



**Universidad**  
Zaragoza

## **Trabajo Fin de Grado**

**Aplicación multiplataforma para la distribución  
del catálogo de productos y servicios de la  
empresa Oleazara**

Grado en Ingeniería Informática

**AUTOR: Carla Pascual Real**

**TUTOR: Joaquín Ezpeleta**

**Dpto. de Informática e Ingeniería de Sistemas  
Escuela de Ingeniería y Arquitectura (EINA)  
Universidad de Zaragoza**

**FEBRERO 2021**

# ÍNDICE

1. Introducción.....	1
1.1. Objetivo del proyecto.....	1
1.2. Fases de realización.....	2
2. Análisis.....	2
2.1. Diagrama de caso de uso.....	2
2.2. Requisitos.....	4
2.2.1. Requisitos funcionales.....	4
2.2.2. Requisitos no funcionales.....	4
2.3. Metodología de desarrollo.....	5
2.4. Base de datos.....	6
2.5. Aplicaciones similares.....	7
2.6. Framework a utilizar.....	10
3. Diseño.....	11
3.1. Arquitectura.....	11
3.2. Elección tecnologías y servicios.....	13
3.3. Mockups software.....	13
4. Implementación.....	14
4.1. Arquitectura hardware.....	14
4.2. Arquitectura software.....	14
4.3. Herramientas y tecnologías.....	14
4.4. Aspecto final de la aplicación.....	15
5. Pruebas.....	16
6. Conclusiones.....	21
6.1. Dificultades encontradas.....	21
6.2. Aportaciones realizadas.....	21
Anexo I.....	22
Anexo II.....	23
Anexo III.....	25
Anexo IV.....	27
Bibliografía.....	33

# 1- INTRODUCCIÓN

En la actualidad, las aplicaciones móviles son una manera muy efectiva para las empresas de llegar a sus clientes, especialmente en el campo del e-commerce. El motivo es que es un método muy sencillo debido al acceso rápido y cómodo mediante un dispositivo móvil, sin necesidad de desplazamiento y desde cualquier ubicación.

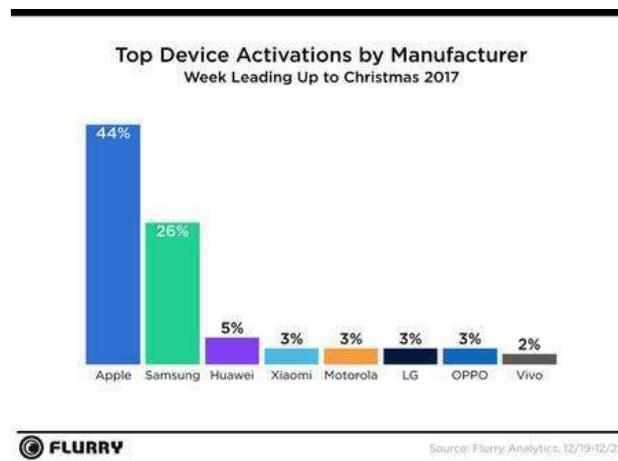
La necesidad de una empresa de desarrollar una aplicación móvil se debe al requerimiento de una estrategia para su crecimiento, además del propósito de mantener su economía [1].

Debido a esto, la empresa Oleazara ha querido hacer crecer su empresa mediante la solicitud de desarrollo de una aplicación móvil de esta, la cual contendrá multitud de funcionalidades.

## 1.1- OBJETIVO DEL PROYECTO

El objetivo principal del proyecto es desarrollar una aplicación para la empresa Oleazara, que sea accesible tanto para iOS como para Android, con la finalidad de visualizar el catálogo de los productos de la empresa y la información general de esta, además de dar la posibilidad de suscripción a su newsletter.

Flurry Analytics realizó un rastreo de 2.100 millones de aplicaciones entre todas las tiendas online [2], obteniendo que el 44% se localizan en dispositivos Apple, siendo el resto Android (Figura 1), por lo que sería muy útil que la aplicación funcionase en ambos sistemas operativos.



**Figura 1:** Estudio de aplicaciones móviles de Flurry Analytics

La aplicación móvil consistirá en un catálogo de los productos de la empresa de Aragón denominada *Oleazara*, entre los cuales se encuentran aceites de oliva virgen extra del Bajo Aragón, mieles, aceitunas o patés, cuyo negocio no tiene aplicación móvil y requiere de una. Además de ser catálogo, se podrán realizar en ella suscripciones a la newsletter de la empresa o acceder a información de *Oleazara* que sea de interés, como política de privacidad o modo de producción, entre otras, además de poder buscar productos concretos de los que se quiera recibir información mediante un buscador.

## 1.2- FASES DE REALIZACIÓN

Los pasos seguidos para la realización del proyecto son cinco, comunes con la mayoría de desarrollos software, y son los siguientes:

- **Estudio:** Investigación sobre todas las tecnologías posibles a utilizar para encontrar la óptima, además de buscar aplicaciones similares para encontrar elementos necesarios para el proyecto.
- **Análisis:** Establecimiento de los requisitos funcionales y no funcionales que debe cumplir la aplicación móvil.
- **Diseño:** Se procede a establecer la arquitectura de la aplicación y las tecnologías a utilizar, además de diseñar las pantallas principales necesarias junto con el flujo entre ellas para mostrarlas al dueño del negocio y posteriormente continuar con el resto de pantallas y sus flujos.
- **Implementación:** Se irá programando la aplicación enseñando al cliente los avances hasta conseguir la aplicación deseada final.
- **Pruebas:** Se realizan una serie de pruebas para encontrar y solucionar problemas técnicos y se le hace demostración de la aplicación al cliente para tener su *feedback*.

Una vez terminado el desarrollo de la aplicación, se procederá a su publicación y a una fase de mantenimiento constante.

- **Publicación:** Es la fase final. Se configura toda la aplicación para lanzarla.
- **Mantenimiento:** El mantenimiento es necesario para controlar periódicamente el funcionamiento de la aplicación, puesto que se van lanzando nuevos smartphones y puede provocar una pérdida de compatibilidad. Dado que se trata de un TFG, la fase de mantenimiento será posterior a su realización, por lo que no se refleja en este documento.

# 2- ANÁLISIS

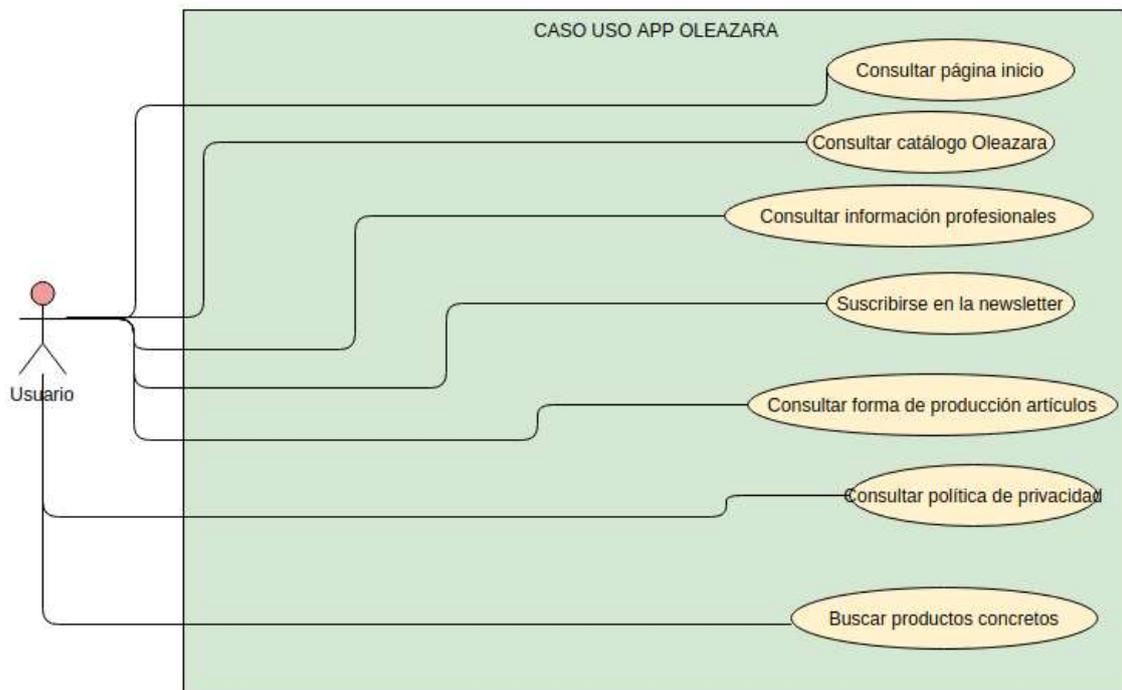
Esta es la primera fase para el desarrollo de una aplicación móvil. Esta fase consistirá en el diagrama de caso de uso de la aplicación y el establecimiento de los requisitos necesarios que se necesitan para este proceso, junto con la elección de una metodología adecuada para su desarrollo o el estudio de aplicaciones con funciones similares a las que pide Oleazara.

## 2.1- DIAGRAMA DE CASO DE USO

Los diagramas de caso de uso son representaciones gráficas de las interacciones de cada tipo de rol del diagrama, además de las interconexiones entre sus diferentes escenarios. Hay tres elementos principales en los diagramas de caso de uso [3]:

- **Actor:** Se representa como una figura humana esquemática.
- **Sistema:** El sistema al que hace referencia el caso de uso se representa como un rectángulo.
- **Caso de uso:** Se representa como una elipse.

El diagrama del caso de uso de la aplicación *Oleazara* sería el siguiente:



**Figura 2:** Diagrama caso de uso Oleazara.

En la representación anterior se puede observar al usuario junto con una serie de casos de uso posibles. Dichos casos de uso son los que se muestran a continuación:

- **Consultar página inicio:** La aplicación debe disponer de una página de inicio que aparezca una vez se inicialice la aplicación, y que esta cuente con información sobre la empresa que sea de interés.
- **Consultar catálogo Oleazara:** Se deberá poder acceder a un catálogo de todos los productos de la empresa, en el cual se podrá observar el precio, imagen, título y descripción de cada uno de los productos.
- **Consultar información de los profesionales:** La aplicación también deberá de poseer un acceso a una interfaz en la cual se muestre la información sobre los profesionales que forman la empresa.
- **Suscribirse a la newsletter:** Se deberá poder realizar una suscripción a la newsletter de la empresa desde la aplicación, con la finalidad de que el usuario pueda recibir información sobre las novedades de los productos de Oleazara.
- **Consultar forma de producción de los artículos:** La aplicación debe disponer también de una interfaz en la que se transmita la manera en que se elaboran los productos de Oleazara.
- **Consultar la política de privacidad:** Se ha pedido también que dicha aplicación permita acceder a la política de privacidad de Oleazara.

- **Buscar productos concretos:** Se va a disponer de la posibilidad de realizar una búsqueda del producto o productos que se requieran, con el fin de facilitar las necesidades del usuario.

## 2.2- REQUISITOS

Los requisitos en un proyecto software definen las características y las funciones del sistema 'target' [4]. Hay requisitos tanto funcionales como no funcionales. Representan en resumen las expectativas que posee el desarrollador con la aplicación a tratar de manera descriptiva.

### 2.2.1- REQUISITOS FUNCIONALES

Los requisitos funcionales son aquellos basados en la interacción del usuario con el software.

#### 2.2.1.1 - APLICACIÓN

##### **RF1: Pantalla principal**

1. La pantalla principal almacenará un resumen sobre todo lo que se puede realizar al utilizar la aplicación.
2. La pantalla principal contendrá también enlaces a las redes sociales (Instagram, Youtube y Twitter), y un acceso al menú. También dispondrá de una barra superior con un botón de búsqueda.

##### **RF2: Catálogo**

1. El catálogo mostrará la información de cada producto accediendo a la base de datos de la empresa. Se mostrará por producto el título, descripción, imagen y precio.

##### **RF3: Información general**

1. Se dispondrá de varios accesos a información general sobre la empresa, como política de privacidad, modo de producción o información sobre los profesionales.

##### **RF4: Buscador**

1. Se localizará un buscador que facilite la búsqueda de productos concretos sin necesidad de acceder a todo el catálogo.

##### **RF5: Suscripción:**

1. Existirá la posibilidad de suscripción a la newsletter de Oleazara con validación del email mediante expresión regular.

### 2.2.2- REQUISITOS NO FUNCIONALES

Los requisitos no funcionales son los requisitos [5] que especifican criterios que se pueden utilizar para juzgar las operaciones en vez de comportamientos específicos.

##### **RNF1: Interfaz y usabilidad**

1. La aplicación debe ser fácil de descargar e instalar.
2. La aplicación tendrá una interfaz intuitiva y sencilla para entenderla sin complicación, de tal manera que no requiramos de esfuerzo para su manejo.

3. El tiempo de aprendizaje de la aplicación será menos de media hora.
4. La aplicación debe ser fácil de analizar y corregir sus posibles fallos.

#### **RNF2: Mantenibilidad y portabilidad**

1. Disponibilidad para cualquier tipo de dispositivo Android o iOS y para la mayoría de versiones.
2. Será necesario la conexión a Internet para el acceso a la base de datos, utilizada para visualizar los productos de la empresa.
3. La aplicación tendrá que funcionar en recursos de potencia media.

#### **RNF3: Eficiencia y rendimiento**

1. El funcionamiento de la aplicación será ágil y se obtendrán tiempos de respuesta cortos (en torno a un segundo).

#### **RNF4: Seguridad**

1. La aplicación tendrá que ser segura.

## 2.3- METODOLOGÍA DE DESARROLLO

La metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información [6]. Hay varios tipos de metodologías, dependiendo del tipo de desarrollo que se vaya a llevar a cabo. Para este tipo de aplicación, se ha optado por la **metodología ágil Extreme Programming**.

### EXTREME PROGRAMMING

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos [7].

**Extreme Programming** se encarga de potenciar las relaciones del equipo de desarrollo como clave del éxito mediante el aprendizaje continuo, el trabajo en equipo y el buen clima de trabajo.

En esta metodología destaca la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes.

Sus características son las siguientes:

- Se considera al equipo de proyecto como el principal factor de éxito del proyecto.
- Software que funciona por encima de una buena documentación.
- Interacción constante entre el cliente y el equipo de desarrollo.
- Planificación flexible y abierta.
- Rápida respuesta a cambios.

Se ha escogido este tipo de metodología ya que se va a tener una comunicación habitual con el cliente, puesto que son personas con estrechos lazos con el programador. Esto se ha realizado de la siguiente manera: el cliente comunicó inicialmente sus objetivos con la aplicación, y el programador cada vez que realiza un objetivo del cliente, se lo muestra para obtener alguna petición de cambio o la aceptación.

Otra característica de este tipo de metodología es un equipo con muy pocos programadores, que es el caso de este proyecto, puesto que solo se cuenta con un programador.

La planificación será flexible y abierta, ya que solo va a haber un único programador que se encargue de todo, por lo que va a ser fácil si en un momento dado hay que aplicar cualquier cambio en el proyecto.

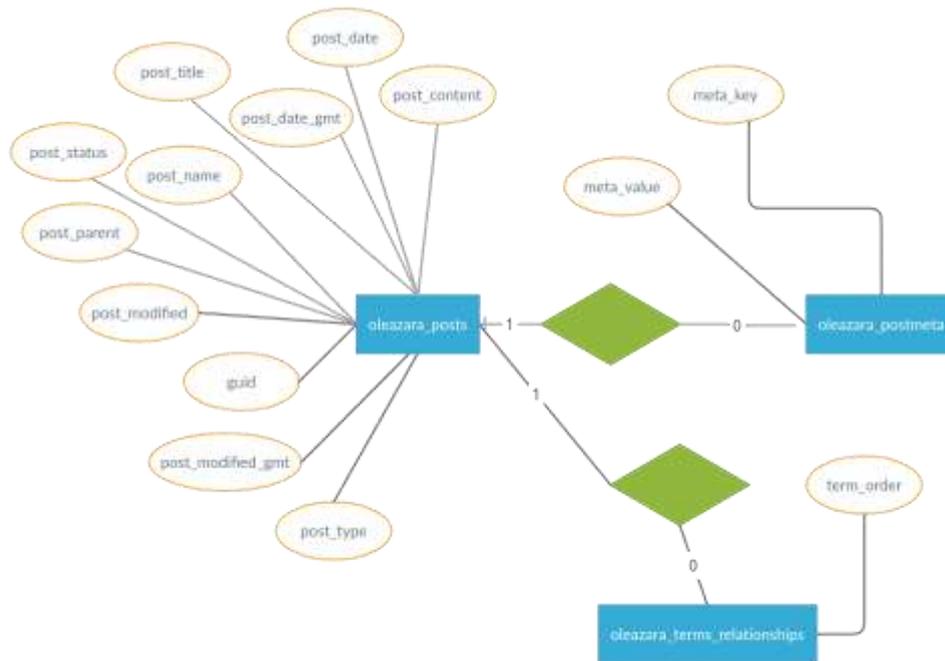
En el **Anexo I** se muestra la manera en que se ha aplicado esta metodología al desarrollo de la aplicación.

## 2.4 - BASE DE DATOS

Para la utilización de una base de datos que almacene los productos que se van a mostrar en el catálogo de la aplicación, se dispone de dos posibilidades.

La primera de ellas sería construir una nueva base de datos propia para la aplicación. Para ello, se requeriría crear una colección en esta introduciendo todos los productos existentes de la empresa. Esta idea daría lugar a un inconveniente, debido a que en el momento que la empresa actualice su inventario (añadiendo, eliminando o modificando algún producto), se necesitaría conocer dichos cambios, para proceder a modificarlos en la base de datos de la aplicación, lo cual supondría un esfuerzo innecesario.

Debido a esto, la idea escogida sería acceder desde la aplicación a la misma base de datos que dispone actualmente la empresa, con lo que se sincronizaría la aplicación con los productos de la web existente cada vez que se abre el catálogo. En la siguiente figura se muestra el esquema Entidad-Relación obtenido de la base de datos existente. Se incluyen en el diagrama únicamente las tablas utilizadas para las consultas, de las más de 100 que configuran la base de datos:



**Figura 3:** Esquema E/R Oleazara.

La tabla *oleazara\_postmeta* tiene un campo llamado “*post\_id*”, el cual representa el id del producto (clave ajena de la tabla *oleazara\_posts*). Esta tabla almacena información variada (atributo *meta\_value*), entre la cual nos interesa la descripción, la imagen del producto o el precio.

*Oleazara\_terms\_relationships* tiene un campo llamado “*term\_taxonomy\_id*”, el cual representa la categoría de los productos: 15 si se trata de aceites, 72 para aceitunas, 124 para las mermeladas y 70 para los patés. Para relacionarlo con la tabla de productos, se almacena una clave ajena llamada “*object\_id*”.

De la tabla *oleazara\_post* interesa el título del producto, y es el punto de conexión entre las tablas anteriores.

Las consultas consisten en obtener los identificadores de los productos según la categoría establecida desde la tabla *oleazara\_term\_relationships*, y con los ids obtenidos, obtener la información para cada uno de los productos.

## 2.5- APLICACIONES SIMILARES

Para continuar con la fase de análisis, se ha realizado una búsqueda exhaustiva de aplicaciones que realicen funciones similares a las requeridas, con la finalidad de ver qué pueden o no pueden hacer. Dichas aplicaciones han sido: *Zara*, *Stradivarius*, *Shein* y *Privalia*, las cuales son aplicaciones móviles de tiendas, disponibles tanto para iOS como Android.

- **Zara:** Esta aplicación (figura 3) procede a la tienda de ropa “Zara” de Inditex. En ella se ofrece la posibilidad de ver todos los productos de la tienda, además de poder comprar de manera online y de otros muchos beneficios. Tiene los siguientes apartados:

1. Pantalla principal donde se ven las novedades de mujer, hombre o niño.
2. Menú en el que están las categorías de mujer, hombre o niño, habiendo a su vez subcategorías en cada uno de estos apartados.
3. Opción de “Mi lista”, donde se pueden guardar las prendas favoritas.
4. Opción de búsqueda de prendas.
5. Icono de “Perfil”, donde se encuentra un inicio de sesión, o si ya se ha iniciado sesión se dispondrá de un menú con los apartados: “Mis compras”, “Solicitudes de devolución”, “Wallet”, “Ayuda”, “Ajustes” y “Newsletter”, además de la posibilidad de cerrar sesión, de compartir la aplicación o de puntuar esta.
6. El icono de “Cesta de compra” solo aparece una vez hayas añadido al carrito algún producto.

- **Stradivarius:** La tienda de ropa “Stradivarius”, también de Inditex, tiene una aplicación similar (figura 4), ya que ambas son de la misma empresa. En ella puedes navegar por los productos de la marca “Stradivarius” y comprar lo que se desee. Tiene las siguientes interfaces:

1. Pantalla principal donde puedes acceder a las novedades de la tienda.
2. Menú desplegable en el que se muestran las múltiples categorías de la tienda para dar la posibilidad de ver sus productos. Además, también se encuentran las opciones de suscripción a la newsletter, búsqueda, política de privacidad o términos y condiciones, entre otras.
3. Carrito. En esta interfaz se observan los productos añadidos al carrito y se ofrece la posibilidad de realizar la compra.
4. Lista de favoritos. En Stradivarius también se dispone de una interfaz donde se almacenan los productos favoritos del usuario.
5. Opción de perfil. Ofrece la posibilidad de iniciar sesión, crear cuenta, o si se ha iniciado sesión ver opción de “Mis compras”, “Mis datos personales”, o “Configuración”, además de otras.



**Figura 3:** Aplicación Zara



**Figura 4:** Aplicación Stradivarius

- **Shein:** Shein es una plataforma internacional de comercio online de tipo B2C (negocio a consumidor). En esta (figura 5) se venden todo tipo de artículos de ropa y de complementos. Destaca por los clientes por tener un precio muy reducido y ser de calidad media. Esta aplicación ofrece las siguientes ventajas:
  1. Pantalla de inicio donde se muestran anuncios de la aplicación, además de una ventana emergente de suscripción de la newsletter.
  2. Opción de agregar productos a favoritos para disponer de ellos siempre que se requiera.
  3. Botón de búsqueda.
  4. Apartado de “Bolsa”. En este se almacena el carrito del usuario.
  5. Botón de “Categorías”. Clicando sobre este, aparece una pantalla con los productos divididos por las distintas categorías (como en la figura 6).
  6. Botón de “Nuevo”. En esta opción se permite visualizar los productos novedosos de la tienda.
  7. Inicio sesión/registro. En el apartado de Perfil, tras iniciar sesión, se dispone de los cupones de descuento del cliente junto con sus puntos, los pedidos realizados, o servicio de atención al cliente.
  
- **Privalia:** Esta aplicación (figura 6) no dispone de productos propios, sino que se basa en vender productos procedentes de distintos tipos de marcas con descuentos que duran unos pocos días. Tiene un mercado más amplio, puesto que sus productos no son solo ropa, sino que cualquier marca que quiera puede aparecer en esta aplicación. Tiene las siguientes funcionalidades:
  1. Pantalla inicial donde aparecen las marcas disponibles ordenadas por novedad, pudiendo filtrarlas también según las categorías siguientes: “*Moda & Beauty*”, “*Infantil*”, “*Hogar & Tech*”, y “*Viajes*”.
  2. Botón de “Catálogo”, en el que clicando se dispone de una lista de múltiples categorías, pero únicamente categorías textiles.
  3. Apartado de mi cuenta, donde se encuentra el carrito, el inicio de sesión en el caso de que no se hubiera realizado, o información del perfil en el caso de que sí se hubiera accedido, como “*Mis pedidos*”, “*Mis datos*”, “*Ayuda*” o “*Información legal*”, entre otras, además del cierre de sesión.



**Figura 5:** Aplicación Shein



**Figura 6:** Aplicación Privalia

Tras analizar las aplicaciones anteriores, se ha observado que la mayoría de ellas dividen sus productos por categorías, además de la existencia de un sistema de búsqueda, con el objetivo de facilitar al cliente acceder a los productos deseados. Además, también se detecta que suele ser común ofrecer la posibilidad de suscripción a la newsletter, ya que con esto se consigue mostrar a los clientes fieles los productos nuevos que se disponen y así tentarles de seguir comprando artículos.

Otra idea que se ha visto en todas las aplicaciones mostradas, es el inicio de sesión, el cual sirve para almacenar tus datos personales (como tarjeta de crédito), para facilitar las siguientes compras, además de acceder a tus pedidos anteriores o servicio al cliente. Esto no va a ser útil, puesto que la aplicación a desarrollar no cuenta con servicio de compra, sino que es meramente informativa, por lo que se descarta este aspecto ya que es innecesario.

## 2.6- FRAMEWORK A UTILIZAR

Se necesita un framework que permita la posibilidad de desarrollar una aplicación móvil que sea accesible tanto desde iOS como desde el sistema Android.

Hay múltiples framework que permiten esta idea. Alguno de ellos son los siguientes [8]:

- **Flutter:** Flutter es un framework de código abierto desarrollado por Google para crear aplicaciones nativas de forma fácil, rápida y sencilla. Su principal ventaja radica en que genera código 100% nativo para cada plataforma, con lo que el rendimiento y la UX es totalmente idéntico a las aplicaciones nativas tradicionales.
- **Ionic:** Ionic es bastante popular entre los framework de desarrollo de aplicaciones, probablemente el que más. Ionic es básicamente una red front-end que ayuda a construir aplicaciones móviles nativas con HTML, CSS3 y Javascript. La ventaja significativa de Ionic es que se pueden utilizar cientos de elementos de interfaz de usuario predeterminados como formularios, filtros, hojas de acción, vistas de lista, barras de pestañas y menú de navegación en su diseño.
- **React Native:** React Native integra los beneficios de JavaScript y ReactJS proporciona a los desarrolladores la posibilidad de escribir módulos en lenguajes Objective-C, Swift o Java. Otro gran detalle es que proporciona animaciones extremadamente suaves ya que el código se convierte en vistas nativas antes de ser renderizado.
- **Xamarin:** A través de su API te permite generar apps estéticamente similares a las nativas y permite incluir las bibliotecas de Objective-C, Java y C++ sin ningún problema y presenta una adaptación total al editor Visual Studio de la propia Microsoft.

Después del análisis de estas, se ha escogido Flutter como *framework* a utilizar, puesto que se considera que ofrece el mejor rendimiento de todos los *frameworks* explicados ya que este es idéntico a las aplicaciones nativas tradicionales, al igual que la experiencia del usuario, lo que produce la preferencia de este *framework* respecto al resto de los explicados.

# 3- DISEÑO

En este apartado se va a describir todo el proceso de diseño de la aplicación Oleazara, basada en los requisitos explicados en el tema de análisis.

## 3.1- ARQUITECTURA

La arquitectura escogida se basa en el modelo cliente-servidor. Hay tres tipos de comunicación cliente-servidor en el diseño de la aplicación.

Uno de ellos se basa en el proceso de acceso a la base de datos para obtener el catálogo (figura 7a). Este tipo de modelo utiliza un servidor con el que se accederá a la base de datos mediante HTTP, siendo la aplicación el cliente que ejecuta la petición. Dicho acceso se realiza mediante una petición GET con HTTP a un fichero PHP que ejecuta todo el acceso a la base de datos de la empresa, devolviendo un JSON con la información pedida. Esta petición se establece en el lenguaje DART del framework Flutter de la siguiente manera:

```
var url = 'https://oleazaradatabase.000webhostapp.com/aceites.php';  
http.Response response = await http.get(url);
```

Hay cuatro ficheros a los que se hace petición para obtener información de la base de datos: para obtener el JSON del catálogo de aceites, del catálogo de mermeladas, del catálogo de patés, o del catálogo de aceitunas.

Estos ficheros que realizan las consultas están localizados en un servidor para poder acceder desde cualquier ubicación. Para ello, se ha creado un usuario de Oleazara en el web hosting llamado **000WebHost**, y en el cual se han almacenado los cuatro ficheros PHP.

Estas peticiones se van a realizar una vez que el cliente acceda en el menú al catálogo de aceites, mermeladas, patés o aceitunas respectivamente, pintando el JSON en la interfaz resultante.

También se realizará el mismo modelo en el momento en que el cliente (la aplicación) manda en una petición GET mediante HTTP el email introducido por el usuario en el formulario de suscripción (figura 7b). Esta petición se enviará a una función de la plataforma **Firestore**, creada mediante un fichero JavaScript. Para crear funciones en Firestore, es necesario registrarse con un usuario y contraseña.

Esta función recibirá el correo electrónico introducido por el usuario, lo añadirá a la lista de suscriptores de Mailchimp de la empresa, y devolverá un "ok". Se ejecuta de la siguiente forma:

```
var url = "https://us-central1-oleazaraapp.cloudfunctions.net/mailChimp?email="+this.email.text;  
http.Response response = await http.get(url);
```

Para acceder a la cuenta de Mailchimp de Oleazara, se han requerido de sus credenciales (usuario y contraseña), para que la adición se establezca en su lista de suscriptores, y no en una independiente, lo que es más eficiente, ya que de esta manera solo existe una única lista.

Por último, se realiza una consulta a la base de datos de Oleazara para buscar los productos concretos que se filtran en el apartado de búsqueda (figura 7c), para cuya ejecución no se ha

empleado un fichero exterior. Esto se consigue debido a la biblioteca *mysql1* de Dart, que nos permite hacer la consulta de manera muy sencilla:

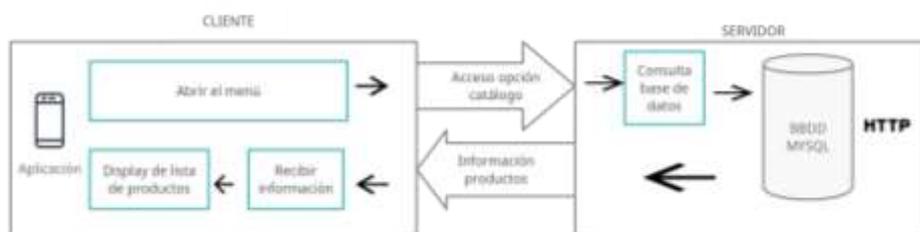
```

HtmlEscape html= const HtmlEscape();
query=html.convert(query);
query=query.replaceAll('%', '\\%');
var conn = await MySqlConnection.connect(settings);
var datos=await conn.query('SELECT item.post_title AS titulo, img2.meta value AS imagen, descr.meta value AS descripcion,
prec.meta value AS precio FROM oleazara_posts item JOIN oleazara_postmeta img JOIN oleazara_postmeta img2 JOIN oleazara_postmeta descr
JOIN oleazara_postmeta prec JOIN oleazara_term_relationships r WHERE ((item.post title LIKE '%$query%' OR descr.meta value LIKE
'%$query%') AND (r.term taxonomy_id=124 OR term taxonomy id=15 OR term taxonomy id=70 OR term taxonomy id=72) AND r.object_id=item.ID
AND prec.post_id=item.ID AND img.post_id=item.ID AND descr.post_id=item.ID AND img.meta key= " thumbnail id" AND
img.meta value=img2.post id AND img2.meta key= " wp_attached_file" AND prec.meta_key= " price" AND descr.meta_key =
"rank_math_description") GROUP BY titulo, imagen, descripcion, precio');

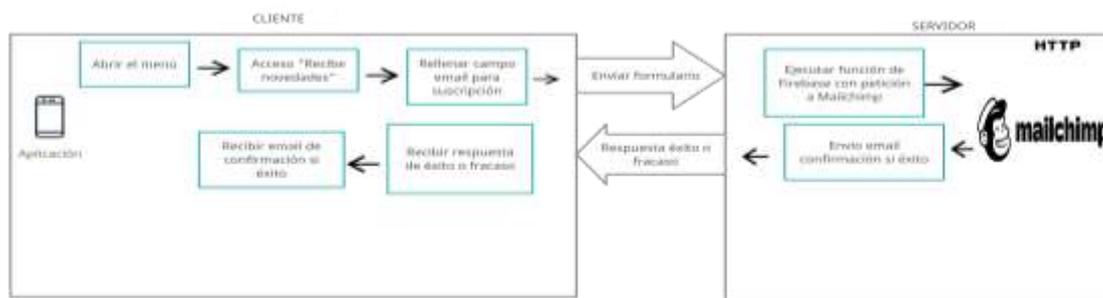
```

Definiéndose previamente en la variable “settings” la configuración de la base de datos, y escapando los caracteres problemáticos para evitar cualquier ataque de inyección SQL a nuestra base de datos. Se almacenará en “datos” una lista con los productos obtenidos.

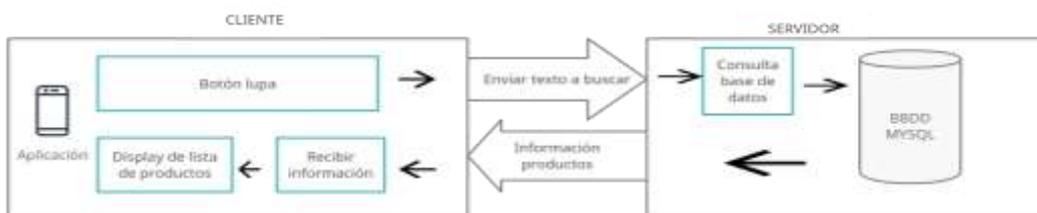
En el **Anexo II** se localiza el código de los ficheros que realizan las consultas a la base de datos para el catálogo y la comunicación con la herramienta Mailchimp.



**Figura 7a:** Comunicación cliente-servidor catálogo.



**Figura 7b:** Comunicación cliente-servidor suscripción.



**Figura 7c:** Comunicación cliente-servidor búsqueda.

## 3.2- ELECCIÓN TECNOLOGÍAS Y SERVICIOS

Se han barajado múltiples hostings para almacenar los ficheros PHP y JavaScript a los que hay que hacer petición desde la aplicación.

Para almacenar los ficheros PHP en un servidor, se ha encontrado un Web Hosting llamado **000WebHost**, que te permite almacenar ficheros y poder hacerles petición desde cualquier parte del mundo. Este Web Hosting permite ubicar ficheros PHP, MySQL y cPanel, y además, completamente gratis.

Para la comunicación con los suscriptores de Oleazara, se ha utilizado el servicio Mailchimp. Esta elección ha sido así puesto que en la página web de la empresa lo utilizan, y de esta manera, los nuevos suscriptores de la aplicación se podrán añadir a la lista ya existente de la empresa, sin necesidad de duplicar suscriptores de móvil y suscriptores de web. Todo ello mediante la conexión a Mailchimp con las mismas credenciales que se utilizan en la página web.

Para la petición a Mailchimp, se ha escogido un JavaScript que cree una función de Firebase, para que esta función utilice la biblioteca de Mailchimp de JavaScript y pueda realizar las peticiones correspondientes.

Se ha probado a desarrollar este fichero sin una función de Firebase, pero se ha invertido mucho tiempo sin éxito, por lo que se ha establecido de esta manera, ya que se ha observado que utilizando una función de Firebase se resuelve de manera sencilla y rápida.

## 3.3- MOCKUPS SOFTWARE

La aplicación tendrá el mismo aspecto tanto para Android como para iOS. Se muestra en la siguiente figura un mockup de ejemplo. El resto de mockups se encuentran en el **Anexo III**.



**Figura 8:** Página inicio Oleazara

# 4- IMPLEMENTACIÓN

En esta etapa del ciclo de vida de un sistema de información hay que proceder a la instalación del hardware y software elegidos, crear las aplicaciones correspondientes, someterlas a pruebas, crear la documentación pertinente y capacitar a los usuarios [9].

## 4.1- ARQUITECTURA HARDWARE

El hardware utilizado para desarrollar la aplicación es el siguiente:

- **SMARTPHONE ANDROID XIAOMI MI 9 T:** Utilizado para ir haciendo pruebas sobre los avances del desarrollo conectándolo al ordenador que contiene el código de la aplicación.
- **ORDENADOR PORTÁTIL:** Ordenador portátil HP con sistema operativo Windows 10 Home. Procesador Intel(R) Core(™) i7-6500U CPU 2.50 GHz. Con tipo de sistema operativo de 64 bits y memoria RAM de 8 GB. Utilizado para realizar todo el desarrollo y depuración de la aplicación Oleazara.

## 4.2- ARQUITECTURA SOFTWARE

Se va a necesitar conexión a Internet para acceder a la base de datos y realizar la suscripción a la newsletter del usuario.

A la base de datos ya existente de Oleazara se accede mediante la IDE DataGrip. Desde esta IDE, se pueden observar las tablas que tiene la empresa, y ejecutar las consultas que sean necesarias para probar que se obtiene lo que se quiere.

El acceso a la base de datos es únicamente de lectura, ya que solo se utilizará para leer los productos de las distintas categorías de la empresa.

También se ha escogido el IDE Android Studio para la programación de la aplicación, puesto que es la que se conoce debido a ser utilizada durante el grado de Ingeniería, y de esta manera resulta más fácil que otro IDE que nunca se ha estudiado.

## 4.3- HERRAMIENTAS Y TECNOLOGÍAS

Se van a usar para el diseño de este proyecto múltiples tecnologías y herramientas para lograr el objetivo de la aplicación. Estas herramientas y tecnologías van a ser:

**ANDROID STUDIO:** El proyecto va a ser desarrollado en este entorno de desarrollo integrado (IDE).

**FLUTTER:** El framework empleado para el desarrollo de programación de la aplicación va a ser Flutter.

**PHP:** Utilizado para realizar la conexión con la base de datos de Oleazara.

**FIREBASE:** : Plataforma en la nube de Google que va a ser necesaria para almacenar una función procedente de un fichero JavaScript y así poder realizar peticiones con él desde nuestra aplicación.

**MAILCHIMP:** Servicio que almacena todos los suscriptores de la newsletter de la empresa, el cual habrá que utilizar para realizar la adición del nuevo suscriptor.

**JAVASCRIPT:** Utilizado para realizar la comunicación con Mailchimp y añadir a los usuarios necesarios.

**ANDROID SDK:** Contiene herramientas necesarias y gratuitas para facilitar el desarrollo de la aplicación en Android Studio y emuladores para ver el avance del proyecto.

**DATAGRIP:** Utilizado para acceder a la visualización de la base de datos de Oleazara y así poder saber qué atributos tiene cada tabla y hacer las consultas necesarias.

**00WEBHOST:** 00WebHost es uno de los hostings más grandes de la web. Se va a utilizar para obtener un dominio que ejecute el fichero PHP mediante una petición que se realizará desde nuestra aplicación.

## 4.4- ASPECTO FINAL DE LA APLICACIÓN

Una vez terminado todo el desarrollo de la aplicación cumpliendo todos los requisitos y necesidades, se adjuntan en el **Anexo IV** todas las pantallas resultantes. Aquí se muestra como ejemplo la pantalla de un producto de la categoría “*Patés*” de la empresa:



**Figura 9:** Interfaz de los productos de la aplicación.

Se muestran todos los productos separados por un recuadro, en el que se puede ver el título del producto, su descripción, su precio y su imagen, información obtenida del JSON que se recibe tras el acceso a la base de datos.

## 5- PRUEBAS

Una vez concluido el desarrollo de la aplicación, se procede a realizar una serie de pruebas para comprobar el correcto funcionamiento de la aplicación y el cumplimiento de todos los requisitos establecidos.

Las pruebas han sido realizadas tanto en dispositivos Android como en dispositivos iOS.

En Android las pruebas han sido establecidas en dispositivos físicos, los cuales han sido los siguientes:

Xiaomi Mi 9T, OnePlus Nord, Redmi Note 8 Pro y Huawei p10 Lite.

Para las pruebas en iOS, ha sido más complicado, ya que es más exclusivo este sistema operativo y no hay posibilidad de descargar la aplicación desarrollada a menos que seas desarrollador de iOS y pagues la cuota anual. Sí se ha conseguido instalarla en un dispositivo (iPad) mediante la conexión USB de este con el ordenador que almacena el desarrollo y gracias al programa Xcode. Puesto que solo se ha podido en un dispositivo físico, se han realizado varias pruebas en los siguientes simuladores: iPhone 12 Pro Max, iPhone 12, iPhone 11, iPhone X y iPhone 8.

Tras ellas, se ha conseguido comprobar que el comportamiento de la aplicación era el adecuado y que se cumplían todos los requisitos. Esta comprobación en la cumplimentación de los requisitos se ha conseguido según lo explicado a continuación:

## 5.1- APLICACIÓN

### 5.1.1 PANTALLA PRINCIPAL

Prueba	PA01
Descripción	Resumen de la aplicación y acceso a las redes sociales
Requisito asociado	RF1
Condiciones iniciales	-
Resultado obtenido	Se muestra correctamente el resumen y los enlaces de Instagram, Facebook y Youtube funcionan a la perfección.

### 5.1.2 CATÁLOGO

Prueba	PA02
Descripción	Acceso al catálogo de los productos
Requisitos asociados	RF2
Condiciones iniciales	Conexión a Internet
Resultado obtenido	Se muestran todos los productos de la categoría indicada correctamente con su descripción, título, imagen y precio.

### 5.1.3 INFORMACIÓN GENERAL

Prueba	PA03
Descripción	Se muestra la información de la empresa en distintas opciones de menú: Quiénes somos, Producción y Política de privacidad.
Requisitos asociados	RF3
Condiciones iniciales	-

Resultado obtenido	Se encuentran en el menú las tres opciones de información, en las cuales se muestra correctamente lo pedido.
--------------------	--

#### 5.1.4 BUSCADOR

Prueba	PA04
Descripción	Se localizará un buscador que facilite la búsqueda de productos concretos sin necesidad de acceder a todo el catálogo.
Requisitos asociados	RF4
Condiciones iniciales	Conexión a Internet
Resultado obtenido	Se comprueba que se realiza la búsqueda correctamente de manera automática cada vez que escribes una letra, realizando el escapado de caracteres problemáticos.

#### 5.1.5 SUSCRIPCIÓN

Prueba	PA05
Descripción	Existirá la posibilidad de suscripción a la newsletter de Oleazara con validación del email mediante expresión regular.
Requisitos asociados	RF5
Condiciones iniciales	Conexión a Internet
Resultado obtenido	Se comprueba que se muestra un mensaje de error en la interfaz cuando se introduce un email vacío o un texto sin estructura de email, y que se recibe un mensaje de éxito cuando es un email correcto y no ha habido error en la suscripción, además de recibir un email de confirmación de Mailchimp.

#### 5.1.6 INTERFAZ Y USABILIDAD

Prueba	PA06
Descripción	<ul style="list-style-type: none"> <li>- La aplicación debe ser fácil de descargar e instalar.</li> <li>- La aplicación tendrá una interfaz intuitiva y sencilla para entenderla sin complicación, de tal</li> </ul>

	<p>manera que no requiramos de esfuerzo para su manejo.</p> <ul style="list-style-type: none"> <li>- El tiempo de aprendizaje de la aplicación será menos de media hora.</li> <li>- La aplicación debe ser fácil de analizar y corregir sus posibles fallos.</li> </ul>
Requisitos asociados	RNF1
Condiciones iniciales	-
Resultado obtenido	<p>Se comprueba que la aplicación se instala de manera sencilla con un enlace, y que es muy sencilla de entender, puesto que tiene elementos comunes como un menú desplegable con las opciones claras, o una lupa para referirse a un buscador. Los fallos que se han ido encontrando se han reparado rápidamente.</p>

#### 5.1.7 MANTENIBILIDAD Y PORTABILIDAD

Prueba	PA07
Descripción	<ul style="list-style-type: none"> <li>- Disponibilidad para cualquier tipo de dispositivo Android o iOS y para la mayoría de versiones.</li> <li>- Será necesario la conexión a Internet para el acceso a la base de datos, utilizada para visualizar los productos de la empresa.</li> <li>- La aplicación tendrá que funcionar en recursos de potencia media.</li> </ul>
Casos de uso asociados	RNF2
Condiciones iniciales	Conexión a Internet
Resultado obtenido	<ul style="list-style-type: none"> <li>- Se han hecho pruebas en diferentes dispositivos con diferentes versiones, tanto de Android como iOS, y se ha comprobado que funciona correctamente en todos ellos.</li> <li>- Con conexión a Internet se visualizan correctamente los productos de Oleazara en la opción de "Catálogo" o con el buscador.</li> </ul>

### 5.1.8 EFICIENCIA Y RENDIMIENTO

Prueba	PA08
Descripción	- El funcionamiento de la aplicación será ágil y se obtendrán tiempos de respuesta cortos (en torno a un segundo).
Requisitos asociados	RNF3
Condiciones iniciales	-
Resultado obtenido	Se observa que todo el manejo de la aplicación y cambio de interfaces utiliza menos de un segundo para su realización.

### 5.1.9 SEGURIDAD

Prueba	PA09
Descripción	La aplicación tendrá que ser segura.
Casos de uso asociados	RNF4
Condiciones iniciales	-
Resultado obtenido	El acceso a la base de datos mediante el catálogo y el proceso de suscripción a la newsletter será tan segura como lo sea HTTP, y el acceso a la base de datos mediante el buscador lo será tanto como lo sea la biblioteca <i>mysql</i> de Flutter. El almacén del correo electrónico para los suscriptores es Mailchimp, por lo que este tendrá la seguridad de dicha herramienta.

# 6- CONCLUSIONES

Para concluir finalmente, este apartado explicará las dificultades encontradas y las conclusiones del desarrollo del trabajo, además de la posible ampliación futura de la aplicación.

## 6.1- DIFICULTADES ENCONTRADAS

La dificultad primordial encontrada en el transcurso de este trabajo es la programación en el lenguaje Dart, puesto que ha sido desconocido inicialmente por el desarrollador, y ha resultado muy complicado su entendimiento, ya que es una programación completamente distinta a las estudiadas en el grado.

Además, también ha sido muy costoso el conectar la aplicación a Mailchimp y a la base de datos, puesto que no se podía en cualquier lenguaje debido a la utilización del framework Flutter y al desconocimiento de petición a ficheros que tienen que albergarse en algún servidor para poder producirse, y la manera de recibir y tratar la información de estas peticiones.

## 6.2- APORTACIONES REALIZADAS

Tras este trabajo se ha conseguido la aportación de una aplicación móvil funcional para la empresa Oleazara, válida tanto para dispositivos iOS como dispositivos Android, que es capaz de consultar el catálogo de los productos de la empresa, de suscribir a nuevos usuarios, de consultar información de esta y de buscar productos concretos mediante una serie de consultas.

Posteriormente, se pretenderá transformar la aplicación a tienda, además de poder loguearse para almacenar los pedidos anteriores e información interesante del usuario, como nombre, dirección, o tarjeta de crédito si así se desea. También se querrá obtener su disponibilidad en Play Store y App Store, para poder llegar a cualquier cliente que disponga de dispositivo Smartphone y esté interesado en los productos.

# ANEXO I:

## Aplicación XP Programming

Se ha aplicado en el proyecto la metodología Extreme Programming. Esta metodología se basa en un proyecto con pocos programadores (un único en nuestro caso), y una comunicación habitual con el cliente.

El proyecto ha tenido de duración dos meses y medio, y se ha realizado una reunión por semana (10 reuniones) con el cliente hasta conseguir la aplicación final que deseaban.

Se realizaron cuatro reuniones en Julio, cuatro en Septiembre y dos en Octubre.

**Primera reunión:** Se muestran las ideas, deseos y sugerencias del cliente al desarrollador sobre la aplicación, estableciendo sus requisitos.

**Segunda reunión:** Se muestran los mockups al cliente para recibir su feedback. El cliente pide varios cambios en algunos de los mockups para que sea más llamativo para el público, como aumento del tamaño de texto, cambio en las imágenes y aumento de margen.

**Tercera reunión:** Se enseña de nuevo los mockups finales arreglados con los cambios propuestos en la reunión anterior. Se recibe un feedback positivo del cliente.

**Cuarta reunión:** El programador le aporta al cliente sus avances en la aplicación en un simulador Android. Se dispone por el momento una aplicación llamada "oleazara", con el logo de la tienda, y una pantalla inicial con el mensaje "Bienvenidos a "Oleazara", que dura 2 segundos, y tras ella el texto de la pantalla inicial y tres iconos de las distintas redes sociales que posee la empresa (Instagram, Facebook y Youtube).. El cliente pide aumentar el margen de este texto, aumentar la duración de la pantalla de inicio a 3 segundos, y establecer los iconos de las redes sociales con enlaces, para que clicando sobre estos se pueda redireccionar a las redes de la empresa.

**Quinta reunión:** Se muestra la aplicación poseída hasta el momento de la reunión anterior, con los cambios sugeridos, además del menú desplegable con las distintas funciones. Al cliente le gusta y acepta todo lo desarrollado hasta el instante.

**Sexta reunión:** Se dispone de la aplicación desarrollada con la adición de las interfaces de los cuatro catálogos (las cuales tienen opción en el menú cada una de ellas). En estas se muestra una lista con la información de cada uno de los productos, separados unos de otros por un recuadro. Se recibe un feedback positivo, con la sugerencia de destacar el precio en negrita y aumentarlo de tamaño.

**Séptima reunión:** Se aplican los cambios, y se muestran las nuevas interfaces de suscripción de usuario, entre las que consta la principal (al clicar en la opción del menú "RECIBE NOVEDADES", la interfaz obtenida al dejar el campo vacío al darle al botón de "Guardar", la que se recibe al introducir un email erróneo, y por último, la establecida al introducir correctamente la información (donde se recibe un mensaje de éxito). El cliente muestra su "ok" tras la reunión, por lo que el desarrollador continuará con más aspectos de la aplicación para la siguiente reunión.

**Octava reunión:** El programador muestra sus avances al cliente, entre los que se encuentran las interfaces de “QUIÉNES SOMOS”, “PRODUCCIÓN” y “POLÍTICA DE PRIVACIDAD” (opciones del menú). El cliente pide cambios en los espaciados entre párrafo y párrafo, y sugiere el establecimiento de una barra de búsqueda para acceder de manera sencilla a la información del producto deseado.

**Novena reunión:** Se muestran los cambios al cliente añadidos a petición suya en la reunión anterior. El cliente acepta el desarrollo de la aplicación, y pide el final de este y la entrega de la aplicación final en la siguiente reunión, pidiendo también ser mostrada tanto en un dispositivo iOS como en un dispositivo Android, ya que hasta ahora se ha mostrado en simuladores.

**Décima reunión:** Se muestra al cliente la aplicación final, ya instalada en un dispositivo iOS y en un dispositivo Android para observar que es exacta en ambos y que funciona correctamente, ofreciéndoles el APK para Android de la aplicación, puesto que para iOS se necesitaría de más dinero y el cliente ha decidido esperar hasta que decidan establecerlo finalmente también para iOS.

## ANEXO II:

### Ficheros consultas a la BBDD y acceso a Mailchimp

Se realizan distintas peticiones a unos ficheros que establecen la comunicación con la base de datos de Oleazara. Para ello, se necesita un fichero común llamado *conexion.php* que realice la conexión con la base de datos. Este fichero tiene el siguiente contenido:

```
<?php
    $servername = //el servidor de la BBDD
    $username = // El usuario de la base de datos de la empresa
    $password = //La contraseña para acceder a la base de datos con el usuario anterior
    $dbname = //El nombre de la base de datos de la empresa
    $connection = new mysqli($servername, $username, $password, $dbname);
    if (!$connection) {
        echo "connection failed!";
        exit();
    }
?>
```

Tras definir este fichero, ya se pueden realizar las diferentes consultas a la base de datos. Por ejemplo, para obtener el JSON de los aceites de la empresa, se le hace petición al siguiente fichero PHP:

```

<?php
    include("conexion.php");
    mysqli_set_charset($connection,"utf8");
    $queryResult = $connection->
query("SELECT item.post_title as titulo, img2.meta_value as imagen, descr.meta_value as descripcion, prec.meta_value as precio
FROM oleazara_posts item JOIN oleazara_postmeta img JOIN oleazara_postmeta img2 JOIN oleazara_postmeta descr
JOIN oleazara_postmeta prec JOIN oleazara_term_relationships r WHERE ( r.term_taxonomy_id=72 AND r.object_id=item.ID AND
prec.post_id=item.ID AND img.post_id=item.ID
AND descr.post_id=item.ID AND img.meta_key= 'thumbnail_id' AND img.meta_value=img2.post_id AND img2.meta_key= 'wp_attached_file'AND
prec.meta_key= 'price' AND descr.meta_key= 'rank_math_description') GROUP BY item.post_title;");
    $result = array();
    while ($fetchdata=$queryResult->fetch_assoc()) {
        $result []= $fetchdata;
    }
header('Content-Type: application/json'); // Establecer encabezado
echo json_encode($result, JSON_UNESCAPED_UNICODE);
?>

```

Con este fichero, muestra un JSON de los aceites de la empresa, con los campos:título, imagen, descripción y precio. Para los demás tipos de catálogo, se cambia el id de la taxonomía, teniendo cada uno un id único para diferenciarlos.

El fichero JavaScript utilizado para realizar la conexión con el servicio Mailchimp, tiene el siguiente contenido:

```

const functions = require('firebase-functions');
const Mailchimp = require('mailchimp-api-v3');
var cors = require('cors');
const corsHandler = cors({ origin: true });
const _mailChimpId = //El id de Mailchimp de Oleazara
const _apiKey = //La API KEY del usuario de la empresa de Mailchimp
let mailchimp=Mailchimp;
try {
    mailchimp = new Mailchimp(_apiKey);
} catch (err) {
    console.log(err);
}
exports.mailChimp = functions.https.onRequest((req, res) => {
    corsHandler(req, res, async () => {
        const email = req.query.email;
        console.log(email);
        try {
            console.log("adding user");
            console.log(email);
            const results = await mailchimp.post(
                '/lists/' + _mailChimpId + '/members',
                {
                    email_address: email,
                    status: "pending",
                }
            );
            res.status(200).send('Success adding the user to the mailchimp list');
        } catch (error) {
            res.send(error);
        }
    });
});

```

Con la petición a este fichero, se realiza el acceso al usuario de Mailchimp de Oleazara y se aplica la adición del nuevo suscriptor, el cual se recibe en el campo req.query.email.

Devuelve un error si no se ha podido completar la adición, y un estado 200 si ha habido éxito en la petición.

# ANEXO III:

## Mockups aplicación

En este anexo se muestran todos los mockups realizados para la aplicación a desarrollar. Estos mockups son los siguientes:



Figura 10: Producción.



Figura 11: Menú desplegable



Figura 12: Catálogo producto



Figura 13: Interfaz suscripción newsletter



Figura 14: Suscripción newsletter fallida.

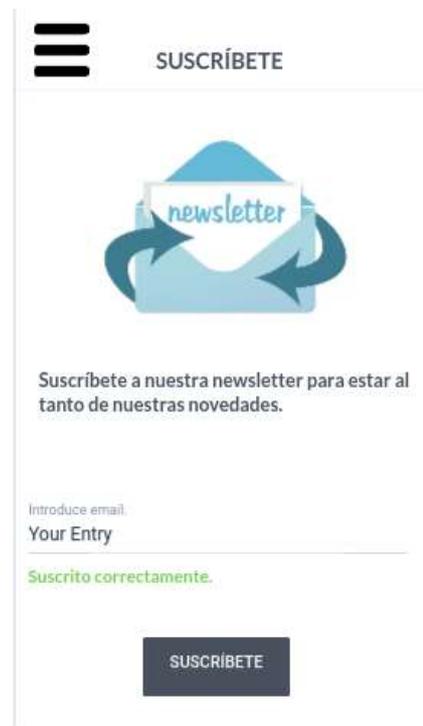


Figura 15: Suscripción newsletter exitosa.



Figura 16: Información sobre los profesionales.

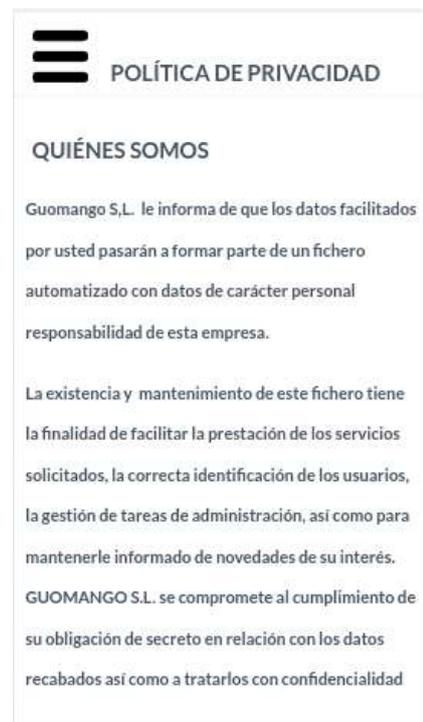


Figura 17: Política de privacidad.

# ANEXO IV:

## Aplicación Oleazara

El resto de pantallas de la aplicación que se han obtenido tras su desarrollo e implementación, son las siguientes:

- **Logo:** Se ha establecido el siguiente logo:



**Figura 18:** Logo Oleazara.

- **Pantalla de inicialización:** Esta es la pantalla que aparece al abrir la aplicación. Dura 3 segundos:



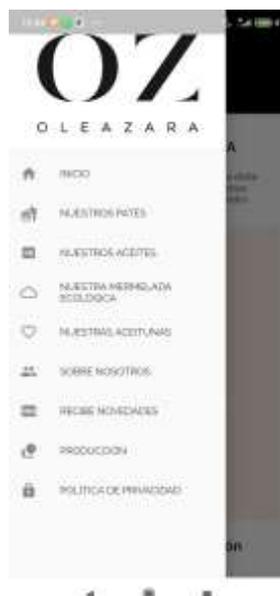
**Figura 19:** Iniciación Oleazara.

- **Pantalla inicio:** Pantalla que se observa tras la pantalla de iniciación, o cuando se clica en la pestaña "Inicio" del menú.



**Figura 20:** Home Oleazara.

- **Menú:** Aparece la siguiente pantalla al tocar en las tres rallas de menú de cualquier interfaz de la aplicación:



**Figura 21:** Menú Oleazara

**Pantallas de catálogo:** En el menú hay cuatro categorías para mostrar el catálogo: mostrar los patés, los aceites, las mermeladas ecológicas o las aceitunas. En la siguiente imagen se muestra la estructura de los patés (las cuatro tienen la misma estructura al mostrar el catálogo):



**Figura 22:** Catálogo Oleazara

**Pantalla suscripción newsletter:** Pantalla al clicar en el menú sobre “Recibe novedades”.



**Figura 23:** Suscripción newsletter

**Suscripción newsletter fallida:** La siguiente pantalla aparece cuando se introduce un email inválido en el campo email del formulario de suscripción.



**Figura 24:** Suscripción newsletter fallida.

**Suscripción exitosa:** Pantalla que se muestra cuando el email es correcto y el mensaje de confirmación es enviado.



**Figura 25:** Suscripción newsletter exitosa.

**Mensaje de confirmación:** El siguiente correo le llega al usuario tras suscribirse en la newsletter para confirmar la suscripción.



**Figura 26:** Email confirmación Oleazara.

**Sobre nosotros:** Pantalla en la que se muestra información sobre los profesionales que forman la empresa.



**Figura 27:** Sobre nosotros.

**Producción:** Pantalla que describe todo el proceso de producción de los distintos artículos que se venden en Oleazara.



Figura 28: Producción.

**Política de privacidad:** Pantalla que muestra la política de privacidad de la empresa.



Figura 29: Política de privacidad.

# BIBLIOGRAFÍA

[1] La importancia de las apps móviles en las empresas. [fecha de consulta: 2 de junio de 2020]  
<https://economyatic.com/importancia-apps-moviles-empresas/>

[2] Los iPhones se imponen al resto de smartphones como regalo estrella, pero ni el X ni los 8 son los más vendidos [fecha de consulta: 2 de junio de 2020]  
<https://www.xatakamovil.com/mercado/los-iphones-se-imponen-al-resto-de-smartphones-como-regalo-estrella-pero-ni-el-x-ni-los-8-son-los-mas-vendidos>

[3] Ciclo de vida de un sistema de información: fases y componentes [fecha de consulta: 10 de junio de 2020]  
<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/ciclo-de-vida-de-un-sistema-de-informacion-fases-y-componentes#:~:text=Fase%20de%20implementaci%C3%B3n,y%20capacitar%20a%20los%20usuarios.>

[4] Wikipedia: Requisito no funcional [fecha de consulta: 26 de junio de 2020]  
[https://es.wikipedia.org/wiki/Requisito\\_no\\_funcional](https://es.wikipedia.org/wiki/Requisito_no_funcional)

[5] Wikipedia: Metodología de desarrollo del software [fecha de consulta: 26 de junio de 2020]  
[https://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software)

[6] Metodología XP Programación Extrema [fecha de consulta: 5 de julio de 2020]  
<https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>

[7] El diagrama de casos de uso en UML [fecha de consulta: 12 de julio de 2020]  
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>

[8] Los 14 Mejores Framework de Desarrollo de aplicaciones multiplataforma (web y móvil) 2021 [fecha de consulta: 27 de julio de 2020]  
<https://ciberninjas.com/mejores-sdk-multiplataforma-2019-20/>

[9] Ciclo de vida de un sistema de información [fecha de consulta: 9 de agosto de 2020]

<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/ciclo-de-vida-de-un-sistema-de-informacion-fases-y-componentes#:~:text=Fase%20de%20implementaci%C3%B3n,y%20capacitar%20a%20los%20usuarios.>