



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

BackEnd NomadWay: una aplicación para la gestión de rutas turísticas

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: García Riego, Nicolás

Tutor: Herrero Cucó, Carlos

2020/2021



Resumen

Cada vez más personas utilizan internet para buscar rutas y lugares de interés para visitar durante sus viajes turísticos en sitios lejanos, pero, en el proceso, tienen que perder mucho tiempo filtrando mucho contenido no deseado e irrelevante para la visita, como tiendas o establecimientos privados, además de la gran cantidad de publicidad que solemos encontrar en cualquier página web o aplicación.

También algunos de los usuarios emplean de forma habitual redes sociales para subir sus experiencias durante sus viajes que podrían ser de utilidad para otras personas en su misma situación, pero estos servicios no suelen poseer un buen filtrado o búsqueda de publicaciones.

Nuestro objetivo es la creación de una solución para todos los problemas mencionados anteriormente, ofreciendo rutas turísticas variadas, además de lugares de interés validados por nuestros colaboradores, filtros en el mapa útiles para destacar sitios en particular como parques o museos, localizaciones globales, una red social propia integrada en la aplicación donde compartir tus fotos y destinos y un sistema de puntos para ser canjeados por descuentos únicos en los establecimientos destacados.

Se ha realizado este proyecto en el marco de Start.inf, el espacio de emprendimiento de la ETSInf, aplicando la metodología ágil de Scrum y Lean Startup. Hemos concretado una idea de negocio, realizado un estudio de mercado y creado una proyección de ingresos y gastos. Además de un mapa de características y la cronología de desarrollo de los tres productos mínimos viables o MVP junto con sus respectivos experimentos de validación con los early adopters.

Las tecnologías aplicadas para el desarrollo de esta aplicación móvil, tanto para Android como para IOS, han sido Flutter, toolkit para crear el cliente y lógica, Dart, lenguaje utilizado para la programación, Firebase como base de datos y servicio de almacenamiento en la nube y Google Cloud como proveedor de los servicios necesarios para la implementación del mapa y obtención de lugares de interés.

Palabras clave: Flutter, metodología ágil, emprendimiento, red social, rutas, viaje turístico.

Abstract

More and more people use the internet to search for routes and places of interest to visit during their tourist trips in faraway places, but, in the process, they must waste a lot of time filtering a lot of unwanted and irrelevant content for the visit, such as stores or private establishments, in addition to the large amount of advertising that we usually find on any website or application.

Also, some of the users regularly use social networks to upload their experiences during their trips that could be useful for other people in the same situation, but these services do not usually have a good filtering or search of publications.

Our goal is to create a solution for all the above-mentioned problems, offering varied tourist routes, as well as places of interest validated by our collaborators, useful map filters to highlight sites such as parks or museums, global locations, a social network integrated into the application to share your photos and destinations and a points system to be redeemed for unique discounts in the featured establishments.

This project has been carried out within the framework of Start.inf, the entrepreneurship space of ETSInf, applying the agile methodology of Scrum and Lean Startup. We have concretized a business idea, conducted a market study, and created a projection of income and expenses. In addition to a feature map and the chronology of development of the three minimum viable products or MVP along with their respective validation experiments with early adopters.

The technologies applied for the development of this mobile application, both for Android and IOS, have been Flutter, toolkit to create the client and logic, Dart, language used for programming, Firebase as database and cloud storage service and Google Cloud as provider of the necessary services for the implementation of the map and obtaining places of interest.

Keywords : Flutter, agile methodology, entrepreneurship, social network, routes, tourist trip.

Tabla de contenidos

1.	Introducción	11
2.	Evaluación de la idea de negocio	12
2.1.	Idea de negocio.....	12
2.2.	Estudio de mercado	13
2.3.	Análisis DAFO.....	15
2.4.	Proyección de ingresos y gastos.....	17
2.5.	Modelo de negocio y Lean Canvas	21
2.6.	Conclusiones de la evaluación	22
3.	Desarrollo de la idea de negocio	24
3.1.	Mapa de características	24
3.2.	Cronología.....	27
3.2.1.	Desarrollo del primer MVP.....	28
3.2.2.	Desarrollo del segundo MVP	28
3.2.3.	Desarrollo del tercer MVP	29
3.3.	Desarrollo primer Experimento.....	29
3.4.	Desarrollo segundo Experimento	40
4.	Aspectos técnicos	49
4.1.	Técnicas y herramientas utilizadas.....	49
4.2.	Entornos de desarrollo.....	53
4.3.	Base de datos.....	54
4.3.1.	Estructura de la base de datos.....	54
4.3.2.	Modelo de datos	56
4.3.3.	Base de Datos Local.....	57
4.4.	Patrones de diseño.....	57
4.4.1.	Proxy	57
4.4.2.	Adaptador	58
4.4.3.	Observador	59
4.5.	Implementación de las API.....	61
4.5.1.	Maps SDK de Android.....	61
4.5.2.	Geolocation	63
4.5.3.	Geocoding	64
4.5.4.	Directions	65
4.5.6.	Places.....	65



4.6.	Extractor de países y ciudades del mundo.....	67
4.7.	Desafíos técnicos.....	69
4.7.1.	Exceso de peticiones a Google.....	69
4.7.2.	Lentitud de carga.....	70
4.7.3.	Inicio de sesión.....	71
5.	Conclusiones	72
6.	Trabajo Futuro.....	73
7.	Referencias	74
Anexo 1: Protección de datos.....		75
1.	Registro de actividades de tratamiento.....	76
2.	Análisis de riesgo y evaluación de impacto.....	76
3.	Notificación de brechas de seguridad.....	77
4.	Consentimiento	77
5.	Cláusulas de información	77
6.	DPO.....	77
7.	Código de conducta y certificados	77
8.	Información a los propietarios de los datos.....	78
9.	Plazo de conservación de los datos	78
10.	Auditorías periódicas.....	78

Lista de Tablas

Tabla 1, Productos competidores	14
Tabla 2, DAFO.....	16
Tabla 3, Precio patrocinios.....	17
Tabla 4, Ingresos del 1° al 10° trimestre.....	18
Tabla 5, Ingresos del 11° al 20° trimestre.....	18
Tabla 6, Personal contratado del 1° al 10° trimestre.....	19
Tabla 7, Personal contratado del 11° al 20 trimestre	19
Tabla 8, Gastos del 1° al 10° trimestre	21
Tabla 9, Gastos del 11° al 20° trimestre	21



Tabla de Ilustraciones

Ilustración 1, Resultado trimestral	20
Ilustración 2, Resultado trimestral acumulado	20
Ilustración 3, Lean Canvas	23
Ilustración 4, Mapa de características	24
Ilustración 5, Backlog	25
Ilustración 6, Tablero Kanban	26
Ilustración 7, Flujo de Trabajo	27
Ilustración 8, Edad	30
Ilustración 9, Frecuencia de uso de dispositivos móviles	31
Ilustración 10, ¿Utilizas alguna aplicación que te ayude en tus viajes?	31
Ilustración 11, ¿Cuáles son estas aplicaciones?	31
Ilustración 12, ¿Con qué frecuencia usas estas aplicaciones?	32
Ilustración 13, ¿Crees que la aplicación es fácil de usar?	32
Ilustración 14, Visibilidad del estado del sistema	32
Ilustración 15, Relación entre el sistema y el mundo real	33
Ilustración 16, Control y libertad del usuario	33
Ilustración 17, Consistencia y estándares	33
Ilustración 18, Prevención de errores	34
Ilustración 19, Reconocimiento antes de recuerdo	34
Ilustración 20, Flexibilidad y eficiencia de uso	34
Ilustración 21, Estética y diseño minimalista	35
Ilustración 22, Reconocer, diagnosticar y recuperarse de errores	35
Ilustración 23, Fotos de las publicaciones que se han hecho en ese sitio	35
Ilustración 24, Enfoque como red social que tiene la aplicación	36
Ilustración 25, ¿Prefieres que los lugares los puedas guardar en local?	36

Ilustración 26, ¿Cuántos lugares de interés te gustaría que tuviera cada ciudad?	36
Ilustración 27, Aplicación tiene todas las ciudades del mundo.....	37
Ilustración 28, Me interesaría más que la aplicación solo tuviese 1 o 2 ciudades.....	37
Ilustración 29, ¿Echas de menos algún filtro en el mapa?	37
Ilustración 30, ¿Querías ver tus propias publicaciones en el feed?.....	38
Ilustración 31, ¿querías que te apareciesen todas las rutas guardadas?	38
Ilustración 32, ¿Querías que te apareciesen todas las rutas creadas?.....	38
Ilustración 33,¿Te gustaría tener un chat privado?.....	39
Ilustración 34, ¿Estarías interesado en usar esta aplicación?	39
Ilustración 35, Edad 2	41
Ilustración 36, Frecuencias de uso de dispositivos móviles 2.....	41
Ilustración 37 ¿Utilizas alguna aplicación que te ayude en tus viajes? 2.....	41
Ilustración 38, ¿Cuáles son esas aplicaciones? 2	42
Ilustración 39,¿Con que frecuencias utilizas estas aplicaciones? 2.....	42
Ilustración 40, ¿Crees que la aplicación es fácil de usar? 2	42
Ilustración 41, Visibilidad del estado del sistema 2	43
Ilustración 42, Relación entre el sistema y el mundo real 2.....	43
Ilustración 43, Control y libertad 2	43
Ilustración 44, Consistencia y estándares 2.....	44
Ilustración 45, Prevención de errores 2.....	44
Ilustración 46, Reconocimiento antes de recuerdo 2.....	44
Ilustración 47, Flexibilidad y eficiencia de uso 2.....	45
Ilustración 48, Estética y diseño minimalista 2.....	45
Ilustración 49, Reconocer, diagnosticar y recuperarse de errores 2	45
Ilustración 50, ¿Te gustaría poder hacer rutas grupales?	46
Ilustración 51, ¿Te gustaría mantener fotos antiguas de perfil?.....	46
Ilustración 52, Compartir rutas y publicaciones fuera de la aplicación	46
Ilustración 53, ¿Te gustaría poder comentar en los lugares de interés?	47

Ilustración 54, ¿Te gustaría poder valorar los lugares de interés?	47
Ilustración 55, ¿Te gustaría poder crear un perfil privado?.....	47
Ilustración 56, ¿Estarías interesado en usar esta aplicación? 2	48
Ilustración 57, Hello World en Flutter	50
Ilustración 58, Hello World en Dart.....	51
Ilustración 59, Google Cloud Consola	52
Ilustración 60, GitHub.....	52
Ilustración 61, Firebase	53
Ilustración 62, Esquema de la base de datos	55
Ilustración 63, Esquema del Storage de Firebase.....	55
Ilustración 64, Modelo de datos	56
Ilustración 65, Patrón Proxy.....	58
Ilustración 66, Primera implementación observador.....	59
Ilustración 67, Segunda implementación de observador.....	60
Ilustración 68, Tercera implementación de observador	60
Ilustración 69, Api key Google Cloud	61
Ilustración 70, Widget Google Maps	62
Ilustración 71, Método al iniciar el mapa.....	62
Ilustración 72, Mapa iniciado.....	63
Ilustración 73, Obtención de la posición actual	63
Ilustración 74, Obtención de un Stream de posición.....	64
Ilustración 75, código de Geocoding.....	65
Ilustración 76, Consulta de direcciones.....	65
Ilustración 77, Consulta de lugares de interés.....	66
Ilustración 78, Método de obtención de imágenes	67
Ilustración 79, Ejemplo de uso SPARQL.....	68
Ilustración 80, Bucle extractor de País/Ciudad	68

1. Introducción

Muchas personas utilizan sus vacaciones para viajar a ciudades desconocidas en diferentes lugares del mundo, pero en la planificación de este o durante el propio viaje, se encuentran indecisos sobre los lugares que deberían de visitar, perdiendo mucho tiempo en búsquedas de internet y, a veces, sin sacar conclusiones claras.

Las aplicaciones o páginas web más comunes de planificación de viajes suelen enturbiar la búsqueda con una gran cantidad de publicidad que aparece durante el uso normal de esta o con demasiados lugares como restaurantes o tiendas que no tienen nada que ver con una visita turística.

También, muchos de estos viajeros comparten sus experiencias en las redes sociales, como los lugares que han visitado, qué les ha parecido la ciudad, por dónde han viajado... Esta información sería muy valiosa para el resto de las personas que quieran visitar ese mismo lugar, sin tener que estar cambiando de aplicación ni buscando entre cientos y miles de publicaciones hasta encontrar alguna que le pueda servir.

Por último, muchas aplicaciones no agradecen ni recompensan de ninguna forma a sus usuarios más activos, sintiéndose a veces desmotivados a seguir usándola o sencillamente cambian a otra que si lo haga.

La aplicación NomadWay ofrece las siguientes soluciones a estos problemas:

- Un mapa localizado de una ciudad cualquiera del mundo donde estén presentes una gran cantidad de lugares de interés específicos de ese lugar, dando la opción de filtrar estos lugares, crear directamente una ruta o cambiar rápidamente a otro país o ciudad distinta.
- Una lista de rutas creadas por usuarios colaboradores que ayudarían a generar contenido único y de calidad pudiendo ser accedido por el resto de los usuarios gratuitamente.
- Subir publicaciones a cada lugar de interés visitado junto con una foto, alguna descripción si se desea y la posibilidad de comentar en otras publicaciones, interactuando con el creador de esta.
- Un perfil personal, que se podría personalizar con la imagen que uno elija y dónde se verían de forma organizada las publicaciones o rutas que haya creado o guardado el usuario. También se podría añadir amigos y seguidores para chatear de forma privada con ellos o seguir las rutas que realice.

- La opción a las empresas de publicitarse en la aplicación para aparecer en el mapa y ofertar descuentos en su establecimiento, sin llegar a ser intrusivo o molesto ya que existirá la opción de desactivar los lugares promocionados.
- Un sistema que ofrecería puntos al utilizar la aplicación, tanto al concluir rutas como al crear publicaciones, que podrán ser canjeados para obtener descuentos exclusivos en los distintos establecimientos que se hayan publicitado en nuestra aplicación.

2. Evaluación de la idea de negocio

Este apartado del trabajo de fin de grado es común para Nicolás García Riego y Alberto Puerta Millet [1].

2.1. Idea de negocio

NomadWay es una aplicación desarrollada para Android que se centra en el sector del turismo y permite planificar nuestros viajes y conocer todo el mundo, ya que ofrece una cobertura global. Por otro lado, cabe destacar la vertiente de red social que incorpora, lo que incita a compartir nuestras aventuras con amigos, tanto a través de las fotos que se han hecho durante el viaje como a través de las rutas que se han realizado y creado durante el mismo.

NomadWay ofrece una forma fácil de preparar viajes desde la misma aplicación sin tener que buscar en un gran número de páginas web, ya que permite ver de manera fácil y rápida desde un mapa los lugares de interés de la ciudad a la que se quiera viajar. Además, muestra las rutas que han creado no solo amigos, sino también los colaboradores y el resto de los usuarios de la aplicación. También permitirá crear rutas propias para compartirlas y lugares de interés para marcar lugares importantes para el usuario.

Por otro lado, la aplicación permite crear tres perfiles de usuario distintos, usuario normal, colaborador y empresa. Los dos primeros son muy similares, sin embargo, la mayor diferencia es que los colaboradores ofrecen un plus de calidad y fiabilidad en sus publicaciones, pueden crear nuevos puntos de interés para todos los usuarios y puede publicar rutas como colaborador que aparecerán en un apartado especial para resaltarlas. Para poder convertirse en colaborador tienen que ser verificados y confirmados por los administradores.

El perfil de empresa está enfocado a aquellas personas que se quieran patrocinar en la aplicación. Se les permitirá introducir sus locales como lugares nuevos en la aplicación y podrán definir los descuentos que quieran ofrecerles a los usuarios

2.2. Estudio de mercado

En este apartado se estudia el estado actual del mercado de aplicaciones móviles enfocadas en la planificación de viajes y el turismo y para determinar posibles competidores. Esto se hace analizando las características y funciones que ofrecen estas otras apps y cuáles no, para más tarde generar una respuesta a las necesidades causadas por la ausencia de estas características e innovar en la creación de otras funcionalidades que no se encuentran en el resto.

Durante este estudio, se encontró una serie de competidores, explicados brevemente a continuación, que ofrecen una experiencia similar, a partir de estos se creó una tabla comparativa de características, Tabla 1, indicando aquellas que poseen y las que no.

- Google Maps¹: Aplicación muy conocida que permite conocer todos los lugares de todo el mundo, centrada en la obtención de direcciones e indicaciones a modo de navegación GPS.
- Alicante Runaway²: Centrada únicamente en el turismo en Alicante, impidiendo su uso fuera de dicha provincia. Es muy completa en cuanto a ubicaciones y rutas turísticas.
- Touristfy!³: Propone rutas que realizar por la ciudad, dando puntos por su realización y subiendo puestos en el ranking de esta. De momento solo tiene una ruta en Madrid.
- Passporter⁴: Red social para compartir las fotos que realices durante tus viajes y las rutas que hayas realizado
- CityMapper⁵: Aplicación muy similar a Google Maps, pero que se caracteriza por la integración de sistemas de muchos tipos de transporte, autobuses, metro, motos eléctricas....

De este análisis se extraen las características que diferencian a NomadWay del resto de productos, como las que tienen en común.

¹ <https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=es&gl=US>

² <https://play.google.com/store/apps/details?id=com.runawaytrips.guides.alicante&hl=es&gl=US>

³ <http://www.touristfy.com/#>

⁴ <https://play.google.com/store/apps/details?id=com.passporterapp.android&hl=es&gl=US>

⁵ <https://play.google.com/store/apps/details?id=com.citymapper.app.release&hl=es&gl=US>

Todas las aplicaciones tienen en común que se centran en el turismo, son multiplataforma y gratuitas. Algunas se centran más en el apartado social, mientras que otras lo hacen en el propio turismo o servicio de transporte.

<i>Característica</i>	<i>Passporter</i>	<i>Google Maps</i>	<i>Alicante Runaway</i>	<i>Touristfy!</i>	<i>City Mapper</i>	<i>NomadWay</i>
Red social	✓	✗	✗	✓	✗	✓
Perfil personalizable	✓	✓	✗	✓	✗	✓
Necesita conexión a internet	✓	✗	✓	✓	✓	✓
Compartir imágenes	✓	✓	✗	✗	✗	✓
Creación de rutas	✗	✓	✗	✗	✗	✓
Sistema de valoraciones	✗	✓	✗	✗	✗	✓
Sistema de recompensas	✗	✗	✗	✓	✗	✓
Recomendación de lugares interés	✗	✓	✗	✓	✗	✗
Desafíos	✗	✗	✗	✓	✗	✗
Permite patrocinios	✗	✗	✗	✗	✗	✓
Notificaciones	✓	✓	✗	✓	✓	✓
Inicio de sesión integrado	✓	✗	✓	✓	✓	✓
Internacional	✓	✓	✗	✗	✓	✓
Ofertas	✗	✗	✓	✗	✗	✓
Anuncios	✗	✗	✓	✗	✗	✗
Filtrado de lugares.	✗	✓	✓	✗	✗	✓
Creación de lugares de interés	✗	✓	✗	✗	✓	✓
Funciones premium	✓	✗	✗	✗	✓	✗
Multiplataforma	✓	✓	✓	✓	✓	✗
Tutorial de uso	✓	✗	✗	✗	✓	✗
Diferentes métodos de transporte	✗	✓	✗	✗	✓	✗
Sistema de autobuses y metros	✗	✓	✗	✗	✓	✗

Tabla 1, Productos competidores

Al mismo tiempo, se encuentran otros servicios informativos y de búsqueda sobre turismo, pero, al encontrarse únicamente en página web y no en aplicaciones, además de que estas no ofrecen una red social ni forma de comunicación entre el publicador y el lector, se ha decidido no incluirlas en la tabla comparativa por falta de similitudes. Algunos de estas páginas competidoras son:

- Spain.info⁶, página web especializada en el turismo de España, localizada en múltiples idiomas, que ofrece información sobre las ciudades más importantes, lugares que visitar dentro de estas ciudades, formas de viajar hasta el destino, transporte público, fiestas, exposiciones y eventos... En general una página completa pero únicamente enfocada a España.
- LonelyPlanet⁷, página web de turismo global, únicamente en inglés, que ofrece algo de información de los lugares más conocidos del planeta, como las atracciones turísticas principales y las rutas más conocidas. No ofrece mucha información, pero tiene un mayor alcance que un único país.

NomadWay destaca por su enfoque combinado de red social, pudiendo crear y personalizar el perfil del usuario, añadir amigos y seguidores, y pudiendo compartir fotos y experiencias, con el del turismo, dando la libertad a los propios usuarios de crear y organizar su propia ruta, pudiendo añadir también sus propios lugares de interés, todo ello en un ecosistema global de localizaciones.

También se aporta un sistema de recompensas único por usar la aplicación e interactuar con otros usuarios, que se podrán usar en descuentos que las empresas promocionadas oferten

2.3. Análisis DAFO

Para la detección de factores internos, debilidades y fortalezas, y de factores externos, amenazas y oportunidades, en la situación de la empresa, se emplea a continuación un análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades).

Con los resultados obtenidos en la Tabla 2 se desarrolla una serie de estrategias futuras para aprovechar los puntos fuertes y mitigar los débiles.

⁶ <https://www.spain.info/es/>

⁷ <https://www.lonelyplanet.com/spain/valencia-and-murcia>

<p style="text-align: center;">Debilidades</p> <ul style="list-style-type: none"> • Poca experiencia tanto en el desarrollo como en la gestión de una empresa. • Escasez de usuarios de forma inicial. • Pocos recursos para la realización del proyecto. • Falta de apoyo por negocios de forma inicial. 	<p style="text-align: center;">Amenazas</p> <ul style="list-style-type: none"> • Lanzamiento del producto durante una pandemia en un momento en el que los viajes están muy restringidos y la gente tiene miedo a desplazarse. • Crisis económica causada por la pandemia, lo que también dificulta los viajes a corto y medio plazo. • Productos ya consolidados en el mercado con una gran base de usuarios que sean reacios a cambiar de plataforma.
<p style="text-align: center;">Fortalezas</p> <ul style="list-style-type: none"> • Una red social activa. • Sistema de puntos para canjear por recompensas. • Descuentos en negocios y locales. • Motivación para estar más activo. • Promueve el turismo. • Aplicación de uso gratuito. 	<p style="text-align: center;">Oportunidades</p> <ul style="list-style-type: none"> • Aplicación con poca competencia. • Crecimiento rápido en caso de éxito. • Producto interesante y muy útil para viajar. • Sinergia con los negocios que decidan confiar en nosotros.

Tabla 2, DAFO

La mayor fortaleza interna es la de promover el turismo de forma internacional junto con el añadido de una red social, pero la peor debilidad es la poca experiencia en el desarrollo y gestión de una empresa ya que es algo totalmente nuevo.

En los factores externos, se encuentra la mayor fortaleza en la poca competencia actual de aplicaciones de turismo y el rápido crecimiento de las redes sociales, las amenazas principales que se encuentran serían la pandemia global y los usuarios que se nieguen a utilizar otra red social diferente a la habitual.

2.4. Proyección de ingresos y gastos

Para hacer el estudio de la proyección económica del proyecto de aquí a cinco años, se han calculado, de manera aproximada, el presupuesto inicial y los gastos e ingresos que aparecerán a lo largo del desarrollo. Para ello, se ha tenido en cuenta en cuenta las instalaciones, el personal, el material y la infraestructura utilizada durante el desarrollo y el mantenimiento del producto.

Para la financiación de la idea se va a seguir un modelo de negocio por suscripción como se indica en la Figura 3 (Lean Canvas), en el que las empresas (restaurantes o empresas relacionadas con el ocio principalmente) pagaran una mensualidad.

Este pago se puede traducir en un patrocinio de descuento, que consiste en que las empresas creen y publiciten descuentos únicos en la plataforma, o en un patrocinio de posicionamiento, que consiste en que los locales de dichas empresas aparezcan en la aplicación como puntos que se puedan visitar y se puedan incluir en rutas por los colaboradores y usuarios. El precio de esta suscripción varía dependiendo de la cantidad de locales que quieras particionar, como se puede ver en la Tabla 3.

	Por 1 local	Por 10 locales	Por 50 locales	Por 100 locales
Patrocinio Descuento	25,00 €	183,33 €	833,33 €	1.333,33 €
Patrocinio Posicionamiento	20,00 €	133,33 €	583,33 €	1.000,00 €

Tabla 3, Precio patrocinios

Por otro lado, el número de suscripciones que se conseguirán se ha calculado a partir del número de usuarios que se espera que la aplicación vaya consiguiendo a lo largo de los años y, además, parece que el patrocinio de posicionamiento es más interesante a pesar de ser más caro.

Se ha estimado, que el mínimo número de usuarios al que se debe llegar a lo largo de los primeros cinco años asciende a aproximadamente un 0.3% de los usuarios con los que cuentan las aplicaciones que compiten más directamente con NomadWay, es decir, un total de 2.200.000 usuarios (1.000.000.000 de Google Maps, 800.000.000 CittyMapper, 20.000 Passporter).

En la Tabla 4 y en la Tabla 5 se pueden ver desglosadas las aproximaciones de ingresos que se esperan en los próximos cinco años.

BackEnd NomadWay: una aplicación para la gestión de rutas turísticas

Número de Trimestre	1	2	3	4	5	6	7	8	9	10
Interacciones con la app	Año 1 / T1	Año 1 / T2	Año 1 / T3	Año 1 / T4	Año 2 / T1	Año 2 / T2	Año 2 / T3	Año 2 / T4	Año 3 / T1	Año 3 / T2
Usuarios (*1000)	2	8	32	128	512	768	1152	1440	1620	1721
Patrocinio Descuento por 1 local	6	16	64	256	1024	1536	2304	2880	3240	3442
Patrocinio Descuento por 10 locales	0	0	2	8	32	48	72	90	101	108
Patrocinio Descuento por 50 locales	0	0	1	2	8	12	18	23	25	27
Patrocinio Descuento por 100 locales	0	0	0	1	2	3	5	6	6	7
Patrocinio Posicionamiento por 1 local	8	20	80	320	1280	1920	2880	3600	4050	4303
Patrocinio Posicionamiento por 10 local	0	1	4	16	64	96	144	180	203	215
Patrocinio Posicionamiento por 50 local	0	0	1	3	13	19	29	36	41	43
Patrocinio Posicionamiento por 100 local	0	0	0	1	4	6	9	11	13	13
Ingresos Trimestrales	Año 1 / T1	Año 1 / T2	Año 1 / T3	Año 1 / T4	Año 2 / T1	Año 2 / T2	Año 2 / T3	Año 2 / T4	Año 3 / T1	Año 3 / T2
Patrocinio Descuento por 1 local * Precio	150,00 €	400,00 €	1.600,00 €	6.400,00 €	25.600,00 €	38.400,00 €	57.600,00 €	72.000,00 €	81.000,00 €	86.050,00 €
Patrocinio Descuento por 10 local * Precio	0,00 €	0,00 €	366,66 €	1.466,64 €	5.866,56 €	8.799,84 €	13.199,76 €	16.499,70 €	18.516,33 €	19.799,64 €
Patrocinio Descuento por 50 local * Precio	0,00 €	0,00 €	833,33 €	1.666,66 €	6.666,64 €	9.999,96 €	14.999,94 €	19.166,59 €	20.833,25 €	22.499,91 €
Patrocinio Descuento por 100 local * Precio	0,00 €	0,00 €	0,00 €	1.333,33 €	2.666,66 €	3.999,99 €	6.666,65 €	7.999,98 €	7.999,98 €	9.333,31 €
Patrocinio Posicionamiento por 1 local * Precio	160,00 €	400,00 €	1.600,00 €	6.400,00 €	25.600,00 €	38.400,00 €	57.600,00 €	72.000,00 €	81.000,00 €	86.060,00 €
Patrocinio Posicionamiento por 10 local * Precio	0,00 €	133,33 €	533,32 €	2.133,28 €	8.533,12 €	12.799,68 €	19.199,52 €	23.999,40 €	27.065,99 €	28.665,95 €
Patrocinio Posicionamiento por 50 local * Precio	0,00 €	0,00 €	583,33 €	1.749,99 €	7.583,29 €	11.083,27 €	16.916,57 €	20.999,88 €	23.916,53 €	25.083,19 €
Patrocinio Posicionamiento por 100 local * Precio	0,00 €	0,00 €	0,00 €	1.000,00 €	4.000,00 €	6.000,00 €	9.000,00 €	11.000,00 €	13.000,00 €	13.000,00 €
Total Ingresos	310 €	933 €	5.517 €	22.150 €	86.516 €	129.483 €	195.182 €	243.666 €	273.332 €	290.492 €

Tabla 4, Ingresos del 1º al 10º trimestre

Número de Trimestre	10	11	12	13	14	15	16	17	18	19	20
Interacciones con la app	Año 3 / T2	Año 3 / T3	Año 3 / T4	Año 4 / T1	Año 4 / T2	Año 4 / T3	Año 4 / T4	Año 5 / T1	Año 5 / T2	Año 5 / T3	Año 5 / T4
Usuarios (*1000)	1721	1829	1886	1945	2006	2037	2069	2101	2134	2167	2201
Patrocinio Descuento por 1 local	3442	3658	3772	3890	4012	4074	4138	4202	4268	4334	4402
Patrocinio Descuento por 10 locales	108	114	118	122	125	127	129	131	133	135	138
Patrocinio Descuento por 50 locales	27	29	29	30	31	32	32	33	33	34	34
Patrocinio Descuento por 100 locales	7	7	7	8	8	8	8	8	8	8	9
Patrocinio Posicionamiento por 1 local	4303	4573	4715	4863	5015	5093	5173	5253	5335	5418	5503
Patrocinio Posicionamiento por 10 local	215	229	236	243	251	255	259	263	267	271	275
Patrocinio Posicionamiento por 50 local	43	46	47	49	50	51	52	53	53	54	55
Patrocinio Posicionamiento por 100 local	13	14	15	15	16	16	16	16	17	17	17
Ingresos Trimestrales	Año 3 / T2	Año 3 / T3	Año 3 / T4	Año 4 / T1	Año 4 / T2	Año 4 / T3	Año 4 / T4	Año 5 / T1	Año 5 / T2	Año 5 / T3	Año 5 / T4
Patrocinio Descuento por 1 local * Precio	86050	91450	94300	97250	100300	101850	103450	105050	106700	108350	110050
Patrocinio Descuento por 10 local * Precio	19799,64	20899,62	21632,94	22366,26	22916,25	23282,91	23649,57	24016,23	24382,89	24749,55	25299,54
Patrocinio Descuento por 50 local * Precio	22499,91	24166,57	24166,57	24999,9	25833,23	26666,56	26666,56	27499,89	27499,89	28333,22	28333,22
Patrocinio Descuento por 100 local * Precio	9333,31	9333,31	9333,31	10666,64	10666,64	10666,64	10666,64	10666,64	10666,64	10666,64	11999,97
Patrocinio Posicionamiento por 1 local * Precio	86060	91460	94300	97260	100300	101860	103460	105060	106700	108360	110060
Patrocinio Posicionamiento por 10 local * Precio	28665,95	30532,57	31465,88	32399,19	33465,83	33999,15	34532,47	35065,79	35599,11	36132,43	36665,75
Patrocinio Posicionamiento por 50 local * Precio	25083,19	26833,18	27416,51	28583,17	29166,5	29749,83	30333,16	30916,49	30916,49	31499,82	32083,15
Patrocinio Posicionamiento por 100 local * Precio	13000	14000	15000	15000	16000	16000	16000	16000	17000	17000	17000
Total Ingresos	290492	308675,25	317615,21	328525,16	338648,45	344075,09	348758,4	354275,04	359465,02	365091,66	371491,63

Tabla 5, Ingresos del 11º al 20º trimestre

Los empleados contratados a finales del quinto año ascienden a un total de 51 (un Director Ejecutivo, un Director de Tecnología, un Director de Marketing, un Director Financiero, 12 desarrolladores senior, 14 técnicos de soporte, 3 administrativos, 10 personas de marketing y 8 desarrolladores junior). Al principio del proyecto, durante el primer año, no se dispone ni Director de Marketing ni Director Financiero, estos llegarán al comienzo del segundo año.

La plantilla va creciendo de manera gradual durante los tres primeros años (como se puede ver en la Tabla 6 y la Tabla 7), pero con la llegada del cuarto año, debido a los buenos resultados que se tendrán se lleva a cabo una gran ampliación de personal para mejorar las condiciones de los trabajadores y reducir la carga de trabajo.

Esto implica una serie de cambios extra ya que, al duplicar el personal, hay que hacer un cambio a unas oficinas con mucho más espacio y una inversión en material de oficina y ordenadores bastante grande. Finalmente, a principios del quinto año, se vuelve a aumentar la plantilla, pero de manera más reducida.

Personal contratado	Año 1 / T1	Año 1 / T2	Año 1 / T3	Año 1 / T4	Año 2 / T1	Año 2 / T2	Año 2 / T3	Año 2 / T4	Año 3 / T1	Año 3 / T2
CEO - Director Ejecutivo	1	1	1	1	1	1	1	1	1	1
CTO - Director Técnico	1	1	1	1	1	1	1	1	1	1
CMO - Director de Marketing	0	0	0	0	1	1	1	1	1	1
CFO - Director Financiero	0	0	0	0	1	1	1	1	1	1
Desarrolladores Senior	1	1	1	1	2	2	2	2	4	4
Técnicos de soporte	1	1	2	2	3	3	3	3	6	6
Administrativo	1	1	1	1	1	1	2	2	2	2
Personal de Marketing	1	1	1	1	3	3	3	3	4	4
Desarrolladores Junior	2	2	2	2	3	3	3	3	3	3
Total empleados	8	8	9	9	16	16	17	17	23	23

Tabla 6, Personal contratado del 1º al 10º trimestre

Personal contratado	Año 3 / T3	Año 3 / T4	Año 4 / T1	Año 4 / T2	Año 4 / T3	Año 4 / T4	Año 5 / T1	Año 5 / T2	Año 5 / T3	Año 5 / T4
CEO - Director Ejecutivo	1	1	1	1	1	1	1	1	1	1
CTO - Director Técnico	1	1	1	1	1	1	1	1	1	1
CMO - Director de Marketing	1	1	1	1	1	1	1	1	1	1
CFO - Director Financiero	1	1	1	1	1	1	1	1	1	1
Desarrolladores Senior[1]	4	4	7	7	7	7	12	12	12	12
Técnicos de soporte	6	6	12	12	12	12	14	14	14	14
Administrativo	2	2	3	3	3	3	3	3	3	3
Personal de Marketing	4	4	10	10	10	10	10	10	10	10
Desarrolladores Junior1	3	3	7	7	7	7	8	8	8	8
Total empleados	23	23	43	43	43	43	51	51	51	51

Tabla 7, Personal contratado del 11º al 20º trimestre

Con respecto al marketing, se ha calculado aproximadamente la inversión adecuada que se tendría que llevar a cabo para poder crecer teniendo relevancia en el mercado y atrayendo usuarios de manera rápida y efectiva al principio, y mantenerlos a lo largo de los años. Teniendo en cuenta que nuestras campañas se centraran más en canales digitales de publicidad y marketing como Facebook o Google Ads, por encima de los canales tradicionales de difusión.

Por el lado de la infraestructura, principalmente los costes de almacenamiento Cloud y el uso de las API de Google Cloud Platform, se ha hecho un estudio de cuál sería el coste de mantener estas funcionalidades con los requisitos que tiene actualmente la aplicación, y escalándolos, dependiendo del número de usuarios con los que se cuenta en cada momento. Aunque en este caso no se ha tenido en cuenta, el coste del almacenamiento de datos en la nube, que crece exponencialmente con el número de usuarios, es muy caro, así que en algún momento se tendría que pensar en crear nuestros propios servidores para reducir los costes.



Por último, se han realizado una serie de cálculos, para calcular de manera aproximada la inversión inicial necesaria para poder llevar a cabo el proyecto. Según dichos cálculos, se necesitará una inversión de unos 230.000€. Con este dinero podrán cubrir los cuatro primeros meses de pérdidas y recuperar la inversión inicial en el séptimo trimestre, como se puede ver en la Figura 1 y en la Figura 2.

Con todo esto, se le ofrece al futuro inversor multiplicar su inversión inicial en la startup por 14 a los 3 años. Esto se ha calculado a través de la formula EBITDA, el resultado anual acumulado se multiplica por 10, que permite saber el valor aproximado de la empresa. En nuestro caso, en el tercer año, el EBITDA da 21.369.800 €. Si, como se ha dicho antes, nuestro inversor tiene que multiplicar su inversión por 14, tan solo hará falta darle aproximadamente el 15% de participación en la empresa.

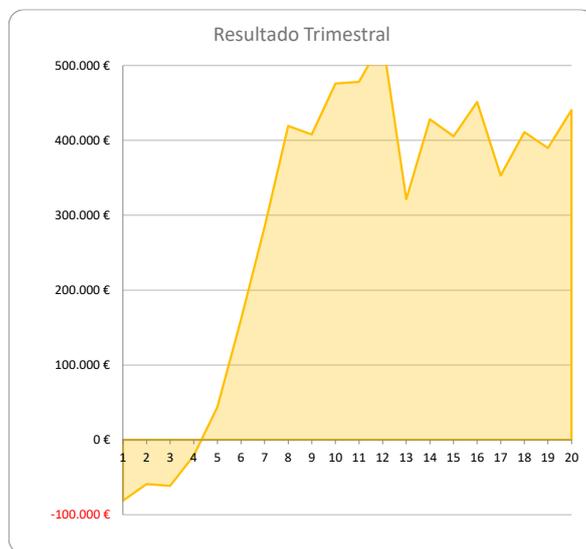


Ilustración 1, Resultado trimestral

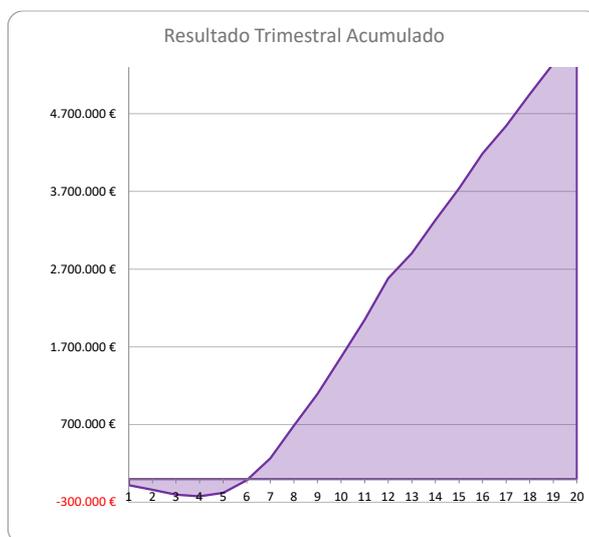


Ilustración 2, Resultado trimestral acumulado

Gastos Anuales	Año 1 / T1	Año 1 / T2	Año 1 / T3	Año 1 / T4	Año 2 / T1	Año 2 / T2	Año 2 / T3	Año 2 / T4	Año 3 / T1	Año 3 / T2
Infraestructura Cloud	800 €	2.100 €	6.300 €	15.000 €	40.000 €	54.000 €	73.500 €	85.000 €	90.000 €	101.000 €
Ordenadores e impresoras	11.000 €	0 €	1.000 €	0 €	6.000 €	0 €	0 €	0 €	7.000 €	0 €
Muebles oficina e instalaciones	6.000 €	0 €	0 €	0 €	3.000 €	0 €	0 €	0 €	2.500 €	0 €
Marketing	5.000 €	0 €	5.000 €	0 €	10.000 €	0 €	20.000 €	0 €	25.000 €	
Alquiler oficina	9.000 €	9.000 €	9.000 €	9.000 €	9.000 €	9.000 €	9.000 €	9.000 €	12.000 €	12.000 €
Internet, electricidad, agua, teléfono, etc.	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	1.500 €	2.000 €	2.000 €
Gestoría	600 €	600 €	600 €	600 €	600 €	600 €	600 €	600 €	800 €	800 €
CEO - Director Ejecutivo	9.000 €	9.000 €	9.000 €	9.000 €	12.000 €	12.000 €	12.000 €	12.000 €	12.000 €	12.000 €
CTO - Director Técnico	9.000 €	9.000 €	9.000 €	9.000 €	12.000 €	12.000 €	12.000 €	12.000 €	12.000 €	12.000 €
CMO - Director de Marketing	0 €	0 €	0 €	0 €	9.000 €	9.000 €	9.000 €	9.000 €	9.000 €	9.000 €
CFO - Director Financiero	0 €	0 €	0 €	0 €	9.000 €	9.000 €	12.000 €	12.000 €	12.000 €	12.000 €
Desarrolladores Senior	6.000 €	6.000 €	6.000 €	6.000 €	12.000 €	12.000 €	12.000 €	12.000 €	30.000 €	30.000 €
Técnicos de soporte	4.500 €	4.500 €	9.000 €	9.000 €	21.000 €	21.000 €	21.000 €	21.000 €	42.000 €	42.000 €
Administrativo	4.500 €	4.500 €	4.500 €	4.500 €	4.500 €	4.500 €	9.000 €	9.000 €	9.000 €	9.000 €
Personal de Marketing	6.000 €	6.000 €	6.000 €	6.000 €	18.000 €	18.000 €	18.000 €	18.000 €	24.000 €	24.000 €
Desarrolladores Junior	9.000 €	9.000 €	9.000 €	9.000 €	13.500 €	13.500 €	13.500 €	13.500 €	13.500 €	13.500 €
Total Gastos	81.900 €	61.200 €	75.900 €	78.600 €	181.100 €	176.100 €	223.100 €	214.600 €	302.800 €	279.300 €

Tabla 8, Gastos del 1º al 10º trimestre

Gastos Anuales	Año 3 / T3	Año 3 / T4	Año 4 / T1	Año 4 / T2	Año 4 / T3	Año 4 / T4	Año 5 / T1	Año 5 / T2	Año 5 / T3	Año 5 / T4
Infraestructura Cloud	108.000 €	110.000 €	112.000 €	115.000 €	117.000 €	118.000 €	120.000 €	121.000 €	122.000 €	123.000 €
Ordenadores e impresoras	0 €	0 €	40.000 €	0 €	0 €	0 €	8.000 €	0 €	0 €	0 €
Muebles oficina e instalaciones	0 €	0 €	8.500 €	0 €	0 €	0 €	3.500 €	0 €	0 €	0 €
Marketing	35.000 €		35.000 €	0 €	35.000 €	0 €	35.000 €	0 €	35.000 €	0 €
Alquiler oficina	12.000 €	12.000 €	36.000 €	36.000 €	36.000 €	36.000 €	36.000 €	36.000 €	36.000 €	36.000 €
Internet, electricidad, agua, teléfono, etc.	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	2.000 €	5.000 €	2.000 €	2.000 €	2.000 €
Gestoría	800 €	800 €	800 €	800 €	800 €	800 €	3.000 €	800 €	800 €	800 €
CEO - Director Ejecutivo	12.000 €	12.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €	18.000 €	18.000 €	18.000 €
CTO - Director Técnico	12.000 €	12.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €	18.000 €	18.000 €	18.000 €
CMO - Director de Marketing	12.000 €	12.000 €	12.000 €	12.000 €	12.000 €	12.000 €	15.000 €	15.000 €	15.000 €	15.000 €
CFO - Director Financiero	12.000 €	12.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €	15.000 €
Desarrolladores Senior	30.000 €	30.000 €	52.500 €	52.500 €	52.500 €	52.500 €	90.000 €	90.000 €	90.000 €	90.000 €
Técnicos de soporte	42.000 €	42.000 €	84.000 €	84.000 €	84.000 €	84.000 €	98.000 €	98.000 €	98.000 €	98.000 €
Administrativo	9.000 €	9.000 €	13.500 €	13.500 €	13.500 €	13.500 €	13.500 €	13.500 €	13.500 €	13.500 €
Personal de Marketing	24.000 €	24.000 €	60.000 €	60.000 €	60.000 €	60.000 €	60.000 €	60.000 €	60.000 €	60.000 €
Desarrolladores Junior	13.500 €	13.500 €	31.500 €	31.500 €	31.500 €	31.500 €	36.000 €	36.000 €	36.000 €	36.000 €
Total Gastos	324.300 €	291.300 €	532.800 €	452.300 €	489.300 €	455.300 €	568.000 €	523.300 €	559.300 €	525.300 €

Tabla 9, Gastos del 11º al 20º trimestre

2.5. Modelo de negocio y Lean Canvas

El modelo de negocio que se ha decidido utilizar ha sido el de suscripción, que consiste en la monetización de la aplicación mediante el pago mensual de unos servicios que se ofrecen al cliente, estos servicios están únicamente dirigidos a empresas para poder publicitar sus locales o los descuentos que quieran incorporar, para el resto de los clientes la aplicación será totalmente gratuita, sin banners⁸, pop-ups⁹, videos ni otro tipo de publicidad.

El Lean Canvas es una herramienta de la metodología Lean Startup¹⁰, utilizada para la visualización de modelos de negocio, generar nuevas soluciones a los problemas planteados y verificarlas en el mercado.

⁸ Anuncios con un formato específico, generalmente pequeños, diseñados para llamar la atención.

⁹ Anuncios emergentes, aparecen normalmente al entrar a una página o moverse por esta.

¹⁰ Es una metodología para desarrollar negocios y productos



Para generarlo, se ha seguido el orden indicado en cada casilla para obtener una visión más clara y concisa:

1. Clientes: Tipo de clientes objetivo que podrían ser Early Adopter.
2. Problema: El problema que sufren los clientes actualmente y que NomadWay puede solucionar.
3. Proposición de valor: Describe de forma clara la ayuda que se va a proporcionar que diferencie a la aplicación del resto.
4. Solución: Identificar las características que ayuden a resolver el problema.
5. Canales: En este apartado se especifica la forma de hacer llegar la proposición a los clientes.
6. Ingresos: Se indica el modelo de negocio que se utilizará para obtener beneficios.
7. Costos: Se analizan los costes que va a generar la aplicación, como almacenamiento en servidores, licencias de API, material para desarrollo...
8. Métricas: Identificar las acciones clave que se lleven a cabo durante el uso de la aplicación, usadas para tomar decisiones.
9. Ventaja Competitiva: La sección más importante del documento que especifica las características clave que ponen a NomadWay por delante de los productos competidores, que sean difíciles de copiar y que hacen especiales a la aplicación.

El resultado obtenido al finalizar el proceso de creación ha sido la Figura 3, del cual se puede generar un prototipo o MVP para poder validar las ideas y soluciones propuestas con los posibles clientes.

2.6. Conclusiones de la evaluación

Después de realizar el estudio anterior se puede concluir de NomadWay tiene el potencial necesario para hacerse un hueco en el mercado. Ya que ofrece una manera de conocer el mundo que ninguna aplicación había explorado hasta el momento de manera global.

En primer lugar, se ha realizado un estudio de la competencia en la que se ha analizado a las aplicaciones más importantes, lo que ha permitido comprobar no solo cuales son las características que tienen el resto de las aplicaciones, si no también darse cuenta de las carencias que tiene NomadWay que principalmente está relacionada con la falta de opciones de transporte, y la integración con plataformas de este tipo.

Por otra parte, se ha realizado el análisis DAFO que se podrá utilizar en un futuro como punto de partida para la campaña de marketing basándose en las fortalezas y oportunidades que permita

<p>2 Problema</p> <p>Mucha gente al viajar no sabe qué lugares de la ciudad que visita tiene que ver o qué rutas podría seguir para tener una mejor experiencia turística. Por tanto, emplea mucho tiempo buscando en muchas páginas web o en puntos de información turística de la propia ciudad.</p> <p>También se suele usar una segunda aplicación, que es solo red social para compartir los lugares que has visitado, por donde has ido..., resultando un proceso muy tedioso y poco práctico.</p>	<p>4 Solución</p> <p>Nuestra aplicación ofrece un mapa de la localidad visitada con puntos de interés típicos de la zona o atracciones turísticas famosas. Ofrecerá además la posibilidad de crear rutas o utilizar las rutas ya creadas por otros usuarios y compartirlas. Es decir, por una red social integrada en la propia aplicación.</p>	<p>3 Proposición de valor</p> <p>Ofrecemos localizaciones por la ciudad que ayude al turista a conocer distintas rutas, consiguiendo puntos al participar y al realizarlas.</p> <p>Usaremos un mapa con diversos puntos en los que se podrán subir publicaciones a nuestra red social.</p>	<p>9 Ventaja competitiva</p> <p>Ofrecemos rutas y lugares turísticos de forma global, además de la integración de una red social propia para compartir tus viajes.</p> <p>También ofrecemos descuentos como recompensa por participar activamente.</p>	<p>1 Clientes</p> <p>Nuestro principal objetivo son personas jóvenes que les guste compartir y enseñar sus experiencias por redes sociales y aquellas que viajen interesadas en hacer turismo por la ciudad.</p> <p>Los primeros usuarios de la aplicación serán aquellos que participen activamente en otras redes sociales y viajen con regularidad.</p> <p>Sin embargo, no renunciamos a la siguiente franja de edad, es decir, aquellos. Usuarios que, a pesar de no ser nativos digitales, se han adaptado al uso de las aplicaciones móviles y son muy activos en redes sociales. </p>
<p>7 Costos</p> <p>Los costes irán en aumento según va avanzando el desarrollo, ya que aumenta el personal y el volumen de usuarios, esto conlleva un aumento de gastos en salarios, hardware, gestión, administración, servicios cloud...</p> <p>Con el aumento del personal, también se tiene que actualizar la oficina a una de mayor capacidad y mejores prestaciones, teniendo que comprar más equipo y nuevo mobiliario.</p> <p>Además, también hay que contemplar la subida de sueldo de los empleados a lo largo de los años.</p>		<p>6 Ingresos</p> <p>Como modelo de negocio usaremos el modelo de suscripción, en el que las empresas pagan para que se las posicione en la plataforma de manera que aparezcan en el mapa de los usuarios, y/o pagan para introducir descuentos a los que solo se pueden acceder a través de nuestra aplicación.</p> <p>Durante el primer año de desarrollo se tendrán pérdidas, sin embargo, en el primer trimestre del segundo año se empezará a generar más dinero del gastado por mes. Habrá que pedir un préstamo inicial de aproximadamente 280.000€ y los beneficios totales ascenderán a 6.500.000€ aproximadamente.</p>		

Ilustración 3, Lean Canvas



crecer en el mercado y ganarse un sitio entre las aplicaciones que ya están asentadas en el mercado.

En el Lean Canvas se han plasmado los problemas que se encuentran en el resto de las aplicaciones del mercado, como la necesidad de recurrir a muchas fuentes para encontrar lugares a visitar en las ciudades perdiendo así una gran cantidad de tiempo y la necesidad de utilizar una segunda aplicación para compartir esos viajes. También, se retrata la solución que presenta NomadWay a estas carencias y los canales a través de los que se va a conseguir distribuir y los costos e ingresos que se espera a lo largo de la vida de esta.

Además, se ha decidido que el modelo de negocio de la aplicación sea por suscripción, que es el que mejor se adapta a este tipo de servicio. Sin embargo, la suscripción no será para los usuarios de la aplicación, si no para las empresas que quieran patrocinarse en esta, beneficiándose de posicionamiento en el mapa o permitiéndoles anunciar descuentos en la aplicación.

3. Desarrollo de la idea de negocio

Este apartado del trabajo de fin de grado es común para Nicolás García Riego y Alberto Puerta Millet [1].

3.1. Mapa de características

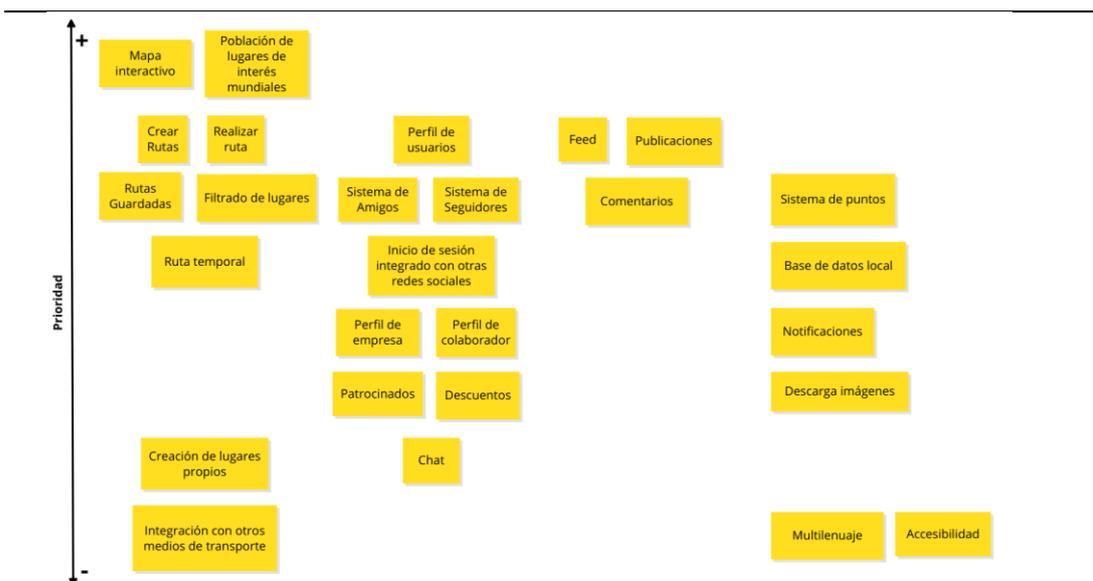


Ilustración 4, Mapa de características



En este mapa se presentan las características que se deseen desarrollar en una gráfica, estando ordenadas en el eje vertical por la prioridad en la que deberían de ser implementadas en una primera instancia.

Están divididas en su eje horizontal por la sección de la aplicación a la que pertenecen, el mapa, la red social, el contenido público y sistemas internos.

Un ejemplo de interpretación sería que, se puede ver en la Figura 4, el mapa interactivo y la población de lugares de interés mundiales serían las características más importantes de toda la aplicación, por otro lado, el multilinguaje o la accesibilidad son las menos prioritarias.

A partir de estas características se puede determinar el alcance de cada MVP que se va a desarrollar, incluyendo aquellas funciones que se encuentren más arriba. También se especifican las que son menos importantes aunque se pueden quedar almacenadas en el Product Backlog¹¹, Figura 5, o ser descartadas.

Orden	UT	Estructura - Temas	Creación	Finalización	Línea de trabajo	Sprint
110	2576	Multilinguaje	23/02/2021		InstaGo	Backlog
1020	2580	Tienda	23/02/2021	19/06/2021	InstaGo	Backlog
1020	2581	Patrocinados	23/02/2021	19/06/2021	InstaGo	Backlog
1020	2582	Descuentos	23/02/2021	19/06/2021	InstaGo	Backlog
1030	2701	Integración inicio sesión otras plataformas	22/03/2021	03/05/2021	InstaGo	Backlog
100	2702	Métricas usuario	22/03/2021		InstaGo	Backlog
1020	2703	Añadir seguidores	22/03/2021	03/05/2021	InstaGo	Backlog
1025	2704	Crear perfil Empresa	22/03/2021	19/06/2021	InstaGo	Backlog
1020	2705	Crear perfil Colaborador	22/03/2021	09/05/2021	InstaGo	Backlog
70	2707	Comentarios en sitios	22/03/2021		InstaGo	Backlog
1010	2708	Filtros	22/03/2021	26/04/2021	InstaGo	Backlog
1040	2709	Crear Ruta temporal	22/03/2021	19/05/2021	InstaGo	Backlog

Ilustración 5, Backlog

Para mantener un control de las características que se están desarrollando y se van a desarrollar, además de mantener una gestión ágil del trabajo, siguiendo un flujo de trabajo especificado, se ha decidido usar la herramienta Worki por la experiencia de uso que se consiguió durante el curso académico.

¹¹ Listado de todas las tareas que se vayan a realizar en un proyecto.





Ilustración 6, Tablero Kanban

Esta permite crear un tablero Kanban¹², Figura 6, para controlar el estado en el que se encuentra cada funcionalidad según el flujo de trabajo creado para el proyecto, además de crear y gestionar los sprint sobre los que se rige el desarrollo y las historias de usuario que incluyen las propias características a implementar. Además, también genera de forma automática las gráficas de seguimiento, como las del tiempo invertido en cada estado o por colaborador y las de burndown en puntos o retrabajo.

El flujo de trabajo, Figura 7, consiste en las siguientes etapas:

1. Registrar: Fase inicial de la característica, se acaba de introducir en la herramienta.
2. Esperar Prioridad: Siguiendo estado en el que se queda esperando una funcionalidad para ser especificada y desarrollada según su prioridad.
3. Especificar Requisitos: Terminar de completar la descripción de la historia de usuario, añadir información extra, elaborar un mockup de las interfaces y establecer las pruebas que se deben realizar para validar el correcto funcionamiento de la funcionalidad desarrollada son algunas de las tareas a completar durante esta fase.
4. Esperar Sprint: Fase de espera hasta que sea introducida en un sprint para ser implementada.
5. Programar: Implementar la historia de usuario en el código de la aplicación, se deberá tener en cuenta el diseño de este, pudiendo incluir patrones, o prevenir retrabajo, además de realizar las primeras pruebas de validación. Si es necesario, se podrá volver a estimar el tiempo necesario para su desarrollo.
6. Aplicar Pruebas de Aceptación: Penúltimo estado en el que se terminan de probar de forma exhaustiva las pruebas de validación designadas en la tercera etapa, siempre se intentará buscar errores y forzar la aplicación para encontrarlos, como, por ejemplo, introducir un valor textual en un campo numérico o introducir más elementos de los permitidos en una lista.
7. Terminar: Último estado en el que se encuentra una funcionalidad que ya ha sido especificada, implementada y validada.

¹² Herramienta utilizada para visualizar el flujo de trabajo.

% Progreso	Operador	Actividad	Esti...	Rol
0	Secuencia	Registrar	<input type="checkbox"/>	
5	Secuencia	Esperar Prioridad	<input type="checkbox"/>	Desarrollador
34	Secuencia	Especificar Requisitos	<input type="checkbox"/>	Desarrollador
35	Secuencia	Esperar Sprint	<input type="checkbox"/>	Desarrollador
85	Secuencia	Programar - InstaGO	<input checked="" type="checkbox"/>	Desarrollador
99	Secuencia	Aplicar Pruebas de Aceptación	<input type="checkbox"/>	Desarrollador
100	Secuencia	Terminar	<input type="checkbox"/>	

Ilustración 7, Flujo de Trabajo

Finalmente, al terminar de desarrollar todas las características planeadas para un mínimo producto viable, se crean unos experimentos para ser validados por los clientes.

3.2. Cronología

Como ya se ha comentado anteriormente, NomadWay es una aplicación que se ha desarrollado en el marco de un TFG en emprendimiento. Esto implica que se ha seguido el ciclo de vida de este tipo de proyectos. Este consiste en los siguientes pasos:

1. Generación de la idea de negocio: en esta fase se desarrolla la idea, para ello se pueden utilizar una gran cantidad de técnicas como el Brainstorming¹³.
2. Evaluación y puesta en marcha: una vez se tiene la idea, se debe de realizar el análisis de mercado para comprobar la viabilidad de la esta. El análisis DAFO, Lean Canvas, estudio de productos competidores y proyección de ingresos gastos, son algunos de los estudios que se realizan para este propósito.
3. Desarrollo de MVP hasta la llegada de la entrega: cada vez que se finaliza alguno de estos se hace un experimento para validar el trabajo realizado con usuarios reales, los early adopters¹⁴.

En este caso se han realizado tres MVP en los que se va a profundizar a continuación.

¹³ Técnica grupal para generar nuevas ideas.

¹⁴ Son los primeros clientes que adoptan un producto determinado.

3.2.1. Desarrollo del primer MVP

Este MVP se comenzó el día 13 de octubre de 2020, se realizó en el marco de la asignatura de PIN, por lo que no entra en el trabajo de fin de carrera. Sin embargo, vale la pena comentarlo para saber cuál era el punto de salida de la aplicación para el TFG.

El objetivo de PIN, entre otros, era desarrollar y presentar una solución software en la feria de proyectos del 17 de diciembre de 2020. Para ello, se utilizó una metodología ágil, usando el concepto de sprint para ir ampliando la versión que se iba presentar más tarde en la feria. En este caso se hicieron tres sprints de aproximadamente 20 días.

- Primer sprint: en este se implementaron el mapa, los anuncios, la lista de rutas, los retos, las rutas guardadas y los lugares de interés.
- Segundo sprint: se creó la página de ruta, el perfil de usuario y amigos, el feed, las publicaciones, los comentarios y generados automático de rutas, además de mejoras de algunos aspectos del anterior sprint y algún fleco.
- Tercer sprint: se implementó el buscador de amigos, se pobló la aplicación de lugares de interés, se añadió la opción de crear rutas, se creó el sistema de puntos, los logros y la opción premium.

Después de todo esto, se presentó la aplicación como InstaGo en la feria de proyectos. Allí se recibió el feedback de varios profesionales del sector que ayudó a pivotar y cambiar el rumbo de la aplicación hacia un futuro más prometedor.

3.2.2. Desarrollo del segundo MVP

Con todo lo aprendido en la feria de proyectos, se planteó que cambios se podían hacer ya dentro del contexto del TFG. Con ello, se descartaron varias ideas que no aportaban o diluían la experiencia que se quería ofrecer a nuestros usuarios. Las ideas que se descartaron fueron: los anuncios, los retos, los logros y la opción premium. Es más, hasta se cambió el modelo de negocio, que paso de ser una fusión de freemium y publicitario a un modelo por suscripción.

Este MVP comenzó el día 25 de abril de 2021 y acabó el día 20 de abril de 2021. En él se hizo un lavado de cara estético completo a la aplicación para darle un aspecto más moderno y fácil de usar. Por otro lado, se implementó un nuevo sistema para poblar la base de datos y que en el mapa solo apareciese una ciudad, aunque pudieses cambiar entre ellas en todo momento y también se le añadieron los filtros para poder explorar las ciudades de manera más cómoda.

También se creó un inicio de sesión integrado para poder registrarte en la aplicación a través de Google, Facebook y Twitter. Se implementó un nuevo sistema para la creación de rutas mucho más intuitivo y que permitía hacer rutas temporales sin tener que guardarlas. Y por último se creó un sistema de notificaciones.

3.2.3. Desarrollo del tercer MVP

Con el primer experimento se recopiló mucha información de los early adopters, y este segundo MVP se centró principalmente en plasmar ese feedback que se había recibido en la aplicación para mejorar su experiencia, los principales añadidos que los pedían y que se desarrollaron en este MVP fueron: mantener la sesión abierta, actualizar la ruta a tiempo real, ver las rutas creadas y guardadas de otras ciudades en su perfil y que aparecieran fotos que se habían hecho otros usuarios en los lugares de interés.

Otras funcionalidades que se implementaron para este MVP para seguir cumpliendo con el backlog que tenías definido fueron las siguientes:

- Unificar el mapa para mejorar el rendimiento y que al utilizarlo la experiencia fuese más fluida.
- Crear un chat privado para poder hablar con tus amigos.
- Crear un sistema de alertas para que la aplicación te de feedback cuando la estás usando.
- Crear el perfil de empresa, junto con el sistema de la tienda, los descuentos y los lugares patrocinados.
- Crear lugares propios para el usuario.

3.3. Desarrollo primer Experimento

Al finalizar el desarrollo del primer MVP, se preparó un primer experimento para validar las características, previamente seleccionadas, con los early adopters elegidos. Debido a las medidas de Covid-19, no era viable realizarlo en un lugar físico agrupando a tanta gente, por lo que al final se decidió generar una apk que sería enviada a los usuarios para ser instalada y probada por ellos, junto a guía con los pasos a seguir durante el uso de esta para poder validar de forma segura una serie de funciones, esta era:

- Créate una cuenta.
- Explora los puntos de interés de tu ciudad.
- Crea una ruta y publícala.
- Hazte amigo o sigue a alguien.
- Habla a través del chat privado a tus amigos.
- Cambia tu perfil.
- Explora las rutas de tus amigos y otros usuarios.
- Hazte colaborador (opcional).
- Intenta hacer una ruta y publicar una foto en alguno de los sitios en lo que hayas estado.
- Explora los descuentos en tu ciudad.

Además, se añadió un formulario de Google Forms para que lo rellenasen.

Las preguntas que se incluyeron en este estaban organizadas en varios grupos para extraer diferente información específica: datos personales, hábitos de uso, usabilidad de la aplicación, valoración de la aplicación y sugerencias.

Todas estas preguntas estaban explicadas de forma individual en el formulario para evitar confusiones, pero al obtener las gráficas de respuesta de cada una, la explicación ha sido omitida.

Las respuestas obtenidas, ordenadas por categoría, fueron las siguientes:

Datos Personales:

Edad
20 respuestas

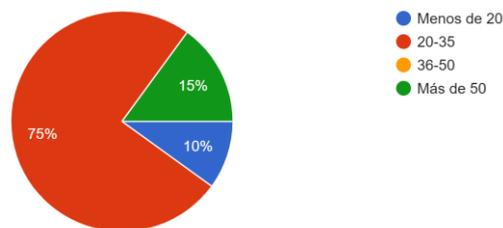


Ilustración 8, Edad

Hábitos de Uso:

Frecuencia de uso de dispositivos móviles
20 respuestas

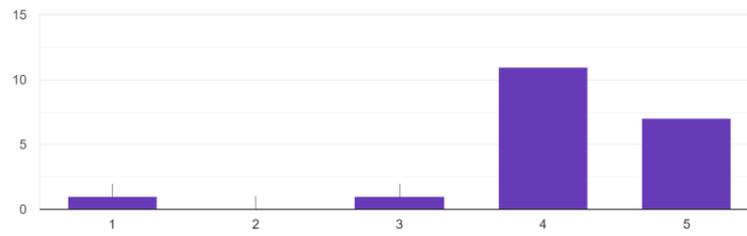


Ilustración 9, Frecuencia de uso de dispositivos móviles

¿Utilizas alguna aplicación que te ayude en tus viajes?
20 respuestas

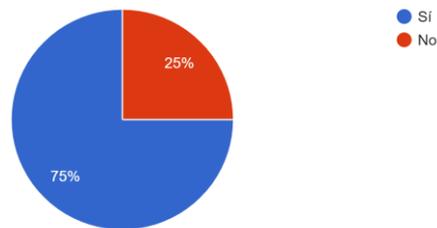


Ilustración 10, ¿Utilizas alguna aplicación que te ayude en tus viajes?

¿Cuáles son estas aplicaciones?
15 respuestas

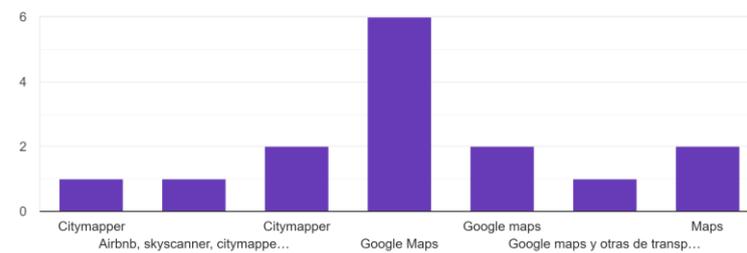


Ilustración 11, ¿Cuáles son estas aplicaciones?

¿Con que frecuencia utilizas estas aplicaciones?
19 respuestas

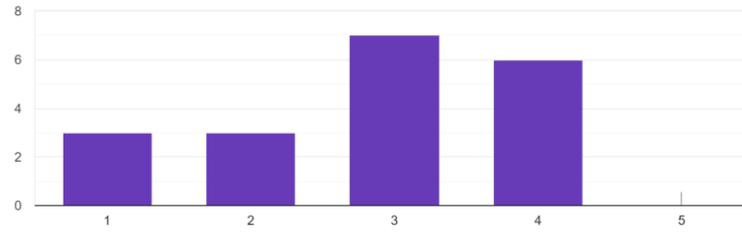


Ilustración 12 ¿Con qué frecuencia usas estas aplicaciones?

Usabilidad de la aplicación:

¿Crees que la aplicación es fácil de usar?
20 respuestas

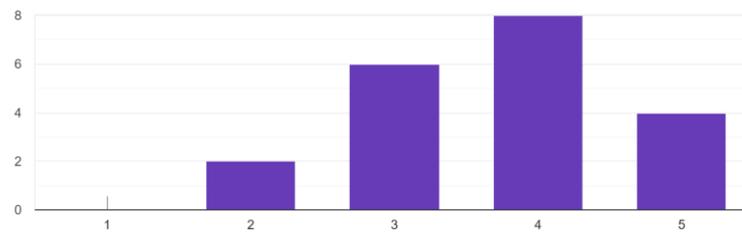


Ilustración 13, ¿Crees que la aplicación es fácil de usar?

Visibilidad del estado del sistema
20 respuestas

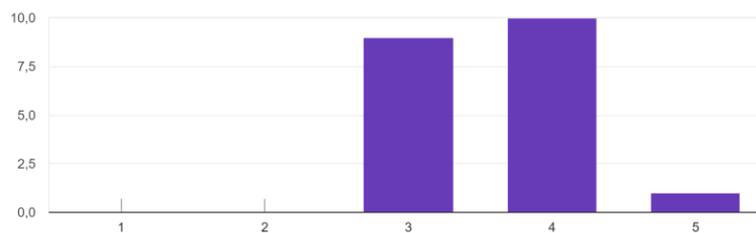


Ilustración 14, Visibilidad del estado del sistema

Relación entre el sistema y el mundo real
20 respuestas

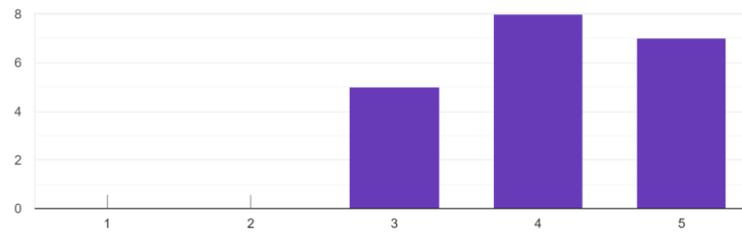


Ilustración 15, Relación entre el sistema y el mundo real

Control y libertad del usuario
20 respuestas

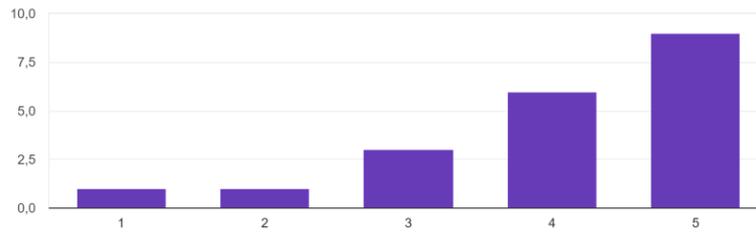


Ilustración 16, Control y libertad del usuario

Consistencia y estándares
20 respuestas

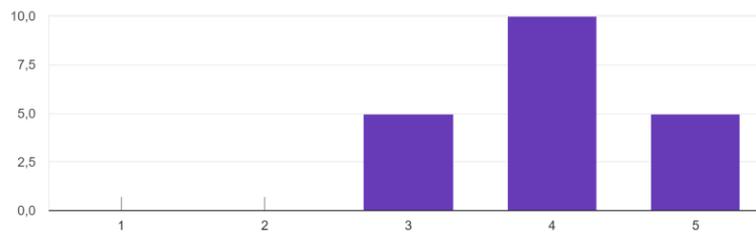


Ilustración 17, Consistencia y estándares

Prevención de errores
20 respuestas

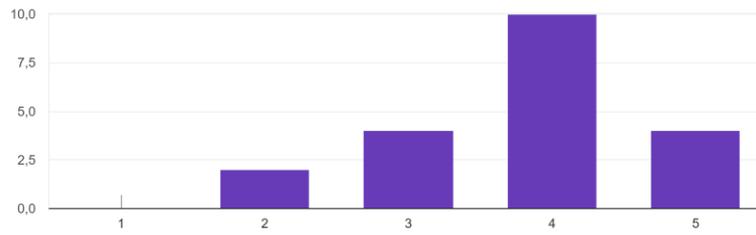


Ilustración 18, Prevención de errores

Reconocimiento antes de recuerdo
20 respuestas

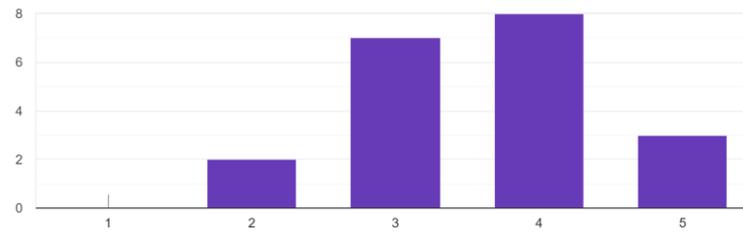


Ilustración 19, Reconocimiento antes de recuerdo

Flexibilidad y eficiencia de uso
20 respuestas

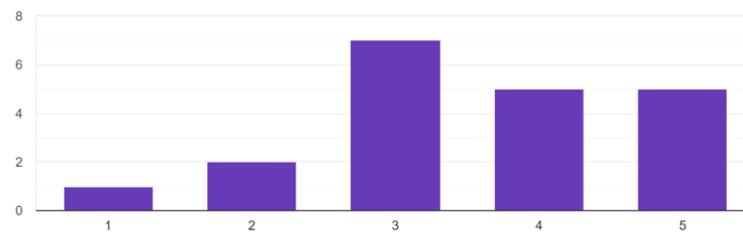


Ilustración 20, Flexibilidad y eficiencia de uso

Estética y diseño minimalista
20 respuestas

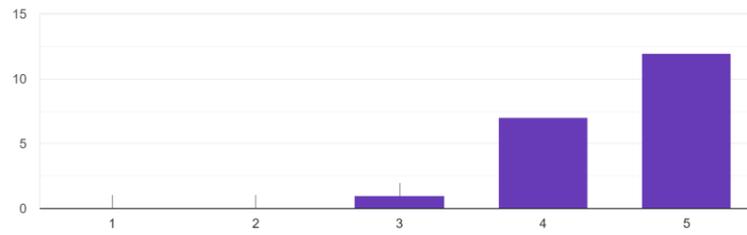


Ilustración 21, Estética y diseño minimalista

Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de errores
20 respuestas

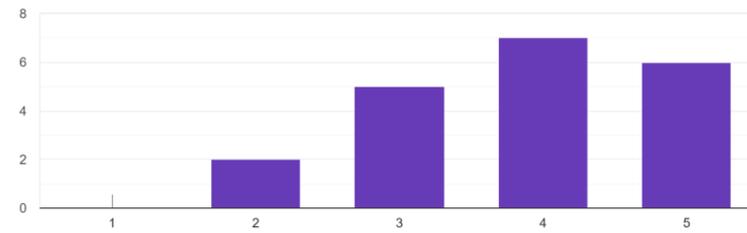


Ilustración 22, Reconocer, diagnosticar y recuperarse de errores

¿Te gustaría ver desde los lugares de interés fotos de las publicaciones que se han hecho en ese sitio?
20 respuestas

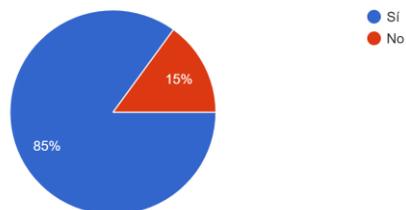


Ilustración 23, Fotos de las publicaciones que se han hecho en ese sitio

Valoración ideas de la aplicación:

¿Te parece interesante el enfoque como red social que tiene la aplicación?
20 respuestas

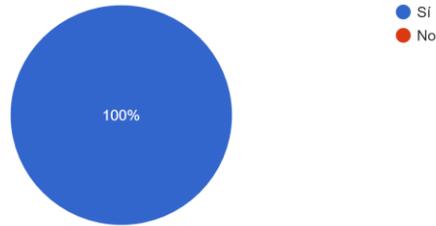


Ilustración 24, Enfoque como red social que tiene la aplicación

¿Prefieres que los lugares de las ciudades los puedas guardar en local en tu teléfono (para mejorar los tiempos de carga de esa ciudad) o que no se guarde nada en local aun sabiendo que irá mas lento?
20 respuestas

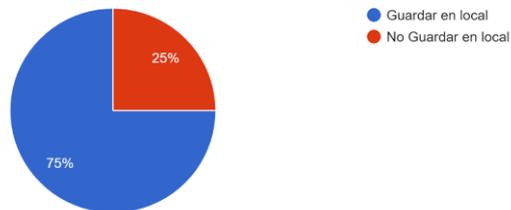


Ilustración 25, ¿Prefieres que los lugares los puedas guardar en local?

¿Cuántos lugares de interés te gustaría que tuviera cada ciudad?
20 respuestas

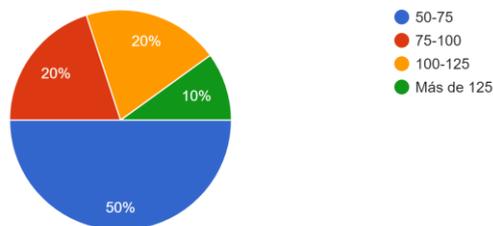


Ilustración 26, ¿Cuántos lugares de interés te gustaría que tuviera cada ciudad?

Me interesa mucho que la aplicación tenga todas las ciudades del mundo
20 respuestas

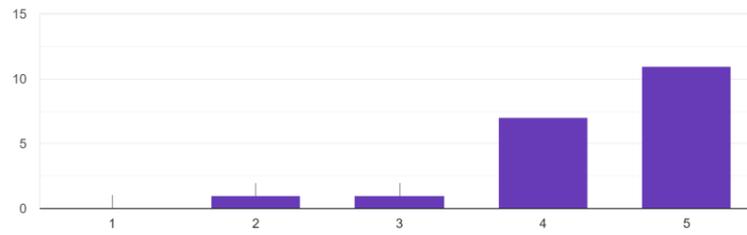


Ilustración 27, Aplicación tiene todas las ciudades del mundo

Me interesaría más que la aplicación solo tuviese 1 o 2 ciudades, pero que mostrara todos los lugares de interés de la ciudad sin excepción
20 respuestas

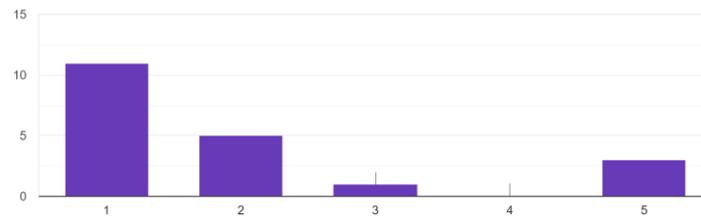


Ilustración 28, Me interesaría más que la aplicación solo tuviese 1 o 2 ciudades

¿Echas de menos algún filtro en el mapa? ¿Cuáles pondrías?

6 respuestas

Parques y hoteles o albergues

Hoteles y hostales,

Talleres

Transporte público y comisarias

Transporte publico

Ilustración 29, ¿Echas de menos algún filtro en el mapa?

¿Querías ver tus propias publicaciones en el feed?
20 respuestas

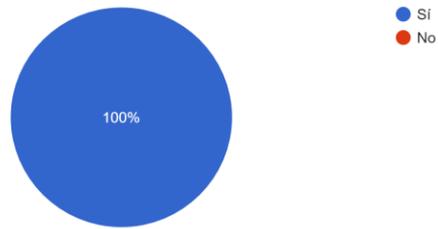


Ilustración 30, ¿Querías ver tus propias publicaciones en el feed?

En tu perfil, ¿querías que te aparecieran todas las rutas guardadas o solo las de la ciudad en la que te encuentras?
20 respuestas

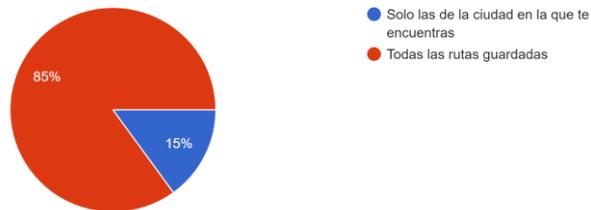


Ilustración 31, ¿querías que te aparecieran todas las rutas guardadas?

En tu perfil, ¿querías que te aparecieran todas las rutas creadas o solo las de la ciudad en la que te encuentras?
20 respuestas

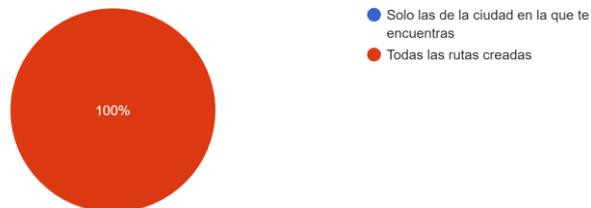


Ilustración 32, ¿Querías que te aparecieran todas las rutas creadas?

¿Te gustaría tener un chat privado en el que hablar con tus amigos?
20 respuestas

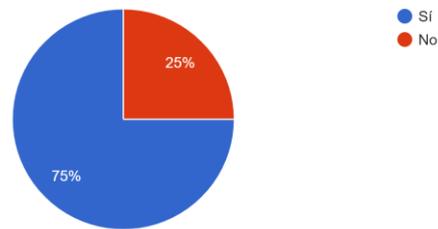


Ilustración 33, ¿Te gustaría tener un chat privado?

Sugerencias:

En caso de que viajaras a una ciudad que no conoces. ¿Estarías interesado en usar esta aplicación?
20 respuestas

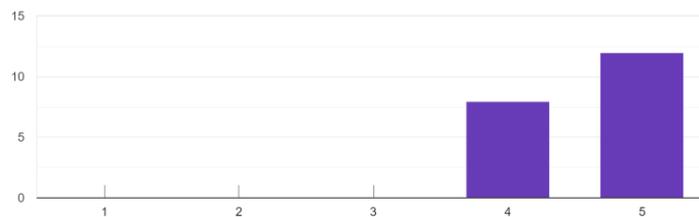


Ilustración 34, ¿Estarías interesado en usar esta aplicación?

En la pregunta de ¿Qué cambios o mejoras querías que implementáramos en la aplicación?, se obtuvieron las siguientes respuestas:

- Mantener iniciada la sesión
- Recalcular la ruta según tu posición
- Si realizas la foto directamente desde la app no se guarda en tu móvil y no hay forma de guardarte la foto realizada.
- Cambiar los iconos para representar mejor las acciones
- No está claro como entrar al lugar de interés
- Al volver para atrás después de guardar una ruta vuelve a la creación
- Editar la ruta creada

Como resultado de este experimento se obtuvieron ideas para nuevas funcionalidades que se podrían implementar como la creación de nuevos lugares de interés y otras que se deberían de mejorar y arreglar, como la descarga de imágenes o mantener iniciada la sesión.

También se consolidaron las nuevas historias de usuario que se iban a incluir en el siguiente MVP como el chat y la visualización de todas las rutas creadas independientemente de la localidad y se validaron las que ya estaban incluidas como la cantidad de filtros y lugares de interés.

Por último, en la parte de verificación de la usabilidad del sistema, se encontró una falta de feedback por parte de la aplicación hacia el usuario para confirmar las acciones que iba realizando o notificaciones de realización de estas como añadir a un amigo o seguidor, y la falta de claridad o de ejecución de ciertas funciones, como entrar a un lugar de interés o iconos ambiguos.

3.4. Desarrollo segundo Experimento

Con respecto al segundo experimento se utilizó una dinámica muy similar a la del anterior, constando de un análisis de usabilidad, haciendo las mismas preguntas con el objetivo de comprobar si se ha mejorado con respecto a anterior cuestionario, y una validación de las funcionalidades que se pretenden introducir en el futuro, para comprobar si los early adopters están interesados en ellas o no.

Se volvió a realizar con las mismas limitaciones debido a la situación sanitaria, por lo que se distribuyó la aplicación entre unos cuantos usuarios y se les proporciono una lista de funcionalidades que se quieren validar, que fueron las siguientes:

- Créate una cuenta.
- Explora los puntos de interés de tu ciudad.
- Crea una ruta y publícala.
- Crea tu propio lugar de interés.
- Hazte amigo o sigue a alguien.
- Habla a través el chat privado a tus amigos.
- Cambia tu perfil.
- Explora las rutas de tus amigos y otros usuarios.
- Hazte colaborador (opcional).
- Intenta hacer una ruta y publicar una foto en alguno de los sitios en lo que hayas estado.
- Explora los descuentos en tu ciudad.
- Cambia de ciudad.

Por último, se les volvió a compartir un Formulario de Google para que respondieran y las respuestas que se obtuvieron fueron las siguientes:

Datos Personales:

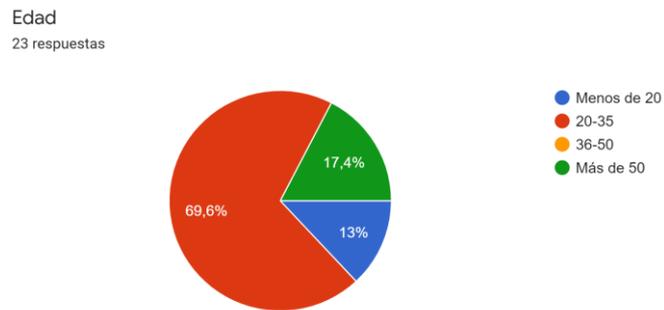


Ilustración 35, Edad 2

Hábitos de Uso:

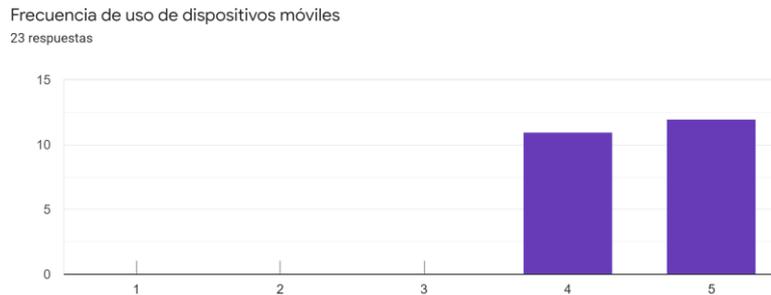


Ilustración 36, Frecuencias de uso de dispositivos móviles 2

¿Utilizas alguna aplicación que te ayude en tus viajes?
23 respuestas

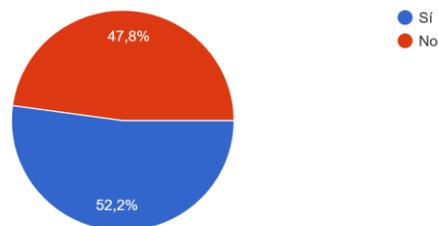


Ilustración 37 ¿Utilizas alguna aplicación que te ayude en tus viajes? 2

¿Cuáles son estas aplicaciones?

9 respuestas

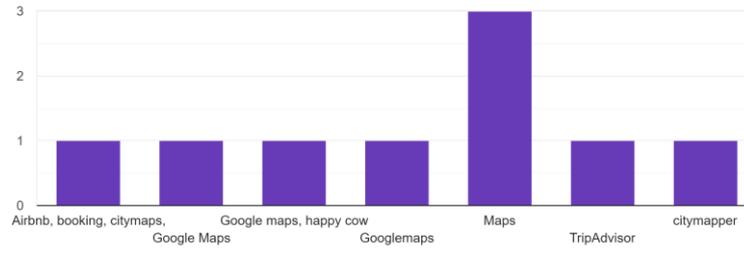


Ilustración 38, ¿Cuáles son esas aplicaciones? 2

¿Con que frecuencia utilizas estas aplicaciones?

19 respuestas

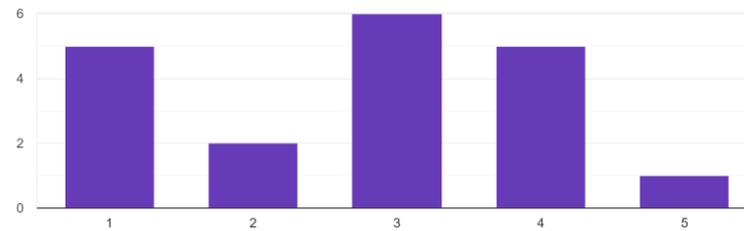


Ilustración 39, ¿Con que frecuencias utilizas estas aplicaciones? 2

Usabilidad de la aplicación:

¿Crees que la aplicación es fácil de usar?

23 respuestas

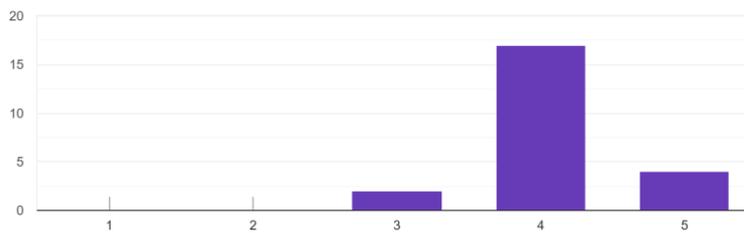


Ilustración 40, ¿Crees que la aplicación es fácil de usar? 2

Visibilidad del estado del sistema
23 respuestas

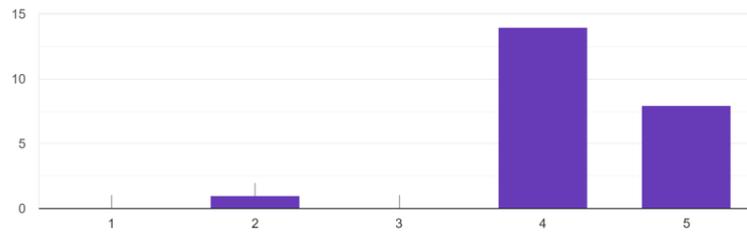


Ilustración 41, Visibilidad del estado del sistema 2

Relación entre el sistema y el mundo real
23 respuestas

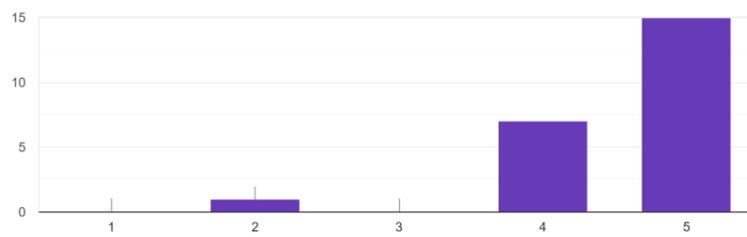


Ilustración 42, Relación entre el sistema y el mundo real 2

Control y libertad del usuario
23 respuestas

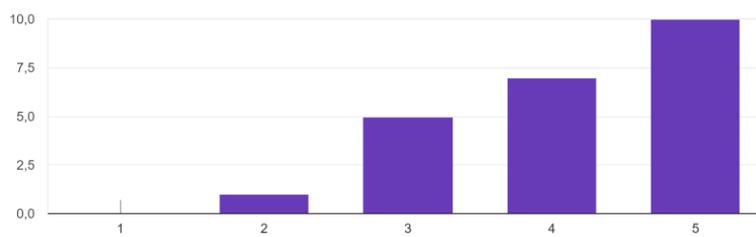


Ilustración 43, Control y libertad 2

Consistencia y estándares
23 respuestas

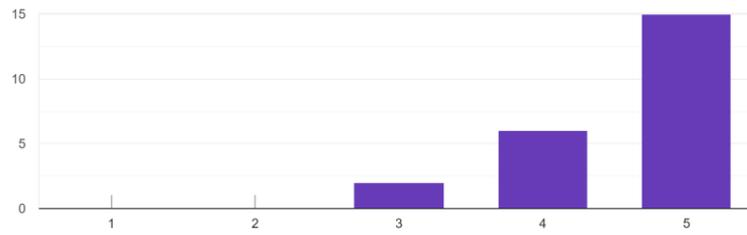


Ilustración 44, Consistencia y estándares 2

Prevención de errores
23 respuestas

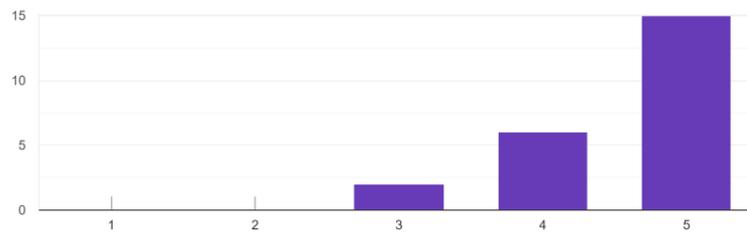


Ilustración 45, Prevención de errores 2

Reconocimiento antes de recuerdo
23 respuestas

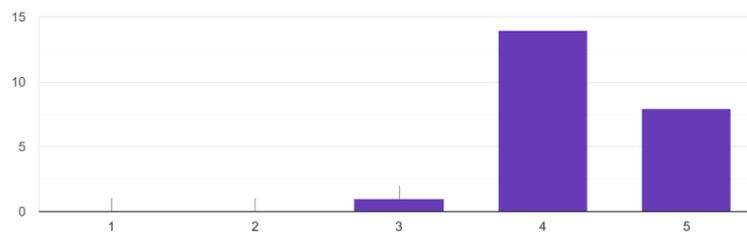


Ilustración 46, Reconocimiento antes de recuerdo 2

Flexibilidad y eficiencia de uso
23 respuestas

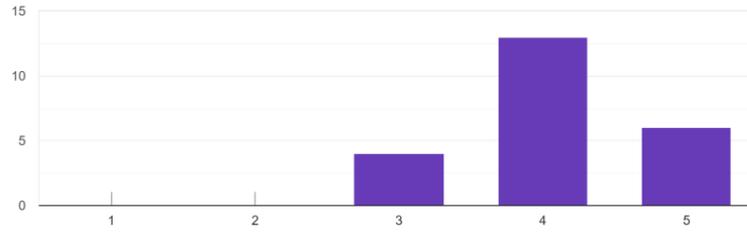


Ilustración 47, Flexibilidad y eficiencia de uso 2

Estética y diseño minimalista
23 respuestas

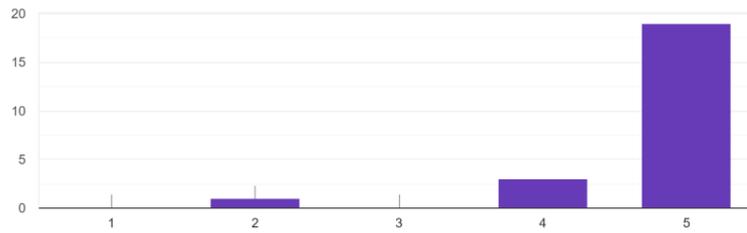


Ilustración 48, Estética y diseño minimalista 2

Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de errores
23 respuestas

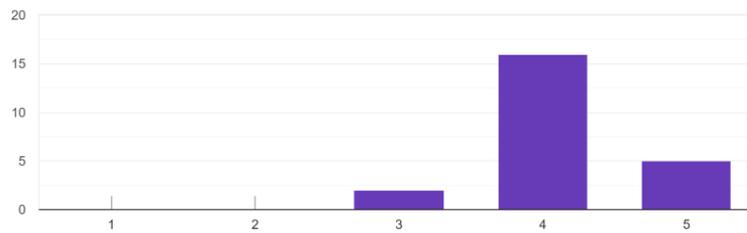


Ilustración 49, Reconocer, diagnosticar y recuperarse de errores 2

Valoración ideas de la aplicación:

¿Te gustaría poder hacer rutas grupales? Con tus amigos
23 respuestas

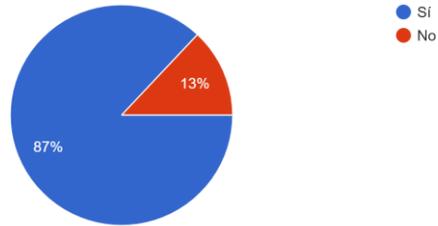


Ilustración 50, ¿Te gustaría poder hacer rutas grupales?

¿Te gustaría mantener fotos antiguas del perfil?
23 respuestas

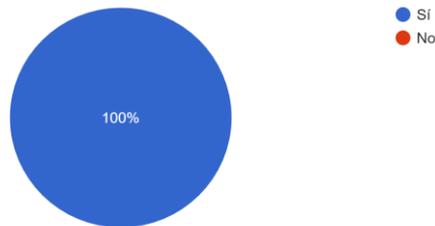


Ilustración 51, ¿Te gustaría mantener fotos antiguas de perfil?

¿Te gustaría poder compartir rutas y publicaciones fuera de la aplicación?
23 respuestas

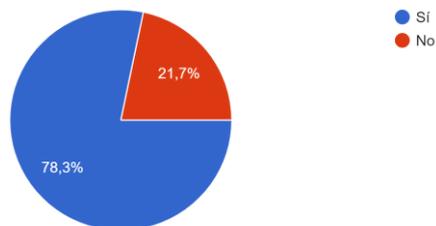


Ilustración 52, Compartir rutas y publicaciones fuera de la aplicación

¿Te gustaría poder comentar en los lugares de interés?
23 respuestas

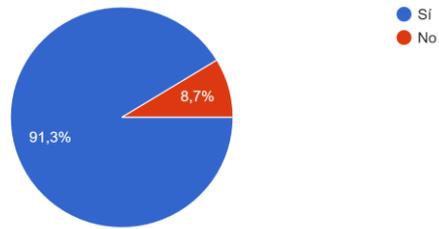


Ilustración 53, ¿Te gustaría poder comentar en los lugares de interés?

¿Te gustaría poder valorar los lugares de interés?
23 respuestas

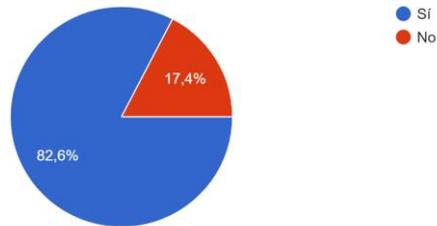


Ilustración 54, ¿Te gustaría poder valorar los lugares de interés?

¿Te gustaría poder crear un perfil privado? Para que no puedan ver tus publicaciones y rutas a no ser que lo permitas
23 respuestas

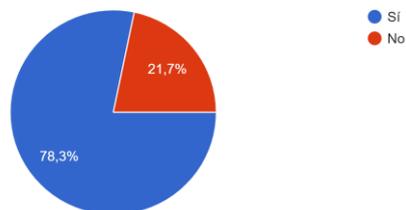


Ilustración 55, ¿Te gustaría poder crear un perfil privado?

Sugerencias:

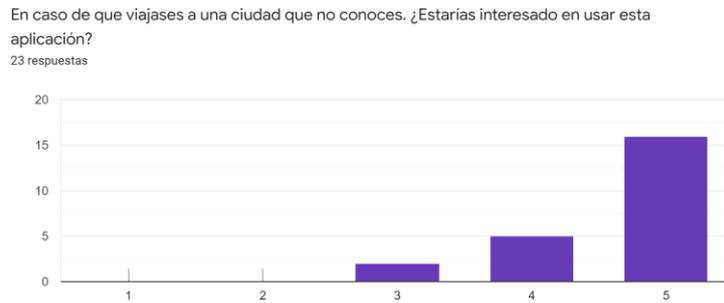


Ilustración 56, ¿Estarías interesado en usar esta aplicación? 2

En la pregunta de ¿Qué cambios o mejoras querrías que implementáramos en la aplicación?, se obtuvieron las siguientes respuestas:

- Mayor fluidez y velocidad
- Marcar las zonas de ocio, cultura, etc....
- Iconos más grandes.
- Implementar una hoja de ruta que te diga que calles, por donde girar y distancias, estilo lo que tiene Google Maps
- Botón en los filtros para reiniciar la búsqueda cuando tienes muchos filtros puestos
- Horarios de visita y precios

Como resultado de este experimento se obtuvieron ideas para nuevas funcionalidades que se pudieran implementar como la creación de zonas de actividades y la implementación de una hoja de ruta.

También se consolidaron las nuevas historias de usuario que se iban a incluir en el siguiente MVP como el perfil privado, y la inclusión de un sistema de valoraciones y comentarios en los lugares de interés, además de la inclusión de la posibilidad de compartir rutas y publicaciones fuera de la aplicación.

Por último, en la parte de verificación de la usabilidad del sistema se ha conseguido el objetivo de mejorar la usabilidad en comparación al anterior MVP. Ya que se ha conseguido mejorar los resultados en todos los aspectos quitando algún caso aislado. Con respecto a este punto, la mayor queja a sido que se necesita mejorar la fluidez de la aplicación para mejorar la experiencia.

4. Aspectos técnicos

En este apartado se muestran y se explican todas las técnicas y herramientas utilizadas, los entornos en los cuales se ha desarrollado, la arquitectura y diseño, además de los desafíos encontrados junto a métricas generales de la aplicación.

4.1. Técnicas y herramientas utilizadas

NomadWay ha sido desarrollada en su totalidad utilizando el framework Flutter¹⁵, creado y mantenido por Google, diseñado para la creación de apps nativas para móviles y web con el mismo código para todas las plataformas.

Se utilizó esta herramienta por la facilidad que otorga para la creación de interfaces, ya que tiene preconfiguradas unas opciones gráficas que se adaptan al sistema nativo en el que se esté desarrollando, además de la posibilidad de reutilización y generalización de código, pudiendo utilizar widgets y vistas en diferentes partes de la aplicación.

Por último, al ser desarrollado por Google y tener bastante popularidad en este momento, se puede encontrar una gran cantidad de información para resolución de dudas e ideas de implementación, además de tener una gran documentación detallada con varios ejemplos de uso [2].

¹⁵ <https://flutter.dev/>

```

1. import 'package:flutter/material.dart';
2.
3. void main() => runApp(MyApp());
4.
5. class MyApp extends StatelessWidget {
6.   @override
7.   Widget build(BuildContext context) {
8.     return MaterialApp(
9.       title: 'Hello World en Flutter',
10.      home: Scaffold(
11.        appBar: AppBar(
12.          title: const Text('Hello World'),
13.        ),
14.        body: const Center(
15.          child: Text('Hello World'),
16.        ),
17.      ),
18.    );
19.  }
20. }

```

Ilustración 57, Hello World en Flutter

El lenguaje de programación utilizado durante todo el desarrollo ha sido Dart¹⁶, también creado y mantenido por Google, necesario para poder desarrollar en Flutter. Está pensado para ofrecer una alternativa más moderna a JavaScript¹⁷, pues tienen bastante similitudes y ambos están pensados para el desarrollo de clientes, aunque también se puede utilizar para el desarrollo de la lógica, unificando los lenguajes de front-end¹⁸ y back-end¹⁹ a uno solo.

También se encuentra optimizado para el desarrollo de interfaces web, utilizando un compilador JIT (Just in time), que permite el Hot Reload, para la recarga al instante de la aplicación agilizando el desarrollo, y DevTools, poderosa suite de herramientas y debugger de código de la que se hablará en detalle más adelante.

Cuando la aplicación se encuentra desplegada en el dispositivo utilizará el compilador AOT (Ahead of time), ofreciendo un corto tiempo de inicio y optimizando la memoria del dispositivo [3].

¹⁶ <https://dart.dev/>

¹⁷ <https://developer.mozilla.org/es/docs/Web/JavaScript>

¹⁸ Parte visual del programa, aquella que ven los clientes.

¹⁹ Parte trasera del programa, encargada de la lógica que realiza la funcionalidad principal.

```
1. void main() {  
2.   print('Hello, World!');  
3. }
```

Ilustración 58, Hello World en Dart

La suite de DevTools²⁰ comprende herramientas de debugger de código y de interfaces gráficas, carga de CPU, vista del uso de la memoria, vista de rendimiento de la aplicación y de las peticiones a internet y una consola para mostrar registros.

Se ha utilizado esta herramienta para observar y medir los rendimientos en las distintas vistas y métodos utilizados durante el desarrollo, detectando loops de conexiones a internet que ralentizaban y bloqueaban la aplicación, además de métodos que no eran eficientes e impedían la carga de los widgets.

El debugger de esta suite estaba ya implementado en la extensión de Dart para Visual Studio Code²¹, entorno de desarrollo del que se hablará en el apartado siguiente, agilizando el análisis de código para identificar fallos y bugs [4].

Para las funcionalidades principales de la aplicación se han utilizado las API proporcionadas por Google Cloud²², utilizando el servicio gratuito que ofrecen durante el desarrollo.

Concretamente, se han utilizado las siguiente API:

- Directions, para la creación de rutas de un punto a otro.
- Geocoding y Geolocator, para la obtención de coordenadas y direcciones, además de la localización precisa del usuario.
- Maps SDK for Android, para la obtención de un mapa a tiempo real de la localización.
- Places, para obtener todos los lugares de interés y sus detalles.

También se utilizó durante un tiempo AdMob para la creación de anuncios para monetizar la aplicación, aunque finalmente se optó por cambiar de modelo de negocio.

Todas estas API cuentan con una gran y extensa documentación de uso con claros ejemplos de implementación [5].

²⁰ <https://flutter.dev/docs/development/tools/devtools/overview>

²¹ <https://code.visualstudio.com/>

²² <https://cloud.google.com/>

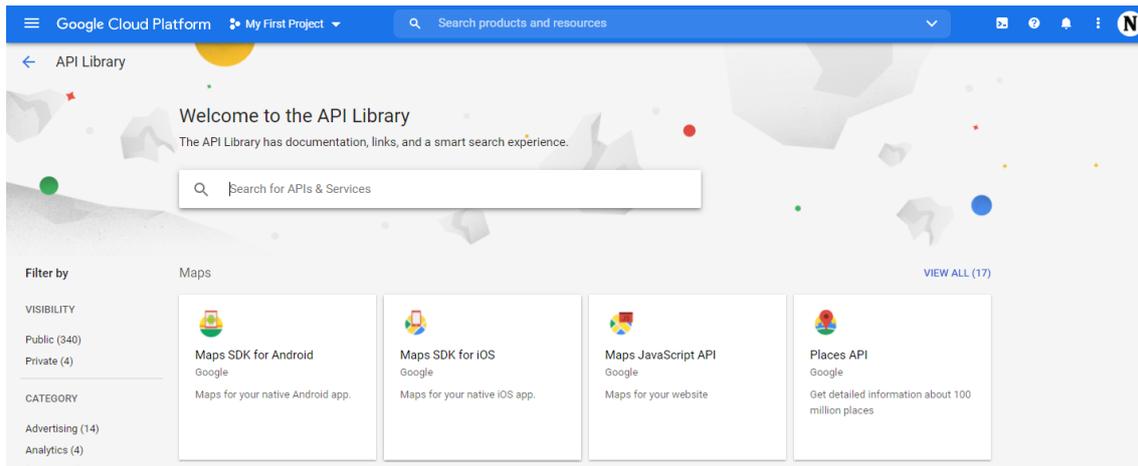


Ilustración 59, Google Cloud Consola

Como control de versiones se ha utilizado la herramienta de Git²³, es la más reconocida y estándar en la industria, sencillo de utilizar y de código abierto. De repositorio se utiliza GitHub²⁴, ofrece un servicio gratuito excelente y suficiente para el almacenamiento de código desarrollado, utiliza Git para controlar las versiones [6].

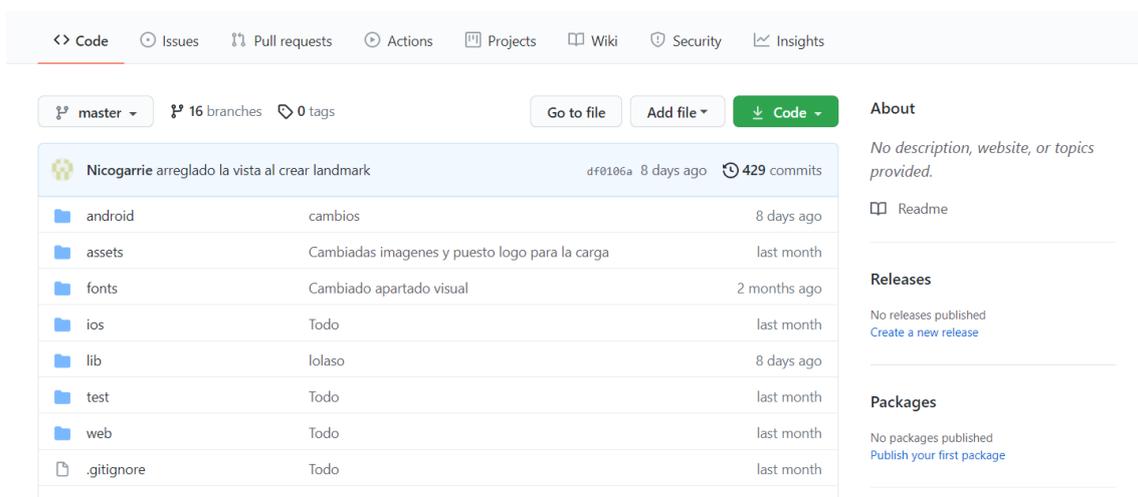


Ilustración 60, GitHub

Por último, para la implementación de la base de datos, se utiliza Firebase²⁵, servicio de nube online también creado por Google.

Con la ayuda de esta plataforma, se ha conseguido externalizar la base de datos, obteniendo una mayor libertad e interacción entre desarrolladores, pudiendo compartir la información de forma constante, pues es un servicio que siempre está disponible en internet.

²³ <https://git-scm.com/>

²⁴ <https://github.com/>

²⁵ <https://firebase.google.com/?hl=es>

Se ha utilizado el plan gratuito que ofrecen, alojando espacio de sobra para todas las fotos utilizadas y guardadas durante el desarrollo, pero el desarrollo ha sido lastrado con el poco ancho de banda que se concedía al día, ya que se bloqueaba hasta el día siguiente [7].

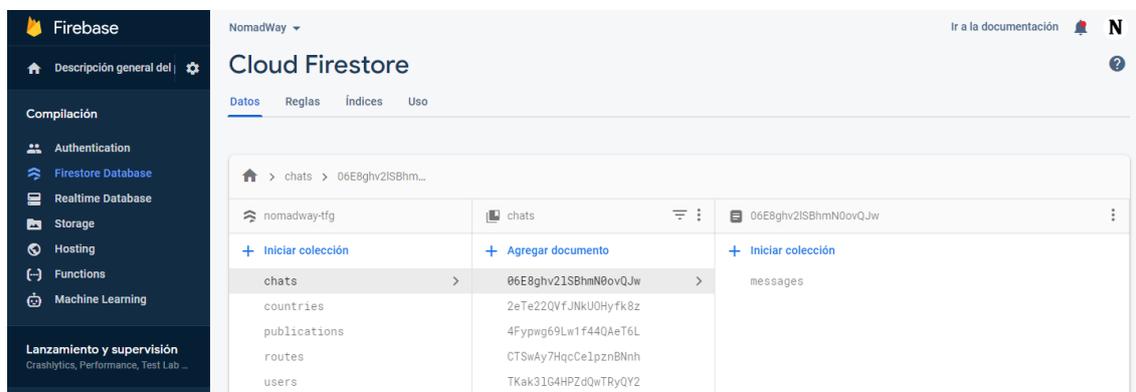


Ilustración 61, Firebase

4.2. Entornos de desarrollo

El entorno de desarrollo principal elegido para trabajar con la aplicación ha sido Visual Studio Code, elegido por su facilidad de uso y su velocidad para iniciar y creación de código, también posee una extensa cantidad de extensiones, que permiten configurar el entorno para adaptarse a distintos tipos de desarrollo.

Por ejemplo, se ha utilizado la extensión oficial de Flutter y Dart, que facilita el desarrollo ágil ofreciendo integración total del debugger de Dart Dev Tools, tanto de código como de interfaz gráfica, también reconoce estructuras y clases de objetos propios del framework, permitiendo realizar las refactorizaciones más comunes, como la envoltura de fragmentos de código en otros contenedores, práctica muy común trabajando con Flutter. Por último, también ofrece atajos a la ejecución y despliegue de la aplicación en los emuladores creados en el siguiente entorno de desarrollo [8].

El segundo y último entorno ha sido Android Studio²⁶, en el que no se ha creado código, pero necesario para poder desplegar la aplicación en nuestros móviles o en emuladores de Android generados a medida.

²⁶ <https://developer.android.com/>



Durante las primeras etapas de desarrollo se utilizó un emulador para la ejecución de la app, pero, a medida que se iban desarrollando nuevas funcionalidades, se detectó un rendimiento limitado que se solucionó desplegando directamente en móviles, obteniendo un acercamiento más fiel al despliegue en los dispositivos del usuario final [9].

4.3. Base de datos

A continuación, se explican y detallan los modelos de datos que se han utilizado para el desarrollo y la estructura de la base de datos implementada.

4.3.1. Estructura de la base de datos

Para la base de datos se ha utilizado Cloud Firestore, que es una base NoSQL, lo que le permite obtener una gran escalabilidad horizontal, permitiendo tener una gran cantidad de documentos en las colecciones (alternativa a los registros y tablas respectivamente de una base de datos SQL), estando optimizado para una gran velocidad en la creación y consulta de dichos documentos.

Estos poseen una gran flexibilidad, permitiendo modificarlos con facilidad añadiendo nuevos campos o alterando los tipos de datos que almacenan. También sus consultas ofrecen una velocidad proporcional a los resultados obtenidas con las mismas, no de la cantidad de datos que contengan las colecciones.

Por último, al ser una base de datos online en la nube con sincronización a tiempo real, se pueden obtener actualizaciones rápidamente mientras se tuviera internet, aunque también permite el almacenamiento en caché de los datos consultados para poder realizar consultas sin estar conectados.

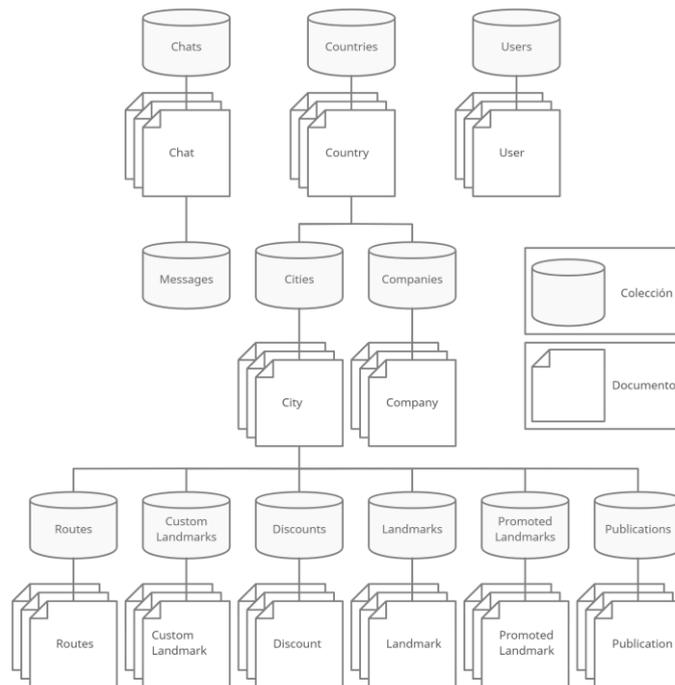


Ilustración 62, Esquema de la base de datos

Por otro lado, se utiliza el servicio de almacenamiento Storage para guardar datos más pesados que necesitarían muchos recursos para poder ser almacenados en Firestore, como las imágenes de los lugares registrados en la aplicación o las fotos subidas de los usuarios.

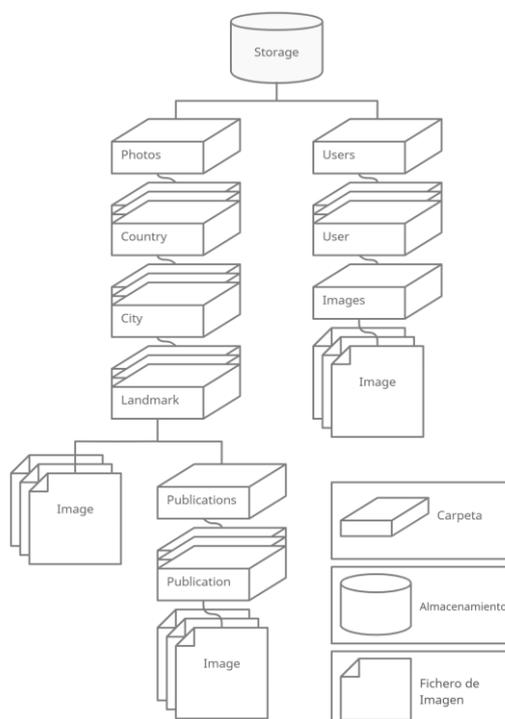


Ilustración 63, Esquema del Storage de Firebase

4.3.2. Modelo de datos

El modelo de datos utilizado en la arquitectura de la base de datos varía del resto debido a las características propias de NoSQL y de su flexibilidad. Por ejemplo, las tablas tradicionales se transforman en colecciones donde se almacenan los documentos, que serían el equivalente a las filas en SQL, estos documentos pueden almacenar otras colecciones o directamente información, permitiendo la agregación con total libertad de nuevos campos o la modificación de los tipos de estos.

Entre estos tipos destacan las referencias, que serían el equivalente a las claves ajenas de las bases de datos más tradicionales, pudiendo acceder fácilmente a otros documentos en distintas colecciones.

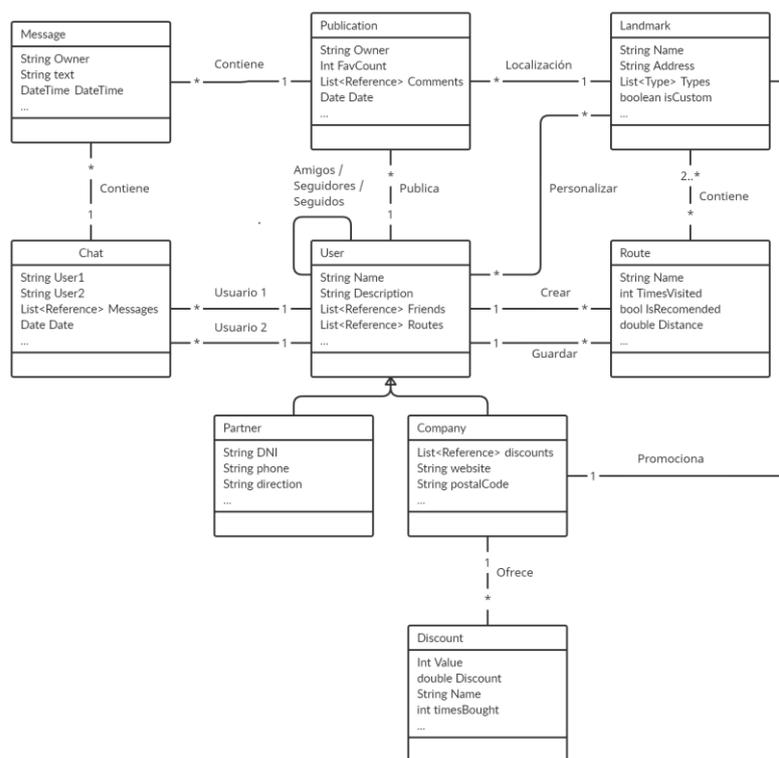


Ilustración 64, Modelo de datos

4.3.3. Base de Datos Local

Por último, se ha utilizado un paquete de desarrollo creado por la comunidad de Flutter y publicado en Pub.dev²⁷, Hive²⁸, con el objetivo de crear un pequeño almacenamiento local de información en forma de Json con un funcionamiento similar al de un Map, relacionando claves con un valor, de forma que se consigue una gran eficiencia en la obtención y almacenamiento de los datos que se quiera.

También posee un sistema propio de encriptación para proteger los datos almacenados ya que hay información sensible como información de los lugares de interés o identificadores de otros usuarios [10].

4.4. Patrones de diseño

A continuación, se explican algunos patrones de diseño que se han implementado durante el desarrollo de la aplicación.

4.4.1. Proxy

Proxy es un patrón estructural que controla el acceso desde la aplicación a un objeto mediante un intermediario, creando este objeto solamente cuando sea necesario de utilizar en nuestra aplicación se ha utilizado este patrón para controlar el acceso a la base de datos mediante unas clases llamadas Service que proporcionan una interfaz para la manipulación de datos en esta.

Estos Service son los encargados de crear, borrar, modificar, buscar... todos los elementos necesarios de base de datos, para ello, se ha implementado una serie de clases que contienen una instancia a la información y una serie de métodos que ejecutan la lógica necesaria para su funcionamiento.

²⁷ <https://pub.dev/>

²⁸ <https://pub.dev/packages/hive>



```
1. void addLandmarkToLocality(Landmark landmark) {
2.     var docRef = dbInstance.doc(
3. 'countries/${landmark.country}/cities/${landmark.locality}/landmarks/${landmark.id}
4. ');
5.     docRef.set({
6.         'name': landmark.name,
7.         'location': GeoPoint(landmark.latitude, landmark.longitude),
8.         'description': landmark.description,
9.         'address': landmark.address,
10.        'locality': landmark.locality,
11.        'country': landmark.country,
12.        'icon': landmark.icon,
13.        'types': landmark.types,
14.        'isCustom': landmark.isCustom,
15.        'isPromoted': landmark.isPromoted,
16.    });
17. }
```

Ilustración 65, Patrón Proxy

En total se han creado siete clases distintas de Service para gestionar distintos aspectos de la aplicación, como la tienda, los lugares de interés o los usuarios.

Como principales ventajas destacan la facilidad para controlar el acceso a la base de datos, pues una vez se destruye la ventana dónde se utiliza este proxy la instancia se descarta, y la facilidad de inclusión de nuevas clases Service en el caso de que se requieran sin modificar el propio servicio que accede a base de datos.

Por último, como principal inconveniente se quiere mencionar la lógica extra que supone crear todas las nuevas clases que se van a encargar de controlar el acceso a la base de datos.

4.4.2. Adaptador

Adaptador es un patrón estructural que permite la comunicación entre dos sistemas incompatibles, en la aplicación, se ha tenido que implementar para poder utilizar los servicios de Google ya que aún no están disponibles de forma nativa.

Para ello se ha utilizado principalmente librerías creadas por otros usuarios de la comunidad de Flutter, recogidas en la página de Pub.Dev, que sirven para adaptar o crear funcionalidad extra.

También se ha tenido que adaptar la conexión a los servicios de Google pues el adaptador que se estaba usando no era capaz de soportar y proporcionar los resultados que se necesitaban

para el correcto funcionamiento de la aplicación, de estas implementaciones se habla más adelante en la API de Google Places.

Para finalizar con este patrón, se quiere destacar la principal ventaja que es la de poder utilizar unos servicios que de otra forma serían inutilizables y la mayor desventaja que sería el código extra necesario para poder implementar correctamente un adaptador, además de conocer la documentación del servicio, especialmente si es de terceros como es este caso, puede conducir a situaciones de error o desconocimiento.

4.4.3. Observador

Observador es un patrón de diseño de comportamiento que permite escuchar los eventos definidos que ocurran en un objeto indicado y notificar del cambio a otros objetos que estén suscritos.

En NomadWay se desarrolla una pequeña implementación de este patrón para notificar los eventos especificados como el envío de un mensaje, la recepción de otro, añadir amigos o seguidores... al generador de notificaciones, observando cambios en base de datos en una lista especial de cada usuario a la que se añaden estas notificaciones.

```
1. Stream collectionStream = FirebaseFirestore.instance
2.     .collection('users/${globals.userLogged.id}/notifications')
3.     .snapshots();
4. return collectionStream.listen((event) async {
5.     if (event.docs.isNotEmpty) {
6.         for (var docChange in event.docChanges) {
7.             DocumentSnapshot doc = docChange.doc;
8.             //Sacas los datos del doc y los usas en la notificacion
9.             NotificationService.notify(doc);
10.        }
11.    }
12. });
```

Ilustración 66, Primera implementación observador

La segunda implementación de este patrón ha sido la de suscribirse a la escucha de eventos de cambio de posición del usuario para poder tener más controlada la realización de la ruta ya

que, cuando el usuario se está aproximando al final de esta, debería de aparecerle la opción de finalizarla y obtener sus puntos.

```

1.     GeolocatorPlatform.instance.getPositionStream().listen((event) {
2.         currloc = event;
3.     });

```

Ilustración 67, Segunda implementación de observador

Como tercera y última implementación que se quiere mencionar es la del Widget StreamBuilder, que es un fragmento de código ya creado por Flutter que proporciona de forma rápida y fácil un observador que notifica al resto de componentes hijo sobre los cambios que se han realizado en el Stream que está escuchando, se ha utilizado para obtener los nuevos mensajes en un chat en tiempo real o para conseguir las nuevas publicaciones que se hayan creado recientemente.

```

1.  StreamBuilder<dynamic>(
2.      stream: chatService.getChats(),
3.      builder: (context, AsyncSnapshot<dynamic> snapshot) {
4.          if (snapshot.hasError) {
5.              return RedError(snapshot.error);
6.          }
7.          if (!snapshot.hasData) {
8.              EasyLoading.show(
9.                  status: "Cargando chats...",
10.                 maskType: EasyLoadingMaskType.black);
11.             return Container();
12.         }
13.         List<String> chats = snapshot.data;
14.         return ChatList(chats);
15.     },
16. );

```

Ilustración 68, Tercera implementación de observador

Como aspectos positivos se destaca la facilidad de notificación de cambios al resto de componentes que dependen de estos y como aspecto negativo mencionar el control especial que requieren este tipo de suscripciones en Flutter pues, aunque la ventana que lo utilizase fuese destruida, la suscripción sigue activa y va a seguir consumiendo recursos y notificando, aunque ya no sea necesario.

4.5. Implementación de las API

En este apartado se explica en mayor profundidad el uso que se le han dado a las API de Google y cómo se han implementado en este proyecto.

API Keys

<input type="checkbox"/>	Name	Creation date ↓	Restrictions	Key		
<input type="checkbox"/>	✓ NomadWay	21.10.2020	8 APIs	AIzaSyCGDL...1cX2LtYCEU		

Ilustración 69, Api key Google Cloud

Para empezar a utilizar el servicio de Google Cloud, que proporciona las claves para poder realizar peticiones a sus servidores, se tuvo que registrar NomadWay bajo el plan gratuito que ofrece para usar en el límite de tres meses trescientos euros en peticiones, ya que este servicio cobra por cada cierta cantidad de llamadas a sus librerías.

Una vez registrados y añadida NomadWay a su sistema, se proporcionó una clave de uso global para todos sus servicios, teniendo que ser estos activados de forma individual, y que se introduciría en cada petición para comprobar su veracidad.

Dado que estas API están pensadas para ser utilizadas en otros lenguajes más comunes como Java, JavaScript o Kotlin²⁹, se tuvieron que utilizar adaptadores de estas librerías encontrados en Pub.Dev, página donde la comunidad o desarrolladores oficiales suben paquetes para adaptar funcionalidades externas a Dart.

4.5.1. Maps SDK de Android

La primera API que se utiliza, y la más básica de todas, es la que permite utilizar el mapa interactivo de Google en un dispositivo Android, para implementarla se utiliza un paquete llamado `google_maps_flutter`, desarrollado por el equipo oficial de Flutter, que adapta toda la API a código Dart [11].

Al activarla en el Cloud, se empezó a poder usar en la aplicación en forma de Widget, un fragmento de código en Flutter que contiene front y back end y es reutilizable en distintos lugares. A este se le añaden los parámetros básicos como la posición actual, que se obtienen de la siguiente

²⁹ <https://kotlinlang.org/>

API Geolocation o del propio GPS del móvil, una lista de marcadores que señalan los lugares en el mapa y método de inicio del que se obtiene su controlador, utilizado para controlar sus atributos y métodos de manipulación, como cambiar el tipo de mapa o cambiar el zoom o el ángulo de la cámara.

```
1. GoogleMap(  
2.   onMapCreated: _onMapCreated,  
3.   initialCameraPosition: _initialCameraPosition,  
4.   myLocationEnabled: true,  
5.   Marker s: _markers,  
6. ),
```

Ilustración 70, Widget Google Maps

Al controlador obtenido al realizar el método de creación se le actualiza el diseño del mapa que se quiere usar con un Json creado en la herramienta de modificación de diseño de mapas propio de Google, ya que el de por defecto de Google Maps viene con todas las localizaciones marcadas en este y se quiere que solo apareciesen los de nuestra base de datos.

```
1. void _onMapCreated(var controller) async {  
2.   var mapStyle = await rootBundle.loadString('assets/gMapStyle.json');  
3.   this.controller = controller;  
4.   this.controller.setMapStyle(mapStyle);  
5.   loadPlaces(); //método para cargar los marcadores  
6. }
```

Ilustración 71, Método al iniciar el mapa

Por último, se cargan los lugares de interés de la base de datos para ser mostrados en el mapa y se actualiza el estado del mapa para ser enseñados. Obteniendo el resultado visto en la Figura 72.


```

1. GeolocatorPlatform.instance.getPositionStream().listen((event) {
2.     updatedPosition = event;
3. });

```

Ilustración 74, Obtención de un Stream de posición

4.5.3. Geocoding

Esta es otra API con un uso muy simple pero es uno de los más importantes para que la aplicación sea global, es una librería que transforma la posición geográfica del usuario en una dirección de texto, pudiendo localizar fácilmente dónde se encuentra este de forma automática y con una gran precisión y flexibilidad, ofreciendo la dirección completa incluyendo calle, ciudad, provincia, comunidad y país además de la opción de cambiar el idioma en el que se muestra este resultado, pudiendo adaptar fácilmente los lugares de interés y las localizaciones mundiales al posible idioma seleccionado [13].

Para su implementación se decidió no utilizar un paquete externo pues el simple uso de lanzar una consulta y obtener una respuesta de servidor pareció que sería fácil de manejar con código propio, y así fue, con una petición Http³⁰ y un posterior análisis del Json³¹ devuelto por esta sería suficiente para la funcionalidad requerida:

```

1. Future<List<String>> fetchLocality(LatLng location) async {
2.     http.Response response = await http.get(Uri.parse(
3. 'https://maps.googleapis.com/maps/api/geocode/json?latlng=${location.latitude},${lo
4. cation.longitude}&result_type=locality&language=es&key=${globals.googleApiKey}'))
5.     .timeout(Duration(seconds: 5), onTimeout: error);
6.     if (response.statusCode == 200) {
7.         List<String> result = [];
8.         var json = jsonDecode(response.body);
9.         result.add(json['results'][0]['address_components'][0]['long_name']);
10.        result.add(json['results'][0]['address_components'][3]['long_name']);
11.        return result;
12.    } else {
13.        error();
14.    }

```

³⁰ Petición que se realiza por un cliente para iniciar una acción en un servidor.

³¹ <https://www.json.org/json-es.html>

En este método se observa como se lanza una petición a los servidores de Google con la información requerida, especificando un tiempo límite para devolver la respuesta que ejecute el método error en el caso de que se pase de este, y procesando la respuesta Json obtenida comprobando los campos que se necesitan y se genera una lista que sería la devuelta como retorno de la función.

4.5.4. Directions

La librería de direcciones es utilizada para ofrecer al usuario el camino a seguir durante la realización de una ruta o al dirigirse hacia un lugar de interés. Es una de las API más costosas en el servicio de mapas de Google y por ello se llevó una buena parte del presupuesto de prueba de Google Cloud [13].

Su funcionamiento consiste en la generación de puntos geométricos que, al unirse, crearían las líneas en el mapa que marcan el camino utilizando el callejero de Google Maps.

Para su implementación se utilizó la librería de Pub.Dev llamada `flutter_polyline_points`, que adapta el uso del servicio, utilizando dos posiciones en el mapa y la forma de viajar, y genera una lista de puntos utilizada por el mapa mencionado anteriormente.

Un ejemplo de su uso en nuestra aplicación es:

```
1. recorrido = await polylinePoints.getRouteBetweenCoordinates(  
2.     googleAPIKey,  
3.     PointLatLng(a.latitude, a.longitude),  
4.     PointLatLng(b.latitude, b.longitude),  
5.     travelMode: TravelMode.walking);
```

4.5.6. Places

La última librería utilizada está dividida en dos partes, la primera de obtención de lugares de interés, aspecto fundamental para nuestra aplicación tanto para hacerla global como para generar los sitios que visitar y la segunda que consiste en la obtención de las imágenes propias de dichos lugares [15].



Para la implementación de la primera parte de la API ha sido muy importante revisar bien la documentación que proporciona Google, pues informa de los campos que se devuelven en cada llamada y cómo se puede alterar la consulta para mostrar un tipo de información u otro como, por ejemplo, la obtención de lugares de interés de un lugar del mundo concreto, el tipo del lugar y el idioma y tipo en el que es generado el documento final.

```
1. https://maps.googleapis.com/maps/api/place/textsearch/json?query=place+of+interest+in+${globals.city}+,$${globals.country}&language=es&key=$apikey
```

Ilustración 77, Consulta de lugares de interés

El desarrollo en NomadWay ha sido complicado, para empezar, cada consulta solo devuelve veinte resultados, por lo que se tuvo que encadenar con la clave única de búsqueda que generaba, una sucesión de consultas hasta el máximo permitido de ochenta resultados,

Después, al usar una librería llamada `flutter_google_places`, que ofrecía una funcionalidad básica del servicio, tenía problemas para realizar consultas más complejas u obtener unos mejores resultados y, por lo tanto, se tuvo que descartar. También esta librería dio problemas de conexión con el servidor de Google, ya que devolvía los resultados de forma lenta y, a veces, impedía el resto de las búsquedas y bloqueaba la aplicación.

Por último, se decidió implementar una función que genera consultas genéricas que se pueden modificar para ser amoldadas a ciertas necesidades, pudiendo cambiar fácilmente entre idiomas, lugar, tipos y los propios lugares como estaciones de tren o museos.

Para ello se utilizó una cadena de texto parametrizada que sería la que después se introduciría en una petición Http contra los servicios de Google, después, la respuesta obtenida sería examinada para extraer la clave única de búsqueda si la hubiese y volver a lanzar la consulta para obtener los siguientes resultados y, por último, se procesarían los lugares devueltos para ser transformados en nuestro propio modelo de lugares de interés.

Este proceso se repetirá únicamente cuando un usuario acceda a una ciudad que no esté registrada en nuestra base de datos, ya que cada vez que se realiza el proceso se almacenan los resultados y no hace falta estar lanzando peticiones constantemente a los servidores de Google, reduciendo el tiempo de carga y el coste de uso de Google Cloud.

La segunda parte consiste en la obtención de las fotos de los lugares de interés, que son conseguidas mediante una referencia que es devuelta en la petición explicada antes y es utilizada

después en otra petición a la API de Photos, incluida en el servicio de Places, que devuelve un array de bytes que conforman las imágenes.

Como la primera parte, utilizar la librería de Pub.dev generó los mismos inconvenientes y también resultó en la creación de un propio adaptador al servicio proporcionado, aunque este servicio fue más simple de implementar pues solamente devuelve una lista con los arrays de bytes.

```
1. http.Response response = await http.get(Uri.parse(  
2. 'https://maps.googleapis.com/maps/api/place/photo?maxwidth=$width&maxheight=$height  
   &photoreference=$photoreference&key=$apikey'));  
3.   if (response.statusCode == 200) {  
4.     return Uint8List.fromList(response.body.codeUnits);  
5.   } else {  
6.     throw Exception("Failed to fetch Photo");  
7.   }
```

Ilustración 78, Método de obtención de imágenes

4.6. Extractor de países y ciudades del mundo

En este apartado se explicará cómo se obtuvieron cada lugar país del mundo y sus ciudades correspondientes mediante la utilización de Wikidata, una base de datos libre, gratuita y global que sirve de almacenamiento de objetos estructurados y que fue creada y es mantenida por el grupo Wikimedia, creadores también de otras páginas como Wikipedia.

Utilizar esta base de datos no es simple ni intuitivo, pues utiliza un lenguaje de consulta bastante distinto al habitual utilizado en otras bases de datos SQL llamado SPARQL, por lo que se tuvo un periodo de aprendizaje utilizando la documentación propia de la página, donde enseñaba el funcionamiento básico que se necesitasen para obtener la información que se quiera.

Este lenguaje funciona de una manera especial, pues se tienen que concatenar diferentes propiedades de los datos que conforman la base de datos para generar el resultado, la complicación principal fue encontrar el número de la propiedad que hace referencia a cada propiedad de la información que se buscaba.

```

1. select distinct ?item ?itemLabel ?itemDescription where {
2.     ?item wdt:P31 wd:Q5; # Any instance of a human.
3.         wdt:P19 wd:Q60; # Who was born in New York City.
4.
5.     SERVICE wikibase:label { bd:serviceParam wikibase:language "es" }
6. }
7. ORDER BY DESC(?itemLabel)

```

Ilustración 79, Ejemplo de uso SPARQL

Finalmente, optimizando la búsqueda para poder el superar el límite de tiempo, se consiguió desarrollar una gran consulta que obtendría todas las ciudades del mundo que sigan existiendo, pues muchas ciudades de la base de datos eran históricas o que habían cambiado de nombre y era un registro antiguo, que tuviesen una población mayor a diez mil habitantes, para descartar pequeños pueblos o registros de ciudades históricas que no solían tener población, que tuviesen una posición geométrica determinada, ya que hacía falta para geolocalizarlas en nuestro mapa, y que tuviesen traducción al español, puesto que NomadWay solo posee dicho idioma.

Con el resultado en Json que genera la consulta, se desarrolla un extractor de información externo al resto de la aplicación, formado por tablas de llave valor, que introducen una nueva llave o país en el caso de que no exista aún y añaden posteriormente el valor o ciudad para traducir los resultados a los modelos de datos y organizarlos mejor, obteniendo como resultado un Json ordenado alfabéticamente por países que a su vez contienen a las ciudades que forman parte de él.

```

1. for (var city in decodedJson['results']['cities']) {
2.     if (!countries.containsKey(city['countryLabel']['value'])) {
3.         countries[city['countryLabel']['value']] = [];
4.     }
5.     countries.update(city['countryLabel']['value'], (value) {
6.         String position = city['coordinates']['value'];
7.         position = position.substring(6, position.length - 1);
8.         value.add(City(city['cityLabel']['value'], position.split(" ")[1],
9.             position.split(" ")[0]));
10.         return value;
11.     });
12. }

```

Ilustración 80, Bucle extractor de Países/Ciudad

Este Json fue introducido en la aplicación en dos selectores, cargando primero el de países y, posteriormente cuando se haya seleccionado un país, carga el segundo con las ciudades que le pertenecen. Para finalizar, una vez estén las dos selecciones realizadas, carga un campo adicional que será la posición central de la ciudad, utilizado para posicionar correctamente la cámara cuando se seleccione la opción de cambiar de ciudad.

4.7. Desafíos técnicos

Algunos desafíos técnicos encontrados que han dificultado o han supuesto un reto durante el desarrollo del proyecto han sido aquellos que se van a exponer a continuación, además de los ya mencionados en los apartados anteriores.

4.7.1. Exceso de peticiones a Google

Durante los inicios del desarrollo, para obtener los lugares cercanos al usuario se hacía cada vez una petición al servicio de búsqueda por proximidad, teniendo que esperar para obtener y procesar los resultados, tanto como los lugares de interés como las imágenes asociadas, duplicando el tiempo de espera necesario, además de generar unos costes muy altos que lastraban nuestro presupuesto gratuito.

Para solucionarlo se decidió realizar una gran carga con lugares de interés en la base de datos utilizando el mismo servicio de proximidad, pero cambiando la posición de búsqueda muy poco a poco y guardar los resultados para luego mostrarlos según su proximidad al usuario, sin tener que utilizar el servicio de Google.

Esta solución ayudó bastante con la cantidad de peticiones a sus servicios, pues se redujo a cero en esa ciudad en concreto, pero los lugares obtenidos seguían siendo demasiado aleatorios ya que la librería que se utilizaba para adaptar el servicio de Google no filtraba correctamente por tipos y llenó la base de datos de tiendas o empresas privadas que no deberían de mostrarse debido a la idea de negocio implementada en la aplicación de enseñar únicamente lugares públicos de interés exceptuando a las propias empresas que se publicitaran.



El problema de la carga masiva correcta de la aplicación se corrigió implementando nuestro propio adaptador para utilizar el servicio de Google, lo que mejoró de forma notable la lógica interna de carga de datos, otorgando una mayor flexibilidad al parametrizar la búsqueda y obtener unos resultados más afines a nuestros requisitos con la desventaja de conseguir una menor cantidad de lugares de interés ya que el límite de la búsqueda era de ochenta sitios.

4.7.2. Lentitud de carga

Los lugares de interés e imágenes siempre han sido un lastre en los tiempos de carga del sistema, ya que al principio para obtenerlos se utilizaban peticiones encadenadas a servicios de terceros que en ocasiones tardaban en responder además del procesamiento extra que supone para adaptar los resultados a nuestra aplicación.

Con la carga de estos lugares en base de datos y luego extraerlos para ser mostrados se mejoró mucho el tiempo de carga de la aplicación, pero la descarga de imágenes del almacenamiento en la nube seguía siendo igual de lento que si de una petición al servicio se tratase.

La mejora final que se implementó fue la de la crear una base de datos local llamada Hive que almacena en forma de clave valor los lugares de interés y sus imágenes asociadas en el almacenamiento interno del dispositivo del usuario, saltándose las consultas a internet.

Con esto se consiguió una carga casi instantánea de los lugares de interés y fotos asociadas una vez hayan sido descargadas. Dado que esta mejora supone un coste extra de almacenamiento en un dispositivo de terceros, se da la opción de descargar la información o no, pudiendo eliminarla en un futuro si se deseease.

Finalmente, durante el inicio de sesión y la creación de una nueva cuenta de usuario, se está llevando a cabo una primera descarga de los lugares de interés e imágenes para que el tiempo de espera real una vez se esté dentro de la aplicación sea el menor.

4.7.3. Inicio de sesión

Para el inicio de sesión en redes sociales externos como Google, Facebook y Twitter, se tuvo que utilizar una librería externa de autenticaciones para cada servicio, dificultando en cada uno la implementación dado que cada una requería de una información distinta para poder utilizar una cuenta durante desarrollo y despliegue final.

Por ejemplo, para utilizar Google de inicio de sesión se necesita generar una clave específica, una para desarrollo por dispositivo y otra global para despliegue y producción, para cada sistema y añadirla a Firebase, además de volver a generar la configuración Json y actualizarla en nuestra aplicación para que detectase la nueva clave.

El caso de Facebook dio muchos más problemas, el más importante fue el cambio de nombre de NomadWay, ya que al llamarse previamente InstaGo, incumplía con los requisitos de nombres pues no lo aceptaban por similitudes a Instagram. Tampoco se consiguió generar una clave de despliegue y producción para Facebook y configurar correctamente el proyecto con esta, por lo que actualmente solo funciona en el modo de desarrollador, y únicamente un dispositivo.

Con estos problemas resueltos, se consiguió hacer funcionar el inicio de sesión, pero no conseguía mantener la sesión iniciada en el dispositivo, haciendo obligatorio volver a entrar cada vez que se salga de la aplicación, lo que supuso un incordio y una molestia para el desarrollo y despliegue.

Finalmente se creó en la base de datos local un atributo que guardase el token de inicio de sesión del usuario para poder entrar directamente en la aplicación, con la opción de poder salir de la cuenta y volver a iniciar con otra diferente.

5. Conclusiones

El desarrollo de este proyecto ha durado un total de nueve meses, incluyendo el periodo de implementación durante la asignatura de Pin, y ha evolucionado de forma constante todo este tiempo, desde el aspecto visual, que dio un giro completo a finales del segundo MVP y durante el tercero, hasta la idea de negocio, que empezó siendo freemium con publicidad hasta que acabó como un modelo de suscripción.

Siguiendo la metodología ágil de Scrum y Lean Startup, se ha evaluado la idea de negocio, realizado un estudio de mercado, creado un DAFO y Lean Canvas, y generado una proyección de ingresos y gastos, además de un mapa de características junto a la cronología y desarrollo de los tres MVP que componen esta memoria.

Se realizaron tres experimentos con early adopters para validar las características desarrolladas, de los cuales se han sacado mucha información e ideas para implementar en la aplicación y modificaciones importantes como la del modelo de negocio o añadir los inicios de sesión externos.

También se ha conseguido implementar una aplicación completa para móviles superando todos los desafíos y complicaciones que han salido durante todo el transcurso de este trabajo de fin de grado, desde el diseño de los modelos de los datos utilizados hasta la estructura del sistema de almacenamiento, incluyendo el estudio y búsqueda de la documentación necesaria para poder utilizar servicios de terceros externos a nuestro desarrollo e incluso crear nuestros propios adaptadores para poder utilizarlos de forma efectiva.

Se quiere destacar que se empezó sin tener conocimientos sobre ninguna herramienta que se ha acabado usando ni el lenguaje de programación, por lo que se ha tenido una curva de aprendizaje dura, además de ser un equipo de fullstack por lo que los dos integrantes debían saber implementar tanto funcionalidades de FrontEnd como de BackEnd, pero que ha terminado dando sus frutos y se ha conseguido crear una aplicación completamente funcional y un futuro esperanzador.

Por último, hay que indicar que este producto solo ha sido posible gracias al culmen de los conocimientos obtenidos durante la realización de este grado, desde las asignaturas del primer curso hasta las asignaturas de rama específicas para el desarrollo de software.

6. Trabajo Futuro

Aunque ahora mismo la aplicación se encuentre completamente funcional y se podría empezar a utilizar perfectamente para realizar visitas turísticas en otros países, faltan algunos detalles que la terminarían de completar:

- Encontrar empresas y negocios que quieran publicitarse en nuestra aplicación, añadiendo descuentos reales y localizaciones destacadas interesantes.
- Contratar colaboradores para crear una colección inicial de rutas por las ciudades más importantes de España y comenzar a ser usada por usuarios que quieran realizar turismo.
- Optimizar el guardado local de imágenes para almacenar las fotos con una menor calidad y ser mostradas de miniatura y de placeholder mientras se carga la foto real, disminuyendo así en gran medida el espacio necesitado.
- Mejorar la fluidez y velocidad en la aplicación, dado que se empezó a desarrollar sin tener conocimientos sobre la plataforma, no se sabían las consecuencias que tenía, por ejemplo, recargar completamente el árbol de Widgets, que consume muchos recursos y es el factor principal que lastra a Flutter. Para solucionar este problema se tendrían que extraer los componentes que se quieran actualizar de forma individual o crear zonas específicas de refresco.
- Implementar otro idioma como el inglés, que sería un importante añadido dado que nuestra aplicación tiene un enfoque global y es el idioma principal y secundario que se aprende en todos los países. Para ello se deberían de extraer todos los textos de la aplicación y crear etiquetas para ser referenciadas dependiendo del idioma local seleccionado.
- Definir mejor la estructura de la base de datos y el modelo de datos ya que actualmente hay campos que se podrían eliminar o secciones y colecciones internas que se podrían sustituir por referencias y ahorrar espacio en Firebase.
- Contactar con una empresa que gestione las auditorías de protección de datos, ya que es un aspecto legal fundamental para todas las empresas y dado que la nuestra guarda información personal de sus usuarios, más aún.
- Aprender sobre seguridad informática para aplicarla a nuestra aplicación, ya que no se tiene ningún conocimiento sobre este tema y también es necesario en algunos requisitos de las leyes de protección de datos.



7. Referencias

- [1] Alberto Puerta Millet. FrontEnd NomadWay: una aplicación para la gestión de rutas turísticas.
- [2] Documentación de Flutter. Consultado en octubre de 2020, en: <https://flutter.dev/docs>.
- [3] Documentación de Dart. Consultado en octubre de 2020, en: <https://dart.dev/guides>.
- [4] Documentación de Dart Dev Tools. Consultado en noviembre de 2020, en: <https://flutter.dev/docs/development/tools/devtools/overview>.
- [5] Documentación de Google Cloud. Consultado en octubre de 2020, en: <https://cloud.google.com/docs>.
- [6] Documentación de Git. Consultado en octubre de 2020, en <https://git-scm.com/doc>.
- [7] Documentación de Firebase. Consultado en octubre de 2020, en <https://firebase.google.com/docs>.
- [8] Documentación de Visual Studio Code. Consultado en octubre de 2020, en <https://code.visualstudio.com/docs>.
- [9] Documentación de Android Studio. Consultado en octubre de 2020, en <https://developer.android.com/studio/intro>.
- [10] Documentación de Hive. Consultado en mayo de 2020, en <https://docs.hivedb.dev/#/>.
- [11] Documentación de Maps SDK for Android. Consultado en noviembre de 2020, en: <https://developers.google.com/maps/documentation/android-sdk/overview?hl=es-419> .
- [12] Documentación de Geolocation. Consultado en noviembre de 2020, en: <https://developers.google.com/maps/documentation/geolocation/overview>.
- [13] Documentación de Geocoding. Consultado en mayo de 2020, en: <https://developers.google.com/maps/documentation/geocoding/overview>.
- [14] Documentación de Directions. Consultado en noviembre de 2020, en: <https://developers.google.com/maps/documentation/directions/overview>.
- [15] Documentación de Places. Consultado en noviembre de 2020, en: <https://developers.google.com/maps/documentation/directions/overview>.

Anexo 1: Protección de datos

Este apartado del trabajo de fin de grado es común para Nicolás García Riego y Alberto Puerta Millet [1].

Las empresas que traten con datos personales de sus clientes o de sus empleados, deben de cumplir con la Ley de Protección de Datos Personales. Dado que estas gestiones se realizan de forma informática, se suele coordinar la seguridad con la protección de datos.

La normativa de la LOPD-GDD (Ley Orgánica de Protección de Datos y Garantías de Derechos Digitales), que entró en vigor en 2018, adapta el RGPD (Reglamento General de Protección de Datos), ley de ámbito europeo que se creó en 2016, pero que no entró al sistema jurídico español hasta la creación de la anterior mencionada.

Muchas de las normativas que se incluyen en estas leyes conllevan una mayor exigencia en la seguridad informática, como las que afectan al cloud computing, que también se utilizan en nuestra aplicación. Además, también se deberá tener en cuenta la Ley de Servicios de la Sociedad de la Información y de comercio Electrónico o LSSI-CE. Si se garantiza el cumplimiento de la ley española, se estará cumpliendo también la de ámbito europeo.

Dada la inexperiencia para la implementación de dichas garantías de seguridad y de cumplimiento de leyes, se ha decidido contactar con una empresa externa para que lo coordine y gestione.

La empresa seleccionada para la gestión de la protección de datos ha sido Atico34, que incluye una explicación de lo que supone cumplir las leyes mencionadas y de los pasos a seguir para cumplir con las normativas europeas, además de una breve explicación en cada apartado para entender por qué y cómo se deben realizar. También se ha elegido esta empresa por sus años de experiencia en el comercio electrónico y consultorías de protección de datos.

Ahora se explican los pasos que se deben de llevar a cabo para implementar la ley de protección de datos.

1. Registro de actividades de tratamiento

Este documento recoge y clasifica el tipo y la cantidad de datos que manejas en tu negocio, así como otros aspectos sobre el tratamiento de datos personales. Debe estar actualizado, ya que, en caso de inspección de la AEPD (Agencia Española de Protección de Datos), es uno de los documentos necesarios. Debe incluir la siguiente información:

- Identificación y datos de contacto del responsable y, en su caso, del encargado del tratamiento y del delegado de protección de datos.
- Fines del tratamiento.
- Categorías de interesados y datos.
- Categorías de destinatarios existentes o previstos (incluidos otros países u organizaciones internacionales).
- Transferencias internacionales (si las hay).
- Medidas de seguridad.
- Plazos previstos para la supresión de datos.

2. Análisis de riesgo y evaluación de impacto

También se debe realizar un análisis de los riesgos de las diferentes actividades de tratamiento de datos que se lleven a cabo en el día a día. Este análisis debe tener en cuenta, dónde se almacenan los datos, por cuánto tiempo, la naturaleza de los datos o el número de interesados afectados.

En ciertos casos, debido al riesgo para los derechos y libertades de los interesados, este análisis previo deberá transformarse en una Evaluación de Impacto en la Protección de Datos Personales (EIPD).

Ambos estudios sirven para implementar medidas de seguridad que minimicen o prevengan esos riesgos y amenazas.

3. Notificación de brechas de seguridad

Si se producen brechas de seguridad que pongan en riesgo los datos personales de los interesados y sus derechos y libertades, es obligatorio informar a la AEPD en un plazo máximo de 72 horas. También se debe notificar de estos incidentes de seguridad a los interesados cuyos datos personales hayan podido verse expuestos.

4. Consentimiento

Es necesario y obligatorio recabar el consentimiento de los interesados tanto para recabar sus datos personales como para tratarlos. Este consentimiento debe darse mediante una acción afirmativa.

5. Cláusulas de información

Hay que dar más información al cliente. Le debe quedar claro quién trata sus datos, cómo los trata y por qué los trata. También habrá que incluir nuevas cláusulas de elección del proveedor que realiza un servicio, como la empresa informática o la gestoría que elabora las cuentas.

6. DPO

Solo es necesario nombrar un DPO (Delegado de Protección de Datos) en caso de que tu servicio trate datos a gran escala. Puede ser tanto un empleado interno como un profesional externo. Pero, debe contar con los conocimientos necesarios sobre protección de datos para desempeñar sus funciones.

7. Código de conducta y certificados

Es posible adherirse a códigos de conducta y certificados que promuevan desde el sector de la informática el cumplimiento de la normativa, evaluando de manera externa y objetiva nuestro nivel de cumplimiento normativo en materia de protección de datos.



8. Información a los propietarios de los datos

El consentimiento debe facilitar a los interesados información relevante sobre tratamiento de datos que se va a hacer:

- Responsable del tratamiento
- Finalidad del tratamiento
- Plazo de conservación de los datos
- Cesión a terceros
- Dónde y cómo ejercer sus derechos

9. Plazo de conservación de los datos

La RGPD exige informar sobre los plazos de conservación de los datos de los interesados, pero ninguna ley específica un plazo concreto, cada empresa debe ser la responsable de especificarlo teniendo en cuenta que el plazo debe de ser el mínimo necesario para obtener los resultados esperados de ellos.

A sí mismo, en el caso de que otra ley indique ciertos plazos de conservación, se deberán de tener en cuenta y ser aplicados, por ejemplo, las facturas se deben almacenar un mínimo de cinco años.

10. Auditorías periódicas

Ninguna ley indica explícitamente la obligación de realizar auditorías periódicas, pero dado que las empresas tienen que demostrar constantemente las herramientas y soluciones para el debido cumplimiento de dichas leyes y asegurar la protección de sus usuarios, se recomienda someter a la empresa a estas evaluaciones periódicas con empresas externas.