



eetac

Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE CARRERA

Título: Servicio de impresión y enmarcado fotográfico

Titulación: Ingeniería Técnica de Telecomunicaciones - Telemática

Autor: Jordi Bonet Caño

Director: Toni Oller

Fecha: 7 de septiembre del 2015

Resumen

Durante el 2015, se estima que se van a hacer más de 1 billón de fotografías (ver [21]), de las cuales, el 80% se realizarán desde smartphones. La mayoría de las fotos serán almacenadas y olvidadas en medios digitales. Y sólo unas pocas serán reveladas o imprimidas. Pero, ¿qué ha pasado con los típicos marcos de fotos?

En este trabajo final de carrera, se ha observado que en el mercado de la impresión fotográfica podría existir un nicho o grupo de personas que necesitan un servicio de enmarcado fotográfico.

En consecuencia, durante este trabajo se ha ideado, diseñado y desarrollado un servicio telemático de impresión y enmarcado fotográfico utilizando una arquitectura cliente-servidor y aplicando metodologías ágiles durante su desarrollo.

Como cliente del servicio, se ha desarrollado una aplicación para dispositivos iOS con las funciones y recursos necesarios para realizar pedidos.

Como servidor, se han habilitado cuatro entidades para ofrecer servicios de consulta de información, facilitar la gestión de comunicaciones, además de registrar y procesar la información generada por el usuario.

Para la implementación integral del servicio, se ha realizado un estudio previo de mercado, diseñado el producto, analizado su viabilidad e identificado acciones para su promoción y lanzamiento.

Durante todo el proyecto, se han usado dos metodologías ágiles: Lean Startup para la parte de diseño y Scrum para la gestión del desarrollo integral del servicio.

Como resultados de este trabajo, se ha conseguido crear la versión beta del servicio con las funcionalidades y recursos identificados como vitales para la validación de la hipótesis de negocio definida.

Title: Printing and framed photographic service

Author: Jordi Bonet Caño

Director: Antoni Oller

Date: September 7, 2015

Overview

During 2015, it's expected to take over 1 billion photos. The 80% of these photos will be take using smartphones. Most photos will be stored and forgotten in digital media. And only a few will be printed. But, what happened to the typical picture frames?

In this final work, it has been observed that there may be a niche or group of people who need a framed photo service in the photo printing market.

Consequently, during this work it has been conceived, designed and developed a telematics service and framed photographic printing using a client-server architecture and agile methodologies applied during development.

As client, it has been developed an application for iOS devices with the functions and resources needed to carry out orders.

As server, four entities have been enabled to provide information consultation services, facilitate communications management, in addition to record and process the information generated by the user.

For the full implementation of the service, there has been done a market research, designed the product, analyzed its feasibility and identified actions to promote and launch the service.

Throughout the project, it has been used two agile methodologies: Lean Startup for design part and Scrum to manage the overall development of the service.

As a result of this work, it has been created the beta version of the service with the capabilities and resources identified as vital to its implementation.

Dedico este trabajo a mi hijo Mateo por recordarme la inocencia del ser humano y poner a prueba las metodologías de trabajo ágil, a mi mujer Vanessa por su apoyo incondicional, a mi madre y a todas las personas que tienen por vocación hacer de este mundo un mundo mejor

ÍNDICE

INTRODUCCIÓN	7
OBJETIVOS DEL TRABAJO	8
CAPÍTULO 1. ESTUDIO DE MERCADO	9
1.1. Identificar un nicho de mercado	9
1.1.1. Observación y experiencia	9
1.1.2. Herramientas de análisis de tendencias	9
1.1.3. Analizar productos o servicios ya establecidos en el mercado	10
1.1.4. Cuantificar un nicho de mercado	11
1.2. Hipótesis de negocio	11
1.2.1. Definición	12
1.2.2. Validación previa	12
CAPÍTULO 2. DISEÑO DEL PRODUCTO	13
2.1. Definición de caso de uso	13
2.2. Diseño funcional	13
2.3. Diseño visual	15
2.3.1. Prototipos	15
2.3.2. Diagrama User Journey	19
2.4. Diseño Técnico	19
2.4.1. Funcionamiento interno del sistema	20
2.4.2. Comunicaciones internas del sistema	22
2.4.3. Definición técnica del sistema	23
CAPÍTULO 3. ESTUDIO DE VIABILIDAD	25
3.1. Requisitos previos	25
3.2. Estimación de tareas	25
3.3. Costes y retorno esperado	26
3.3.1. Inversión inicial	27
3.3.2. Costes de desarrollo	27
3.3.3. Costes de mantenimiento	28
3.3.4. Retorno esperado	28
3.3.5. Escenarios económicos	29
3.4. Tiempo de desarrollo	30
3.5. Conclusiones de la viabilidad del proyecto	31
CAPÍTULO 4. DESARROLLO DEL PRODUCTO	33
4.1. Desarrollo del cliente	33

4.1.1.	Pasos previos para programar en Swift	33
4.1.2.	Modelos, Vistas y Controlador (MVC)	34
4.2.	Desarrollo del Servidor	38
4.2.1.	Servidor API REST	38
4.2.2.	Servidor de notificaciones	40
4.2.3.	Servidor de e-mail	40
4.2.4.	Servidor Business Intelligence	40
4.3.	Configuración del back office	41
4.3.1.	Panel de gestión de pedidos	41
4.3.2.	Panel de notificaciones.....	42
4.3.3.	Panel de gestión de e-mails	43
4.3.4.	Panel de Business Intelligence.....	43
4.4.	Testeo del sistema	45
 CAPÍTULO 5. PROMOCIÓN Y LANZAMIENTO.....		47
5.1.	Promoción	47
5.1.1.	Conocer al consumidor.....	47
5.1.2.	Analizar los medios de comunicación consumidos.....	47
5.1.3.	Definir el objetivo de la campaña	48
5.1.4.	Elaborar acciones para promocionar el producto en esos medios	48
5.1.5.	Monitorizar y optimizar campañas de promoción.....	49
5.2.	Planificar el lanzamiento del servicio.....	49
5.2.1.	Antes del lanzamiento	49
5.2.2.	Día previo al lanzamiento	50
5.2.3.	Día del lanzamiento	50
5.2.4.	Después del lanzamiento	51
 CAPÍTULO 6. CONCLUSIONES DEL TRABAJO		52
6.1.	Conclusiones por objetivos del trabajo	52
6.1.1.	Definir un plan de proyecto.....	52
6.1.2.	Implementar una versión de prueba.....	52
6.1.3.	Integrar y poner a prueba metodologías ágiles.....	53
6.1.4.	Probar la versión de prueba en un entorno controlado	53
6.1.5.	Probar la versión de prueba en un entorno real.....	54
6.2.	Conclusiones generales	54
6.2.1.	Una buena planificación ahorra tiempo y esfuerzo	54
6.2.2.	Observar es vital para identificar oportunidades	54
6.2.3.	Definir, diseñar y validar antes de programar	54
6.2.4.	La importancia de la capacidad resolutive	54
 BIBLIOGRAFÍA		55
 ÍNDICE DE FIGURAS		57

INTRODUCCIÓN

En 1826, el científico Joseph Nicéphore Niepce, realizó las primeras fotografías de la historia de la humanidad. 100 años más tarde, se estima que el número de fotos realizadas hasta entonces ya alcanzaba los mil millones. Pero el punto de inflexión de la fotografía se encuentra a finales del siglo XX, con el comienzo de la era digital.

Hoy en día, gracias a los avances en electrónica y, sobre todo, gracias a la distribución masiva de los smartphones, el 40% de la población mundial tiene a su alcance un dispositivo electrónico con cámara de fotos. En el 2015, se estima que se realizarán más de un billón de fotos en el mundo de las cuales, el 80% se realizarán desde smartphones.

La mayoría de las fotos son publicadas en redes sociales, compartidas a través de aplicaciones de mensajería y olvidadas en medios de almacenamiento digitales durante años con la intención de recuperarlas en algún momento. Sólo unas pocas son reveladas en papel o imprimidas en formato fotolibro.

En Internet, es posible encontrar excelentes servicios de revelado o impresión de fotos y composición de fotolibros. Pero, actualmente, el enmarcado de fotos parece un formato olvidado. Los servicios existentes son muy limitados, tediosos y confusos.

En consecuencia, el objetivo de este trabajo final de carrera es crear una versión de prueba de un servicio telemático de impresión y enmarcado fotográfico. Una versión que permita, de la forma ágil, comprobar que existe demanda suficiente como para que un servicio del estilo tenga éxito.

Para crear este servicio de impresión y enmarcado fotográfico, se ha realizado un estudio previo de mercado, identificado un nicho, definida y validada una hipótesis previa de negocio y analizado la viabilidad del servicio; además de ideado, diseñado y desarrollado la versión beta del servicio usando una arquitectura cliente-servidor y aplicando metodologías ágiles durante su desarrollo

Como metodologías ágiles, se han usado Lean Startup para la definición de la hipótesis y Scrum para planificar y gestionar el desarrollo del servicio.

Como estructura, este trabajo está dividido en un total de seis capítulos que representan los pasos seguidos para la creación del servicio. Cada capítulo incluye una parte teórica donde se identifica su objetivo y una parte aplicada al servicio a desarrollar.

En el primer capítulo, se ha realizado un estudio de mercado con el objetivo de identificar una oportunidad y elaborar la hipótesis de negocio.

El segundo capítulo, se ha dedicado íntegramente al diseño del servicio usando diferentes herramientas de definición y tomando las primeras decisiones técnicas.

En el tercer capítulo, se ha realizado un estudio de viabilidad para asegurar que el servicio a desarrollar cumplirá con los requisitos deseados.

El cuarto capítulo especifica como se ha desarrollado el producto incluyendo una muestra del desarrollo del cliente, del servidor y del back office.

En el quinto capítulo, se han listado las diferentes acciones de promoción para dar a conocer el servicio desarrollado y definido un procedimiento para asegurar un correcto lanzamiento del producto.

Y por último, el sexto capítulo, se ha dedicado a listar las conclusiones obtenidas durante el desarrollo de este trabajo final de carrera.

OBJETIVOS DEL TRABAJO

Con la intención de focalizar esfuerzos, limitar la envergadura del trabajo y obtener una serie de conclusiones se han definido los siguientes objetivos:

- Definir un plan de proyecto para idear, diseñar y desarrollar un producto o servicio que atienda a una demanda cuantificada en el mercado.
- Implementar una versión de prueba de un servicio telemático de impresión y enmarcado fotográfico basado en una arquitectura cliente-servidor.
- Integrar y poner a prueba metodologías ágiles durante el desarrollo del trabajo final de carrera.
- Probar la versión de prueba en un entorno controlado.
- Probar la versión de prueba en un entorno real para recopilar datos reales y validar la hipótesis de negocio definida.

CAPÍTULO 1. ESTUDIO DE MERCADO

Como punto de partida de este trabajo final de carrera y primer paso del plan de proyecto a definir, se ha realizado un estudio de mercado con el objetivo de identificar un nicho, cuantificarlo y definir la hipótesis de negocio que se deseada validar.

1.1. Identificar un nicho de mercado

Para la identificación de un nicho de mercado o grupo de personas que requieren un servicio o producto para solucionar un problema concreto, se ha realizado un estudio de mercado usando diferentes técnicas listadas a continuación.

1.1.1. Observación y experiencia

Aunque el concepto parezca muy subjetivo, grandes ideas de negocio han sido creadas para solucionar un problema concreto de su creador. La observación y la experiencia de cada persona es una gran fuente de ideas. DropBox, Airbnb, Virgin o Inditex son ejemplo de cómo es posible tener ideas de problemas que el propio fundador se ha encontrado.

Como resultados a aplicar esta primera técnica, se ha identificado que el proceso de enmarcado de un fotoretrato de manera online es, actualmente, un proceso tedioso y potencialmente mejorable. Las soluciones online actualmente disponibles se centran en la impresión en papel u otros formatos como álbumes o libros fotográficos.

1.1.2. Herramientas de análisis de tendencias

Además de la observación y experiencia, se han analizado mecanismos más objetivos que permiten identificar nichos de mercado.

Google Trends (ver [6]), es una herramienta de análisis de intereses que permite analizar la evolución de un nicho concreto y comparar varios nichos entre sí, usando las posibles palabras clave que el consumidor podría usar para localizar el servicio o producto.

Google Adwords, es la herramienta de Google para ofrecer publicidad contextualizada. Con una mínima inversión publicitaria, es posible lanzar una batería de anuncios que promocionen una idea de producto o servicio. Aquel anuncio que obtenga más clics, identifica la idea que genera más interés. Sin vincularse con este servicio, la metodología Lean Startup recomienda este tipo de mecanismo para validar hipótesis antes de cualquier inversión.

Otra de las herramientas, son los servicios de encuestas online como Feebbo o Survio. Este tipo de servicios de encuestas, permiten obtener información de un grupo de personas concreto de forma previa al desarrollo del servicio o producto.

Para este proyecto, se ha elegido Google Trends. Como resultados se ha identificado que las posibles palabras clave que representan al servicio de impresión y enmarcado fotográfico evolucionan favorablemente a lo largo del tiempo.

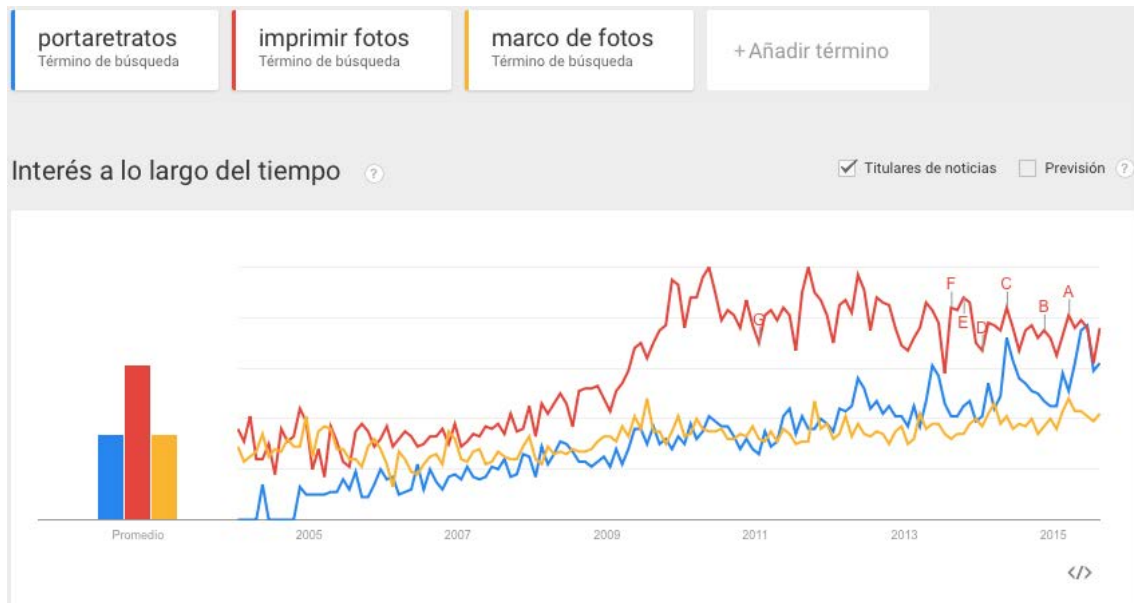


Fig. 1: Gráfica de interés de Google Trends

Como se aprecia en la gráfica de interés de Google Trends (Fig. 1), mientras que el interés de la palabra clave de referencia (imprimir fotos) está estabilizada desde el 2013, las palabras clave que representan al servicio de impresión y enmarcado fotográfico (portaretratos y marco de fotos) reflejan un interés que evoluciona de forma positiva.

1.1.3. Analizar productos o servicios ya establecidos en el mercado

Según Lean Startup (ver [13]), la existencia de grandes empresas ya establecidas que ofrecen un producto o servicio parecido es una indicador fundamental de que existe una demanda importante en el mercado. Analizar esas empresas, identificar sus nichos y probar sus servicios es otra fuente importante de ideas que permiten descubrir un nicho sin atender en un sector de alta demanda.

Un ejemplo de producto de éxito que usó esta técnica es Apple. Según la biografía de Steve Jobs (ver [8]), los responsables de Apple identificaron una fuerte demanda de reproductores digitales de música (los primeros reproductores MP3). Bastó identificar las carencias de los dispositivos actuales a nivel de capacidad, autonomía y usabilidad, para que Apple revolucionara el mercado con su propio reproductor digital de música, el iPod.

Para este proyecto, se ha seleccionado como empresas que ofrecen un servicio similar Hoffmann (ver [19]) y Cheerz (ver [22]). La primera como representante de empresa veterana con más de 90 años en el sector y la segunda como Startup

de éxito con apenas 3 años de vida. Ambas ofrecen importantes servicios de impresión en papel pero sus servicios de enmarcado son limitados.

1.1.4. Cuantificar un nicho de mercado

El concepto “nicho de mercado” incluye en su definición que el número de personas que forman parte del nicho es suficientemente elevado como para que el producto o servicio diseñado sea rentable.

Para ello, se ha cuantificado la población de ese nicho mediante estimaciones basadas en datos ofrecidos por estudios de mercado e informes (Ver [9] y [17]).

Parámetros	Operador	Cantidad
Habitantes con Smartphone conectado a Internet en España		26.250.000 Habitantes
Cuota iPhone en España	8,8%	2.310.000 Dispositivos iOS
% de mujeres con iPhone	40%	924.000 Mujeres con iPhone
Mujeres receptivas al producto	5%	46.200 Usuarios receptivos
Margen de beneficio por venta	10 €	462.000€ Potencial del negocio

Fig. 2: Tabla de parámetros usada para cuantificar el nicho de mercado identificado

Entre estos datos, el margen de beneficio es aquel que nuestro producto o servicio puede ofrecernos. Para conocerlo, es necesario tener una idea previa de sus costes.

Como anotación, aunque los costes por unidad previstos han evolucionado a lo largo del desarrollo de este trabajo, siempre se ha protegido un margen de beneficio por venta de 10€.

Esta decisión probablemente ha encarecido el precio final del producto y probablemente limite el número de ventas en el tiempo. El argumento utilizado para defender este sobrecoste es ofrecer un producto de calidad y con responsabilidad social.

Además, varios referentes del mundo de los emprendedores (ver [4]) afirman que un producto más caro minimiza los gastos logísticos y de post-venta y, en la mayoría de los casos, ofrece beneficios iguales o superiores a vender un producto más económico.

1.2. Hipótesis de negocio

Una hipótesis de negocio es una afirmación sobre cómo se comportará una idea de negocio determinada, sin tener los datos suficientes para afirmar su adecuación a la realidad.

Básicamente, la hipótesis de negocio es la idea que queremos validar en el mercado. Por ejemplo, en el caso de DropBox, la hipótesis a validar podría haber

sido: “La gente necesita compartir documentos sin tener que llevar una memoria USB encima”

1.2.1. Definición

Una hipótesis de negocio debe ser de carácter afirmativo, debe incluir un mensaje preciso, sin ambigüedades y debe incluir los elementos del problema a validar.

Siguiendo estas recomendaciones y usando los datos obtenidos del estudio de mercado, para este proyecto se definió la siguiente hipótesis de negocio:

“Las mujeres con un iPhone necesitan enmarcar fotos. Un servicio que atienda a esta demanda puede obtener un retorno de 462.000€ a nivel nacional”

1.2.2. Validación previa

Según Lean Startup, existen múltiples técnicas para validar previamente hipótesis de negocio antes de iniciar el desarrollo del producto: presentación del producto a nuestro círculo de amistades o familiares, encuestas online a terceros o campaña de Adwords vinculada a una landing page.

Si los resultados obtenidos se aproximan a las estimaciones previas realizadas en la identificación del nicho de negocio, la hipótesis de negocio queda previamente validada y se inicia la siguiente fase del proyecto.

La forma de validar definitivamente una hipótesis de negocio es construir lo que se conoce como PMV (Producto Mínimo Viable) de producto o servicio (ver [11]). Una versión que permite recoger con el mínimo esfuerzo la máxima cantidad de conocimiento validado acerca de la reacción de los consumidores.

Para la validación previa de la hipótesis definida para este proyecto, se ha elegido presentar un prototipo del producto desarrollado en PowerPoint al círculo de amistades y familiares.

Como resultados, se ha obtenido:

Tamaño de la muestra	10 hombres + 10 mujeres
Receptivos a usar el servicio	3 mujeres
Porcentaje de receptivos	15%
Porcentaje de mujeres	100%

Fig. 3: Resultados de la presentación del prototipo

Obteniendo estos datos, se ha dado por validado el 100% de la hipótesis de negocio definida en este proyecto: por un lado, el 100% de las personas receptivas a usar el servicio eran mujeres; por otro, el 30% de las mujeres encuestadas fueron receptivas a usar el servicio. Porcentaje que supera al 5% previsto al cuantificar el nicho de mercado (Fig. 2).

CAPÍTULO 2. DISEÑO DEL PRODUCTO

Una vez realizado el estudio de mercado, identificado el nicho y definida la hipótesis de negocio, se ha diseñado el producto con el propósito de atender a la demanda identificada y validar la hipótesis de negocio previamente definida.

El diseño del producto se ha organizado en cuatro bloques: definición de caso de uso, diseño funcional, dónde se trabaja la idea y se define una posible estructura del producto; diseño visual, dónde se elaboran los primeros prototipos basados en las funcionalidades necesarias y diseño técnico dónde se identifica los requisitos necesarios para su desarrollo.

2.1. Definición de caso de uso

Como caso de uso para el servicio de impresión y enmarcado fotográfico se ha definido el siguiente:

“María acaba de ser abuela. Ya tiene su primer nieto. Un momento inolvidable que quiere recordar siempre. Su hijo, al ver la ilusión de su madre al ser abuela, quiere regalarle algo muy especial. Quiere imprimir la foto del nieto con calidad excelente, enmarcarla con un cuadro que se ajuste a los gustos de su madre y empaquetarlo para regalo. Todo, directamente usando tu iPhone o iPad y sin moverse de casa”.

2.2. Diseño funcional

Para realizar el diseño funcional del producto se ha utilizado mapas mentales como herramienta para representar e identificar las diferentes vistas y funcionalidades del servicio.

Además, se utilizó un código de colores para priorizar cada una de los bloques identificados en el mapa mental. En gris, aquellas vistas y funcionalidades complementarias y en verde, aquellas vistas y funcionalidades esenciales para ofrecer el servicio.

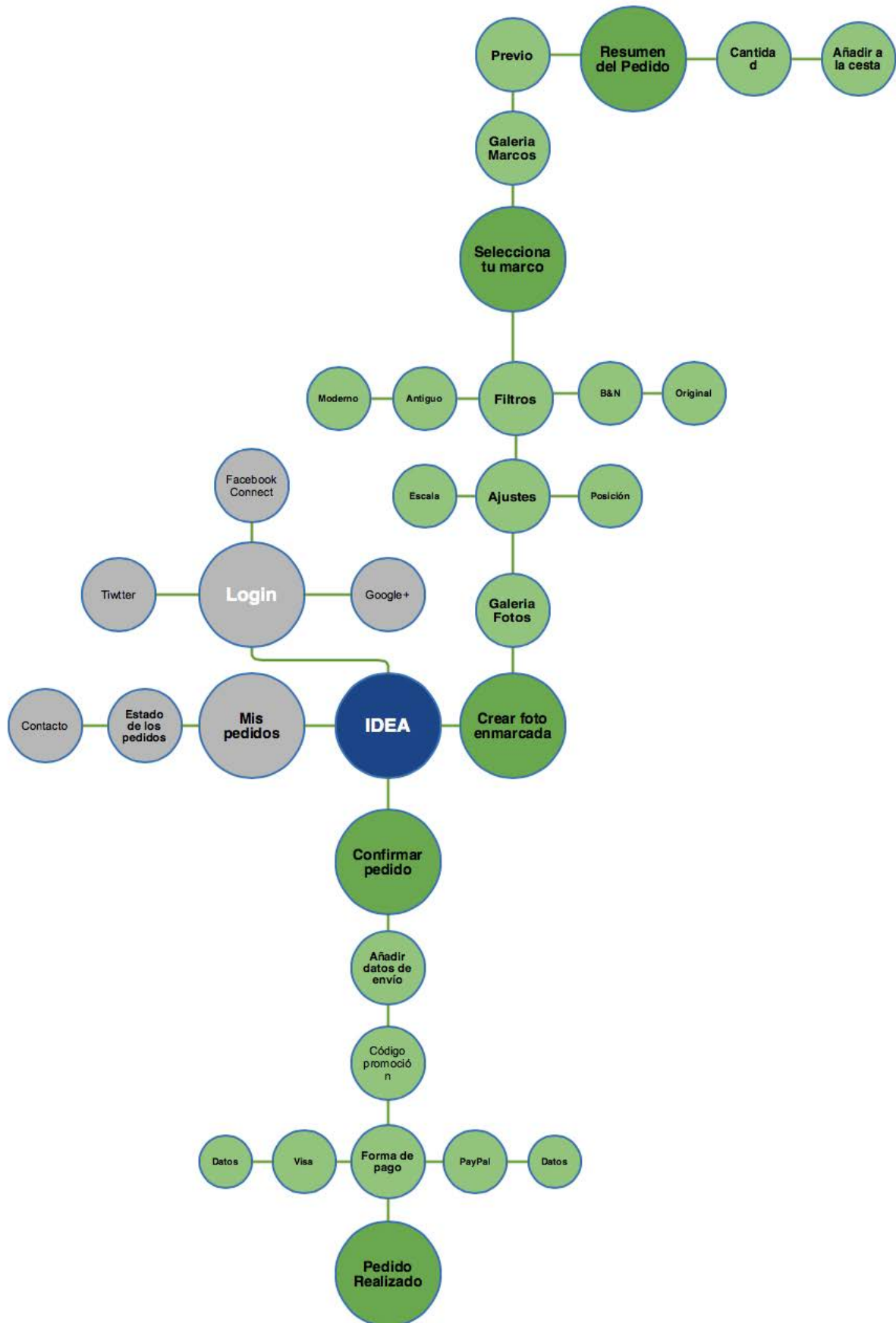


Fig. 4: Mapa mental del servicio de impresión y enmarcado fotográfico

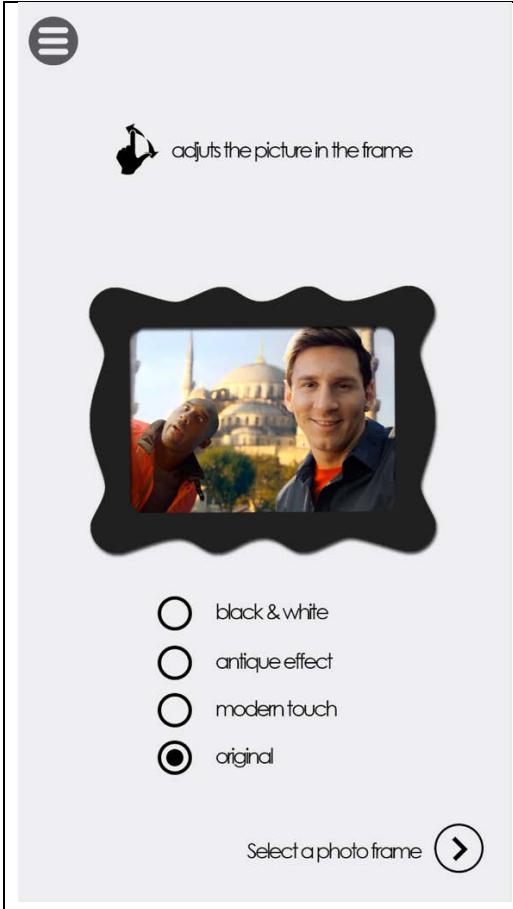
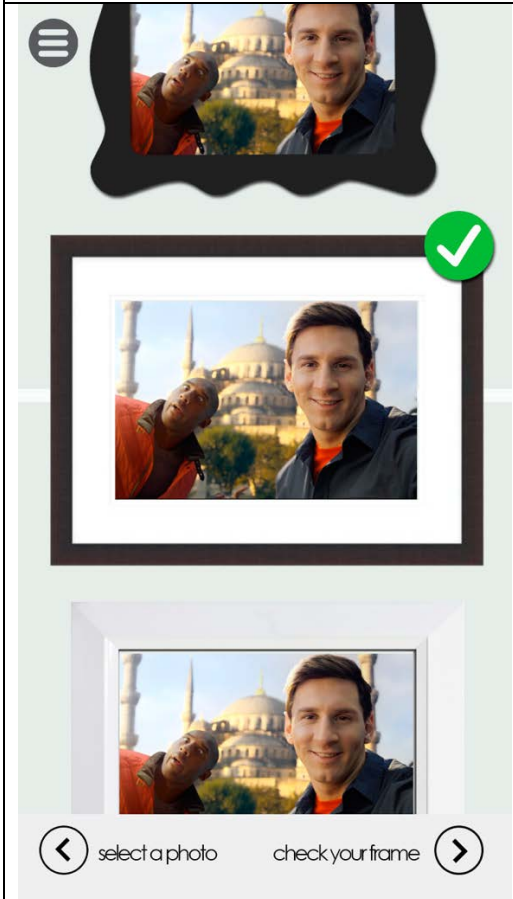
2.3. Diseño visual

Una vez definido el mapa mental del producto a desarrollar e identificadas las diferentes vistas necesarias y sus respectivas funcionales, se han elaborado los primeros prototipos visuales. Además, para asegurar la usabilidad de los prototipos, se ha realizado un diagrama User Journey con usuarios reales.

2.3.1. Prototipos

Para la creación de los prototipos se ha utilizado PowerPoint con una plantilla de elementos visuales de iOS.



 <p>adjusts the picture in the frame</p> <p>black & white</p> <p>antique effect</p> <p>modern touch</p> <p>original</p> <p>Select a photo frame</p>	<h3>Vista 2: Ajustes y Filtros</h3> <p>Segunda vista del producto. Muestra la foto seleccionada por el usuario utilizando la galería de imágenes por defecto en el sistema.</p> <p>La funcionalidad de esta vista es realizar los ajustes pertinentes para que la escala y la posición de la foto seleccionada por el usuario sea la correcta.</p>
 <p>select a photo</p> <p>check your frame</p>	<h3>Vista 3: Selecciona tu marco</h3> <p>Tercera vista del producto. Se muestra la foto seleccionada por el usuario integrada en los diferentes marcos disponibles.</p> <p>La funcionalidad de esta vista es permitir al usuario la selección del marco de un modo rápido, sencillo y visual.</p>



Vista 4: Previo

Cuarta vista del producto. Muestra la foto y el marco seleccionado por el usuario en diferentes decorados simulados.

La funcionalidad de esta vista es mostrar al usuario cómo quedaría su foto enmarcada antes de realizar cualquier tipo de pago.



Vista 5: Resumen del pedido

Quinta vista del producto. Muestra un resumen de la foto y el marco, además de una descripción detallada del producto finalizado.

La funcionalidad de esta vista es informar al usuario de las características finales de la foto enmarcada: tamaño, modelo del marco, coste del envío, garantía, etc.

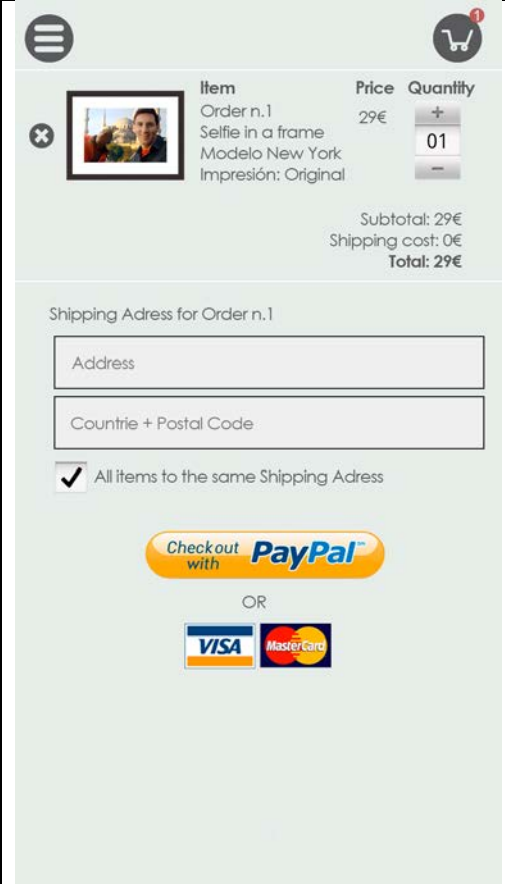
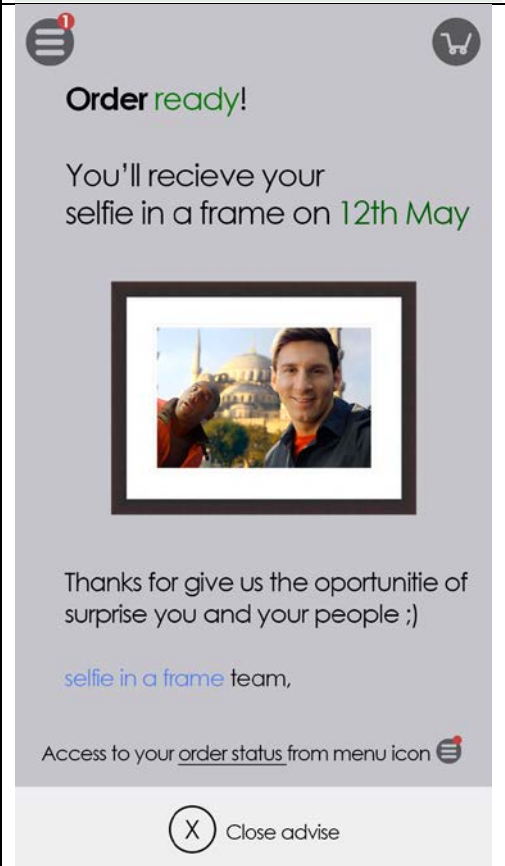
 <p>The screenshot shows a checkout interface for 'Selfie in a frame'. At the top, there is a shopping cart icon with a red notification bubble containing the number '1'. Below it, the item details are listed: 'Order n.1', 'Selfie in a frame', 'Modelo New York', and 'Impresión: Original'. The price is 29€ and the quantity is 01. A subtotal of 29€, a shipping cost of 0€, and a total of 29€ are displayed. Below the pricing, there is a section for 'Shipping Adress for Order n.1' with input fields for 'Address' and 'Country + Postal Code'. A checked checkbox indicates 'All items to the same Shipping Adress'. At the bottom, there are payment options: 'Check out with PayPal' and 'OR' followed by 'VISA' and 'MasterCard' logos.</p>	<h3>Vista 6: Forma de pago</h3> <p>Sexta vista del producto. Muestra las formas de pago disponibles y los campos necesarios para introducir los datos del envío.</p> <p>La funcionalidad de esta vista es permitir al usuario introducir los datos para confirmar el pago y añadir la dirección del envío del producto.</p>
 <p>The screenshot shows a confirmation message on a grey background. At the top, there is a shopping cart icon with a red notification bubble containing the number '1'. The text reads 'Order ready!' in green, followed by 'You'll receive your selfie in a frame on 12th May' in green. Below the text is a framed image of a selfie of a man and a woman in front of a large domed building. At the bottom, there is a thank you message: 'Thanks for give us the oportunitie of surprise you and your people ;)', the 'selfie in a frame team' logo, and a link to 'Access to your order status from menu icon' with a menu icon. A 'Close advise' button with an 'X' icon is at the very bottom.</p>	<h3>Vista 7: Pedido realizado</h3> <p>Séptima vista del producto. Muestra el producto finalizado con un mensaje de pedido realizado con éxito.</p> <p>La funcionalidad de esta vista es informar al usuario que la compra se ha realizado con éxito, además de mostrar una previsión de cuándo recibirá su foto enmarcada.</p>

Fig. 5: Prototipos del servicio de enmarcado fotográfico

2.3.2. Diagrama User Journey

El diagrama User Journey es una herramienta que permite medir la experiencia de un usuario mientras interactúa con un producto. Gracias a esta herramienta, es posible identificar puntos de mejora y realizar los ajustes necesarios para ofrecer una experiencia agradable al usuario.

Para obtener resultados concluyentes, se ha realizado un experimento con usuarios reales interactuando con los prototipos.

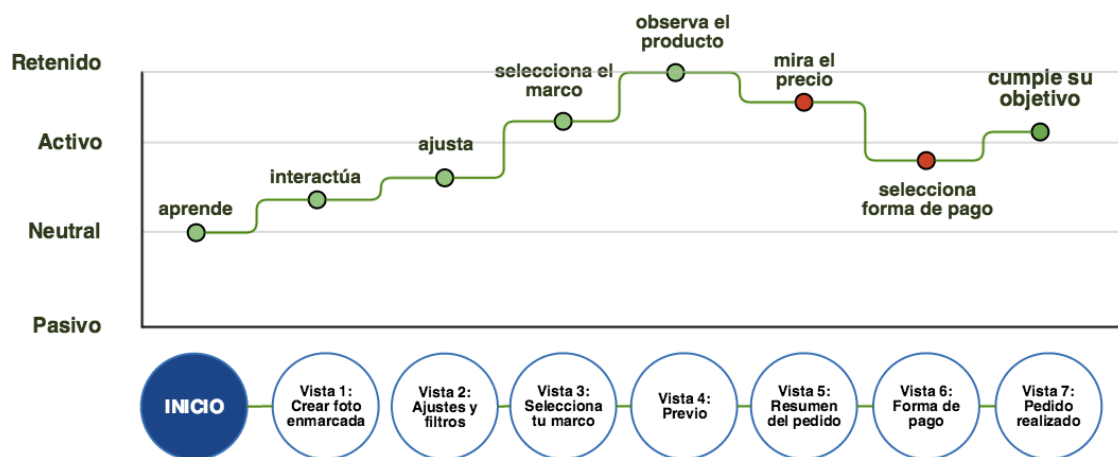


Fig. 6: User Journey de un nuevo usuario

Como resultados del experimento (Fig. 6), se ha identificado que la experiencia del usuario pasa de neutral a usuario retenido durante el proceso de creación de la foto enmarcada. En cambio, al observar el precio y seleccionar la forma de pago, su experiencia adquiere una tendencia negativa que se neutraliza al observar que ha cumplido su objetivo.

Como conclusiones, se ha observado que el precio del producto es elevado y el proceso de pago es tedioso y aburrido. Con la intención de corregir esta experiencia, se ha recalculado de nuevo el precio del producto y ha diseñado una pasarela de pago lo más simple posible.

Al realizar de nuevo el experimento con un usuario que ya conocía el producto, esta tendencia negativa provocada por el precio y el proceso de compra se neutraliza.

2.4. Diseño Técnico

El Diseño Técnico del producto tiene como objetivo definir cómo se va a implementar el servicio o producto deseado. Identifica los materiales, herramientas y tecnologías necesarias para su puesta en marcha.

En el caso de diseño de software, el Diseño Técnico viene dado por su arquitectura técnica o conjunto de tecnologías usada para ofrecer un servicio al usuario.

En este proyecto, al ser un servicio que requiere la gestión de pedidos, se ha elegido una arquitectura cliente-servidor con una capa adicional de back office dedicada a la gestión de pedidos, eventos y mensajes.

2.4.1. Funcionamiento interno del sistema

Para representar el funcionamiento interno del sistema, identificar los bloques necesarios, sus relaciones y entradas y salidas, se ha diseñado el siguiente diagrama de bloques:

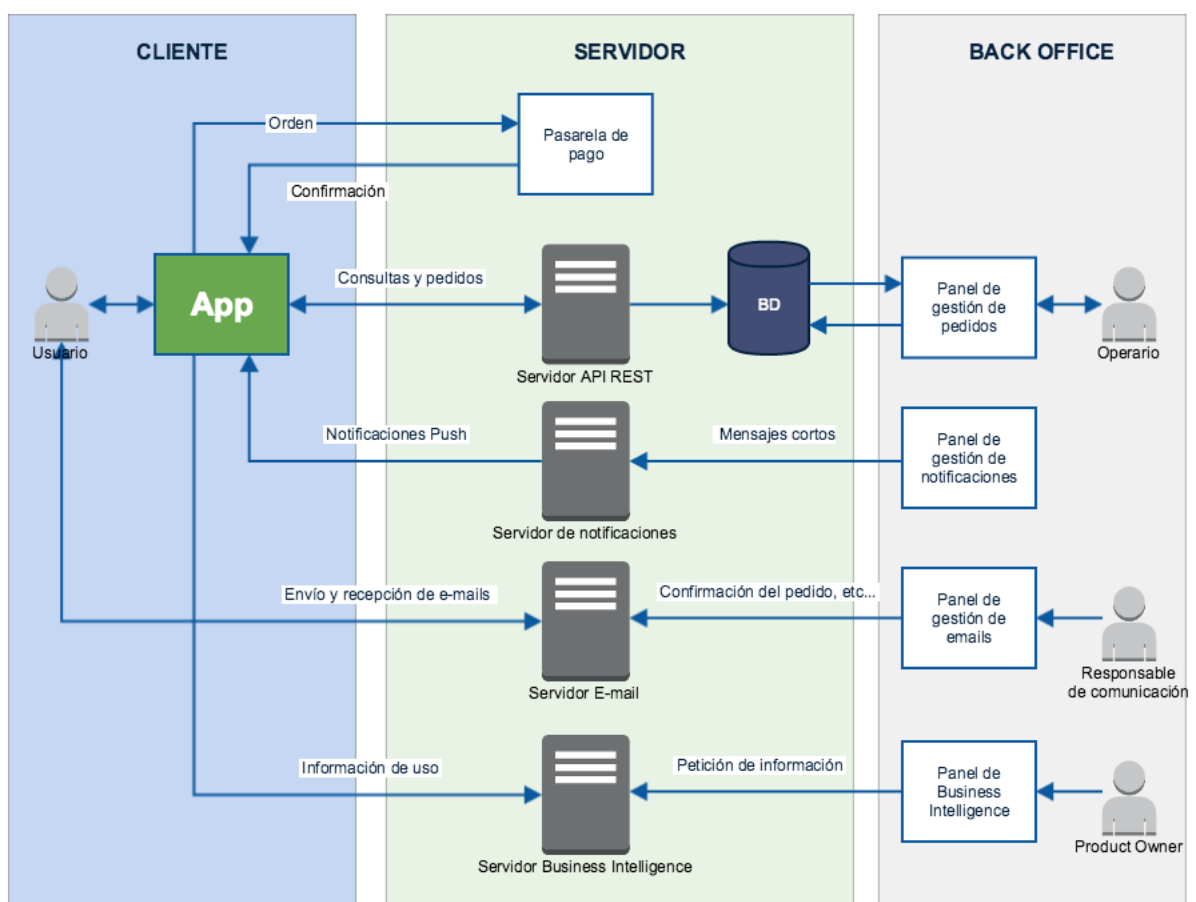


Fig. 7: Diagrama de bloques

Como se observa en el diagrama de bloques (Fig. 7), para este proyecto se ha elegido una arquitectura cliente-servidor con un capa adicional de back office.

Por la parte cliente, se ha seleccionado como medio para interactuar con el usuario una app nativa. Esta app permite al usuario realizar nuevos pedidos, recibir notificaciones y enviar eventos.

Por la parte servidor, se ha elegido cuatro entidades que ofrecen los siguientes servicios:

- Servidor API REST: API o librería de funciones que ofrece un servicio de consultas a un Base de Datos usando el protocolo HTTP.
- Servidor de notificaciones: Servicio que gestiona el envío de mensajes a través de tecnología Push.
- Servidor E-mail: Servidor de correo para el envío y recepción de correos electrónicos con los usuarios.
- Servidor Business Intelligence: Servidor dedicado a la medición y procesado de la información generada por el usuario mientras utiliza el cliente

Como back office, se ha definido una configuración de cuatro paneles independientes con el objetivo de minimizar costes de desarrollo:

- Panel de gestión de pedidos: Panel interno que permite recibir y gestionar los pedidos de los usuarios, además de configurar diferentes características como el precio final de cada operación, el catálogo de marcos disponibles o los códigos de promoción.
- Panel de notificaciones: Panel para el envío y gestión de notificaciones Push
- Panel de gestión de e-mails: Panel para gestionar la comunicación por correo electrónico con el cliente.
- Panel de Business Intelligence: Panel para acceder y analizar la información generada por el usuario mientras utiliza el cliente.

2.4.2. Comunicaciones internas del sistema

Para diseñar las comunicaciones internas del sistema, se han identificado los elementos básicos, definido los mensajes de intercambio de información necesarios y sus respectivas secuencias, mediante el siguiente diagrama de secuencia.

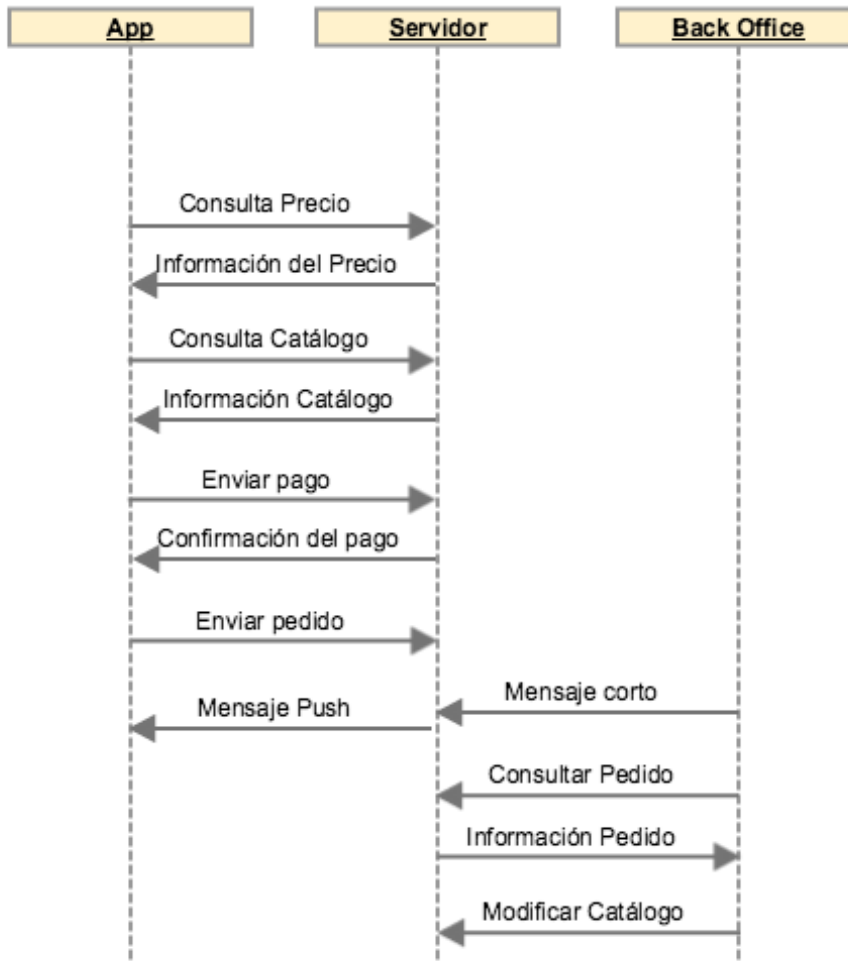


Fig. 8: Diagrama de secuencia

Como se observa en el diagrama de secuencia (Fig. 8), se han definido los mensajes de intercambio de información:

Mensajes del cliente (App) al Servidor:

- Consulta Precio
- Consulta Catálogo
- Enviar o modificar Pedido

Mensajes del servidor al cliente:

- Información del Precio

- Información del Catálogo
- Confirmación de pago
- Mensaje Push

Mensajes del Back Office al Servidor:

- Mensaje corto (Push)
- Consultar Pedido
- Modificar Catálogo
- Modificar Precio

Mensajes del Servidor al Back Office

- Información del pedido

2.4.3. Definición técnica del sistema

Tras diseñar el funcionamiento interno del sistema y definir sus comunicaciones internas, se ha identificado el paquete de tecnologías que se desea usar para su implementación.

Después de analizar diferentes opciones y priorizando aquellas soluciones que pueden minimizar el coste de implementación del producto, para este proyecto se ha optado por la siguiente configuración técnica:

2.4.3.1. Cliente

Para el desarrollo del cliente, se ha elegido Swift (ver [7]) como lenguaje de programación. Un nuevo lenguaje basado en Objective C desarrollado por Apple para agilizar la programación de Apps para dispositivos iOS.

2.4.3.2. Servidor

Para la implementación de todos los servicios ofrecidos por la parte del servidor, se ha optado por la siguiente configuración:

2.4.3.2.1 Servidor API REST

Para levantar este servicio, se ha elegido un paquete de tecnologías JavaScript conocido con el acrónimo MEAN: MongoDB como gestor de Bases de Datos, Express como framework para crear la API REST, AngularJS para crear el panel de gestión de pedidos y Node.JS como framework que proporciona las funcionalidades core del servicio.

2.4.3.2.2 Servidor de notificaciones

Con el objetivo de minimizar costes de desarrollo, se ha elegido un servicio externo de gestión de notificaciones Push ofrecido por Parse (ver [12]). Un servicio gratuito que ofrece su propio sistema de envío de notificaciones Push con su respectivo panel de control.

2.4.3.2.3 *Servidor de correo*

Como servidor de correo, no se ha elegido ninguno. Para minimizar la dimensión del proyecto, se optará por el uso de un servicio de correo electrónico convencional.

2.4.3.2.4 *Servidor Business Intelligence*

Al igual que con el servidor de notificaciones, para medir y procesar la información generada por el usuario mientras utiliza el cliente se ha seleccionado el servicio ofrecido por Flurry (ver [5]). Además, incorpora su propio panel de control para acceder y analizar la información registrada.

2.4.3.2.5 *Pasarela de Pago*

Como pasarela de pago, se ha elegido Paypal (ver [20]) como punto de partida. Su simple integración y su actual soporte de pago con tarjeta permite agilizar el desarrollo del producto considerablemente a cambio de soportar una comisión por operación.

2.4.3.3. *Back Office*

Para la elaboración de los paneles de gestión de pedidos, notificaciones, correos electrónico y Business Intelligence, se ha optado por usar AngularJS, Parse, Gmail y Flurry respectivamente.

CAPÍTULO 3. ESTUDIO DE VIABILIDAD

Siguiendo la metodología Lean Startup para la creación de un producto, se ha realizado un estudio de viabilidad con el objetivo de analizar si se disponen de los recursos necesarios para su implementación y si el diseño del servicio cumple con los retornos esperados.

3.1. Requisitos previos

Uno de los puntos analizados para valorar la viabilidad del un proyecto son los requisitos previos necesarios para su desarrollo. En el caso de ingeniería de software, estos requisitos vienen dados por el acceso a las tecnologías requeridas y por los conocimientos necesarios para su implementación.

En cuanto a tecnología o herramientas para el desarrollo de este proyecto se ha requerido:

- Cuenta de desarrollor iOS de Apple
- XCode 6.4 + Swift 1.2
- Web Storm
- MongoDB + Express + AngularJS + NodeJS
- Flurry SDK
- Parse SDK
- PayPal SDK
- Alamofire Framework

La cuenta de desarrollador iOS se ha adquirido para el desarrollo de este proyecto, Web Storm se ha activado mediante licencia de estudiante y el resto de herramientas han sido de carácter gratuito.

Referente a los conocimientos, al inicio del proyecto se disponía de conocimientos de programación de C# y Java; pero no de Swift basado en Objective-C. Debido a ello, se ha tenido que realizar una formación rápida usando cursos online de Udemy y documentación oficial.

Para la implementación de el Servidor API REST, se ha utilizado una aplicación desarrollada por AlterAid (ver [23]). Además, gracias al soporte ofrecido por los miembros de su equipo, se ha podido configurar el servidor en dos semanas.

3.2. Estimación de tareas

Como otro punto estudiado para analizar la viabilidad del producto, se ha realizado una estimación aproximada del coste de esfuerzo de las tareas necesarias para el desarrollo del producto. De esta manera, se ha valorado si la magnitud del proyecto está al alcance de los recursos disponibles.

Para ello, y siguiendo las buenas prácticas recomendadas por Scrum, se han identificado las tareas, priorizado por valor de negocio, valorado usando puntos

de esfuerzo como unidad de valoración y clasificadas en diferentes Sprint contruidos sobre la marcha.

Aunque Scrum recomienda evitar una unidad temporal, en este proyecto se ha optado por vincular los puntos de esfuerzo a días de desarrollo.

Sprint 1		
User Story	Puntos	Estado
Diseño de Producto: Mindmap	0.5	Hecho
Diseño de Producto: User Journey	0.5	Hecho
Diseño de Producto: Prototipado de vistas	2	Hecho
Diseño de Producto: Diagrama de bloques	0.5	Hecho
Diseño de Producto: Diagrama de secuencia	0.5	Hecho
Definición del Producto - Convencional	0.5	Hecho
Definición del Producto – Business Model Canvas	0.5	Hecho
Diseño financiero: Inversión inicial	0.5	Hecho
Diseño financiero: Forecast	0.5	Hecho
Puntos quemados	6/6	

Sprint 2		
User Story	Puntos	Estado
Feedback sesión de control	0.5	Hecho
Estimación de tareas	0.5	Hecho
Formación Swift 1/3	5	Hecho
Costes de desarrollo	0.5	Hecho
Costes de mantenimiento	0.5	Hecho
Escenarios económicos	1	Hecho
Viabilidad del proyecto	0.5	Hecho
Análisis del mercado	1	Hecho
Adaptación al formato del TFC	3	Hecho
Puntos quemados	12.5/12.5	

Fig. 9: Muestra de las tareas valoradas y priorizadas usando Scrum

3.3. Costes y retorno esperado

Uno de los parámetros más importantes a la hora de estudiar la viabilidad de un producto, ha sido conocer de antemano la inversión inicial requerida, los costes de desarrollo y mantenimiento y estimar el retorno esperado ofrecido por el servicio una vez operativo.

3.3.1. Inversión inicial

La inversión inicial identifica el coste de las herramientas, materiales, mano de obra y otros gastos derivados de la propia producción del producto. Para este proyecto, se han previsto la siguiente inversión inicial:

Costes iniciales			
Herramientas	Unidades	Coste	Subtotal
Cuenta de desarrollador iOS	1	99 €	99 €
XCode 6.4 + Swift	1	0 €	- €
Web Storm	1	0 €	- €
MongoDB + Express + AngularJS + NodeJS	1	0 €	- €
Flurry + Parse + PayPal + Alamofire	1	0 €	- €
Materiales			
Impresora	1	1.200 €	1.200 €
Papel	1	120 €	120 €
Tinta	0	100 €	- €
Marcos	200	4 €	800 €
Cajas	200	4 €	800 €
Hosting + dominio	1	100 €	100 €
Tráfico	1	100 €	100 €
Mano de obra			
Programador Swift + JS (horas)	560	10 €	5.600 €
Otros			
Presupuesto campaña de lanzamiento	1	1.000 €	1.000 €
Total			9.819 €

Fig. 10: Inversión inicial requerida

3.3.2. Costes de desarrollo

Como se observa en la inversión inicial requerida (Fig. 10), el coste más importante son los generados por la mano de obra. 560 horas de programación supone una inversión inicial prevista de 5.600€ aproximadamente. En este caso, estos costes de mano de obra se convierten en una inversión de tiempo, ya que el desarrollo se realiza por cuenta propia. Siendo así, la inversión inicial de este proyecto se ha cuantificado en 5.000€.

3.3.3. Costes de mantenimiento

Los costes de mantenimiento son aquellos gastos necesarios para mantener un producto o servicio en funcionamiento o en un estado determinado. En el caso de ingeniería de software, los costes de mantenimiento suelen estar vinculados a la mano de obra y los servicios subcontratados para mantener el producto en funcionamiento.

En este proyecto, se ha identificado como coste de mantenimiento la cuenta de desarrollador iOS de pago anual, el tiempo dedicado por el desarrollador para mantener el producto en funcionamiento, el desgaste de la impresora y el alojamiento del propio servicio. En total, los costes de mantenimiento de este proyecto son de unos 590€ anuales asumiendo que el tiempo de desarrollo será por cuenta propia.

3.3.4. Retorno esperado

Todo producto tiene un objetivo y gran parte de los productos comerciales que conocemos tienen como objetivo generar valor. Por ello, a la hora de valorar la viabilidad de un producto se ha estimado el retorno o rendimiento esperado.

Para elaborar esta estimación, se ha calculado el coste de producción por unidad y definido el margen de beneficio que se desea aplicar.

Coste producción	6,20 €
Marco	4,00 €
Papel	0,10 €
Tinta	0,10 €
Operario	2,00 €
Coste de envío	6,00 €
Coste publicidad	1,00 €
Coste packaging	5,00 €
Coste total (sin gastos de envío)	12,20 €
PVP	29,00 €
IVA	21%
IVA soportado por unidad	6,09 €
Beneficio por unidad antes de imp.	10,71 €

Fig. 11: Cálculo del coste de producción y beneficio por unidad

Una vez conocido el coste de producción por unidad y el margen de beneficio, se ha calculado el retorno esperado aplicando los parámetros anteriores a una estimación de unidades vendidas.

Unidades vendidas	1	100	500	1.000	5.000	50.000
Coste producción	12 €	1.220 €	6.100 €	12.200 €	61.000 €	610.000 €
Facturación	29 €	2.900 €	14.500 €	29.000 €	145.000 €	1.450.000 €
IVA soportado (21%)	6 €	609 €	3.045 €	6.090 €	30.450 €	304.500 €
Beneficio AI	11 €	1.071 €	5.355 €	10.710 €	53.550 €	535.500 €
Costes						
Operario	2 €	200 €	1.000 €	2.000 €	10.000 €	100.000 €
Marcos	4 €	400 €	2.000 €	4.000 €	20.000 €	200.000 €
Papel	0 €	10 €	50 €	100 €	500 €	5.000 €
Tinta	0 €	10 €	50 €	100 €	500 €	5.000 €
Total	6 €	620 €	3.100 €	6.200 €	31.000 €	310.000 €
Retorno esperado (beneficios)						
Antes de impuestos	11 €	1.071 €	5.355 €	10.710 €	53.550 €	535.500 €
Después impuestos	8 €	803 €	4.016 €	8.033 €	40.163 €	401.625 €

Fig. 12: Cálculo del retorno esperado

Como se observa en el cálculo del retorno esperado (Fig. 12), el servicio tiene previsto un retorno de 8€ por unidad después de impuestos. 2€ por debajo de nuestro objetivo de mantener 10€ de beneficio por unidad debido al impuesto de sociedades.

3.3.5. Escenarios económicos

Una vez calculado el retorno esperado del producto, se han identificado los diferentes escenarios económicos del producto. Estos escenarios permiten medir en todo momento el éxito o fracaso del producto.

Escenario económico	Ventas	Observación
Fracaso	0	No vender ni un solo producto es un identificador claro de fracaso
Negativo	<400	No se recupera la inversión inicial
Neutral	500	Se recupera la inversión inicial pero no el tiempo dedicado
Positivo	1000	Se recupera la inversión y monetizamos el tiempo dedicado
Muy positivo	5000	Beneficios netos de 30.000€
Éxito	50000	Beneficios netos de 400.000€

Fig. 13: Escenarios económicos definidos

La metodología Lean Startup exige un parámetro cuantificable para validar la hipótesis de mercado.

En este proyecto, se ha definido como parámetro el número de ventas. Si se consigue alcanzar el escenario Positivo (1000 unidades vendidas) en un periodo de tiempo previsto de 3 meses, la hipótesis de negocio será validada.

3.4. Tiempo de desarrollo

Cómo último parámetro para valorar la viabilidad del proyecto, se ha estimado el tiempo necesario para su desarrollo usando un diagrama de Gantt como eje temporal y las estimaciones de las tareas como duración.

Nombre	Duración	Inicio	Fin
☐ FASE 2 DEFINICIÓN Y DISEÑO	7d	01/04/2015	09/04/2015
Definición	1d	01/04/2015	01/04/2015
Diseño de producto	6d	02/04/2015	09/04/2015
☐ FASE 3 VIABILIDAD	7d	10/04/2015	20/04/2015
Estimación de tareas	3d	10/04/2015	14/04/2015
Costes	2d	15/04/2015	16/04/2015
Retorno esperado	2d	17/04/2015	20/04/2015
☐ FASE 4 DESARROLLO	81d	21/04/2015	11/08/2015
Frontend	70d	21/04/2015	27/07/2015
Backend	10d	28/07/2015	10/08/2015
Business Intelligence	1d	11/08/2015	11/08/2015
☐ FASE 5 PROMOCIÓN	12d	12/08/2015	27/08/2015
Acciones	10d	12/08/2015	25/08/2015
Costes	1d	26/08/2015	26/08/2015
Retorno esperado	1d	27/08/2015	27/08/2015
☐ FASE 6 LANZAMIENTO	7d	28/08/2015	07/09/2015
Medidas previas	5d	28/08/2015	03/09/2015
Resultados	1d	04/09/2015	04/09/2015
Aprendizaje	1d	07/09/2015	07/09/2015

Fig. 14: Relación de tareas y duración

Como se observa en la relación de tareas y duración (Fig. 14), la fase dedicada al desarrollo consume gran parte del tiempo total del proyecto. En esta fase, se ha tenido en cuenta la curva de aprendizaje habitual en el uso de nuevas herramientas, lenguajes o tecnologías.

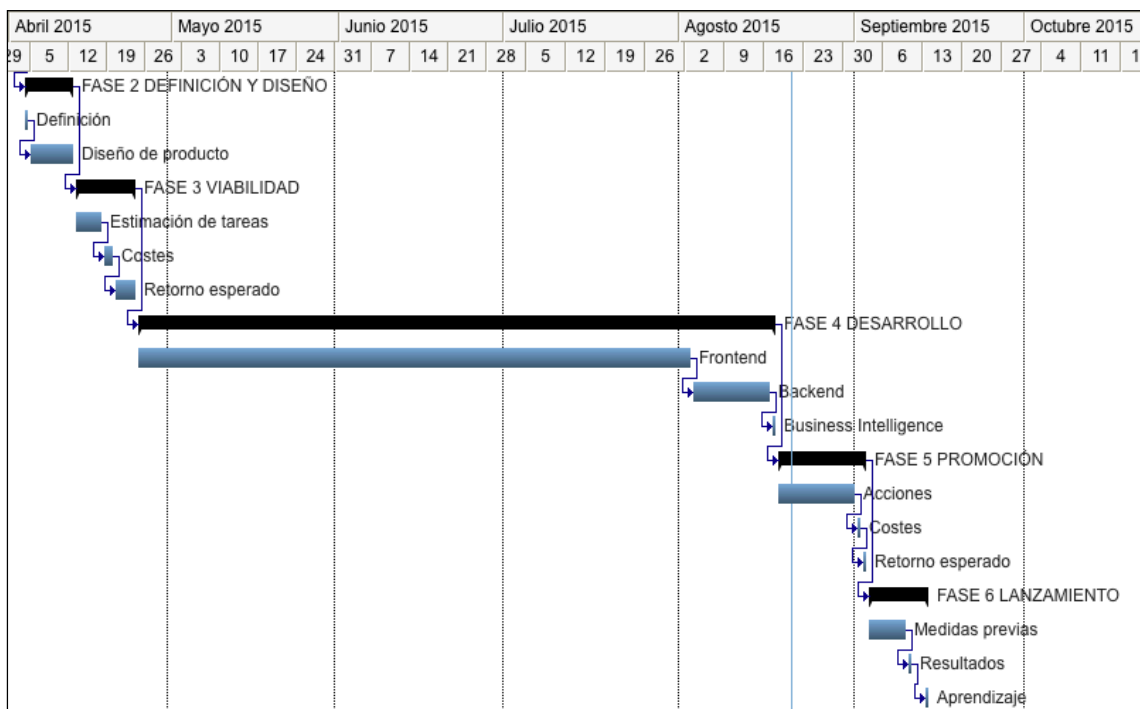


Fig. 15: Diagrama de Gantt del proyecto

Al disponer de un único recurso para la elaboración de este proyecto, la relación de las diferentes fases ha presenta un modelo de desarrollo en cascada (Fig. 15). En consecuencia, el tiempo necesario para el desarrollo integral del proyecto se ha definido en cinco meses y medio.

3.5. Conclusiones de la viabilidad del proyecto

Para identificar la viabilidad del proyecto, se han analizado los resultados obtenidos en los puntos anteriores generando las siguientes afirmaciones:

- Se disponen de todas las herramientas necesarias: entornos de programación y licencias.
- Aún sin disponer de los conocimientos necesarios para su producción, existe un compromiso de aprendizaje que entra dentro de la planificación.
- Teniendo en cuenta que los costes iniciales se reducen gracias al desarrollo del producto por cuenta propia y que los gastos de mantenimiento no nos excesivos, sí es posible soportar los costes identificados.
- Aunque soportar los impuestos de sociedades supone no respetar el objetivo de los 10€ de margen de beneficio, el beneficio por unidad de 8€ se aproxima lo suficiente.
- Al ser un producto relativamente económico y de complejidad baja, la inversión no supone un riesgo importante de endeudamiento.

- Teniendo en cuenta que el escenario de Éxito está vinculado a unos beneficios netos de 400.000€ y que el proyecto no requiere gastos adicionales de personal u oficinas, el escenario de Éxito es muy atractivo.
- Y por último, el tiempo de desarrollo de 6 meses entra dentro de lo previsto para este trabajo.

En definitiva y teniendo en cuenta las afirmaciones anteriores, el desarrollo del proyecto definido se ha dado como viable.

CAPÍTULO 4. DESARROLLO DEL PRODUCTO

Esta fase del proyecto se ha dedicado íntegramente a la construcción o desarrollo del producto usando las tecnologías o herramientas previamente identificadas y siguiendo, dentro de lo posible, la planificación prevista.

Para este proyecto y haciendo una referente a la arquitectura cliente-servidor seleccionada, el desarrollo del producto está dividida en tres partes: desarrollo del cliente (Frontend), desarrollo del servidor (Backend) y puesta en marcha del back office.

4.1. Desarrollo del cliente

El cliente (Frontend) o medio para interactuar con el usuario, es una de las piedras angulares de un un producto de ingeniería de software. Es la interfaz que permite al usuario acceder al valor ofrecido por el producto.

Para este proyecto, se ha elegido como medio o mecanismo de interacción con el usuario una app para iOS desarrollada en Swift.

4.1.1. Pasos previos para programar en Swift

Para programar en Swift se ha requerido disponer de una cuenta de desarrollador de Apple y descargar el entorno de desarrollo integrado XCode.

Una vez la cuenta creada, el acceso, la descarga e instalación del entorno no ha sido compleja. En apenas 10 minutos, es posible hacer el primer “Hello world!” en Swift.

```
println (“Hello world!”)
```

Fig. 16: Código del ejemplo “Hello world!” en Swift

Swift es un lenguaje de programación para Cocoa y Cocoa Touch, frameworks de programación OS X e iOS respectivamente. Un lenguaje de programación presentado a finales del 2014 que promete sustituir a Objective-C como lenguaje de programación oficial para estas plataformas.

La arquitectura de Swift está basada en el patrón “Modelo, Vista, Controlador” agilizando la planificación del desarrollo y facilitando la organización del código generado.

Para su aprendizaje se dedicó un total de 15 días dónde se impartieron dos cursos online (ver [18]), tutoriales y otros ejemplos disponibles en la red.

4.1.2. Modelos, Vistas y Controlador (MVC)

Gracias al patrón o estrategia de diseño de software ofrecida por Swift (Modelo, Vista y Controlador), se ha planificado todo el desarrollo del cliente usando los prototipos del producto como referencia.

Cada prototipo o vista, se ha implementado en Swift mediante el uso de un modelo de datos determinado, el diseño de una vista o interfaz y uno o varios controladores encargados de responder a eventos o acciones del usuario.

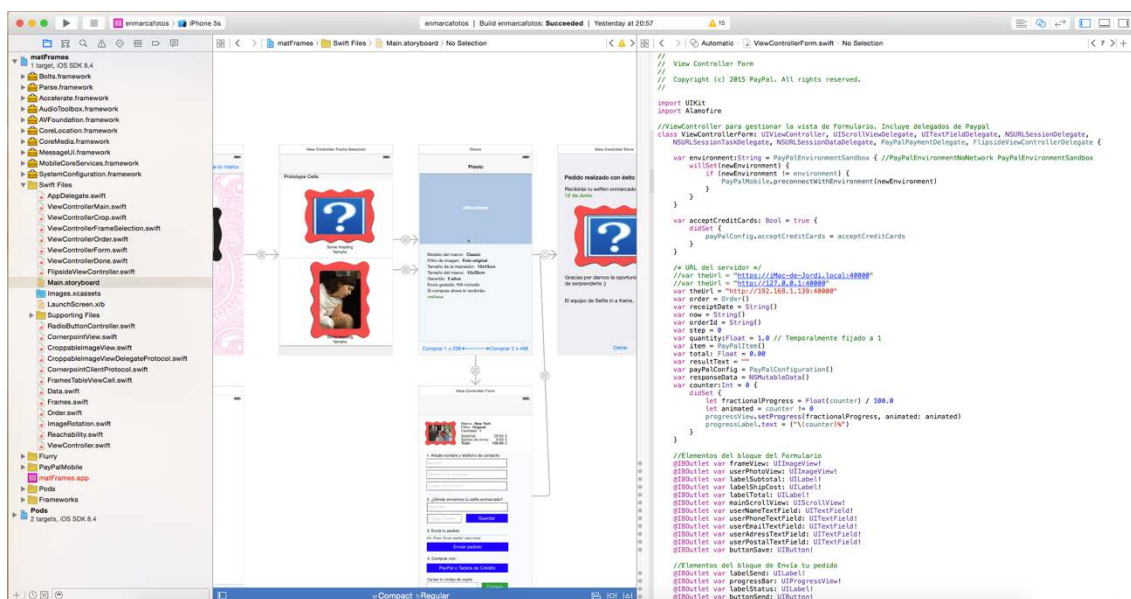


Fig. 17: Pantalla principal de XCode

4.1.2.1. Ejemplo: Vista 6 – Formulario de pago

Como ejemplo de una vista desarrollada para este proyecto, se ha elegido la Vista 6 de Formulario de pago. Una vista que incluye la gestión de elementos del interfaz básicos como campos de texto, comunicación con el servidor API REST para el envío del pedido, registro de eventos de Business Intelligence y comunicación con la pasarela de pago.

4.1.2.1.1 Modelo

El modelo de datos utilizado en esta vista, ha sido el modelo de datos utilizado en toda la aplicación. Este modelo de datos, llamado “Order”, representa todos los datos necesarios para realizar un pedido.

Cada una de las vistas de la aplicación añade al modelo los parámetros seleccionados por el usuario: foto, filtro de impresión, modelo de marco, datos personales y forma de pago. Este modelo de datos se ha transferido de una vista a otra utilizando la función del sistema “prepareForSegue”.

```
class Order{
    var originalPhoto = UIImage()
    var filter: String = "Original"
    var filterPhoto = UIImage()
    var orientation = String()
    var frameSelected = String()
    var framePreview = UIImage()
    var frameSize = String()
        var userName = String()
    var userPhone = String()
    var userEmail = String()
        var userAddress = String()
    var userPostal = String()
        var orderSent = false
    var paid = false
    var uniquePrice:Float = 29.90
    var shipCost:Float = 9.90
    var receiptDate = String()
}
```

Fig. 18: Modelo de datos del proyecto

4.1.2.1.2 Vista

XCode incorpora un editor visual de vistas llamado User Interface Builder. Gracias a este editor, se ha diseñado la vista usando tanto elementos visuales por defecto del sistema como incorporando elementos personalizados.

Con el objetivo de minimizar la envergadura del proyecto, se ha priorizado el uso de elementos visuales por defecto del sistema: Visor de imágenes, campos de texto, etiquetas, barra de navegación, botones, indicadores de progreso, etc.

Aún así, las múltiples opciones de personalización de los elementos por defecto del sistema, han facilitado la elaboración de interfaces de apariencia personalizada.

The screenshot displays a mobile application interface for a photo framing service. At the top, there are three icons: a yellow square with a white circle, a red cube, and a red square with a white circle. Below these is a battery icon. The main content area is divided into several sections:

- Product Preview:** A photo of a man in a red jacket is shown in a black frame with a scalloped edge. To the right of the photo, the following text is displayed:
Marco: **New York**
Filtro: **Original**
Cantidad: **1**
Subtotal: 29.00 €
Gastos de envío: 0.00 €
Total: 150.00 €
- Step 1: Añade nombre y teléfono de contacto**
This section contains three input fields: "Nombre", "Teléfono de contacto", and "Correo electrónico".
- Step 2: ¿Dónde enviamos tu selfie enmarcado?**
This section contains two input fields: "Dirección" and "Código Postal". A blue "Guardar" button is positioned to the right of the "Código Postal" field.
- Step 3: Envía tu pedido**
This section shows a progress indicator at 0% and the text "Pulsa 'Enviar pedido' para iniciar". A large blue "Enviar pedido" button is centered below.
- Step 4: Comprar con:**
A large blue button labeled "PayPal o Tarjeta de Crédito" is centered below.
- Gift Code:** Below the payment options, there is a text label "Canjea tu código de regalo", an input field, and a green "Canjear" button.

Fig. 19: Vista 6 del proyecto creada con User Interface Builder

4.1.2.1.3 Controlador

El controlador define toda la lógica necesaria para que la vista diseñada gestione correctamente nuestro modelo de datos, además de reaccionar a los eventos generados por el usuario. Sintetizando, el controlador es la parte programable de nuestra vista.

Para el controlador de esta vista, han sido necesarias más de 700 líneas de código organizadas en las siguientes funciones.

```
import UIKit
import Alamofire

//Declaración de la clase View Controller para la vista 6
class ViewControllerForm: UIViewController, UIScrollViewDelegate,
UITextFieldDelegate, NSURLSessionDelegate, NSURLSessionTaskDelegate,
NSURLSessionDataDelegate, PayPalPaymentDelegate,
FlipsideViewControllerDelegate

//Declaración de elementos creados con el User Interface Builder
@IBOutlet var frameView: UIImageView!
@IBOutlet var userPhotoView: UIImageView!
@IBOutlet var labelSubtotal: UILabel!

//Función del sistema ejecutada al finalizar la carga de la vista
override func viewDidLoad()

//Función del sistema ejecutada al aparecer la vista en pantalla
override func viewWillAppear(animated: Bool)

//Función para ocultar el teclado al tocar la pantalla
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent)

//Función para verificar y guardar datos de los campos de texto
@IBAction func userDataSaveButton(sender: UIButton)

//Función para enviar el modelo de datos al servidor API REST
@IBAction func startCount(sender: AnyObject)

//Función para realizar un pago por Paypal o tarjeta de crédito
@IBAction func buyPayPal(sender: AnyObject)

//Función para identificar algún cambio en campos de texto
func textFieldDidChange(textField: UITextField) {

//Función del sistema para el paso de parámetros entre vistas
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?)
```

Fig. 20: Código esquematizado del controlador de la vista 6

4.2. Desarrollo del Servidor

El servidor es la parte del producto que atiende a las peticiones del cliente, las procesa y devuelve la información correspondiente.

Para este proyecto, se han incluido en la parte de servidor un total de cuatro entidades dedicadas a ofrecer los servicios de consultas o API REST, servicios de gestión de notificaciones Push, servicios de e-mail y servicios de Business Intelligence o registro de eventos generados por la interacción del usuario con la aplicación.

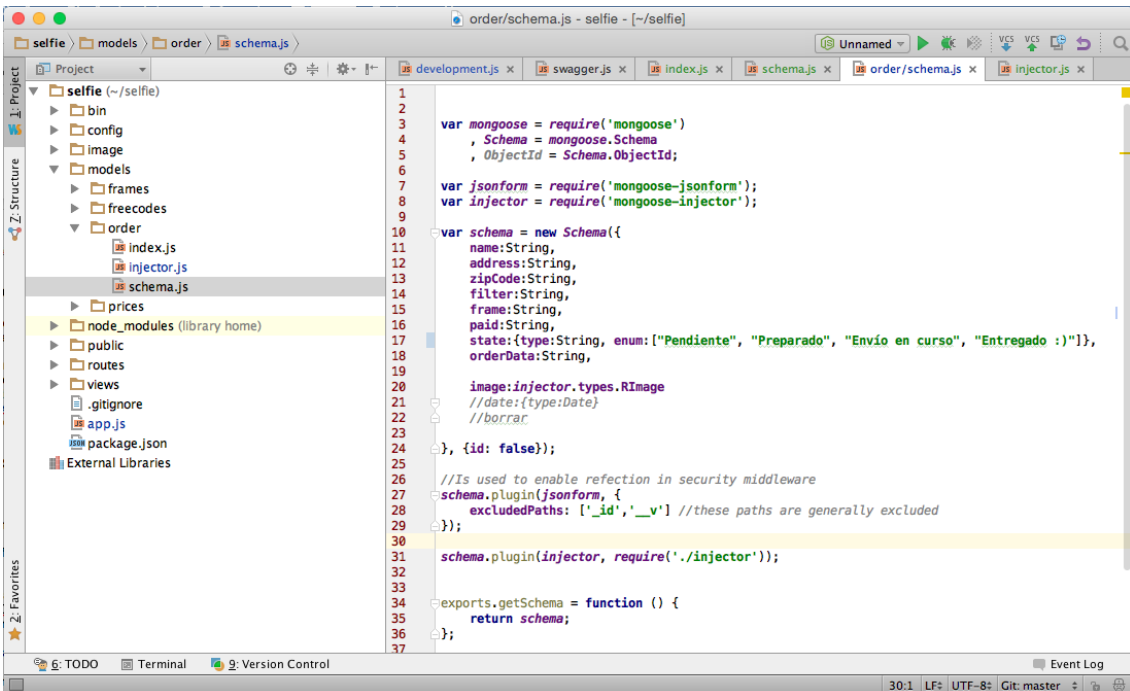
4.2.1. Servidor API REST

Para la elaboración del servicio de consultas y siguiendo lo estipulado en el diseño técnico, se ha configurado el paquete tecnológico conocido con el acrónimo MEAN.

Su configuración o puesta en marcha se ha ejecutado con facilidad gracias al soporte ofrecido por el equipo de desarrollo de AlterAid.

Por un lado, se ha instalado el servicio de gestión de bases de datos ofrecido por MongoDB y usado la aplicación desarrollada en Express por AlterAid para la creación de servicios API REST.

Gracias a esta aplicación, se han creado los diferentes endpoints necesarios para el funcionamiento del producto. Como definición previa, un endpoint es básicamente una URL que permite a un cliente acceder a los servicios de consulta o registro de información ofrecidos por el servidor.



```

1
2
3 var mongoose = require('mongoose')
4   , Schema = mongoose.Schema
5   , ObjectId = Schema.ObjectId;
6
7 var jsonform = require('mongoose-jsonform');
8 var injector = require('mongoose-injector');
9
10 var schema = new Schema({
11   name:String,
12   address:String,
13   zipCode:String,
14   filter:String,
15   frame:String,
16   paid:String,
17   state:{type:String, enum:["Pendiente", "Preparado", "Envío en curso", "Entregado :)],
18   orderData:String,
19
20   image:injector.types.RImage
21   //date:{type:Date}
22   //borrac
23
24 }, {id: false});
25
26 //Is used to enable reflection in security middleware
27 schema.plugin(jsonform, {
28   excludedPaths: ['_id', '.__v'] //these paths are generally excluded
29 });
30
31 schema.plugin(injector, require('./injector'));
32
33
34 exports.getSchema = function () {
35   return schema;
36 };
37

```

Fig. 21: Código del servidor API REST

4.2.1.1.1 Endpoints del Servidor API REST

Para que el cliente pueda realizar las diferentes consultas previamente definidas, se han creado los siguientes endpoints en el servidor API REST:

- Endpoint de consulta de los precios
http://dominio:puerto/prices
- Endpoint de consulta del catálogo de marcos disponibles
http://dominio:puerto/frames
- Endpoint de envío y modificación de pedidos
http://dominio:puerto/orders
- Endpoint de consulta de códigos de promoción
http://dominio:puerto/freecodes

Gracias a estos endpoints y a los métodos HTTP ofrecidos por servicios API REST (GET, POST y PUT), el cliente puede consultar precios al arrancar (GET), descargar el catálogo de marcos (GET), enviar pedidos (POST), modificar parámetros de un pedido (PUT) y consultar los códigos de promoción (GET).

Ejemplo de consulta GET del cliente para obtener los precios:

```
let myUrlPrices = "http://dominio:puerto/prices"
Alamofire.request(.GET, myUrlPrices)
    .responseJSON { (_, _, JSON, _) in
        var prices = JSON!.valueForKey("result") as! NSArray
```

Fig. 22: Ejemplo de consulta GET realizada por el cliente para obtener precios

Como se observa en el ejemplo de consulta GET (Fig. 22), se ha usado la clase Alamofire para realizar la consulta. Alamofire es un framework para Swift que permite agilizar la programación de consultas a servicios API REST usando JSON como estructura de datos.

```
let myUrlOrders = "http://dominio:puerto/orders"
Alamofire.request(.POST, myUrlOrders, parameters: Order)
```

Fig. 23: Ejemplo de consulta POST realizada por el cliente para guardar un pedido

4.2.2. Servidor de notificaciones

Para poder enviar mensajes cortos desde el Back Office del producto al cliente usando tecnología Push, se ha utilizado el servicio ofrecido por Parse.

Para su integración, se ha incluido framework y registrado el servicio en el código del cliente, además de crear los certificados de autenticación requeridos por Apple para el uso de notificaciones Push.

```
import Parse

...

//Registro para las Push Notifications
if application.applicationState != UIApplicationState.Background {

let preBackgroundPush =
!application.respondsToSelector("backgroundRefreshStatus")

let oldPushHandlerOnly =
!self.respondsToSelector("application:didReceiveRemoteNotification:fetchCompletionHandler:")

var pushPayload = false

if let options = launchOptions {
pushPayload = options[UIApplicationLaunchOptionsRemoteNotificationKey] != nil

if (preBackgroundPush || oldPushHandlerOnly || pushPayload) {
PFAnalytics.trackAppOpenedWithLaunchOptionsInBackground(launchOptions,
block: nil)

if application.respondsToSelector("registerUserNotificationSettings:") {

let userNotificationTypes = UIUserNotificationType.Alert |
UIUserNotificationType.Badge | UIUserNotificationType.Sound
let settings = UIUserNotificationSettings(forTypes: userNotificationTypes, categories:
nil)
application.registerUserNotificationSettings(settings)
application.registerForRemoteNotifications()}}
```

Fig. 24: Código de importación de librería Parse y registro del servicio de notificaciones

4.2.3. Servidor de e-mail

Para el servicio de envío y recepción de correos electrónicos no se ha implementado con la intención de minimizar la envergadura del proyecto. En su lugar, se ha optado por un servicio de gestión de correo convencional. Una cuenta de Gmail especialmente creada para contactar con los usuarios.

4.2.4. Servidor Business Intelligence

Para medir y procesar la información generada por el usuario mientras utiliza el cliente, se ha elegido el servicio ofrecido por Flurry. Un servicio de analítica especializado en aplicaciones que se encarga de registrar parámetros como número de instalaciones del cliente, frecuencia de uso, número de sesiones, duración de cada sesión, páginas vistas, entre otros.

Además de estos parámetros por defecto, Flurry permite crear eventos propios para medir aquellas acciones personalizadas que pueden ser clave para analizar el funcionamiento del producto y actuar en consecuencia.

A diferencia que con Parse, para empezar a medir los parámetros por defecto ofrecidos, ha sido necesario importar su librería (Flurry.h + libFlurry) al proyecto de nuestro cliente e inicializarlo.

```
Flurry.startSession("XMMQBVDXYCWCJ2KB9Z65")  
Flurry.logEvent("Started Application")  
Flurry.logAllPageViewsForTarget(UINavigationController)
```

Fig. 25: Código de iniciación de Parse

Para este proyecto, además de los parámetros por defecto, se han creado una serie de eventos con la intención de conocer con detalle como interactúa el usuario con el cliente.

Para crear un evento personalizado, basta con añadir la siguiente línea de código en la parte del cliente deseado.

```
Flurry.logEvent("Nombre del evento personalizado")
```

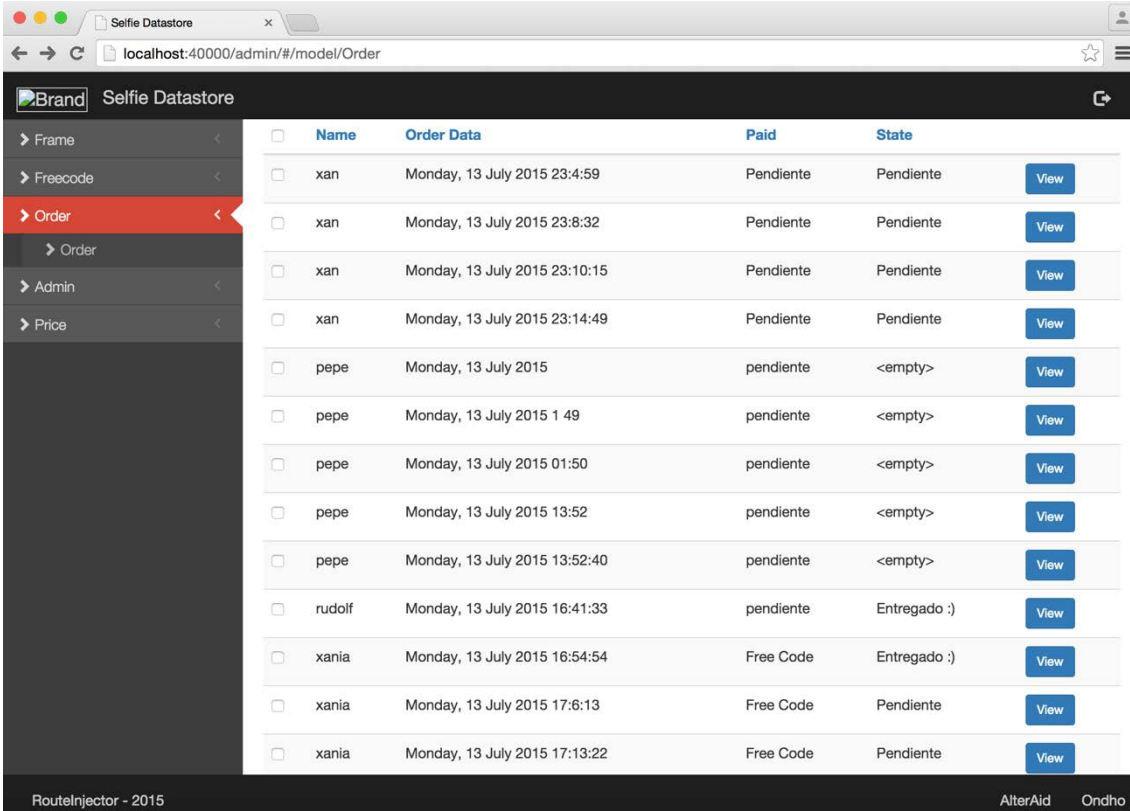
Fig. 26: Ejemplo de registro de un evento personalizado

4.3. Configuración del back office

Para la gestión de nuestro producto, se identificó la necesidad de disponer un back office con cuatro paneles independientes que faciliten la gestión de pedidos, el envío de notificaciones Push, la gestión de la comunicación con usuarios vía correo electrónico y un último panel para acceder y analizar la información generada por el usuario mientras utiliza el cliente.

4.3.1. Panel de gestión de pedidos

Como Panel de gestión de pedidos se ha utilizado el entorno gráfico desarrollado en Angular JS por AlterAid. Gracias a este, es posible listar y acceder a los detalles de todos los pedidos enviados por los usuarios, cambiar el estado de los pedidos, modificar los precios, editar el catálogo de marcos disponibles y gestionar los códigos de promoción.



<input type="checkbox"/>	Name	Order Data	Paid	State	
<input type="checkbox"/>	xan	Monday, 13 July 2015 23:4:59	Pendiente	Pendiente	View
<input type="checkbox"/>	xan	Monday, 13 July 2015 23:8:32	Pendiente	Pendiente	View
<input type="checkbox"/>	xan	Monday, 13 July 2015 23:10:15	Pendiente	Pendiente	View
<input type="checkbox"/>	xan	Monday, 13 July 2015 23:14:49	Pendiente	Pendiente	View
<input type="checkbox"/>	pepe	Monday, 13 July 2015	pendiente	<empty>	View
<input type="checkbox"/>	pepe	Monday, 13 July 2015 1 49	pendiente	<empty>	View
<input type="checkbox"/>	pepe	Monday, 13 July 2015 01:50	pendiente	<empty>	View
<input type="checkbox"/>	pepe	Monday, 13 July 2015 13:52	pendiente	<empty>	View
<input type="checkbox"/>	pepe	Monday, 13 July 2015 13:52:40	pendiente	<empty>	View
<input type="checkbox"/>	rudolf	Monday, 13 July 2015 16:41:33	pendiente	Entregado :)	View
<input type="checkbox"/>	xania	Monday, 13 July 2015 16:54:54	Free Code	Entregado :)	View
<input type="checkbox"/>	xania	Monday, 13 July 2015 17:6:13	Free Code	Pendiente	View
<input type="checkbox"/>	xania	Monday, 13 July 2015 17:13:22	Free Code	Pendiente	View

Fig. 27: Panel de gestión de pedidos ofrecido por AlterAid

4.3.2. Panel de notificaciones

Gracias a la integración de Parse en el proyecto, es posible enviar mensajes a través de notificaciones Push desde su propio panel de control.

Además de enviar mensajes, el panel de notificaciones de Parse permite configurar una serie de parámetros adicionales para optimizar el ratio de apertura de mensajes por parte de los usuarios.

Por ejemplo, es posible segmentar el envío por sistema operativo, programar envíos periódicos o fechas concretas, cambiar el formato de la notificación e incluso hacer experimentos para mejorar resultados.

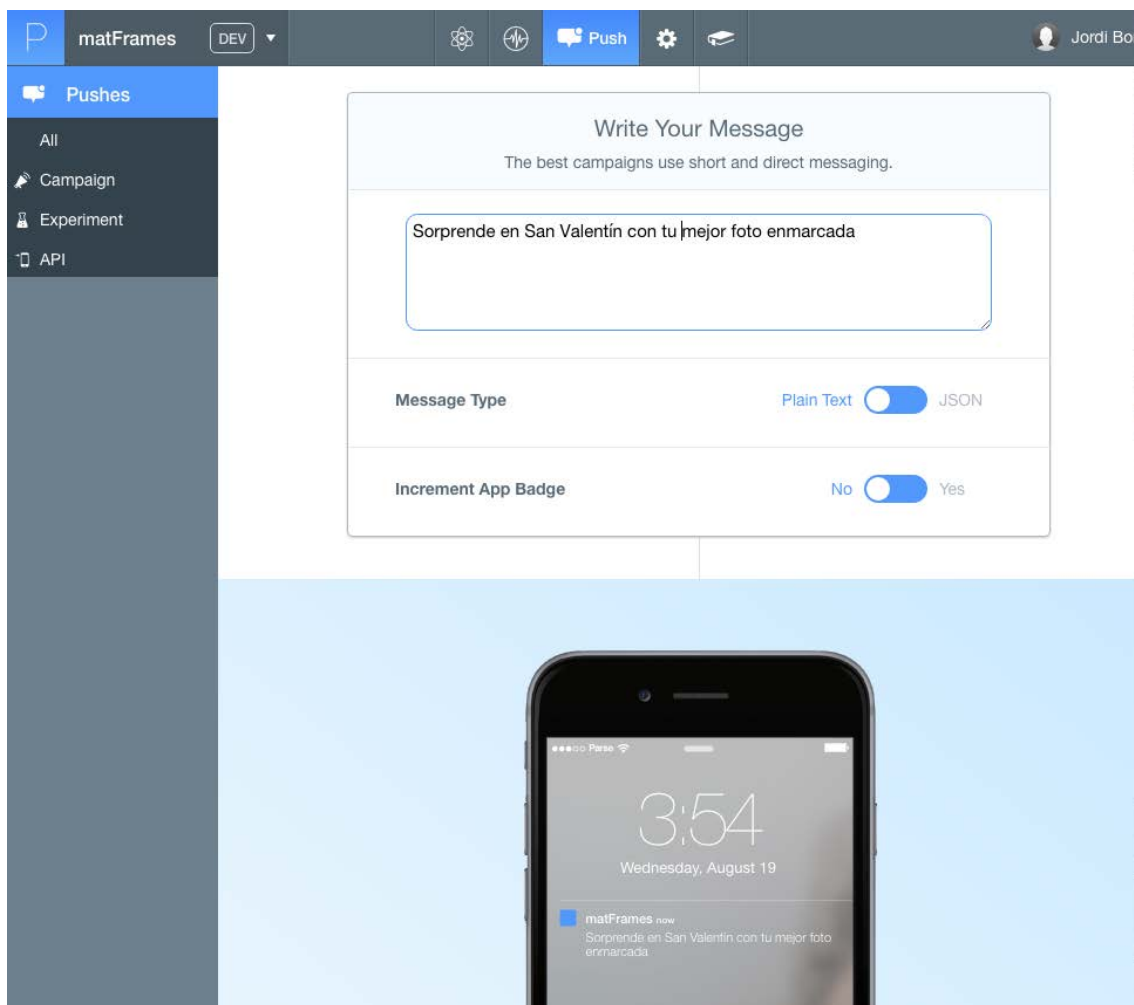


Fig. 28: Panel de notificaciones ofrecido por Parse

4.3.3. Panel de gestión de e-mails

Como se ha definido anteriormente, para la comunicación a través de e-mail con los usuarios se ha seleccionado una cuenta concreta de Gmail. La integración de un servidor de correo en nuestro sería viable una vez validada hipótesis de negocio.

4.3.4. Panel de Business Intelligence

Al igual que Parse, Flurry dispone de su propio panel para el análisis de la información generada por el usuario (información conocido en el sector de la analítica web como Key Parameter Identificators o KPIs).

El análisis de la evolución de estos KPIs permiten cuantificar la interacción del usuario con nuestro cliente (e incluso modelo de negocio) con la intención de corregir posibles carencias o identificar oportunidades.

Gracias al panel de Flurry, es posible conocer datos como número de usuarios únicos, usuarios activos, sesiones producidas, frecuencia de uso, niveles de retención, eventos producidos, etc.

Además de sus funcionalidades como herramienta de analítica, Flurry también permite monitorizar errores producidos en el cliente (como errores de conexión), controlar campañas de promoción y su nivel de cumplimiento; y crear nuevas campañas para conseguir nuevos usuarios

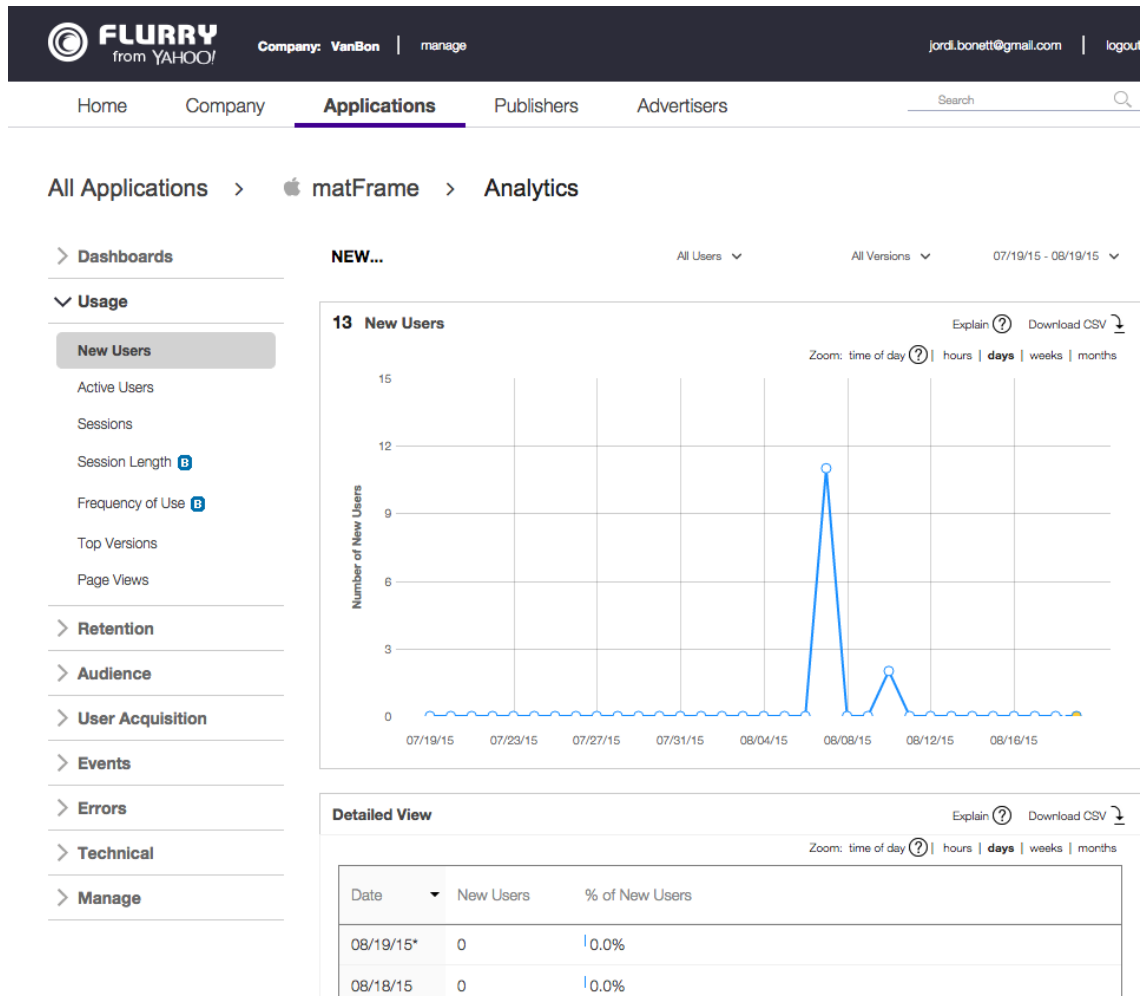


Fig. 29: Panel de analítica o Business Intelligence ofrecido por Flurry

4.4. Testeo del sistema

Una vez el producto desarrollado, se han realizado una serie de pruebas para asegurar la estabilidad y funcionalidad del servicio.

Prueba	Descripción	Resultado
Unitaria	Comprobar que los módulos del sistema funcionan correctamente de forma independiente	Correcto
Regresión	Comprobar el funcionamiento que los módulos del sistema ejecutados conjuntamente	Correcto
Sistema	Asegurar que el sistema funciona correctamente	Correcto
Stress	Comprobar el sistema funciona correctamente en diferentes escenarios de carga	Correcto
Carga	Validar el tiempo de respuesta de las consultas	Correcto
Volumen	Verificar que el espacio disponible en la base de datos es suficiente	Correcto
Recuperación	Comprobar el procedimiento de recuperación del sistema	Correcto
Compatibilidad	Identificar problemas de compatibilidad con diferentes versiones de iOS	Correcto
Integridad de Datos	Asegurar que el sistema no genera datos corruptos en la base de datos	Correcto
Seguridad	Verificar que los niveles de seguridad del sistema son los requeridos	Error
Ciclo de Negocio	Comprobar que el sistema atiende sin problemas a las necesidades del negocio	Correcto
Usabilidad	Asegurar que las diferentes interfaces del sistema se entienden sin problemas	Correcto
Instalación	Validar y verificar que el cliente se instala correctamente en los dispositivos requeridos	Correcto
Configuración	Validar y verificar que el cliente funciona correctamente en los dispositivos requeridos	Error
Funcional	Verificar que no existen errores al usar el producto de manera convencional	Correcto
Campo	Comprobar que el sistema funciona correctamente en un escenario real	Correcto
Alfa	Comprobar que el producto funciona correctamente bajo un entorno controlado	Correcto
Beta	Realizar la validación del producto por parte del usuario	Pendiente

Fig. 30: Pruebas realizadas para testear el sistema

Gracias a las pruebas realizadas para testear el sistema (Fig. 30), se ha identificado que los endpoints ofrecidos por el Servidor API REST eran de

carácter público comprometiendo la confidencialidad de los datos registrados en los pedidos.

Además de este problema de seguridad, gracias a la prueba de configuración se ha comprobado que en dispositivos iPad Retina, las vistas no están correctamente maquetadas.

Como solución para mejorar la seguridad, se ha optado por habilitar un sistema de autenticación basado en tokens en el Servidor API REST. Acción que modifica la estructura de los endpoints del servicio (<http://dominio:puerto/prices+token>)

Como solución al error de configuración, se ha planificado el rediseño de las vistas en el backlog del proyecto.

CAPÍTULO 5. PROMOCIÓN Y LANZAMIENTO

En este capítulo del proyecto se ha definido de las diferentes acciones de promoción del servicio y el procedimiento a seguir para asegurar un lanzamiento del servicio libre de sorpresas.

5.1. Promoción

El objetivo de la promoción de un producto es identificar, planificar y ejecutar una batería de acciones para informar a los consumidores de la existencia del producto.

Para ello, se ha requerido conocer al consumidor, analizar los medios de comunicación que suelen utilizar, elaborar una lista de acciones en esos medios, monitorizar el nivel de cumplimiento de la campaña y realizar las correcciones o ajustes correspondientes.

5.1.1. Conocer al consumidor

Para que una campaña de promoción sea eficiente, los anuncios sólo deben mostrarse al tipo de consumidor de nuestro producto. De ahí que conocerlo sea muy importante para que las decisiones tomadas sean las correctas.

En este proyecto, el perfil del cliente o consumidor se ha definido como una mujer de entre 20 a 60 años, de poder adquisitivo medio-alto y residente en España. Un perfil de usuario cuantificado, según el estudio de mercado (Fig. 2), en 46.200 usuarios receptivos.

5.1.2. Analizar los medios de comunicación consumidos

Una vez conocido al consumidor, se han identificado que tipo de medios de comunicación suelen atender: blogs, webs, revistas especializadas, radio, canales de TV., etc.

Tras realizar un ejercicio de observación, se ha identificado que los medios de comunicación posiblemente consumidos por el perfil de consumidor son:

- Blogs o webs de bebés
- Blogs o webs de artículos de belleza
- Blogs o webs de decoración
- Redes sociales: Facebook e Instagram
- Apps de comercio electrónico como LetsBonus
- Canales de TV tipo Antena3 o Divinity
- Emisoras de radio Casena Ser, M80, los 40 principales

5.1.3. Definir el objetivo de la campaña

En la mayoría de los casos, el objetivo de una campaña de promoción de un producto es darlo a conocer o estimular su demanda en el mercado. Es decir, incrementar sus ventas.

En otros casos, el objetivo es reforzar la imagen de marca que hay detrás de un producto o gama de productos determinado.

Pero en ambos casos, definir correctamente el objetivo de una campaña, es fundamental para conseguir los resultados esperados.

Teniendo en cuenta que el medio de comunicación con el usuario elegido es una aplicación para dispositivos iOS, se ha definido como objetivo principal de la campaña conseguir que los 46.200 usuarios receptivos instalen la aplicación y realicen un pedido.

5.1.4. Elaborar acciones para promocionar el producto

Tras identificar los medios de comunicación atendidos por el perfil del cliente y definido el objetivo de la campaña, se han elegido las acciones que están al alcance de los recursos disponibles.

Debido al presupuesto limitado para promoción, se han descartado aquellos medios de comunicación con audiencias masivas como Canales de TV o Radio.

En consecuencia, se han definido las siguientes acciones de promoción:

1. Crear landing page con información del producto
2. Identificar bloggers especializados en consumibles para bebés, artículos de belleza y decoración
3. Enviarles códigos de promoción para que prueben el producto de manera gratuita a cambio de un breve artículo en su blog
4. Crear cuentas de Facebook e Instagram y añadir contenidos periódicos sobre ejemplos de fotos enmarcadas, consejos sobre decoración, marcos especiales para festividades populares, etc.
5. Crear un concurso en Facebook para incrementar el número de "comparticiones"
6. Habilitar una campaña de 500€ de presupuesto en una campaña de Facebook correctamente "targetizada" para que se muestre sólo a las personas que cumplen con el perfil de cliente identificado
7. Habilitar una segunda campaña de 500€ en Admob (versión de Adwords especializada en aplicaciones) con las mismas características que en Facebook
8. Preparar una oferta de liquidación en LetsBonus reduciendo el precio del producto un 40%

5.1.5. Monitorizar y optimizar campañas de promoción

Los resultados de una campaña de promoción pueden variar considerablemente dependiendo las decisiones tomadas. Por ello, se ha identificado como importante controlar o monitorizar el nivel de cumplimiento (los resultados obtenidos) de cada una de las acciones definidas.

Es decir, para poder mejorar los resultados de una campaña de promoción es necesario monitorizar el parámetro que cuantifica el objetivo de la campaña: las ventas, las instalaciones, el registro de nuestros usuarios, los pedidos, etc.

Para este proyecto, gracias a la integración de Flurry, es posible controlar los siguientes parámetros:

- Número de impresiones del anuncio
- Número de instalaciones obtenidas
- Número de pedidos obtenidos
- Ratio de conversión de la campaña o nivel de cumplimiento

De esta manera, dependiendo los resultados de cada campaña es posible identificar mejoras y aplicarlas inmediatamente para mejorar su conversión y, en consecuencia, mejorar la efectividad de nuestras acciones de promoción.

5.2. Planificar el lanzamiento del servicio

Con la intención de evitar problemas de última hora, carencias en el servicio una vez activo y maximizar el impacto de la campaña de promoción, se ha planificado un procedimiento dónde se listan las pruebas y acciones a realizar antes, durante y después del lanzamiento.

5.2.1. Antes del lanzamiento

Como acciones a ejecutar antes del lanzamiento, se ha definido verificar el correcto funcionamiento del sistema e implementar las diferentes acciones de promoción definidas.

5.2.1.1. Prueba funcional del sistema

Consiste en controlar la estabilidad del sistema en cada modificación y cambio realizado antes del lanzamiento mediante una prueba funcional.

5.2.1.2. Crear landing page o página de promoción

El objetivo de esta acción es crear la página de promoción dónde se presenta el producto, se identifican sus características, sus beneficios y se muestra el día del lanzamiento.

5.2.1.3. Crear cuentas en redes sociales

Acción que requiere la creación de las cuentas de Facebook e Instagram lo antes posible para publicar contenidos previos y aumentar, en consecuencia, el número de seguidores.

5.2.1.4. Contactar con los medios seleccionados

El objetivo de esta acción es contactar previamente con los medios para presentarles con anterioridad y en exclusiva el servicio.

5.2.1.5. Crear anuncios

Crear anuncios en Adwords y Facebook con anterioridad y programar su activación con la intención de liberar tiempo y recursos para los días previos al lanzamiento.

5.2.2. Día previo al lanzamiento

Como acciones del día previo del lanzamiento, se ha definido realizar de nuevo un testeo del sistema y recordar a los medios el lanzamiento inminente del servicio.

5.2.2.1. Prueba funcional del sistema

De nuevo, validar la estabilidad del sistema en cada modificación y cambio realizado antes del lanzamiento mediante una prueba funcional.

5.2.2.2. Recordar la publicación a los medios

El objetivo de esta acción es avisar a los medios seleccionados para que preparen y planifiquen la publicación de los artículos conseguidos.

5.2.2.3. Recordar el lanzamiento en redes sociales

Publicar en redes sociales un preaviso de lanzamiento para generar sentimientos de fidelidad y exclusividad entre los seguidores.

5.2.3. Día del lanzamiento

El día del lanzamiento es un día clave para el producto o servicio desarrollado. Como procedimiento, se han planificado las siguientes acciones:

5.2.3.1. Prueba funcional del sistema en entorno controlado

Antes de lanzar el producto o subirlo a producción, realizar una última prueba funcional en el sistema para identificar problemas de último momento y agilizar su resolución.

5.2.3.2. Lanzar el producto o servicio

Acción que identifica el lanzamiento, subida o activación del servicio una vez verificado que el sistema funciona correctamente en un entorno controlado.

5.2.3.3. Prueba funcional del sistema en entorno real

Realizar una última prueba funcional del servicio en su entorno real para evitar problemas de configuración.

5.2.3.4. Activar y verificar la campaña de promoción

Activar las campañas de promoción en los medios de comunicación elegidos, verificar que las campañas se muestran correctamente y dar el OK a los medios para la publicación de los artículos previamente preparados.

5.2.3.5. Atención total

El objetivo de esta acción es atender dudas de los primeros usuarios y verificar que el sistema funciona correctamente en diferentes situaciones de carga.

5.2.4. Después del lanzamiento

Una vez el servicio activo o lanzado, se ha definido una serie de acciones para analizar logs de errores, controlar el rendimiento de la campaña, estudiar los datos recopilados por las herramientas de analítica y seguir ciclos de mejora.

5.2.4.1. Analizar datos del servicio

Consiste en revisar logs de errores y priorizar los errores encontrados para planificar o no su resolución.

5.2.4.2. Analizar conversiones de la campaña

Controlar el rendimiento de la campaña con la intención de identificar posibles acciones para mejorar su conversión.

5.2.4.3. Aplicar mejoras

Desarrollar las mejoras priorizadas y planificarlas en los siguientes Sprints de desarrollo.

5.2.4.4. Seguir ciclos de mejora

Recopilar información del rendimiento del producto, identificar mejoras, priorizarlas desde un punto de vista de negocio, desarrollar las mejoras de mayor impacto y ofrecerlas a los usuarios de forma constante y periódica.

CAPÍTULO 6. CONCLUSIONES DEL TRABAJO

Después de seis meses dedicados al estudio de metodologías ágiles, herramientas para analizar mercados, procedimientos para la creación de productos o servicios y aprender lenguajes y otros recursos técnicos para la elaboración del servicio desarrollado en este trabajo, se han obtenido una serie de conclusiones que se han decidido clasificar en dos puntos: conclusiones por objetivos del trabajo y conclusiones generales.

6.1. Conclusiones por objetivos del trabajo

Una vez finalizado el trabajo final de carrera y gracias a la consecución de los objetivos definidos, se han obtenido las siguientes conclusiones:

6.1.1. Definir un plan de proyecto

Respecto al primer objetivo del trabajo, se ha conseguido desarrollar el plan de proyecto reflejado en esta propia memoria. Un plan de proyecto que divide en cinco fases el desarrollo de un producto o servicio: estudio de mercado, diseño de producto, estudio de viabilidad, desarrollo y promoción.

Gracias al estudio de mercado, ha sido posible identificar y cuantificar un nicho de mercado lo suficientemente atractivo para que el servicio o producto definido tenga los retornos esperados.

La fase de diseño ha sido vital para minimizar costes de desarrollo identificando carencias del servicio antes de su desarrollo y diseñando su arquitectura técnica utilizando servicios ofrecidos por terceros.

Con el estudio de viabilidad se ha podido valorar en tiempo y esfuerzo la envergadura del trabajo y realizar los ajustes previos para acotar en tiempo las características o requisitos del servicio.

Una vez tomadas las decisiones a nivel de negocio, diseñado el producto y analizada su viabilidad, en la fase de desarrollo se ha desarrollado el servicio usando Scrum como metodología de trabajo.

Con la fase de promoción, ha permitido identificar una serie de acciones de promoción y seleccionar las viables teniendo en cuenta los recursos disponibles para este trabajo. Todo con la intención de maximizar la visibilidad del servicio una vez puesto en marcha.

6.1.2. Implementar una versión de prueba

Como segundo objetivo del trabajo, se ha conseguido desarrollar una versión beta del servicio telemático de impresión y enmarcado fotográfico mediante el uso de una arquitectura cliente-servidor y usando servicios ofrecidos por terceros.

Gracias al uso de servicios de tercero, se ha podido acotar considerablemente la envergadura del proyecto. Implementar desde cero toda la arquitectura del servicio hubiera supuesto duplicar el coste de esfuerzo del trabajo pasando de los 6 meses dedicados a un año como mínimo y tumbando, en consecuencia, la viabilidad del trabajo.

Aún así, durante la fase de desarrollo se consumió un 30% más de tiempo de lo planificado. Utilizar Swift como lenguaje de programación y MEAN como modelo para el servidor, supuso soportar una curva de aprendizaje que se extendió más de lo esperado. Pero una vez superada esta curva, el uso de estas tecnologías han agilizado la implementación de los paneles de gestión (Back Office) del servicio.

Por último, trabajar con una arquitectura cliente-servidor mediante el uso de un servicio API REST para la consulta y gestión de información ha sido realmente eficiente. Una vez configurado el servicio de API REST, añadir nuevas interacciones entre el cliente y el servidor ha sido rápido y sencillo. Una solución que asegura la estabilidad y escalabilidad del servicio.

6.1.3. Integrar y poner a prueba metodologías ágiles

El uso de metodologías ágiles en este trabajo final de carrera ha permitido controlar constantemente la envergadura del proyecto, además de focalizar los esfuerzos en cada fase sólo en aquellas funcionalidades o requisitos que realmente han sido vitales para el servicio final.

De todos modos, tanto Scrum como Lean Startup son metodologías que proponen una serie de buenas prácticas para mejorar la eficiencia del trabajo en equipo y desarrollo de productos o empresas respectivamente. Propuestas que han sido usadas y adaptadas a este trabajo ya que en su formato original hubieran incrementado el coste de esfuerzo.

De todos modos, el uso de metodologías ágiles ha sido vital para que este trabajo final de carrera pudiera finalizarse en la ventana de tiempo disponible.

6.1.4. Probar la versión de prueba en un entorno controlado

Una vez finalizada la versión de prueba o versión beta del servicio telemático de impresión y enmarcado fotográfico, ha sido posible probar su funcionamiento en un entorno controlado.

Gracias a estas pruebas, se ha podido localizar carencias en el formulario de datos del usuario, habilitar un nuevo modelo de datos para soportar códigos de descuento y mejorar la vista de ajustes con una vista adicional de edición para ajustar aquellas fotografías no tomadas desde el propio dispositivo.

Además, las pruebas realizadas en un entorno controlado han permitido validar el funcionamiento correcto de las diferentes entidades del servidor y afirmar que el cliente presenta una interfaz sencilla e intuitiva permitiendo realizar un pedido con facilidad.

6.1.5. Probar la versión de prueba en un entorno real

Debido al desvío de la planificación en la fase de desarrollo y a la parte logística requerida en el servicio no planificada con exactitud, las pruebas del servicio en un entorno real quedan pendientes a la hora de crear la memoria de este trabajo final de carrera.

6.2. Conclusiones generales

Además de las conclusiones vinculadas con los objetivos definidos al inicio de este trabajo final de carrera, se han obtenido las siguientes conclusiones genéricas. No por ello, menos importantes:

6.2.1. Una buena planificación ahorra tiempo y esfuerzo

Además de la capacidad resolutive como pilar fundamental para el desarrollo de servicios o productos, una buena planificación ha permitido evaluar constantemente la envergadura del proyecto, buscar alternativas para agilizar el desarrollo y seguir la planificación definida previamente. Gracias al uso de Scrum y Lean Startup ha sido posible priorizar y optimizar el tiempo requerido para la ejecución de cada uno de los pasos del proyecto, maximizando la eficiencia de los recursos disponibles.

6.2.2. Observar es vital para identificar oportunidades

Al igual que la observación es uno de los pasos definidos en el método científico, en el desarrollo de un producto o servicio también ha sido un paso básico para la toma de decisiones. Gracias a la observación, se ha identificado el nicho de mercado, definido la hipótesis de negocio y diseñado el servicio correspondiente para cubrir la demanda prevista.

6.2.3. Definir, diseñar y validar antes de programar

Disponer de una definición, diseño y validación previa del servicio, ha permitido agilizar y acelerar considerablemente su desarrollo ya que la gran mayoría de las decisiones más complejas ya estaban tomadas. Gracias a estos pasos previos, se ha evitado cambios drásticos de arquitectura, funcionalidades o herramientas que podrían haber retrasado y encarecido considerablemente el desarrollo integral del servicio.

6.2.4. La importancia de la capacidad resolutive

Solucionar problemas es, sin duda, la habilidad adquirida en la carrera de ingeniería técnica más útil para desarrollar productos o servicios. Durante este proyecto, se han realizado ejercicios de documentación, análisis y definición, para solucionar los múltiples problemas encontrados durante su desarrollo.

Siendo así, sin la capacidad resolutive adquirida en la carrera de ingeniería técnica de telecomunicaciones, hubiera sido muy difícil la consecución de este trabajo.

BIBLIOGRAFÍA

- [1] Álvarez, A. *Métodos Ágiles y Scrum*. Madrid, España: Anaya.
- [2] Brandt, R. L. (2011). *un click*. San Francisco, EEUU: Gestión 2000.
- [3] *Business Studies*. (s.f.).
<http://www.bbc.co.uk/schools/gcsebitesize/business/marketing/brandingandpackagingrev2.shtml>
- [4] Ferriss, T. (2012). *La semana laboral de 4 horas*. Nueva York, EEUU: RBA Libros.
- [5] *Flurry.com*. <http://www.flurry.com/>
- [6] *Google Trends*. <https://www.google.es/trends/>.
- [7] Inc., A. *The Swift Programming Language*.
https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/GuidedTour.html
- [8] Isaacson, W. (2011). *Steve Jobs: La Biografía*. Palo Alto, EEUU: Debate.
- [9] Kantar. *Kantar World Panel*. <http://www.kantarworldpanel.com/>
- [10] *Mean Stack*. Obtenido de Mean.io: <http://learn.mean.io/>
- [11] Osterwalder, A. (2014). *Value Proposition Design*. Winterthur, Suiza: Wiley.
- [12] *Parse*. Obtenido de Parse.com: <https://www.parse.com/>
- [13] Ries, E. (2012). *El método Lean Startup*. San Francisco, EEUU: Deusto.
- [14] Ries, E. (s.f.). *Startup Lessons Learned*.
<http://www.startuplessonslearned.com>
- [15] Rodgers, A. M. (2013). *Métodos de investigación para el diseño de producto*. Dublin, Irlanda: Blume.
- [16] Scheid, J. *Project Management Methodologies*
<http://www.brighthubpm.com/methods-strategies/67087-project-management-methodologies-how-do-they-compare/>
- [17] Telefónica, F. (2014). *Sociedad de la información 2014*.
http://www.fundaciontelefonica.com/arte_cultura/publicaciones-listado/pagina-item-publicaciones/?itempubli=323
- [18] Villalvazo, J. *Desarrollo iOS con Swift*. Obtenido de Udemy:
<https://www.udemy.com/desarrollo-ios-con-swift/>

[19]Hoffmann. *Hoffman.es*. <http://www.hofmann.es/>.

[20]Paypal. *Developer Paypal*.
<https://developer.paypal.com/docs/integration/mobile/mobile-sdk-overview/>

[21]Stevens, M. *Vsauce*. <http://vsauce.com>
https://www.youtube.com/watch?v=4e_kz79tjb8&feature=youtu.be

[22]Cheerz. <https://www.cheerz.com/es>

[23]AlterAid. <http://www.alteraid.com/>

ÍNDICE DE FIGURAS

Fig. 1: Gráfica de interés de Google Trends.....	10
Fig. 2: Tabla de parámetros usada para cuantificar el nicho.....	11
Fig. 3: Resultados de la presentación del prototipo.....	12
Fig. 4: Mapa mental del servicio de impresión y enmarcado fotográfico	14
Fig. 5: Prototipos del servicio de enmarcado fotográfico.....	18
Fig. 6: User Journey de un nuevo usuario.....	19
Fig. 7: Diagrama de bloques	20
Fig. 8: Diagrama de secuencia.....	22
Fig. 9: Muestra de las tareas valoradas y priorizadas usando Scrum	26
Fig. 10: Inversión inicial requerida.....	27
Fig. 11: Cálculo del coste de producción y beneficio por unidad.....	28
Fig. 12: Cálculo del retorno esperado	29
Fig. 13: Escenarios económicos definidos	29
Fig. 14: Relación de tareas y duración	30
Fig. 15: Diagrama de Gantt del proyecto.....	31
Fig. 16: Código del ejemplo "Hello world!" en Swift.....	33
Fig. 17: Pantalla principal de XCode	34
Fig. 18: Modelo de datos del proyecto	35
Fig. 19: Vista 6 del proyecto creada con User Interface Builder.....	36
Fig. 20: Código esquematizado del controlador de la vista 6	37
Fig. 21: Código del servidor API REST	38
Fig. 22: Ejemplo de consulta GET realizada para obtener precios.....	39
Fig. 23: Ejemplo de consulta POST realizada para guardar un pedido.....	39
Fig. 24: Código de importación de Parse y registro de notificaciones	40
Fig. 25: Código de iniciación de Parse	41
Fig. 26: Ejemplo de registro de un evento personalizado.....	41
Fig. 27: Panel de gestión de pedidos	42
Fig. 28: Panel de notificaciones ofrecido por Parse	43
Fig. 29: Panel de analítica o Business Intelligence ofrecido por Flurry	44
Fig. 30: Pruebas realizadas para testear el sistema.....	45



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANEXOS

Título: Servicio de impresión y enmarcado fotográfico

Titulación: Ingeniería Técnica de Telecomunicaciones - Telemática

Autor: Jordi Bonet Caño

Director: Toni Oller

Fecha: 7 de septiembre del 2015

ANEXO I: ACRÓNIMOS Y TECNICISMOS

API REST: Servicio de gestión de información que atiende a consultas realizadas mediante el uso del protocolo http. Siendo así, se usan urls para realizar consultas o registro de información. Como respuesta, API REST ofrece los datos en diferentes formatos: texto plano, XML, JSON, etc.

Back Office: Conjunto de herramientas de apoyo para gestionar un servicio: gestión de pedidos, comunicación con usuarios, gestión del catálogo, configuración de consultas o endpoints.

Business Intelligence: Conjunto de estrategias y aspectos relevantes enfocados a la administración y creación de conocimiento sobre el medio mediante el análisis de parámetros cuantificables.

Endpoint: URL que permite a un cliente acceder a los servicios de consulta o registro de información ofrecidos por un servidor API REST.

Framework: Desde la perspectiva de diseño de software, un framework es un conjunto estandarizado de funciones y otros recursos para desarrollar aplicaciones o servicios.

Frontend y Backend: En diseño de software se conoce como frontend la parte del software que interactúa con el o los usuarios y el backend como la parte que procesa la entrada desde el frontend. La separación del sistema en frontend y backend es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas. La idea general es que el frontend sea el responsable de recolectar los datos de entrada del usuario y los transforma ajustándolos a las especificaciones que demanda el backend para poder procesarlos.

Hipótesis de negocio: Afirmación sobre cómo se comportará una idea de negocio sin tener los datos suficientes para afirmarlo con certeza.

KPIs: Del inglés key performance indicator, indicador clave de rendimiento es un parámetro cuantificable que mide el rendimiento de un proceso determinado: número de usuarios, número de impresiones, número de ventas, número de visitas, etc.

Landing Page: Página web dónde llega el usuario de manera orgánica o referenciada a través de otra web o un anuncio. En la mayoría de los casos esta página web es una extensión del anuncio de promoción. Muestra los detalles del producto o servicio ofrecido.

Lean Startup: Metodología ágil de gestión que ayuda a simplificar la creación de modelos de negocio recurrentes, rentables y escalables. Se compone de una serie de principios que tienen como objetivo minimizar la probabilidad de fracaso de una Startup, producto o servicio. A diferencia de las herramientas de gestión tradicional, Lean Startup propone una estrategia constante de aprendizaje. Una

alternativa que sustituye los rígidos y costosos planes de negocios por una estructura escalable basada en experimentos cuantificables.

MEAN: Es un paquete de software de código abierto para construir sitios y aplicaciones web de forma dinámica. El acrónimo MEAN se refiere a Mongo DB, Express, Angular.js y Node.js

Nicho de mercado: Término de mercadotecnia utilizado para referirse a una porción de un segmento de mercado en la que los individuos poseen características y necesidades homogéneas, y estas últimas no están del todo cubiertas por la oferta general del mercado.

Notificaciones Push: Mensajes que se envían de forma directa a dispositivos móviles. Las notificaciones Push ayudan a los desarrolladores independientes y

Scrum: Metodología o proceso de trabajo en el cual se aplican una serie de buenas prácticas para mejorar la productividad de un equipo de desarrollo. Se caracteriza por fomentar el desarrollo incremental, por trabajar con equipos autosuficientes y por solapar diferentes fases de desarrollo. Una de las principales características del Scrum son sus entregas parciales y regulares del producto final denominados Sprints.

Sprint: Ciclo de desarrollo dónde se trabaja cada una de las diferentes entregas parciales del producto final.

Targetización: Definición de los criterios por los cuales un anuncio se mostrará al usuario.

User Interface Builder: Herramienta de desarrollo que simplifica de un modo visual la creación de entornos o interfaces de usuario.

User Journey: Herramienta que permite medir la experiencia de un usuario mientras interactúa con un producto. Gracias a esta herramienta, es posible identificar puntos de mejora y realizar los ajustes necesarios para ofrecer una experiencia agradable al usuario.

ANEXO II: ESTUDIO MEDIOAMBIENTAL

Como segundo Anexo de este trabajo final de carrera, se plantea un estudio medioambiental dónde recopila la información necesaria para evaluar el impacto medioambiental del servicio ofrecido.

Para la evaluación, se ha realizado un informe siguiendo la normativa ISO 14064 y GhG Protocol (ver [1]).

6.3. Descripción breve del servicio

El servicio ofrecido es un servicio de impresión y enmarcado fotográfico dónde el usuario utiliza una aplicación móvil para realizar un pedido. El pedido consta de una foto imprimida en papel fotográfico, un marco de madera MDF, un empaquetado especial y el envío del producto ensamblado a domicilio.

6.4. Explicación de los límites organizacionales de la empresa

La actividad de la empresa se basa en el ensamblaje de elementos producidos por terceros. Por un lado, marcos producidos por un fabricante nacional y por otro, impresiones realizadas mediante una impresora subliminal profesional alimentada por rollos de papel fotográfico suministrados por un proveedor internacional.

De esta manera, como Alcance 1, emisiones directas generadas por la actividad, únicamente se contempla el traslado de los pedidos desde el punto de ensamblaje al punto logístico de envío nacional.

En consecuencia, como Alcance 2, se incluyen las emisiones indirectas generadas por el uso de materiales de terceros como papel, marcos y cajas; además de las emisiones indirectas derivadas al transporte del producto ensamblado o consumo eléctrico.

6.5. Periodo de reporte cubierto

En este estudio se realiza una previsión anual de los gases de efecto invernadero (GEI) generados por la actividad

6.6. Metodología utilizada

Como metodología utilizada en el informe, se ha seguido ISO 14064 y GhG Protocol para comercios pequeños.

6.7. Emisiones totales o huella de carbono

Alcance 1 - Emisiones directas			
Fuente de emisión de GEI			
Consumo de combustible	m ³	FE	Kg CO ₂
Consumo de gas natural	0	0,2	0
Consumo de gasoleo	0	2,68	0
Transporte	Km	FE	Kg CO ₂
Vehículo propiedad del servicio	1.060	0,119	126
Total emisiones directas			126

Alcance 2 - Emisiones indirectas				
Fuente de emisión de GEI				
Consumo de energía eléctrica	Kwh	Kwh anuales (12 horas día)	FE	Kg CO ₂
Ordenador + Servidores	0,3	1314	0,649	853
Impresora	0,05	219	0,649	142
Iluminación	0,02	87,6	0,649	57
Material de ensamblaje	Kg	Kg anuales	FE	Kg CO ₂
Papel fotográfico	18,25	18,25	1,8	33
Marco de madera	1095	1095	1,2	1314
Caja	365	365	1,6	584
Transporte de mercancía	Km	Km anuales	FE	Kg CO ₂
Vehículo de la agencia	100.000	100.000	0,119	11.900
Total emisiones indirectas				14.883

Huella de carbono	15 ton CO₂
--------------------------	------------------------------

FE: Factor de emisión de CO₂ ofrecido por IDEA

GEI: Gases de Efecto Invernadero

Según se observa en la tabla de cálculo de GEI, el 99,99% de los GEI emitidos son generados por las emisiones indirectas (Alcance 2), siendo el transporte de mercancías la principal fuente de emisión de GEI con un 73% del total de CO₂ generado por la actividad.

En conclusión, el servicio de impresión y enmarcado fotográfico tiene una huella de carbono de 15 toneladas de CO₂ al año. Teniendo en cuenta que el nivel de captura de CO₂ media del planeta está definido en 2,6 toneladas de CO₂ por hectárea al año, la huella ecológica del servicio es de 5,7 hectáreas globales.

6.8. Acciones para la reducción de la huella de carbono

Para reducir las emisiones de GEI directas (Alcance 1), se propone agrupar los envíos de manera semanal. Siendo así, se podrían ahorrar mas de 600 Km y reducir, en consecuencia, el 66% del CO₂ generado.

Como acciones para la reducción de las emisiones de GEI indirectas (Alcance 2), se ha seleccionado un proveedor de marcos con sello de gestión forestal responsable (PEFC y FSC) que asegura que la madera usada para los marcos provienen de bosques gestionados de manera sostenible.

Además, con la intención de minimizar las emisiones provocadas por el transporte de los pedidos, se ha diseñado un sistema escalable dónde las regiones con mayor demanda dispondrán de un punto de suministro. De esta manera, se prevé reducir un 33% el CO₂ generado por el vehículo de la agencia de transporte.

6.9. Referencias del Anexo II:

- [1] Ministerio de agricultura y Medio Ambiente
http://www.magrama.gob.es/es/ministerio/servicios/publicaciones/manual_comercios_final_tcm7-230140.pdf
- [2] Footprintnetwork.org
<http://www.footprintnetwork.org/images/uploads/Huella%20ecologica%20de%20Espana.pdf>