo HD, que resultó ser más sencilla de lo esperado.

Otro cambio importante en la planificación surgió con la navegación de la aplicación, en la etapa de desarrollo. Inicialmente, la navegación se planificó como un proceso inherente, dentro de cada bloque de desarrollo, donde se

crearía una pantalla por cada pantalla diseñada. Finalmente, con la intención de reutilizar el código y no crear más objetos de los necesarios, se decidió reutilizar las pantallas que serían iguales, realizando una navegación inteligente. Esta navegación debía saber que mostrar, y como, dependiendo del punto de partida. Este cambio, sin ser exagerado, se mostró con un cambio en el diagrama de planificación de la figura 1. Este ajuste se absorbió reduciendo horas de documentación.

Con todas las funcionalidades desarrolladas, se realizó un último cambio en la planificación de la última etapa. En este caso fue un cambio positivo, en vistas a un mayor conocimiento de la plataforma, y a una previsión en la documentación algo abultada. Se decidió aprovechar para añadir un bloque de desarrollo de mejoras. Estas mejoras aprovechaban ciertas características de la plataforma, y recuperaban algunas propuestas de los usuarios, como la del código de barras. A su vez, se aprovechó para solucionar algún problema menor, detectado en la aplicación.

Gracias a la gestión de la planificación se pudo organizar mejor y asegurar el éxito del proyecto. Sin una planificación presente habría sido más complejo darse cuenta, en cualquiera de los reajustes comentados, hasta que punto una desviación era viable.

En referencia al desarrollo, que el proyecto se centrara en las herramientas oficiales para iPhone ha facilitado mucho el trabajo. Apple ofrece un SDK muy completo y estable, donde tan solo se han tenido que añadir algunas librerías para casos puntuales. Por contra, se ha tenido que usar un lenguaje de programación poco habitual, Swift 2.0, que es el que requieren para su plataforma en la actualidad.

Haber utilizado herramientas nativas impide poder exportar el proyecto de forma directa a otra plataforma. Por contra, gracias a ello la aplicación ofrece un *look & feel* nativo, integrado con la plataforma, y un rendimiento ideal. Tampoco hay que olvidar que, al no tener el soporte del fabricante, las herramientas alternativas no son tan completas, necesitando hacer una mezcla de herramientas de terceros poco homogénea.

A la hora de probar la aplicación, de nuevo la plataforma de Apple se lleva todos los méritos, ofreciendo un simulador que permite simular todos los dispositivos disponibles soportados por Apple. También ofrece la capacidad de conectar un dispositivo físico compatible, sin ningún problema ni complicación, para instalar la aplicación y realizar las pruebas.

Con el desarrollo actual los datos de la aplicación dependen única y exclusivamente del dispositivo. Con la vista puesta en el futuro, sería interesante realizar los cambios necesarios para que, los datos de la aplicación, se pudieran guardar en un servidor. De esta manera, y mediante un sistema de cuentas verificadas, un usuario podría trasladar sus datos a otro dispositivo sin ningún problema.

9. Glosario

- 3D Touch: Tecnología de Apple, incorporada en los dispositivos iPhone 6s, que permite detectar diferentes grados de presión sobre la pantalla.
- Android: Sistema operativo, destinado principalmente a dispositivos móviles con pantalla táctil, desarrollado por Google.
- App: Término inglés abreviado para referirse a aplicaciones para teléfonos móviles inteligentes.
- Bug: Término inglés utilizado para referirse a un error de software.
- Core Data: Tecnología ofrecida por Apple, en sistemas OS X y iOS, para organizar y almacenar datos internamente.
- Framework: Término inglés utilizado para referirse a un conjunto de herramientas de software, destinadas a un uso genérico, empaquetadas y distribuidas para poder ser adaptadas en un desarrollo.
- GIMP: Acrónimo de GNU Image Manipulation Program, programa de libre distribución, destinado a la edición de imágenes.
- Instagram: Red social destinada a subir fotos y videos.
- iOS: Sistema operativo desarrollado por Apple, destinado a los dispositivos móviles con pantalla táctil de la marca, como el iPhone.
- iPhone: Línea de teléfonos inteligentes de Apple.
- Justinmind: Aplicación destinada a la creación de prototipos de alto nivel, para desarrollos web y aplicaciones móviles.
- Look & feel: Expresión en inglés que hace referencia al aspecto y sensaciones que ofrece un elemento.
- MacBook Pro: Línea de ordenadores portátiles, de alta gama, de Apple.
- OS X: Sistema operativo desarrollado por Apple, destinado a los ordenadores de la marca.
- SDK: Acrónimo inglés de “kit de desarrollo de software”.
- Smartphone: Término inglés para referirse a un teléfono inteligente.
- Xcode: Entorno de desarrollo, ofrecido por Apple, para desarrollar aplicaciones destinadas a productos de la marca.

10. Bibliografía

[1] Definición de coleccionar:

Web:

<http://www.oxforddictionaries.com/es/definicion/espanol/leccionar>

Fecha de visita: 03/2016

[2] Lista de colecciones más populares:

Web: https://en.wikipedia.org/wiki/List_of_collectibles

Fecha de visita: 03/2016

[3] Herramientas digitales, para gestionar colecciones, que hay en el mercado:

hobbydb – web

<https://www.hobbydb.com/>

Boxes: Organize. Discover. Buy. Sell. – app

<https://box.es/>

Home Contents – easy home inventory, app iOS

<https://itunes.apple.com/es/app/home-contents-easy-home-inventory/id420151922?l=en&mt=8>

Brickset – web

<http://brickset.com/>

Fecha de visita: 03/2016

[4] Foto con licencia CC. Permite compartir y adaptar, obliga a atribuir la autoría e impide el uso comercial:

Web: <https://www.flickr.com/photos/ankor2/>

Fecha de visita: 04/2016

[5] Foto con licencia CC. Permite compartir y adaptar, obliga a atribuir y a mantener la licencia en caso de modificación e impide el uso comercial:

Web: <https://www.flickr.com/photos/64667396@N00/>

Fecha de visita: 04/2016

[6] Anexo 1 – Encuesta

Documentos adjuntos a la entrega.

PDF: Se pueden consultar las preguntas, las respuestas individuales y el resumen de las respuestas.

[7] Anexo 2 – Prototipo de bajo nivel

Documentos adjuntos a la entrega:

PDF: Descripción y análisis de la creación del prototipo de bajo nivel

[8] Anexo 3 – Prototipo de alto nivel

Documentos adjuntos a la entrega:

PDF: Descripción y análisis de la creación del prototipo de bajo nivel

ZIP: Prototipo de alto nivel navegable, en formato HTML. Se ha probado su funcionamiento con Safari, Firefox e Internet Explorer. No es compatible con Google Chrome.

[9] Anexo 4 – Pantallas finales y navegación

Documentos adjuntos a la entrega:

PDF: Descripción y análisis de las pantallas finales de la aplicación, y de la navegación entre ellas.

[10] Librería ImageScrollView

Creador: Nguyen Cong Huy

Web: <https://github.com/huynguyencong/ImageScrollView>

Licencia: MIT

Fecha de visita: 05/2016

[11] Librería MGSwipeTableCell

Creador: Imanol Fernández

Web: <https://github.com/MortimerGoro/MGSwipeTableCell>

Licencia: MIT

Fecha de visita: 05/2016

[12] Librería RSBarcodes_Swift

Web: https://github.com/yeahdongcn/RSBarcodes_Swift

Licencia: MIT

Fecha de visita: 06/2016

[13] Anexo 5 – Juegos de pruebas

Documentos adjuntos a la entrega:

PDF: Documento con los juegos de pruebas de la aplicación.

[14] Estadísticas del mercado de las apps:

Web: <http://www.businessofapps.com/app-store-statistics-roundup/>

Fecha de visita: 05/2016