

MEMORIA FINAL DEL PROYECTO.

“DINVASHOP”

CICLO FORMATIVO DE GRADO SUPERIOR. ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

AUTORES:

**AARON MARTÍN MARTÍN
JUAN MARÍA FERNÁNDEZ LOZANO**

TUTOR

JULIO CESAR MARTIN ALCOLEA

COORDINADORES

**JOSE LUIS LOPEZ ALVAREZ
LUZ SUSANA CANELO OLIVA**

CURSO

2022 / 2023

I.E.S. CLARA DEL REY

1.	INTRODUCCIÓN.....	1
1.1	DINVASHOP	1
1.2	ABSTRACT	1
2.	¿QUIÉNES SOMOS?.....	2
2.1	EL MODELO DE NEGOCIO.....	2
2.2	BUSINESS MODEL CANVAS.	4
3.	NUESTRA FILOSOFÍA	5
3.1	MISIÓN.....	5
3.2	VISIÓN	5
3.3	VALORES	5
4.	RESPONSABILIDAD SOCIAL.....	5
5.	ANÁLISIS DE LA COMPETENCIA.....	6
5.1	NUESTRO ENTORNO	6
5.2	MACROENTORNO	6
5.3	MICROENTORNO.....	8
5.4	COMPETENCIA	8
5.5	ANÁLISIS DE LOS CONSUMIDORES Y CLIENTES	9
5.6	ANÁLISIS DE LA ENCUESTA.	12
5.7	MAPA DE EMPATIA	14
6.	APLICACIÓN WEB DINVA.	15
6.1	REQUISITOS FUNCIONALES.....	15
6.2	REQUISITOS TÉCNICOS	15
6.3	ORGANIZACIÓN DEL TRABAJO	15
6.3.	ARQUITECTURA LÓGICA.....	16
6.4	HERRAMIENTAS DE DESARROLLO DE DINVASHOP	17
6.4.1	PHP	17
6.4.2	BOOTSTRAP 5 (integrado en html5).....	18
6.4.3	MARIADB	19
6.5.	LA APLICACIÓN WEB	20
6.5.1	BASE DE DATOS	20
6.5.2	WEB	21
7.	DESPLIEGUE DE LA APLICACIÓN WEB	27
7.1	TECNOLOGÍA.	27
7.2	DESPLIEGUE DEL CLUSTER	28
7.2.1	MICROK8S	29

7.2.2 DOCKER.....	29
7.3 CREACIÓN DEL CLUSTER.	30
7.3.1 GRÁFICO DE LA ESTRUCTURA.....	31
7.3.2 DEFINICIÓN DE APLICACIONES Y SERVICIOS.....	31
7.3.3 CONEXIÓN CON LA BASE DE DATOS.....	35
8. CONCLUSIÓN FINAL.....	36
9. BIBLIOGRAFÍA	37
9.1 LIBROS.....	37
9.2 WEBGRAFÍA	37
9.3 CONSULTAS ESTADÍSTICAS	37
9.4 EJEMPLOS Y ANÁLISIS DE LA COMPETENCIA.....	38

1. INTRODUCCIÓN

1.1 DINVASHOP

Dinvashop es una aplicación web en fase de beta, cuya idea principal es ayudar a las personas con intolerancias alimentarias, diabetes, alergias o tendencias alimentarias diferentes a las que estamos acostumbrados.

Nuestro proyecto de empresa surge de nuestras propias experiencias y vivencias con gente afectada por alguna intolerancia o alergia alimentaria, por lo que vimos una necesidad clara que actualmente no está cubierta y no nos fue necesario realizar un *Brainstorming* ya que la idea fue aceptada de buen grado en el momento.

Objetivos

- Analizar el ecosistema de aplicaciones y sitios web orientados a nuestro público objetivo: personas diabéticas, con intolerancias alimentarias, veganas o con alergias.
- Realizar un estudio de mercado / opinión para identificar las necesidades de nuestros usuarios
- Plantear un modelo de negocio que permita dar servicios novedosos a este público a través de una aplicación web o app móvil.
- Desarrollar una herramienta basada en PHP que permita dar estos servicios a los usuarios
- Planificar el despliegue y explotación de esta aplicación web en un entorno moderno y escalable.
- Profundizar en nuestro conocimiento técnico de las tecnologías utilizadas durante el desarrollo.

1.2 ABSTRACT

Dinvashop is a web application in beta-phase, which in resume, tries to help people with food intolerances, diabetes, allergies or different food trends from what we are used to.

Our business project arises from our own experiences and experiences with people affected by some food intolerance or allergy, so we saw a clear need that is currently not covered and it was not necessary for us to carry out a *Brainstorming* since the idea was accepted willingly at the time.

Goals

- Analyze the ecosystem of applications and websites aimed at our target audience: diabetics, people with food intolerances, vegans or allergies.
- Carry out a market / opinion study to identify the needs of our users
- Propose a business model that allows innovative services to be provided to the public through a web application or mobile app.
- Develop a tool based on PHP that allows to provide these services to users
- Plan the deployment and exploitation of this web application in a modern and scalable environment.
- Deepen our technical knowledge of the technologies used during development.

2. ¿QUIÉNES SOMOS?

Somos dos estudiantes de grado superior de Administración de Sistemas Informáticos en Red que decidimos agruparnos para el proyecto común de empresa y hemos decidido crear “DINVA SHOP” cuyo propósito es llegar a ser una Start-up enfocada a la gente Diabética, Intolerante, Vegana y Alérgica. Esta idea, fue presentada al concurso de ClaraEmprende y fue premiada con un segundo puesto de entre más de 20 por un jurado formado por profesionales de empresas de logística, recursos humanos, banca y emprendedores

En principio la idea de aplicación web está pensada y diseñada para dar trabajo a cuatro personas, que era la idea inicial, pero por causas diversas, se llevará a cabo por dos, Aaron y Juan, lo cual supondrá una serie de limitaciones, que iremos describiendo.

2.1 EL MODELO DE NEGOCIO

Nuestro modelo de negocio se basa en el *marketing y publicidad*. Nuestro proyecto no pretende vender productos sino más bien servicios. Actualmente en España se estima que un 25% de la población padece algún tipo de alergia o intolerancia alimenticia, esta gente necesita productos específicos sin determinados alergenicos pero su visibilidad o publicidad en los supermercados es ínfima. Muchos de estos clientes cuando descubren

un producto adecuado tanto económicamente como en términos de salud, acuden sistemáticamente al mismo establecimiento en busca del mismo producto ante el desconocimiento de alternativas y propuestas por parte de otras entidades.

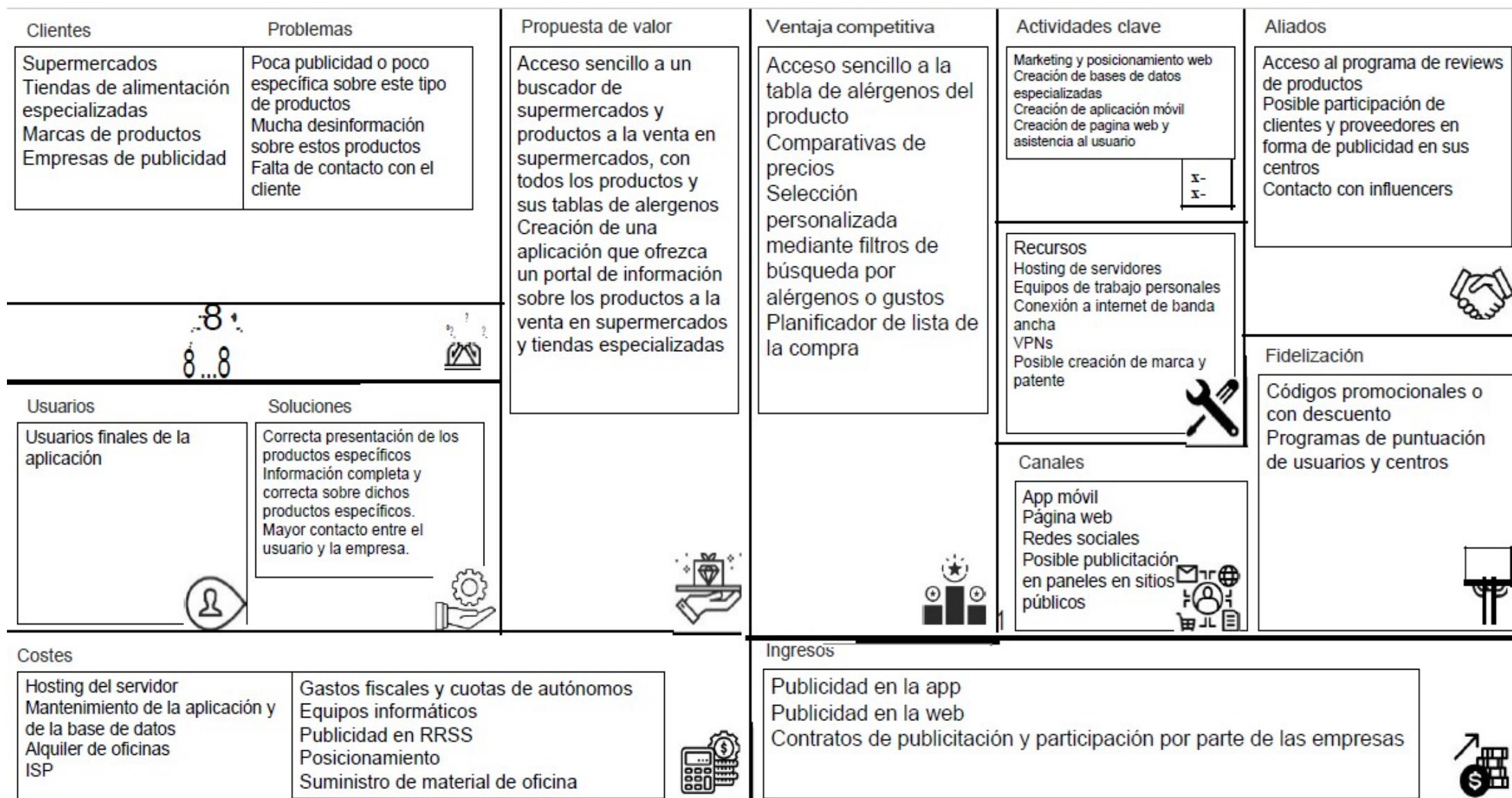
Aquí es donde incide nuestro proyecto, ya que este se basa en una aplicación que mediante el uso de filtros permite al usuario poder conocer alternativas a los distintos productos sin alérgenos en otros supermercados, a la vez que puede compararlos a nivel económico y social mediante las opiniones de otros usuarios.

Nuestro proyecto pertenecería al sector terciario ya que ofrecemos un servicio informático de aplicación móvil/web, de forma más específica con el 6209 ¹

A continuación, presentamos nuestro Business Model Canvas, que es una herramienta para analizar cualquier proyecto en sus inicios, planteando 9 interrogantes a los que hay que dar respuesta.³

2.2

BUSINESS



3. NUESTRA FILOSOFÍA

3.1 MISIÓN

“Cambiar la vida de las personas con problemas alimenticios.”

3.2 VISIÓN

“Ser la aplicación favorita del público y estar en todos los smartphones del mundo, nada puede pararnos”

3.3 VALORES

Innovación.

Responsabilidad.

Sencillez.

Cercanía.

Bienestar.

4. RESPONSABILIDAD SOCIAL.

Tratamos de contribuir con la sociedad en la que vivimos, desarrollando un método que pueda facilitar la vida de las personas que, a diario, encuentran numerosos problemas con su alimentación. Facilitarles el acceso a los diferentes componentes de cada producto y, a su vez, indicarles dónde pueden adquirirlos y de la manera más económica posible para sus bolsillos es nuestro objetivo.

Somos una empresa comprometida con el medio ambiente, toda nuestra base de datos y facturación se realiza de manera digital, por lo tanto, somos una empresa con cero niveles de consumo de papel y nuestros equipos son de empresas que fielmente colaboran con el desarrollo sostenible del planeta.

5. ANÁLISIS DE LA COMPETENCIA

5.1 NUESTRO ENTORNO

Dinvashop surge, como ya hemos comentado, de la necesidad alimentaria específica de personas con alergias e intolerancias. Además de ser útil también para vegetarianos y veganos.

5.2 MACROENTORNO

Como macroentorno entendemos los factores externos genéricos como son:

- La economía actual. Actualmente hay una gran inflación dadas las causas socioeconómicas que estamos viviendo, véase la guerra entre Rusia y Ucrania o la reciente salida de la pandemia causada por el COVID-19.

- Las tecnologías, donde se observa un mayor uso de los Smartphones, así como la creciente influencia de los denominados *Influencers* de las redes sociales, y el impulso que estas han adquirido dada la reciente experiencia con la pandemia.

- Factores demográficos y socioculturales. Se observa una mayor tendencia hacia hábitos alimenticios más respetuosos con el medioambiente (o “*ecofriendly*”) que los de nuestros predecesores, inclinándose hacia tendencias como el veganismo y la ausencia de determinados alérgenos y componentes biológicos de los alimentos.

- En cuanto a los factores políticos, nos encontramos en un momento de esplendor en España para los emprendedores, ya que fue aprobada y ratificada la ley Crea y Crece (anexo 1) y está pendiente de ratificación la ley sobre *Startups* (anexo 2).

Desarrollo creciente de alergias en la población:

Ayudaremos a cubrir la necesidad de estas personas. Según datos de la Fundación² de seguridad alimentaria y prevención de alergias, 520 millones de personas en el mundo

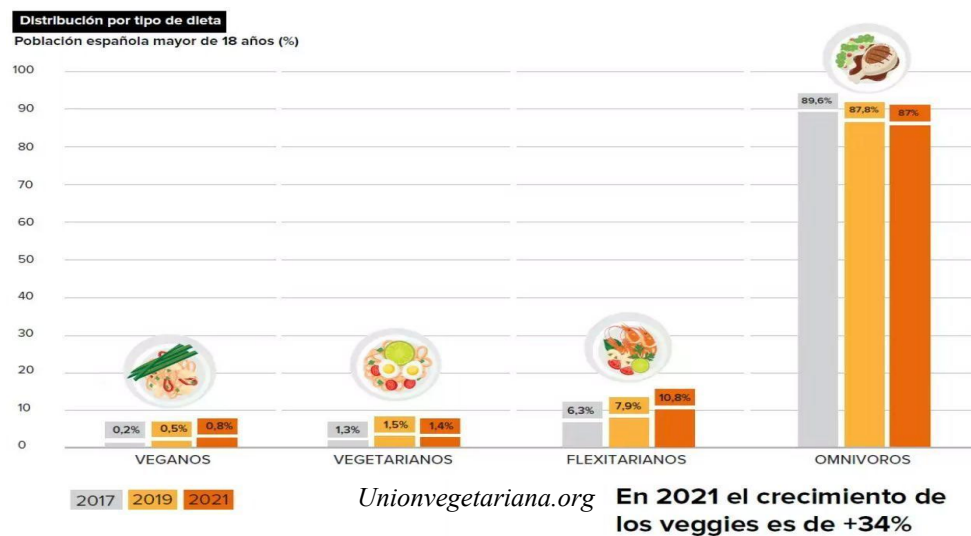
sufren algún tipo de alergia, de los cuales 2 millones son españolas. Los afectados son, mayoritariamente, los menores y adolescentes.

Según nuestros datos un 14,5% de los encuestados sufre alguna alergia.

Popularidad del veganismo/vegetarianismo:

Esta tendencia alimentaria cada vez es más popular entre los jóvenes mayoritariamente.

Según la consultora Lantern³ en su informe “The green revolution”:



Uso generalizado de los Smartphones:

No es necesario investigar demasiado para darse cuenta de esto último. Solo hace falta mirar a nuestro alrededor y observar cuántas personas están utilizando su teléfono. Sin necesidad de atender a datos muy concretos, si la mayoría de la población utiliza recurrentemente su teléfono para diversidad de finalidades. Con gran seguridad nuestro cliente objetivo tiene un Smartphone para poder utilizar nuestra aplicación móvil.

5.3 MICROENTORNO

Nos referimos al microentorno como aquellos factores específicos de nuestro modelo de negocio que pueden afectar a nuestra idea o empresa.

En cuanto al microentorno se refiere existen otras aplicaciones relacionadas con la alimentación y/o alergias e información nutricional como Celicity o Yuka (aunque Celicity está desapareciendo del mercado), aunque en ninguna se ofrece la funcionalidad de usar como filtros todas y cada una de las alergias.

Otra ventaja con la que contamos es que nuestra aplicación se puede desarrollar con nuestro propio trabajo o con poca ayuda a muy bajo coste. Los entornos de desarrollo necesarios son software gratuito y se pueden ejecutar en cualquier ordenador con unos recursos medianamente potentes.

En cuanto al de la app lanzamiento existen múltiples plataformas aparte de Google Play Store y App Store de Apple. Tenemos la posibilidad de cobrar por descargarla u ofrecerla de forma gratuita, pero añadiendo anuncios por los que cobraremos una pequeña cantidad por cada persona que abra el banner y al mismo tiempo incluir la opción de eliminar estos anuncios pagando.

5.4 COMPETENCIA



Celicity puede ser quizás uno de nuestros principales competidores, ya que su aplicación te ayuda a encontrar restaurantes y tiendas especializadas en personas celíacas. Pero no contemplan en principio otros tipos de alergias o tendencias alimenticias. No obstante, llevan años sin actividad en redes sociales.



En materia de veganismo, si observamos más competencia, ya que existen diversas aplicaciones en el mercado, tanto nacional como internacional, pero la gran mayoría de ellas están más enfocadas a

guiarnos a la hora de elegir un restaurante para salir a cenar o comer (Happy Cow).

VHappy si puede hacernos más daño a la hora de la competencia, ya que nos ofrece tanto productos como ubicaciones donde adquirirlos.



Por último, mencionamos VEGAHOLIC por lo curioso de su función, se encargan de buscar y comparar diferentes bebidas alcohólicas y ver si su composición es animal o vegetal. Al igual que Celicity, lleva años sin actividad en redes.



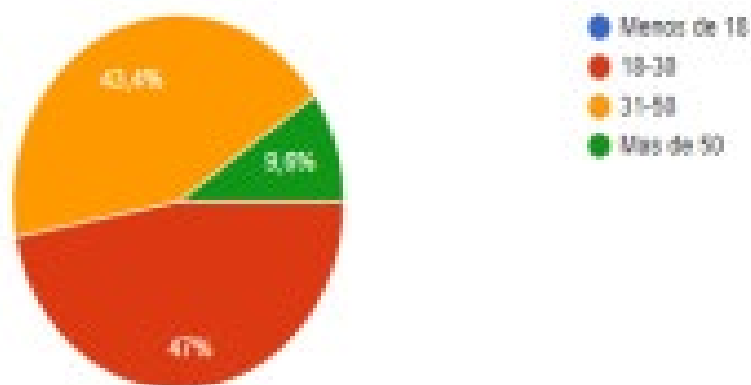
En el mundo de las intolerancias nos parece importante reseñar la competencia que podría generarnos la aplicación “All I Can Eat” con una base de datos muy completa sobre alimentos y sus componentes y los diferentes alérgenos que contienen.

5.5 ANÁLISIS DE LOS CONSUMIDORES Y CLIENTES

Apoyándonos en la encuesta realizada por nuestros propios medios podemos observar que nuestros potenciales clientes son jóvenes mayoritariamente y otra gran parte es menor de 50

¿Podrías decirnos tu edad?

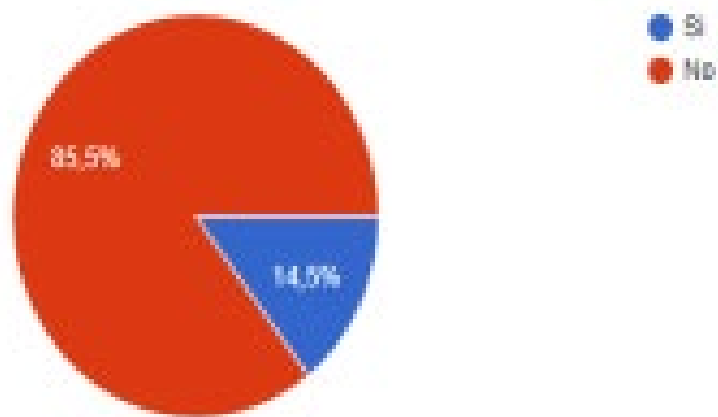
63 respuestas



y un 14.5% es alérgico o intolerante a algún alimento.

¿Eres intolerante o alérgico a algún alimento?

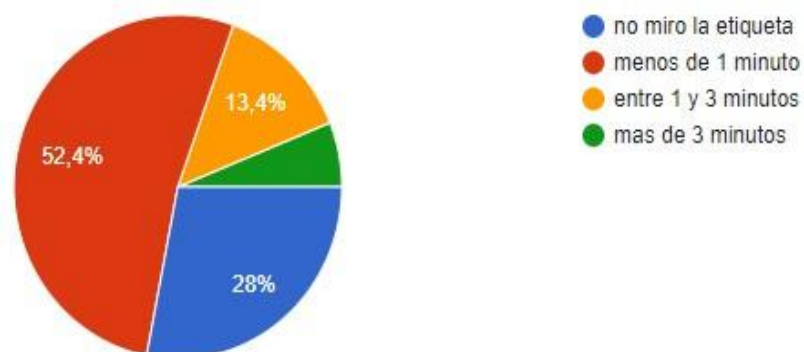
83 respuestas



Vemos una clientela que se preocupa por los ingredientes contenidos en los productos que consumen, ya que más de la mitad de ellos dedican un momento a echar un vistazo a la etiqueta, y un 25% de ellos mira la etiqueta un poco más.

Cuando compras cualquier alimento, ¿miras la etiqueta? Si es así, ¿Cuánto tiempo dedicas a mirar la etiqueta?

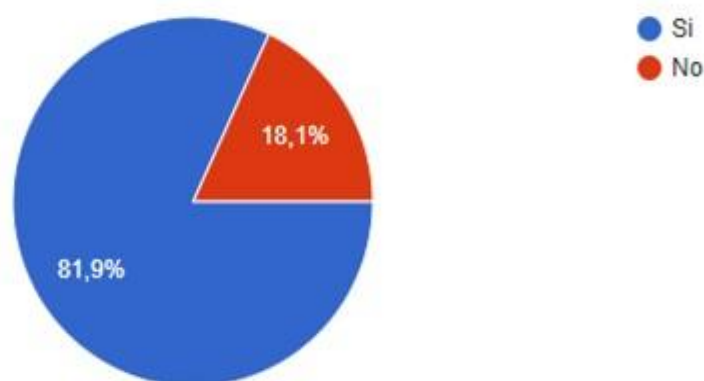
82 respuestas



Además, suelen planificar su compra con una lista y otros pocos ya utilizan una aplicación para ello.

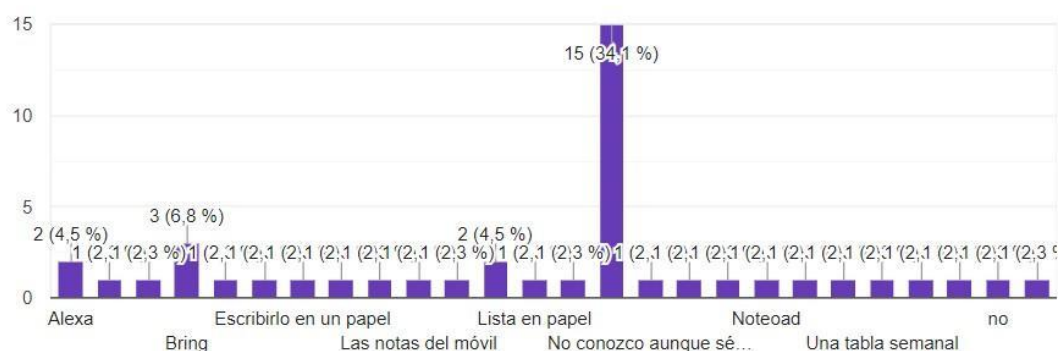
Antes de ir a comprar, ¿Haces una lista?

83 respuestas



¿Conoces algún medio para planificar tu compra? De ser así, ¿Podrías mencionar uno?

44 respuestas

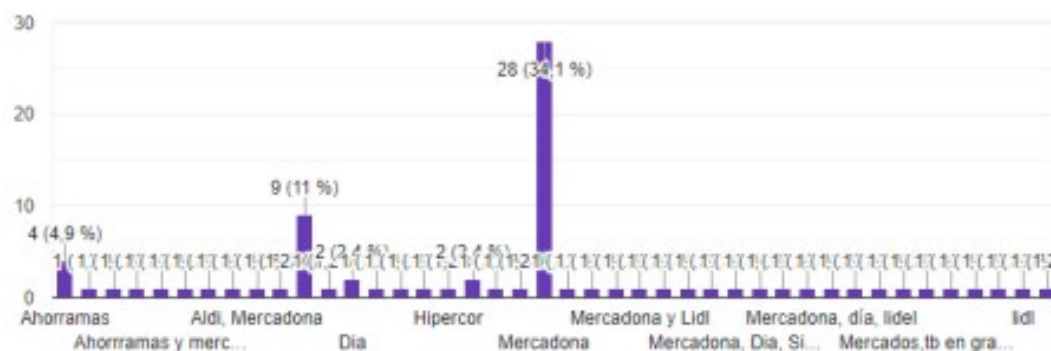


Otro factor a tener en cuenta es la variedad de supermercados en los que compran. Aunque predomina Mercadona, podemos observar una diversidad minoritaria de otras marcas como Ahorramás, Carrefour, Lidl, Hipercor...

¿En qué supermercado sueles comprar?

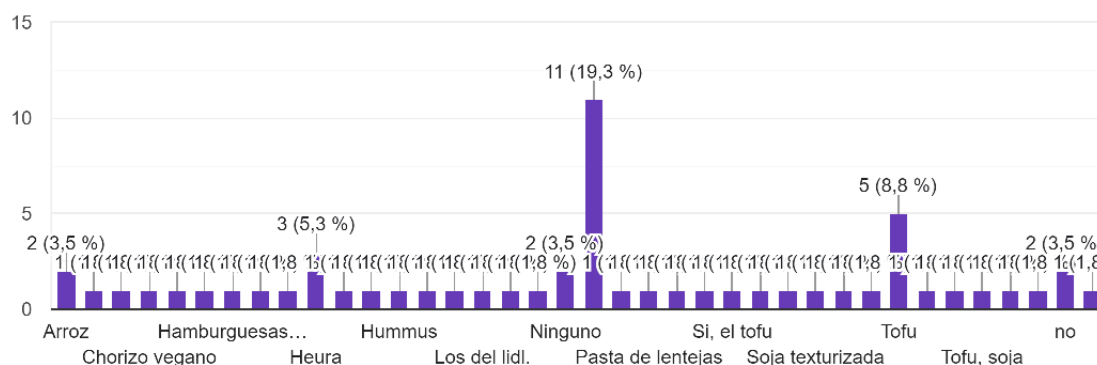
 Copiar

82 respuestas



¿Conoces algún producto vegano? ¿Cual?

57 respuestas



En cuanto al veganismo 57/83 encuestados conocen algún producto vegano.

5.6 ANÁLISIS DE LA ENCUESTA.

Realizando un análisis sobre nuestras encuestas realizadas mediante Google forms vemos que:

Lo primero, tanto la totalidad de aquellos usuarios con alergias y/o intolerancias alimentarias o preferencias/restricciones en las mismas, como aquellos usuarios opuestos a estos últimos, utilizarían nuestra app.

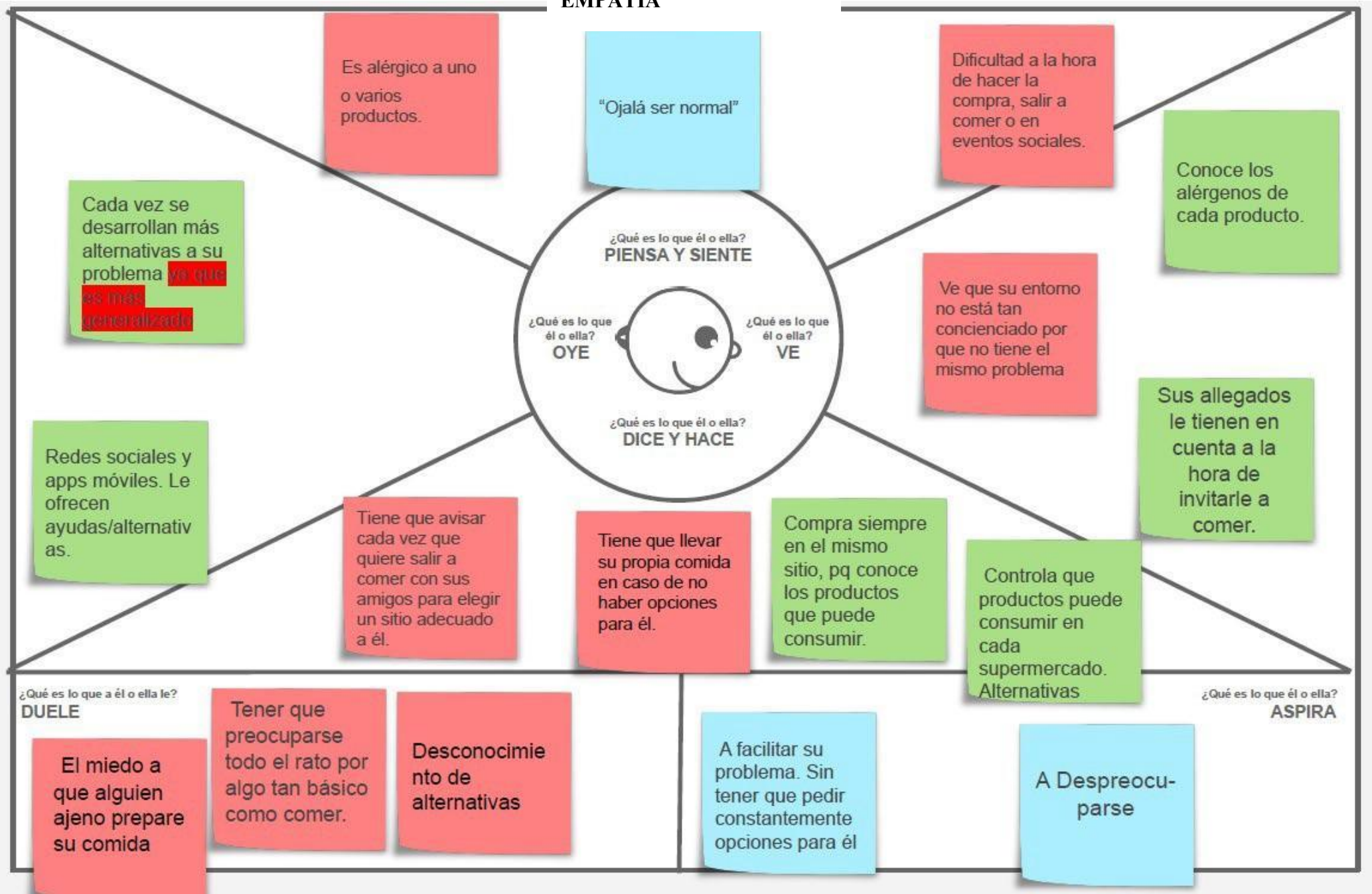
Lo segundo, refiriéndonos al rango de edad, observamos una mayor concentración y diferenciación entre los 20 y 55 años, habiendo casos aislados fuera de dicha franja.

Y tercero y último, observamos que prácticamente el 90% de los usuarios que respondieron nuestra encuesta, dan más valor a los pequeños “extras” o añadidos que a la función base de la aplicación. Demostrando esto, que la población prefiere saber antes ¿por cuánto? que otras cuestiones como ¿dónde? o ¿El qué?

A continuación, presentamos nuestro mapa de empatía, que es una herramienta gráfica que nos permite ponernos en la piel de nuestro cliente objetivo para entender sus necesidades, pensamientos y sentimientos.

5.7 MAPA DE

EMPATIA



6. APLICACIÓN WEB DINVA.

6.1 REQUISITOS FUNCIONALES.

- Mantener una base de datos de productos relacionados con intolerancias, alergias....
- Asociar en la BD los productos con los alérgenos ...
- Mantener una relación de tiendas de proximidad o grandes almacenes en las que estén disponibles
- Permitir a los usuarios listar productos y así obtener tanto los alérgenos asociados como los comercios en el lugar que
- Que los usuarios registrados puedan tener un perfil con sus alergias e intolerancias para facilitar la búsqueda.

6.2 REQUISITOS TÉCNICOS

- Desarrollo con lenguaje PHP.
- Creación y uso de una base de datos de libre distribución como por ejemplo mysql.
- Creación de una aplicación web escalable y portable.
- Despliegue en entornos escalables basados en Docker, Kubernetes u otros sistemas de virtualización.
- Estudio de un posible despliegue en la nube.
- Securitización de los datos.

6.3 ORGANIZACIÓN DEL TRABAJO

Para poder empezar a diseñar la aplicación web y las características, primero vamos a aclarar que, llegados a este punto, lo ideal habría sido, como se comentó anteriormente, haber repartido el trabajo entre cuatro personas, ya que al ser dos, limita mucho los objetivos que podamos conseguir por falta sobre todo de tiempo y de recursos.

Este proyecto será la base sobre la que, en caso de presentarse la ocasión, se desarrollará finalmente la aplicación con fines comerciales.

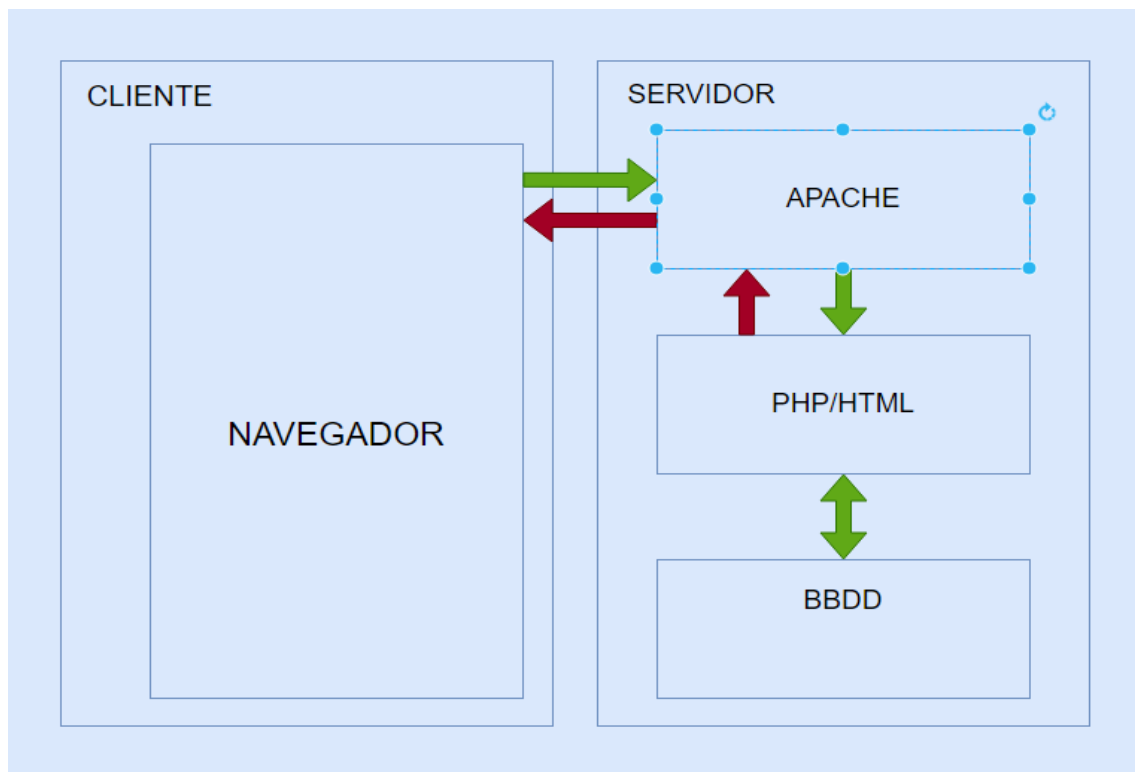
El desarrollo se hará de manera simultánea por dos personas, por lo que, para facilitar el

trabajo, todos los cambios se subirán a un repositorio de GitHub para mantenerlo actualizado y funcional.

Primero instalaremos un servidor web local XAMPP, para poder poner una Base de Datos de prueba en Local y que interprete código de PHP. Una vez definido todo, se estudiará la posibilidad de desplegarlo en la nube.

Como entorno de desarrollo, utilizaremos la herramienta Visual Studio Code, en la cual es necesario instalar el paquete de PHP y de Bootstrap 5 que creara las librerías de estilo que le iremos dando a nuestra app.

6.3. ARQUITECTURA LÓGICA



6.4 HERRAMIENTAS DE DESARROLLO DE DINVASHOP

6.4.1 PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación de código abierto especialmente diseñado para el desarrollo de aplicaciones web dinámicas. Fue creado originalmente por Rasmus Lerdorf en 1994 y, desde entonces, ha evolucionado y se ha convertido en uno de los lenguajes más populares para el desarrollo web.

PHP se ejecuta en el lado del servidor, lo que significa que el código PHP se procesa en el servidor web antes de que se envíe al navegador del usuario. Esto permite generar contenido dinámico en función de la interacción del usuario, datos almacenados en una base de datos u otras condiciones específicas.

Algunas características y ventajas de PHP son:

1. Fácil de aprender: PHP tiene una sintaxis similar a otros lenguajes como C o Java, lo que facilita su aprendizaje para aquellos que ya están familiarizados con la programación estructurada.
2. Amplia compatibilidad: PHP es compatible con la mayoría de los servidores web, como Apache, Nginx e IIS. También es compatible con diversos sistemas operativos, como Windows, Linux y macOS.
3. Integración con bases de datos: PHP puede conectarse a una amplia gama de sistemas de bases de datos, como MySQL, PostgreSQL, Oracle y SQLite, lo que facilita la gestión y manipulación de datos en aplicaciones web.
4. Gran cantidad de recursos y comunidades activas: Existe una amplia variedad de recursos en línea, documentación, tutoriales y comunidades activas que brindan soporte y comparten conocimientos sobre PHP.
5. Frameworks populares: PHP cuenta con una amplia gama de frameworks populares, como Laravel, Symfony y CodeIgniter, que facilitan el desarrollo de aplicaciones web complejas y siguen las mejores prácticas de programación.

6. Capacidad de generar contenido dinámico: Con PHP, puedes generar contenido dinámico en tus páginas web, como mostrar datos de una base de datos, procesar formularios, interactuar con APIs externas y más.

PHP se ha utilizado ampliamente para crear diversos tipos de aplicaciones web, desde sitios web simples hasta plataformas de comercio electrónico, sistemas de gestión de contenido (CMS), redes sociales y mucho más.

6.4.2 BOOTSTRAP 5 (integrado en html5)

Bootstrap 5 es la última versión de un popular framework de desarrollo front-end diseñado para facilitar la creación de sitios web y aplicaciones web responsivas y móviles. Fue lanzado en 2020 y ha ganado una amplia adopción en la comunidad de desarrollo web.

Bootstrap 5 se basa en HTML, CSS y JavaScript, y ofrece una colección de estilos predefinidos, componentes y utilidades que permiten crear interfaces de usuario atractivas y funcionales de manera rápida y sencilla. Algunas características y beneficios de Bootstrap 5 son:

1. Responsividad: Bootstrap 5 está diseñado para crear sitios web que se adapten automáticamente a diferentes tamaños de pantalla, desde dispositivos móviles hasta pantallas de escritorio. Los componentes y diseños de Bootstrap se ajustan y reorganizan de manera fluida para ofrecer una experiencia de usuario óptima en todos los dispositivos.
2. Componentes y estilos predefinidos: Bootstrap 5 proporciona una amplia variedad de componentes reutilizables, como navegación, botones, formularios, tarjetas, carruseles, barras de progreso y mucho más. Estos componentes vienen con estilos predefinidos y se pueden personalizar según las necesidades del proyecto.
3. Diseño flexible y grid system: Bootstrap 5 utiliza un sistema de cuadrícula (grid system) que facilita la creación de diseños de página responsivos y organizados. El grid system de Bootstrap permite dividir una página en filas y columnas, lo que facilita el posicionamiento y alineación de los elementos de la interfaz.
4. Personalización sencilla: Bootstrap 5 proporciona variables CSS y opciones de configuración que permiten personalizar fácilmente los estilos y apariencia de los

componentes, como colores, fuentes, espaciado, bordes, etc. Esto ayuda a que los diseños creados con Bootstrap se adapten a la identidad visual de cada proyecto.

5. **Compatibilidad con navegadores:** Bootstrap 5 es compatible con los principales navegadores modernos, como Chrome, Firefox, Safari, Edge, entre otros, lo que garantiza una experiencia consistente en diferentes entornos de navegación.
6. **Documentación y comunidad activa:** Bootstrap cuenta con una documentación exhaustiva que proporciona guías, ejemplos de uso y referencias de todos los componentes y características. Además, cuenta con una comunidad activa de desarrolladores que comparten recursos, plantillas y soluciones a través de foros y sitios especializados.

Bootstrap 5 es ampliamente utilizado en el desarrollo web para agilizar el proceso de diseño y desarrollo de interfaces responsivas y estilizadas. Al utilizar Bootstrap, los desarrolladores pueden ahorrar tiempo y esfuerzo al aprovechar la base sólida de estilos y componentes proporcionados por el framework.

6.4.3 MARIADB

MariaDB es un sistema de gestión de bases de datos (SGBD) de código abierto y una bifurcación del popular sistema de gestión de bases de datos MySQL. Fue desarrollado por la comunidad de desarrolladores de MySQL después de la adquisición de MySQL por parte de Oracle Corporation en 2010.

MariaDB está diseñado para ser una alternativa compatible y mejorada de MySQL, lo que significa que las aplicaciones y los comandos de MySQL pueden utilizarse con MariaDB sin necesidad de modificaciones importantes. Ofrece un rendimiento similar y una funcionalidad mejorada en comparación con MySQL.

Entre las características clave de MariaDB se incluyen:

7. **Compatibilidad con MySQL:** MariaDB es compatible con la sintaxis y los comandos de MySQL, lo que facilita la migración de aplicaciones y bases de datos existentes de MySQL a MariaDB.
8. **Mejoras de rendimiento:** MariaDB ofrece mejoras de rendimiento en comparación con MySQL, incluyendo un optimizador de consultas más rápido y una mejor gestión de índices.

9. Almacenamiento de datos: MariaDB admite diferentes motores de almacenamiento, incluyendo InnoDB (el motor de almacenamiento predeterminado), Aria, MyISAM y más. Cada motor de almacenamiento tiene sus propias características y beneficios específicos.
10. Funciones y extensiones adicionales: MariaDB incluye funciones y extensiones adicionales que no están presentes en MySQL, como el motor de almacenamiento ColumnStore para análisis de big data, y la capacidad de ejecutar procedimientos almacenados en lenguajes de programación como Python y Java.
11. Seguridad: MariaDB ofrece mejoras en seguridad, como la encriptación de datos en reposo y en tránsito, autenticación de dos factores, y auditoría de acceso.
12. Comunidad activa: MariaDB cuenta con una comunidad de desarrolladores y usuarios activos que contribuyen con mejoras y nuevas características. Además, existe soporte comercial disponible a través de MariaDB Corporation.

En resumen, MariaDB es un sistema de gestión de bases de datos de código abierto que proporciona una alternativa compatible y mejorada a MySQL, con características adicionales, mejoras de rendimiento y seguridad.

6.5. LA APLICACIÓN WEB

6.5.1 BASE DE DATOS

Para almacenar los datos relativos a los productos, supermercados, usuarios, alergen... se define una base de datos creada en MariaDB que contiene toda esa información y sus relaciones.

Esta base de datos tiene las siguientes tablas:

```
+-----+
| Tables_in_tfg          |
+-----+
| alergen                |
| producto_alergeno      |
| producto_supermercado  |
| productos              |
| supermercados          |
| usuarios                |
+-----+
```


La tabla de usuarios mantiene los usuarios existentes en la aplicación.

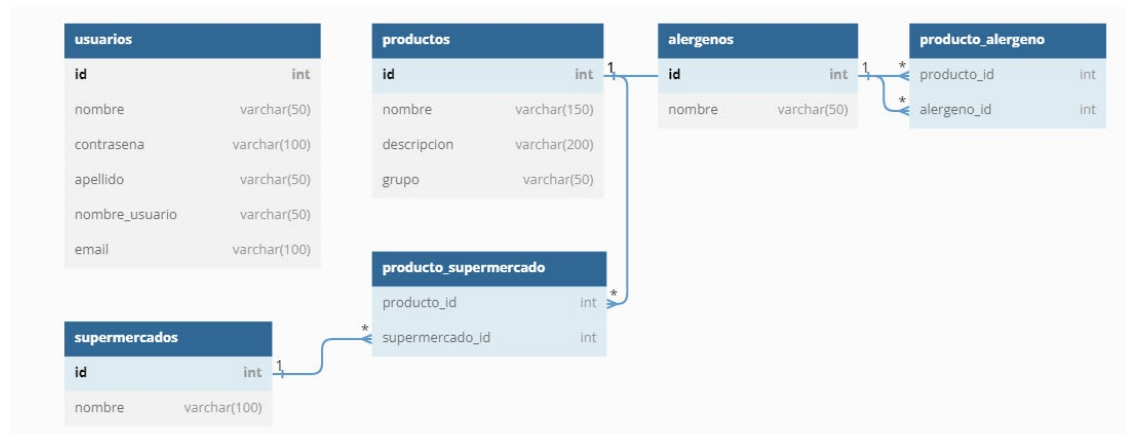
La tabla alérgenos contiene los datos de los 14 alérgenos definidos por la OMS.

La tabla de productos almacena los productos listados en la aplicación.

La tabla supermercados almacena datos sobre algunos supermercados que hemos escogido como ejemplos en nuestra aplicación.

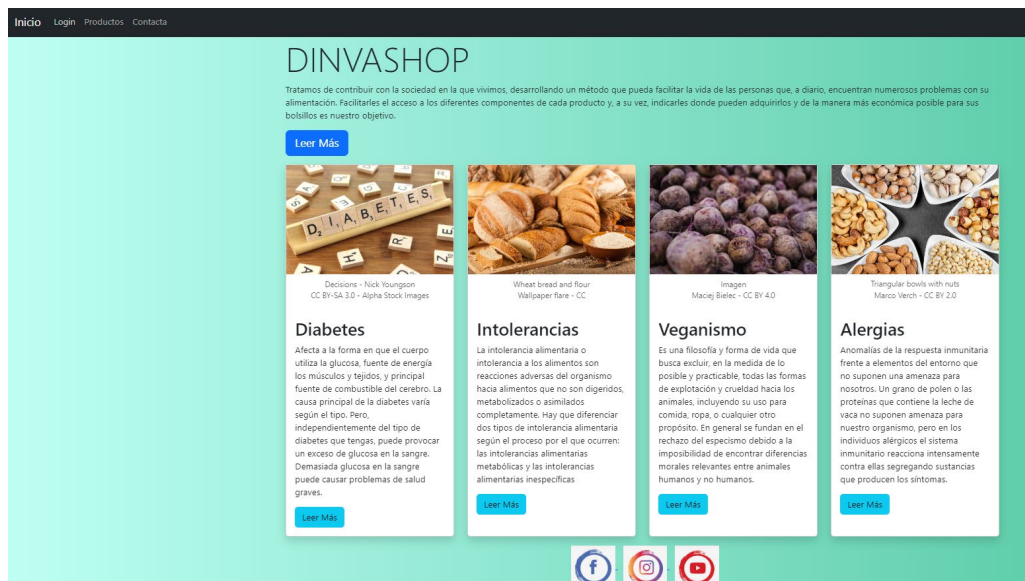
Las tablas producto alergeno y producto supermercado son cruces de las tablas que almacenan datos extras sobre los productos, sus alergenos y en que supermercados se pueden encontrar los productos que no son de marca propia.

El diagrama de las tablas es el siguiente:

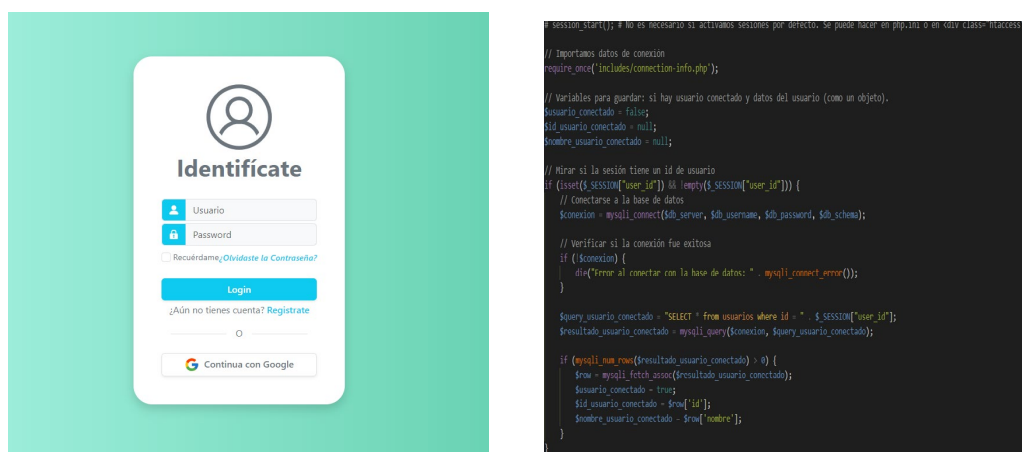


6.5.2 WEB

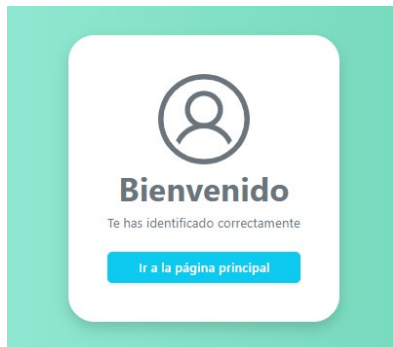
Para crear la aplicación vamos a partir de una página index.html en la cual vamos a desarrollar una barra de navegación, desde la cual poder ir accediendo al resto de páginas, contiene también una breve introducción a lo que es Dinva, y expone los diferentes temas a los que va enfocado la app. Incluye, además, un enlace a las RRSS para que nos conozcan un poco más y se unan a la comunidad de internet.



Después pasamos a crear la página del Login a la base de datos. Está también creada a través de librerías de Bootstrap para darle estilo a los contornos, a la imagen y recuadros para logarse, también contiene las conexiones con la base de datos.



Una vez el usuario se haya logueado en la base de datos le devolverá este mensaje de bienvenida:



Después pasamos al formulario de registro, al cual podemos acceder desde el enlace”
Regístrate”.

A registration form for "DINVASHOP" on a teal background. The form has several input fields: "Nombre" and "Apellidos" (first and last names), "Nombre de usuario" (username), "Email", "Password", "Dirección" (address, with "1234 Main St" as a placeholder), "Ciudad" (city), "País" (country, with a dropdown menu showing "España"), and "Zip". There are two blue buttons at the bottom: "Comprobar datos" (check data) and "Registro" (register).

```
// Consulta para verificar la existencia del nombre de usuario
$consulta = "SELECT COUNT(*) FROM usuarios WHERE nombre_usuario = '$nombre_usuario'";
$resultado = $conexion->query($consulta);
$fila = $resultado->fetch_row();
$existeUsuario = $fila[0] > 0;

if ($existeUsuario) {
    // Mensaje de error si el nombre de usuario ya existe
    $error = "El nombre de usuario ya está en uso.";
    echo "<script>alert('";
} else {
    // Insertar los datos
    $nombre = $_POST["nombre"]
    $insertar = "INSERT INTO usuarios (nombre, apellidos, email, password, nombre_usuario)
    VALUES ('$nombre', '$apellidos', '$email', '$password', '$nombre_usuario')";
    if ($conexion->query($insertar) === TRUE) {
```

Realizamos el insert, con los datos que le ofrecemos, en caso de ya formar parte de la base de datos, nos dará mensaje de error, en caso contrario lo agregará a la misma.

Y con este código de PHP verificaremos que el usuario ingresa sus datos y que no está ya en nuestra base de datos. En caso de no estar, lo añadirá a la base de datos.

```
// Verificar si el formulario ha sido enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtener los datos del formulario
    $nombre = $_POST["nombre"];
    $apellidos = $_POST["apellidos"];
    $email = $_POST["email"];
    $password = $_POST["password"];
    $nombre_usuario = $_POST["nombre_usuario"];

    // Consulta para verificar la existencia del nombre de usuario
    $consulta = "SELECT COUNT(*) FROM usuarios WHERE nombre_usuario = '$nombre_usuario'";
    $resultado = $conexion->query($consulta);
    $fila = $resultado->fetch_row();
    $existeUsuario = $fila[0] > 0;

    if ($existeUsuario) {
        // Mensaje de error si el nombre de usuario ya existe
        $error = "El nombre de usuario ya está en uso.";
        echo "<script>alert('$error');</script>";
    } else {
```

En la página de productos, listamos los últimos artículos añadidos a la base de datos y sus principales alérgenos, lo que es más importante si buscas lo más adecuado para tu

compra.

```
// Insertar los datos del usuario en la base de datos
$insertar = "INSERT INTO usuarios (nombre, apellidos, email, password, nombre_usuario)
VALUES ('$nombre', '$apellidos', '$email', '$password', '$nombre_usuario')";

if ($conexion->query($insertar) === TRUE) {
    // Mensaje de confirmación si se agrega el usuario correctamente
    $mensaje = "Usuario agregado correctamente.";
    echo "<script>alert('$mensaje');</script>";
} else {
    // Mensaje de error si ocurre algún problema al agregar el usuario
    $error = "Error al agregar el usuario: " . $conexion->error;
    echo "<script>alert('$error');</script>";
}
}
```

```
// Consulta para obtener los productos listados por fecha de más reciente a menos
$query = "SELECT
p.id, p.nombre, p.descripcion, p.grupo,
IFNULL(a.nombre, 'Sin alérgenos') AS alergeno
FROM productos p
LEFT JOIN producto_alergeno pa ON p.id = pa.producto_id
LEFT JOIN alergenos a ON pa.alergeno_id = a.id
ORDER BY p.grupo ASC";

$resultado = mysqli_query($conexion, $query);
```

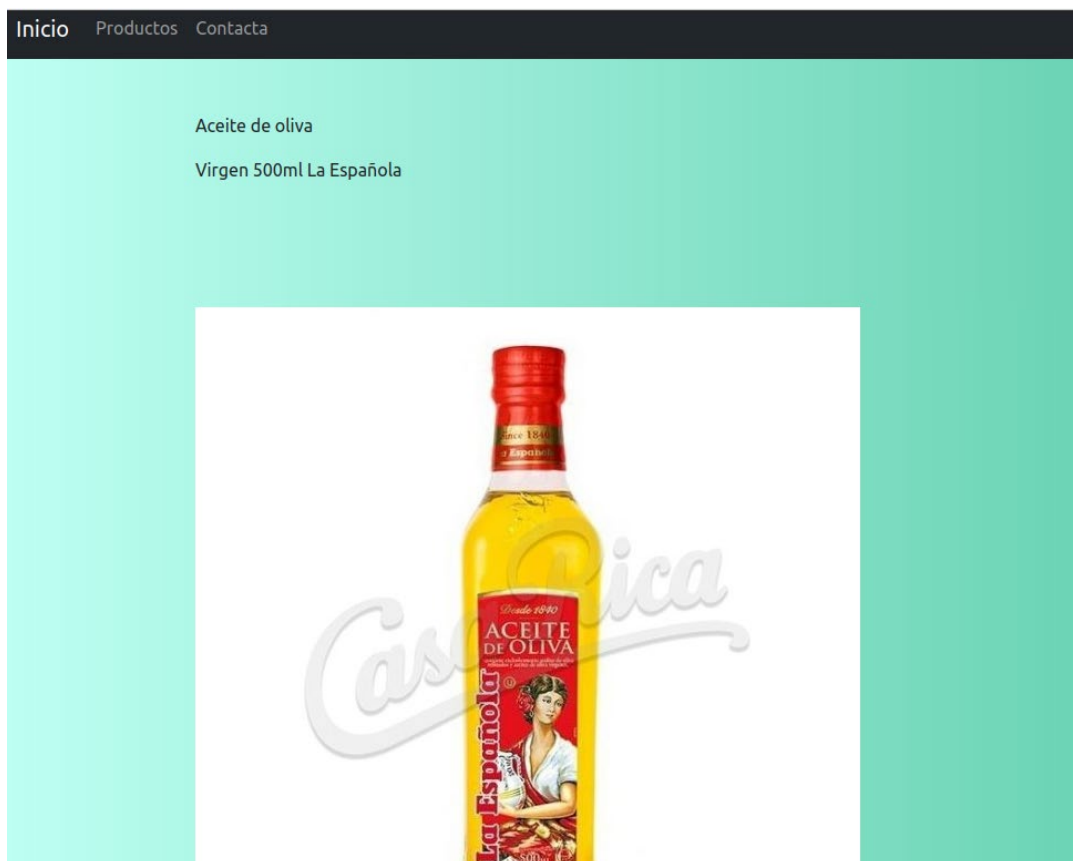
Lanzamos la consulta a la Base de datos y nos mostrará de esta manera los productos.

Te mostramos una gran cantidad de productos, con sus principales alérgenos, para que a la hora de comprar no tengas problema, y sepas con certeza la composición del producto que vas a consumir.

Productos Listados

 <p>Aceite de oliva Virgen extra 500ml La Española Alergenos: Sin alérgenos</p>	 <p>Aceite de oliva Virgen 500ml La Española Alergenos: Sin alérgenos</p>	 <p>Aceite de girasol 500ml La Española Alergenos: Sin alérgenos</p>	 <p>Aceite de oliva Virgen extra 500ml Marca propia Mercadona Alergenos: Sin alérgenos</p>	 <p>Aceite de oliva Virgen extra 500ml Marca propia Carrefour Alergenos: Sin alérgenos</p>
---	---	--	---	--

Cada uno de los productos tiene su propio detalle de producto, aun en una fase muy básica. Por falta de recursos, esta parte de la aplicación web ha quedado en este nivel básico. Es funcional, pero se debe pulir para que sea agradable de cara al usuario.



Y por último el formulario de contacto, que recoge la información del usuario, y la envía a través del servicio SMTP de correo a nuestra dirección de correo de educamadrid para poder recibirlo y ponernos en contacto.

Se utiliza un servicio externo que ofrece EducaMadrid de SMTP. Los correos se envían desde nuestras cuentas de estudiante a una cuenta de gmail creada para el proyecto.

Dinvashop te escucha

Email de contacto

Dinos lo que quieras

Enviar

```

+ ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtener los valores del formulario
    $contacto_usuario = $_POST["email"];
    $mensaje = $_POST["mensaje"];

    // Valores de variables para el envío. Esto se podría poner en un php independiente, como la conexión.
    $mail_to = 'dinvashops@gmail.com'; // El mensaje debe llegar a la cuenta de la tienda
    $mail_from = 'aaron.martin58@educa.madrid.org'; // El mensae se envía desde la cuenta de uno de los dos
    // $mail_from = 'maria.fernandezlozano@educa.madrid.org';
    $mail_name = 'Contacto web Dinvashops'; // Esto es sólo para que el correo recibido sea más "bonito".
    // $mail_user = 'aaron.martin58'; // En educamadrid el usuario es el correo sin @educa.madrid.org
    $mail_user = 'maria.fernandezlozano';

    José Luis López Álvarez, anteayer • Cambiada página contacta.html a contacta.php ...
    $mail_pass = getenv('password'); // Esta es la parte delicada... Hay que sacar esto de un almacén más seguro
    $mail_host = 'smtp01.educa.madrid.org';
    $mail_port = 587;

    $mail = new PHPMailer(true); // Crear correo. El parámetro "true" permite usar try/catch

    try {
        // Datos para el servidor, para la conexión
        $mail->SMTPDebug = 2; // Nivel de debug de la conexión. Poner a 0 para que no salga nada (en pro debe ser 0)
        $mail->isSMTP(); // Enviar usando SMTP
        $mail->Host = $mail_host; // Servidor al que hay que conectar
        $mail->SMTPAuth = true; // Con autenticación (no anónimo)
        $mail->Username = $mail_user; // Usuario
        $mail->Password = $mail_pass; // Password
        $mail->SMTPSecure = 'tls'; // Usar TLS.
        $mail->Port = $mail_port; // Puerto TCP para TLS.

        // Remitente y destinatario
        $mail->setFrom($mail_from, $mail_name);
    }
}

```

7. DESPLIEGUE DE LA APLICACIÓN WEB

Para llevarlo a un entorno lo más similar a una producción real, y aprovechando el conocimiento recientemente adquirido por uno de los integrantes del proyecto en sus prácticas, se decidió montar en contenedores de Kubernetes.

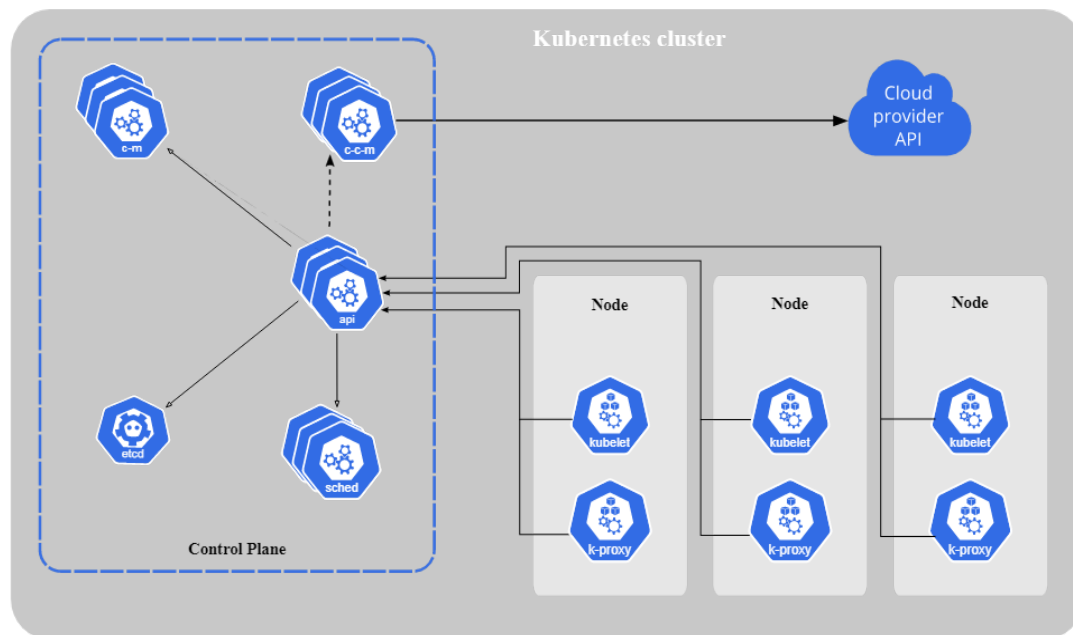
Esta tecnología es bastante nueva, y está en continuo desarrollo. Cada vez más empresas están migrando sus infraestructuras a esta solución, ya que permite escalar los recursos según se vayan necesitando y asegura una portabilidad mayor, así como un menor coste a la empresa. Por ello lo vimos como una oportunidad de ganar conocimiento en este entorno durante

7.1 TECNOLOGÍA.

La virtualización mediante contenedores es una forma de virtualizar recursos diferente a el aprovisionamiento de máquinas virtuales. En vez de virtualizar todos los recursos necesarios para levantar un sistema completo, permite virtualizar los componentes necesarios para nuestros servicios o aplicaciones. De esta manera, se consigue aprovechar mejor los recursos de los que disponemos.

Estos contenedores se organizan en diferentes nodos, que forman un clúster. Existen dos tipos de nodos: master y worker.

Los masters controlan la configuración interna del clúster, y los workers llevan la carga de trabajo de los servicios y las aplicaciones. Aquí es donde se montan los contenedores de las aplicaciones y servicios.



7.2 DESPLIEGUE DEL CLUSTER

Durante el transcurso del proyecto, se ha cambiado el enfoque de este despliegue múltiples veces.

Nuestra primera intención fue aprovechar las cuentas de Azure para montar un cluster con la solución *EKS* de contenedores de Azure. Se llevó a cabo la creación del cluster y primera inicialización, pero según se fue avanzando con la aplicación, nos encontramos con una limitación en las cuentas que no nos permitía crear más de un solo servicio simultáneamente. Ya que nuestra aplicación requiere como mínimo de dos servicios (servidor web y servicio de base de datos), así como de la propia aplicación y base de datos asociada, nos vimos obligados a descartar Azure.

Nuestra siguiente opción fue AWS, de Amazon, que ofrece también un servicio de contenedores; pero de manera similar a Azure, las cuentas gratuitas que podíamos crear estaban altamente limitadas. Además, si se supera ese límite, los precios del servicio son bastante altos, por lo que desechamos la idea nuevamente y pasamos a un despliegue local.

El despliegue local también pasó por varias fases, ya que existen múltiples proveedores open-source de contenedores. Las que más en profundidad se analizaron y probaron fueron Minikube y Microk8s, siendo esta última la elegida finalmente.

Ambas opciones son igual de sencillas, ligeras y rápidas de administrar, pero debido a que minikube requiere de instalar máquinas virtuales en el equipo mediante VirtualBox, nos decantamos por microk8s.

7.2.1 MICROK8S

Microk8s se basa en un servicio que se instala en el sistema operativo, y se encarga de virtualizar sobre el propio sistema, sin necesidad de terceras aplicaciones.

Esta distribución es bastante ligera y compatible con la mayoría de sistemas operativos.

Para poder implantar una aplicación en un contenedor, es necesario “dockerizarla”, es decir, comprimirla en una imagen que tenga todos los componentes necesarios para poder correr en un contenedor. Para ello se utilizó la aplicación Docker.

7.2.2 DOCKER

Docker es una aplicación open-source que permite el almacenamiento de las imágenes que posteriormente se utilizan en los contenedores. Permite crear cuentas para desarrollar nuestras aplicaciones sin coste alguno, además de ofrecer un repositorio donde almacenar dichas imágenes. Docker tiene soporte nativo con Kubernetes, por lo que es compatible con la mayoría de proveedores, exceptuando alguno específico (por ejemplo, RedHat Openshift, que utiliza otros servicios como Podman).

Para dockerizar la aplicación, se crea un fichero Dockerfile sin extensión, en el cual se introducen, en este caso, la imagen base (PHP 7.4-apache), los recursos necesarios de la aplicación (MariaDB), los archivos de trabajo de nuestra aplicación y los puertos que deseamos exponer. El fichero queda de esta manera:

```
# Establece la imagen base con soporte para PHP y Apache
FROM php:7.4-apache
# Instala la extensión MySQL y el cliente de MariaDB
RUN apt-get update
RUN apt-get install -y MariaDB-client
RUN docker-php-ext-install mysqli
RUN rm -rf /var/lib/apt
# Copia los archivos de la aplicación al directorio de trabajo
del servidor web
COPY . /var/www/html/
# Expone el puerto en el que se ejecuta el servidor web
EXPOSE 80
# Define el comando para iniciar el servidor web
CMD ["apache2-foreground"]
```

Una vez creado, se aplica el comando de composición de la aplicación y se sube al repositorio de Docker. De esta manera queda accesible desde cualquier sistema y lugar.

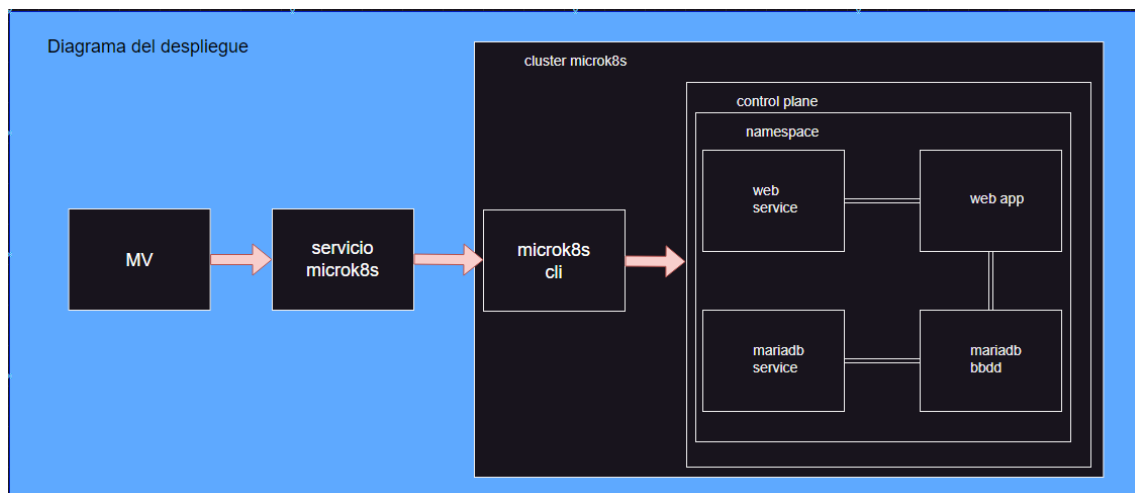
Como se puede apreciar en el fichero, la imagen base de la que está compuesta la aplicación integra PHP y Apache, por lo que ya figuran en el contenedor las dependencias necesarias para levantar este tipo de servicios. Junto a esta base, le añadimos un par de librerías extra para acceder a la base de datos, tanto desde el pod como desde la web.

7.3 CREACIÓN DEL CLUSTER.

Los clústeres de Kubernetes con microk8s funcionan aplicando directamente los ficheros de configuraciones sobre el sistema donde tenemos el servicio de microk8s. Estos ficheros están escritos en formato YAML y contienen la información necesaria para levantar el contenedor.

Existen múltiples objetos de Kubernetes que podemos crear. En nuestro caso, necesitaremos varios deployments, dos servicios y los almacenes de datos de la base de datos. Este último debe ser del tipo PersistentVolume para evitar perder los datos al reiniciar el clúster.

7.3.1 GRÁFICO DE LA ESTRUCTURA



Microk8s levanta un único nodo, al que llama control plane node, que hace las veces de master y worker simultáneamente. Esto es muy útil a la hora de hacer pruebas ya que no requiere demasiados recursos. Es necesario tener instaladas las CLIs de kubectl y docker, además de la de microk8s que se instalan sobre la consola del sistema.

Como en este cluster solo se va a montar una aplicación completa, se utilizará el namespace por defecto. Un namespace es un entorno específico dentro del cluster donde se quiere levantar un servicio determinado, manteniéndose al margen del resto del cluster.

Entraremos en profundidad en este gráfico más adelante.

7.3.2 DEFINICIÓN DE APLICACIONES Y SERVICIOS.

La configuración se hace a través de ficheros YAML.

Estos ficheros contienen la información necesaria para el despliegue de cualquier componente de Kubernetes.

El controlador de recursos del nodo accederá a ellos y los desplegará de la manera especificada.

El deployment de nuestra aplicación web queda de la siguiente manera:

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: dinvashop
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dinvashop
  template:
    metadata:
      labels:
        app: dinvashop
    spec:
      containers:
        - name: dinvashop
          image: dinvashop/dinva-tfg:v2.9.13-beta
          ports:
            - containerPort: 80
          env:
            - name: CREDENCIALES
              valueFrom:
                secretKeyRef:
                  name: mail-pass
                  key: archivo

```

Junto a este deployment, se debe levantar un servicio de ingress para que se pueda acceder a ella desde el navegador.

Esta configuración quedaría así:

```

apiVersion: v1
kind: Service
metadata:
  name: dinvashop
spec:
  selector:
    app: dinvashop
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer

```

Este servicio es del tipo LoadBalancer, que es un objeto de Kubernetes que se encarga de redireccionar el tráfico a las aplicaciones a las que podremos acceder posteriormente. Especificamos el protocolo y el puerto por el cual nos comunicaremos (TCP, 80)

Ahora quedaría definir la base de datos, el servicio de acceso a dicha base de datos y los volúmenes persistentes de la base de datos. Esos ficheros son los siguientes:

Servicio

```
apiVersion: v1
kind: Service
metadata:
  name: mariadb-service
spec:
  selector:
    app: mariadb
  ports:
    - protocol: TCP
      port: 3306
      targetPort: 3306
  type: LoadBalancer
```

Definición del espacio del volumen:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: datos
provisioner: k8s.io/minikube-hostpath
```

Definición del volumen de almacenamiento:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: my-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: datos
  hostPath:
    path: /var/lib/mysql
```

Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mariadb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mariadb
  template:
    metadata:
      labels:
        app: mariadb
    spec:
      containers:
        - name: mariadb
          image: mariadb:latest
          env:
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mariadb-secret
                  key: mariadb-root-password
          volumeMounts:
            - name: mariadb-storage
              mountPath: /var/lib/mysql
      volumes:
        - name: mariadb-storage
          persistentVolumeClaim:
            claimName: mariadb-pvc
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
spec:
  storageClassName: datos
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Junto a estos ficheros de configuración quedaría añadir los ficheros que se encargan de gestionar las contraseñas. Estos ficheros son del tipo Secret, que son archivos con los datos codificados en base64. Estos ficheros se referencian en los deployments que deben acceder a ellos. Se definen los secretos de conexión a la base de datos y contraseñas del mail para el envío de los formularios de contacto.

Base de datos:

```
apiVersion: v1
kind: Secret
metadata:
  name: mariadb-secret
type: Opaque
data:
  mariadb-root-password: VEZHMDEu
```

Mail:

```
apiVersion: v1
kind: Secret
metadata:
  name: mail-pass
type: Opaque
data:
  archivo: TEFJQTA0MTAxNgo=
```

Este tipo de objetos mantienen los datos cifrados para evitar filtrado de información sensible.

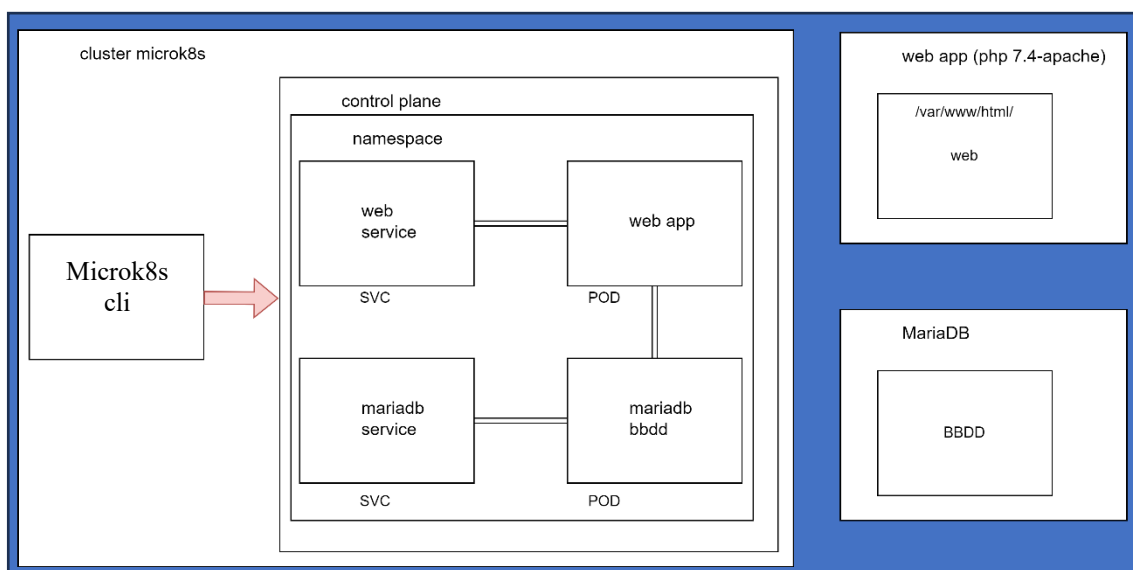
Todos estos archivos están colgados en el GitHub para poder acceder a ellos sin problema

7.3.3 CONEXIÓN CON LA BASE DE DATOS

Una vez creada la infraestructura del cluster, quedaría añadirle los datos a la base de datos.

La aplicación es accesible desde la IP asignada por el web-service. Al estar en local, también es accesible desde localhost ya que este servicio es del tipo LoadBalancer. Este tipo de objetos requieren de un puerto que exponen a través de localhost, que obtenemos haciendo un `kubectl get` del recurso al que queremos acceder. En el caso de la base de datos, debemos conectarnos a ella desde el cliente de mysql/mariadb al puerto 3306, a la ip del pod.

Una vez declarado esto, el gráfico de la aplicación quedaría de la siguiente manera:



El servicio de web se encarga de presentar la aplicación. Dentro del contenedor de la aplicación se encuentran los ficheros php, html y sus correspondientes imágenes que muestra el navegador. Las conexiones a la base de datos desde la web se hacen de manera directa gracias al servicio de la base de datos, que se encarga de que sea accesible desde cualquier punto del namespace en el que nos encontramos.

8. CONCLUSIÓN FINAL.

Como conclusión, creemos que hemos conseguido los objetivos a corto plazo que nos habíamos planteado: creación de la base de datos con los productos y sus relaciones, crear el entorno de la web, dockerizarla y desplegarla en un servicio de Kubernetes.

Si es cierto que se ha logrado de una manera más minimalista, ya que el proyecto lo han desarrollado dos personas de las 4 pensadas en un comienzo, y con conocimientos limitados en el campo del desarrollo web y las aplicaciones web, que hemos tenido que adquirir durante el desarrollo del mismo.

Pero sí que estamos muy orgullosos del resultado que hemos obtenido del despliegue de la aplicación web en Kubernetes; tanto del servicio web como del servicio de base de datos, y nos queda la espinita de no haberlo podido subir a Azure o AWS por nuestras limitaciones en cuanto a presupuesto (ofrecido de manera gratuita por ser estudiantes). Vemos claro que la aplicación, con muchas más horas de creación y desarrollo web y sobre todo con más presupuesto a la hora de garantizar los servicios, podría ser viable.

Como objetivos a futuro quedarían pulir la parte visible de la aplicación, securizar las consultas a la base de datos para evitar inyección SQL, y añadir una personalización específica sobre el usuario, permitiendo guardar sus preferencias y alergias, que podrían ser modificadas a posteriori.

9. BIBLIOGRAFÍA

9.1 LIBROS

María Eugenia Caldas Blanco y María Luisa Hidalgo Ortega (2021) *Empresa e Iniciativa Emprendedora 360°* (Editex)

9.2 WEBGRAFÍA

<https://getbootstrap.com/>

<https://www.w3schools.com/>

<https://docs.docker.com/>

<https://azure.microsoft.com/>

<https://www.youtube.com/watch?v=Gh319dB7pFE>

<https://learn.microsoft.com/en-us/azure/aks/>

<https://docs.aws.amazon.com/eks/index.html>

<https://kubernetes.io/es/docs/home/>

<https://microk8s.io/docs>

<https://www.php.net/manual/es/>

<https://app.diagrams.net/>

<https://www.carrefour.es/>

<https://www.alcampo.es/>

<https://www.lidl.es/>

<https://www.mercadona.es/>

9.3 CONSULTAS ESTADÍSTICAS

[https://unionvegetariana.org/el-veganismo-en-espana-en-cifras-actualizado-en-](https://unionvegetariana.org/el-veganismo-en-espana-en-cifras-actualizado-en-2021/#:~:text=Ahora%20en%20el%20informe%20recién,10%2C8%25%20flexitarianas)

[2021/#:~:text=Ahora%20en%20el%20informe%20recién,10%2C8%25%20flexitarianas](https://unionvegetariana.org/el-veganismo-en-espana-en-cifras-actualizado-en-2021/#:~:text=Ahora%20en%20el%20informe%20recién,10%2C8%25%20flexitarianas)

<https://funsapa.org/alergia-alimentaria/incidencia/#:~:text=En%20el%20mundo%2C%20cerca%20de,en%20los%20%C3%BAltimos%2010%20a%C3%B1os>

9.4 EJEMPLOS Y ANÁLISIS DE LA COMPETENCIA

<https://www.mercadona.es/>

<https://www.nike.com/es/>