



**Escuela Superior  
de Ingeniería y Tecnología**  
Universidad de La Laguna

## Trabajo de Fin de Grado

---

Sistema de información para la  
recopilación y centralización de  
información sobre productos alimenticios.

*Information system for the collection and centralisation of  
information on food products.*

Eduardo Nuez Wehbe

---

La Laguna, 8 de septiembre de 2021

D. **Eduardo Manuel Segredo González**, con N.I.F. 78.564.242-Z profesor Ayudante Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Gara Miranda Valladares**, con N.I.F. 78.563.584-T profesora Titular de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutora

### **C E R T I F I C A ( N )**

Que la presente memoria titulada:

*“Sistema de información para la recopilación y centralización de información sobre productos alimenticios”*

ha sido realizada bajo su dirección por D. **Eduardo Nuez Wehbe**, con N.I.F. 51.147.243-G

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 8 de septiembre de 2021

# Agradecimientos

Quisiera agradecer a mi tutores de TFG, Eduardo Segredo González y Gara Miranda Valladares por todos sus consejos y apoyo que han hecho posible realizar este trabajo. Además agradecer el apoyo de mi familia y amigos que me han acompañado durante estos cuatro años que ha durado el grado y sin ellos no estaría hoy aquí.

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-  
NoComercial-SinObraDerivada 4.0 Internacional.

## **Resumen**

*Actualmente en el mercado existen multitud de aplicaciones relacionadas con el mundo de la alimentación, entre estas se destacan herramientas que velan por la nutrición, planificando las comidas diarias del cliente, bases de datos de composición de alimentos, aplicaciones móviles como Yuka o plataformas para comparar precios de productos entre distintos supermercados.*

*Este tipo de sistemas están creados para un uso específico, por ejemplo, si un usuario está comparando precios de productos en una plataforma, por regla general, esta no tendrá los valores nutricionales del alimento, por lo que si este necesita obtener dicha información deberá acceder a otra aplicación dedicada a este ámbito.*

*Por esta razón el objetivo de este Trabajo de Fin de Grado ha sido el desarrollo de una aplicación web que permita la recopilación y centralización de productos de alimentación con su respectiva información nutricional y su precio e integrar un sistema de E-Commerce donde los pequeños productores puedan promocionar sus productos de manera gratuita.*

*Es por ello que para un correcto funcionamiento de la aplicación, se ha implementado una base de datos no relacional MongoDB, una aplicación web de usuario con Quasar, un servidor API REST hecho con Express.js, un módulo de python para realizar web scraping y un despliegue cloud en Digital Ocean utilizando Docker.*

*El resultado de lo anteriormente descrito ha concluido en la implementación de una plataforma web que agrupa distintas funcionalidades de herramientas enfocadas al sector de la alimentación y un programa para la obtención de precios de productos alimenticios haciendo uso de Bases de Datos de Composición de Alimentos (BDCA) y software de Web Scraping.*

**Palabras clave:** E-Commerce, nutrición, BDCA, Web Scraping, Quasar, MongoDB, Digital Ocean, Docker, API REST, Express.js

## **Abstract**

*Currently on the market there are many applications related to the world of food, among these are tools that ensure nutrition, planning the customer's daily meals, databases of food composition, mobile applications such as Yuka or platforms to compare prices of products between different supermarkets.*

*These types of systems are created for a specific use, for example, if a user is comparing prices of products on a platform, as a general rule, this will not have the nutritional values of the food, so if he/she needs to obtain this information, he/she will have to access another application dedicated to this area.*

*For this reason, the objective of this Final Degree Project has been the development of a web application that allows the collection and centralisation of food products with their respective nutritional information and price and to integrate an E-Commerce system where small producers can promote their products free of charge.*

*That is why for the correct functioning of the application, a non-relational database MongoDB, a user web application with Quasar, a REST API server made with Express.js, a Python module for web scraping and a cloud deployment in Digital Ocean using Docker have been implemented.*

*The result of the above has resulted in the implementation of a web platform that brings together different functionalities of tools focused on the food sector and a software for obtaining prices of food products using Food Composition Databases (FCD) and Web Scraping software.*

**Keywords:** E-Commerce, nutrition, FCD, Web Scraping, Quasar, MongoDB, Digital Ocean, Docker, API REST, Express.js

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivo del proyecto . . . . .	1
1.2. Planificación . . . . .	2
<b>2. Estado del arte</b>	<b>3</b>
2.1. Alimentación y composición de alimentos . . . . .	3
2.1.1. Nutrientes . . . . .	3
2.1.2. Macronutrientes . . . . .	4
2.1.3. Micronutrientes . . . . .	5
2.2. Bases de datos y aplicaciones de nutrición . . . . .	5
2.2.1. Bases de datos . . . . .	5
2.2.1.1. BEDCA . . . . .	6
2.2.1.2. Open Food Facts . . . . .	6
2.2.1.3. EuroFIR . . . . .	7
2.2.1.4. U.S. Department of Agriculture . . . . .	7
2.2.1.5. Agriculture and Food System Institute . . . . .	8
2.2.1.6. INFOODS . . . . .	8
2.2.2. Apps Nutricionales . . . . .	8
2.2.2.1. CalorieKing . . . . .	9
2.2.2.2. Edaman . . . . .	9
2.2.2.3. Oorenji . . . . .	10
2.3. Herramientas para la obtención de precios . . . . .	11
2.3.1. Comparadores de precio . . . . .	11
2.3.1.1. Radarprice . . . . .	11
2.3.1.2. Ofertia . . . . .	12
2.3.1.3. Carritus . . . . .	12
2.3.1.4. Out of Milk . . . . .	13
2.3.1.5. Soysuper . . . . .	13
2.3.2. Software de Web Scraping . . . . .	14
2.3.2.1. Webscraper.io y Data-miner . . . . .	14
2.3.2.2. Parsehub . . . . .	15
2.4. Comparativa de datos . . . . .	16
<b>3. Tecnologías y Arquitectura</b>	<b>18</b>
3.1. Análisis de Tecnologías . . . . .	18
3.1.1. Backend . . . . .	18
3.1.2. Frontend . . . . .	18
3.1.3. Web scraping . . . . .	19

3.1.4. Bases de datos . . . . .	20
3.2. Tecnologías utilizadas . . . . .	20
3.2.1. Lenguajes de programación . . . . .	20
3.2.2. Frameworks . . . . .	20
3.2.3. Base de datos . . . . .	21
3.2.4. Web Scraper . . . . .	21
3.2.5. Librerías . . . . .	21
3.3. Diseño de la arquitectura . . . . .	21
3.3.1. Inyectores de datos . . . . .	23
3.3.1.1. BEDCA . . . . .	23
3.3.1.2. Scrappers . . . . .	23
3.3.1.3. OPF . . . . .	23
3.3.2. MongoDB Atlas . . . . .	23
3.3.2.1. User . . . . .	24
3.3.2.2. Role . . . . .	24
3.3.2.3. Enterprise . . . . .	24
3.3.2.4. ProductCategory . . . . .	24
3.3.2.5. BEDCA . . . . .	25
3.3.2.6. PriceProduct . . . . .	25
3.3.3. Backend y frontend . . . . .	25
3.3.4. Despliegue en la nube . . . . .	25
<b>4. SOFN (Spansih Open Food Network)</b>	<b>26</b>
4.1. Descripción . . . . .	26
4.2. Implementación de la herramienta . . . . .	26
4.2.1. Bases del proyecto . . . . .	27
4.2.2. Módulo de recopilación y centralización de alimentos . . . . .	27
4.2.2.1. Autenticación de usuarios . . . . .	27
4.2.2.2. Datos de OPEN FOOD FACTS . . . . .	28
4.2.2.3. Datos de BEDCA . . . . .	29
4.2.3. Módulo de E-Commerce . . . . .	31
4.2.3.1. Controladores . . . . .	31
4.2.3.2. Dashboard de la Tienda . . . . .	32
4.2.3.3. Acceso a Tiendas . . . . .	33
4.2.4. Módulo de Web Scraping . . . . .	34
4.2.4.1. Obtención de datos . . . . .	34
4.2.4.2. Análisis de los comercios . . . . .	35
4.2.4.3. Implementación de scrapers . . . . .	35
4.2.4.4. Precios de OPF . . . . .	36
<b>5. Conclusiones y futuro del proyecto</b>	<b>38</b>
5.1. Resultados finales . . . . .	38
5.2. Líneas futuras . . . . .	38
<b>6. conclusions and future of the project</b>	<b>40</b>
6.1. Final results . . . . .	40
6.2. Future lines . . . . .	40
<b>7. Presupuesto</b>	<b>42</b>



# Índice de Figuras

1.1. Diagrama de Gantt de la planificación del proyecto . . . . .	2
2.1. Plataforma de BEDCA . . . . .	6
2.2. Vista de Open Food Facts desde el móvil . . . . .	7
2.3. Buscador de productos del Departamento de agricultura de EEUU . . . . .	8
2.4. Página web de Calorie King, presentación de producto . . . . .	9
2.5. Plataforma web de edaman . . . . .	10
2.6. Plataforma web de edaman . . . . .	11
2.7. Aplicación móvil Radarprice . . . . .	12
2.8. Comparativa de la lista de la compra en diferentes tiendas . . . . .	13
2.9. Web Soysuper . . . . .	14
2.10 Web Scraper . . . . .	15
2.11 Ejemplo de uso ParseHub . . . . .	16
3.1. Arquitectura del sistema . . . . .	22
4.1. Logotipo de SOFN . . . . .	26
4.2. Página de registro de SOFN . . . . .	27
4.3. Middleware de autenticación . . . . .	28
4.4. Petición para obtener datos de OPF . . . . .	29
4.5. Selección de los datos del CSV de BEDCA . . . . .	30
4.6. Renombrado de las columnas del CSV de BEDCA . . . . .	30
4.7. Fichero de rutas de la colección Enterprise . . . . .	31
4.8. Perfil de la tienda . . . . .	32
4.9. Productos de la tienda . . . . .	33
4.10 Lista de las tiendas disponibles . . . . .	34

# Índice de Tablas

2.1. Comparativa entre las BDCAs más completas . . . . .	16
2.2. Comparativa entre las Aplicaciones analizadas . . . . .	16
2.3. Comparativa entre los comparadores de precio . . . . .	17
7.1. Presupuesto tecnológico . . . . .	42
7.2. Presupuesto Total . . . . .	43

# Capítulo 1

## Introducción

Las empresas tratan de buscar el desempeño más óptimo para su negocio. En el caso de las empresas dedicadas al sector alimenticio, encuentran este punto óptimo en dos vertientes: la calidad nutricional de sus productos y su precio.

La calidad nutricional es de notable relevancia cuando hablamos de nuestra salud. Acorde a la Organización Mundial de la Salud (OMS) [53] *“llevar a cabo una dieta saludable y equilibrada a lo largo de la vida ayuda a prevenir la malnutrición, así como enfermedades no transmisibles (hipertensión, enfermedades cardiovasculares, diabetes, entre otros trastornos)”*.

Es por ello que consideramos fundamental prestar especial atención a los valores nutricionales de los alimentos junto a sus precios, y en este TFG proponemos diseñar e implementar herramientas informáticas que den soporte a la planificación y evaluación de los alimentos que se quieran adquirir, a la par de mejorar las relaciones entre los productores y distribuidores.

Esta mejora proviene de unas herramientas que permiten informar con eficiencia acerca de la nutrición y permiten la comparación de precios entre los productos. Existen multitud de herramientas informáticas orientadas a estos conceptos y en su mayoría son aplicaciones web o móviles que muestran o bien la información de nutrientes, u ofrecen datos acorde al precio de los alimentos a modo de comparación entre los distribuidores o herramientas enfocadas en el comercio electrónico.

Tras una búsqueda intensiva se ha detectado la ausencia de una plataforma que ofrezca estos servicios de manera centralizada. Esto ha motivado el desarrollo de una aplicación que integre todas estas funcionalidades en una sola plataforma.

### 1.1. Objetivo del proyecto

El objetivo principal del proyecto es diseñar e implementar una aplicación donde los usuarios puedan obtener información nutricional de productos de alimentación con sus precios de los supermercados españoles y además, proveer de un sistema de comercio donde los usuarios puedan crear tiendas y vender sus productos.

Por lo tanto, los objetivos que se han valorado para el desarrollo del TFG son los siguientes:

- Módulo de recopilación y centralización de productos de alimentación junto con su información nutricional, mejorando los sistemas actuales de presentación de los datos.

- Módulo de E-Commerce que proporciona un espacio a los compradores y productores, donde dar de alta sus productos para promocionarlos y comercializarlos.
- Módulo de Web Scraping para la obtención de precios de los productos del primer módulo.

## 1.2. Planificación

El desarrollo del proyecto se ha dividido en las siguientes tareas:

	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
<i>Análisis del estado del arte</i>						
<i>Estudio del web scraping y sus aplicaciones</i>						
<i>Análisis de las herramientas de desarrollo full stack actuales</i>						
<i>Diseño e implementación de la herramienta</i>						
<i>Despliegue de la aplicación en un servicio nube</i>						
<i>Documentación</i>						

Trabajo de 7 horas diarias  
(incluyendo fines de semanas)

Figura 1.1: Diagrama de Gantt de la planificación del proyecto

# Capítulo 2

## Estado del arte

Se ha realizado un análisis y estudio de las diferentes herramientas: apps y bases de datos relacionadas con el campo de la nutrición, venta de productos alimentarios, **APIs** [72] y derivados de estos. Los campos que se han analizado son los siguientes:

- Alimentación y composición de alimentos.
- Bases de datos de composición de alimentos y aplicaciones de nutrición.
- Adquisición de productos y precios.

### 2.1. Alimentación y composición de alimentos

La alimentación es la actividad por la que obtenemos nutrientes a partir de los alimentos. Nuestro organismo los descompone para que puedan viajar por la sangre y llegar hasta las células. Allí se metabolizan, proporcionando la energía que el cuerpo necesita para vivir. La alimentación es una necesidad básica en nuestra vida diaria, y aunque responda a la necesidad vital de nutrirnos, se debe realizar de manera consciente. Por lo que entender qué son los nutrientes es importante para elegir qué comer [34].

Es por ello que es importante comprender la información nutricional de los productos para limitar el consumo de determinados nutrientes que ingeridos en exceso pueden ser perjudiciales para nuestra salud y consumir en mayores cantidades de aquellos que la favorecen. Por estos motivos, analizaremos qué son los nutrientes y su importancia.

#### 2.1.1. Nutrientes

Los nutrientes son sustancias presentes en los alimentos con propiedades específicas sobre el funcionamiento de nuestro organismo, entre ellas para obtener energía. Mediante el proceso de digestión de los alimentos que comemos, se obtienen los nutrientes y se transforman. Existen diversos tipos, por eso la clasificación de los nutrientes se basa en las funciones que estos realizan.

El cuerpo es capaz de fabricar algunos nutrientes a partir de otros, haciendo pequeños cambios o combinándolos para crear otros nuevos. Sin embargo, otros nutrientes imprescindibles no los podemos fabricar y se deben aportar con la alimentación. Estos se denominan nutrientes esenciales.

Estos nutrientes se dividen a su vez en dos grupos: los micronutrientes y los macronutrientes. Mientras los primeros, también conocidos como vitaminas y minerales, trabajan

para asegurar el buen funcionamiento del organismo, los macronutrientes son aquellas sustancias que proporcionan energía, ayudan a reparar y construir las estructuras orgánicas, promueven el crecimiento y regulan los procesos metabólicos, entre otras funciones.

El agua y la fibra no son propiamente nutrientes, pero son necesarios en cantidades adecuadas.

La principal función que han de tener los nutrientes es cubrir tres tipos de necesidades: las energéticas, las estructurales, y las funcionales y reguladoras. Veamos con más detalle los macronutrientes y micronutrientes y qué funciones realizan cada uno de ellos.

### 2.1.2. Macronutrientes

Los macronutrientes son aquellas sustancias que proporcionan energía al organismo para un buen funcionamiento y otros elementos necesarios para reparar y construir estructuras orgánicas, promover el crecimiento y regular procesos metabólicos.

Este grupo está constituido por:

**Proteínas:** Son los componentes de las estructuras de las células. Las necesidades de un adulto sano y sedentario son 0,8-1 g/kg/día. Nos aportan 4 kcalorías/gramo. Se recomienda entre un 10-15 % del total de las calorías de la dieta en forma de proteínas en una dieta equilibrada.

**Grasas:** Son el nutriente energético por excelencia. Este grupo está formado por los triglicéridos, fosfolípidos y colesterol. Constituyen el 30-35 % de una dieta equilibrada. Aportan 9 kcal/gramo.

Los ácidos grasos se clasifican en 3 categorías:

- Grasa saturada (carnes, lácteos, mantequillas).
- Grasas mono y poliinsaturadas (Aceites de oliva, linaza, germen de trigo, pescados, frutos secos y aguacates).
- Grasas trans (Bollerías, frituras y algunos tipos de margarina).

**Hidratos de carbono:** Son una fuente importante de energía y proceden fundamentalmente de los vegetales. La ingesta en un adulto sano recomendada es de 3-5 g/kg/día (se recomienda que no superen del 50 % de la ingesta calórica diaria). Aportan 4 kcal/gramo.

Son los mayores proveedores de energía en términos de actividad física y mental. Los monosacáridos (o azúcares simples) son la piedra angular de esta categoría. Los carbohidratos se dividen según la longitud de sus cadenas en los siguientes grupos:

- Monosacáridos: glucosa (azúcar de maíz), fructosa (azúcar de la fruta) y galactosa (azúcar de la leche)
- Disacáridos: sacarosa y lactosa
- Oligosacáridos: melitosa
- Polisacáridos (o carbohidratos complejos): amilopectina (almidón vegetal), glucógeno (almidón animal), inulina

### 2.1.3. Micronutrientes

Los micronutrientes son sustancias que no aportan energía pero son esenciales para el buen funcionamiento de nuestro organismo.

En este grupo encontramos:

**Vitaminas:** Participan en numerosas funciones vitales del organismo. Son sustancias esenciales para el metabolismo, el desarrollo y el crecimiento normal, además de participar en el proceso de regulación del funcionamiento de las células. Todas las vitaminas se obtienen de los alimentos y, asimismo, el organismo es capaz de sintetizar la vitamina D y la vitamina K en pequeñas cantidades, que son insuficientes para sus necesidades.

Las 13 vitaminas que se conocen están agrupadas en los siguientes grupos:

- Vitamina A
- Vitamina B
- Vitamina C
- Vitamina D
- Vitamina E
- Vitamina K

#### **Minerales**

Son elementos naturales no orgánicos que representan entre el 4 y el 5 por ciento del peso corporal del organismo y que están clasificados en macrominerales y oligoelementos. El ser humano los necesita para mantener el buen funcionamiento del cuerpo y garantizar, entre otros, la formación de los huesos, la regulación del ritmo cardiaco y la producción de las hormonas.

Existen 2 tipos de minerales:

- **Macrominerales:** son necesarios para que las funciones del organismo se desarrollen con normalidad (algunos ejemplos son: calcio, magnesio, fósforo, potasio, azufre, cloro y sodio)
- **Oligoelementos:** son sustancias químicas primordiales para el buen funcionamiento del organismo y que intervienen en el metabolismo (algunos ejemplos son: hierro, manganeso, selenio, cobre, yodo, cobalto, cinc y flúor).

## 2.2. Bases de datos y aplicaciones de nutrición

### 2.2.1. Bases de datos

Las Bases de Datos de Composición de Alimentos (BDCAs) [32] son colecciones de datos sobre contenido nutricional, las cuales derivan de análisis cuantitativos de muestras representativas de alimentos. Las BDCAs tienen múltiples usos: el análisis de nutrientes de los alimentos a partir de encuestas de consumo dietético, etiquetado nutricional e informar de políticas agrícolas sensibles a la nutrición.

La Organización para la Agricultura y la Alimentación (FAO) [43] es el coordinador global de la Red Internacional de Sistemas de Datos Alimentarios (INFOODS) [20],

la cual dispone de un directorio de BDCAs nacionales, regionales e internacionales que pueden ser útiles para aquellos interesados en analizar alimentos y nutrientes, conocer su disponibilidad y consumo, programas de enriquecimiento o suplementación de alimentos. Además, FAO e INFOODS producen directrices para desarrollar BDCAs, emparejar alimentos y convertir datos alimentarios.

Por tanto, realizar un análisis exhaustivo de las BDCAs proporciona una información crucial para el desarrollo de un modelo de datos. Es por ello que a continuación, se detallan las más importantes y sus características principales.

### 2.2.1.1. BEDCA

La Base de Datos Española de Composición de Alimentos (BEDCA) [3] es una red de Centros de investigación públicos, Administración e Instituciones privadas cuyo objetivo es el desarrollo y mantenimiento del contenido de la base de datos.

Actualmente esta base de datos tiene una pequeña cantidad de productos 955, siendo todos de carácter agrícola y con una gran descripción de los valores nutricionales, entre los cuales, aparecen algunos tan inusuales en este tipo de bases de datos como: niacina, riboflavina entre otros.



The screenshot shows the BEDCA website interface. At the top, there is a header with the BEDCA logo, the text "Base de Datos Española de Composición de Alimentos" and "Spanish Food Composition Database", and navigation buttons for "Inicio", "Fuentes", and "Consulta". Below the header, there is a navigation bar with "Página principal" and "Menú de Consulta". A "Consulta anterior" button is visible above the main content area. The main content area is titled "LISTADO DE ALIMENTOS DE LA CONSULTA" and contains a table with the following data:

Id	Nombre	Name
1015	Agua, con gas, embotellada	Sparkling water, bottled
1022	Anís, seco	Anís spirit
681	Café, con leche	Coffee infusion, with milk
1019	Café, en grano o molido, descafeinado	Coffee, powder, decaffeinated
1023	Café, en polvo, soluble	Soluble coffe, decaffeinated, powder
1108	Cerveza, oscura, 8°- 9°	Beer, brown
1107	Coñac	Conyac
1106	Néctar de albaricoque, envasado	Apricot nectar
1027	Néctar de ciruela	Plum nectar
1057	Néctar de frutas exóticas, envasado	Tropical fruit nectar
1061	Néctar de mango, envasado	Mango nectar
1105	Néctar de maracuyá, envasado	Maracuya nectar
1104	Néctar de melocotón	Peach nectar
1103	Néctar de naranja, envasado	Orange nectar
1101	Néctar de pera, envasado	Pear nectar

Figura 2.1: Plataforma de BEDCA

### 2.2.1.2. Open Food Facts

Open Food Facts (OPF) [15] es una base de datos libre sin ánimo de lucro, de colaboración abierta, la cual recopila información de productos alimenticios de todo el mundo. Aparte de la colaboración voluntaria de los usuarios, OPF utiliza una serie de aplicaciones



que aportan datos adicionales a la base de datos entre la que se destaca Yuka [73] como la más importante.

Asimismo OPF ostenta un registro de más de 200.000 productos españoles, indicando la marca, los datos nutricionales más importantes y algunas características diferenciadoras como el país de venta, envases empleados, etiquetas o certificaciones, ingredientes y una larga lista de datos adicionales (entre los cuales no indica el precio).

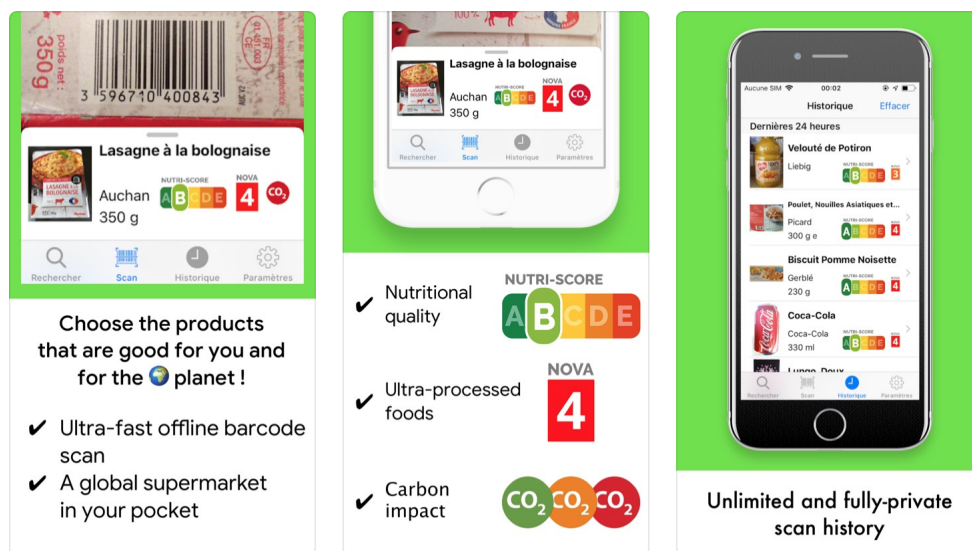


Figura 2.2: Vista de Open Food Facts desde el móvil

### 2.2.1.3. EuroFIR

La European Food Information Resource [13] es una asociación internacional sin ánimo de lucro que apoya el uso de los datos de composición de alimentos existentes y los recursos futuros a través de la cooperación y armonización de la calidad de los datos, la funcionalidad y los estándares globales.

El propósito de la asociación es el desarrollo, la gestión, la publicación y la explotación de datos de composición de alimentos y la promoción de la cooperación y armonización internacionales a través de la mejora de la calidad de los datos, la capacidad de búsqueda de bases de datos de composición de alimentos y las normas.

Para acceder a su API se requiere de un registro en el sistema. Para ello, se debe realizar un pago a través de su página oficial donde se especifique el motivo del uso. Como cliente se puede utilizar como buscador de alimentos, sin embargo tiene poca cantidad de productos españoles.

### 2.2.1.4. U.S. Department of Agriculture

Su propósito es desarrollar y ejecutar políticas de ganadería, agricultura y alimentación. Su meta es entender las necesidades de los productores, promoviendo el comercio agrícola y la producción, trabajando para asegurar seguridad alimentaria, protegiendo los recursos naturales, mejorar las comunidades rurales, y poner fin al hambre [66].

La información nutricional de sus productos es muy completa y con una gran variedad de tipos de alimentos, desde ultraprocesados hasta los más ecológicos. Se caracteriza porque sus productos son exclusivamente americanos.

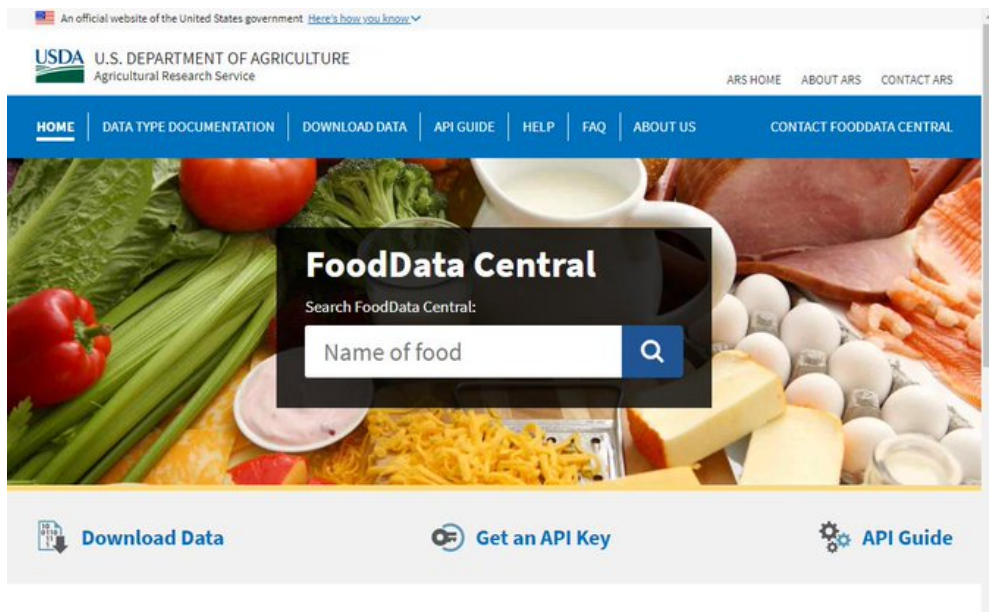


Figura 2.3: Buscador de productos del Departamento de agricultura de EEUU

#### 2.2.1.5. Agriculture and Food System Institute

El AFSI [22] es una organización independiente sin ánimo de lucro que promueve la ciencia para el beneficio público. Su objetivo es lograr sistemas agroalimentarios, sanitarios y medioambientales seguros y sostenibles. Utilizan la investigación aplicada, el desarrollo de capacidades, la educación, las bases de datos y la divulgación para lograr su objetivo.

Tiene un buscador de BDCAs donde se filtra según el país y muestran las bases de datos oficiales.

#### 2.2.1.6. INFOODS

La Red Internacional de Sistemas de Datos sobre Alimentos [20] es una red mundial de expertos especializados en composición de los alimentos cuyo fin es mejorar la calidad, la disponibilidad, la fiabilidad y la utilización de los datos de composición de alimentos. También es un foro mediante el que se pueden realizar y promover actividades internacionales de armonización y respaldo de la composición de los alimentos. Está organizada en varios centros de datos regionales y dispone de un coordinador mundial. En este contexto, INFOODS proporciona lineamientos, estándares, bases de datos, herramientas para el desarrollo de capacidades, asesoramiento sobre políticas y asistencia técnica a nivel de país.

Ofrece una serie de documentos por país en los que aparecen los productos más consumidos con sus componentes alimenticios. Estos documentos se actualizan anualmente.

### 2.2.2. Apps Nutricionales

Este tipo de aplicaciones nos ayudan a cambiar nuestro estilo de vida a uno más saludable. Ofrecen desde dietas específicas para los usuarios, a recomendaciones de productos, pasando por recetas, lugares de compra, etc.

Al ser herramientas con características análogas a este proyecto es de relevancia el

análisis de las mismas. En este caso, se destaca la presentación de los datos nutricionales a la par que los productos sugeridos.

### 2.2.2.1. CalorieKing

Aplicación web que ofrece servicios enfocados a la educación, motivación e inspiración de estilos de vida saludables, cuyo objetivo consiste en lograr resultados positivos para los miembros y usuarios que usan la plataforma. Con este fin, los softwares y herramientas que se encuentran disponibles dentro de esta web están enfocados para el uso diario, ayudando así a cualquier usuario a cambiar su estilo de vida a uno más saludable [5].

Un componente importante de los servicios es el de la BDCA con una información nutricional muy completa.

Como empresa están muy enfocados en ayudar con sus productos a resolver los problemas de la obesidad, mediante programas clínicos y centros de salud.

Posee una API privada de productos de alimentación la cual requiere de una suscripción con pagos mensuales.

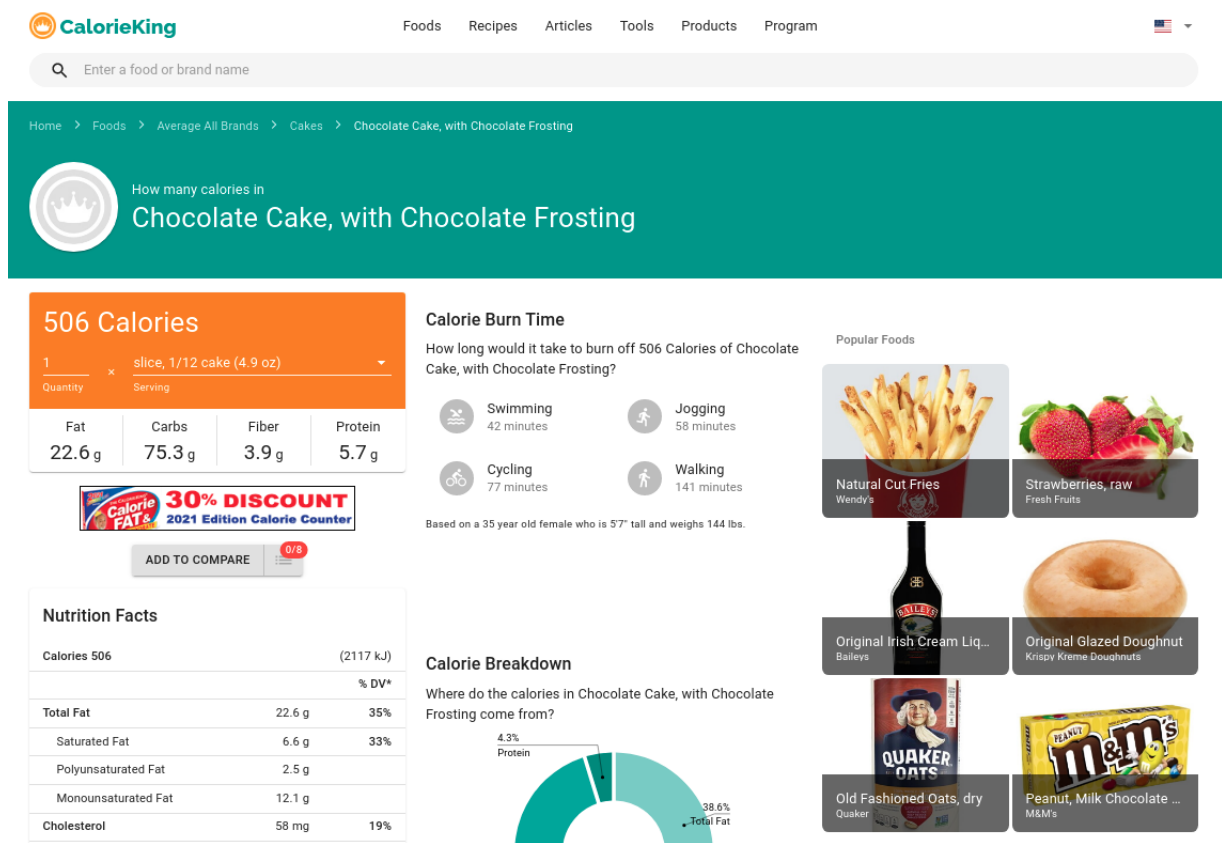


Figura 2.4: Página web de Calorie King, presentación de producto

### 2.2.2.2. Edaman

Ofrece servicios de nutrición y alimentación y alberga distintas bases de datos donde se engloban la búsqueda de productos, recetas y análisis nutricionales [12].

Tiene un identificador de productos, donde los usuarios introducen los alimentos que consumen a diario y esta les dará una serie de recomendaciones alimenticias que pueden seguir para mejorar su calidad de vida.

También agrega un módulo de recetas, que se caracteriza por la participación de los usuarios compartiendo sus recetas, creando una base de datos que se retroalimenta, así como una API con opción gratuita y de pago. Siendo la gratuita una versión limitada que no permite acceder a todo el contenido.

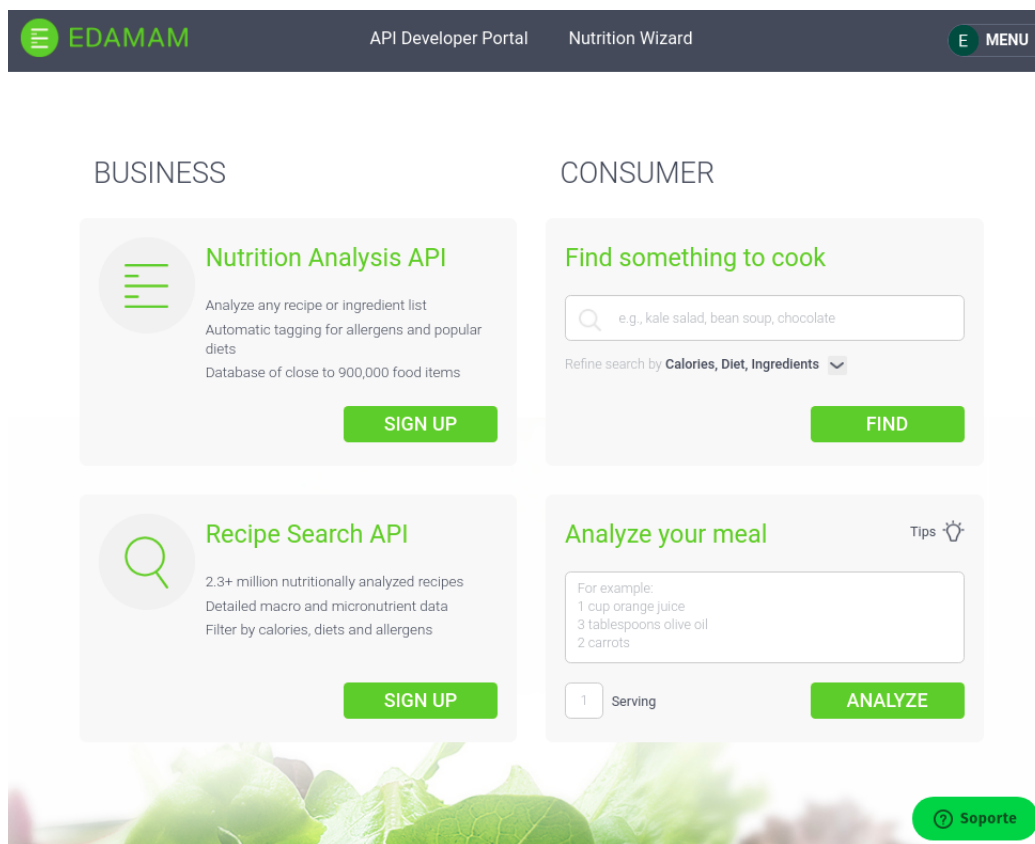


Figura 2.5: Plataforma web de edaman

### 2.2.2.3. Oorenji

Esta aplicación permite configurar el perfil de cada usuario, a partir del cual ofrecerá una gran variedad de servicios y consejos especializados para que los usuarios puedan mejorar su alimentación [41].

Algunos servicios ofrecidos por Oorenji son las dietas personalizadas, los planes nutricionales semanales o diarios adaptados a las necesidades específicas del usuario y una gran variedad de recetas profesionales para cada día de la semana.

Ostenta una funcionalidad muy diferenciadora, donde permite realizar un análisis genético, basado en una toma de saliva, con el cual se integrará información muy útil sobre el usuario al sistema de inteligencia artificial de la aplicación para recibir un servicio aún más personalizado.

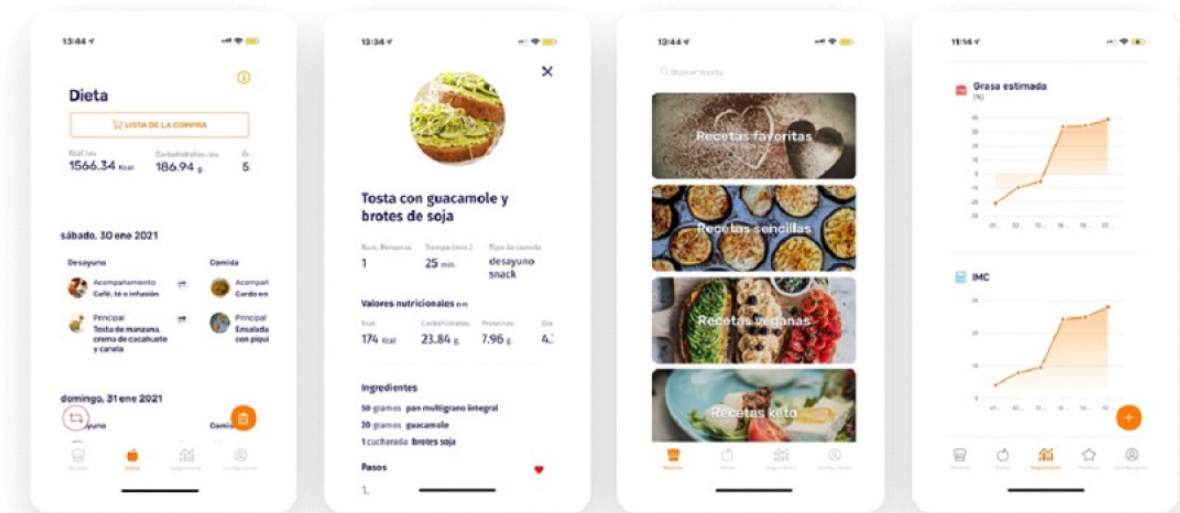


Figura 2.6: Plataforma web de edaman

## 2.3. Herramientas para la obtención de precios

### 2.3.1. Comparadores de precio

Se ha llevado a cabo una investigación de las distintas maneras de obtener los precios de los productos que se encuentran en las BDCAs. Como estos datos no son publicados por la inmensa mayoría de compañías, se ha realizado un estudio de las aplicaciones de precios que existen en el mercado y ver cómo estas acceden a los precios de los productos.

#### 2.3.1.1. Radarprice

RadarPrice [51] es una aplicación disponible para dispositivo móvil, en la cual escaneando el código de barras de cualquier producto indica en qué tienda se puede encontrar más barato, es decir, muestra al usuario en tiempo real una comparativa de precios por tienda para que el usuario pueda elegir dónde comprarlo.

Esta aplicación se nutre de los usuarios para tener una base de datos completa puesto que, al escanear el código de barras del producto este queda registrado y lo busca en otros supermercados para encontrarlo a mejor precio, siendo por ello completamente dependiente del número de usuarios que tenga la aplicación para su correcto funcionamiento.



Figura 2.7: Aplicación móvil Radarprice

### 2.3.1.2. Ofertia

Es un portal que ayuda a los usuarios a preparar sus compras, digitalizando, categorizando y geocalizando todos los folletos de ofertas para que estos puedan beneficiarse de los mejores descuentos. Su objetivo es la preparación y planificación de la compra y el ahorro familiar [40]. De modo que tiene una base de datos en la que almacena los productos que se encuentran en los folletos realizando así la comparativa de precios.

### 2.3.1.3. Carritus

Carritus [6] es un portal que permite comparar y comprar de forma online en cualquiera de los supermercados próximos mediante el código postal y tiene como objetivo conseguir el mejor precio para la cesta de la compra de los clientes, así como ofrecer una manera más sencilla de realizar la compra.

Dentro de la aplicación no especifican cómo obtienen los datos de los productos ni de los precios actualizados.

The screenshot shows the Carritus.com interface for comparing prices across four supermarkets: Mercadona, Caprabo, El Corte Inglés, and Eroski. The total price for each store is displayed at the top, along with the number of items and automatic substitutions. Below, the items are listed in columns, showing their prices and availability at each store.

Tienda	Total Precio	Envío no incluido	Formatos diferentes
MERCADONA	135,83€	(7,21€)	2
caprabo	115,64€	(4,00€)	20
El Corte Inglés	108,65€	(8,20€)	19
EROSKI	114,11€	(5,89€)	13

Items listed include: Carnicería y pescadería (e.g., Jovi 1x292g Escalope pollo fresco emp, Jovi 1x275g Nuggets pollo fresco de 2), Arroz, pastas, harinas y legumbres (e.g., Hacendado 1x560g Pizza fresca 4 quesos fam, Hacendado 1x580g Pizza fresca jamon york m).

Figura 2.8: Comparativa de la lista de la compra en diferentes tiendas

### 2.3.1.4. Out of Milk

Out of Milk [29] es una aplicación para ayudar al usuario a realizar un seguimiento de sus necesidades de compra, el inventario de su despensa y a administrar los elementos de la lista de la compra.

Los usuarios suben el código de barras de los productos que desean tener en su lista de la compra y la aplicación los registra, aunque no presenta una base de datos donde acceder a un alimento en concreto.

El servicio diferenciador que ofrece la aplicación son las recetas que se encuentran disponibles en la aplicación. Contienen una gran variedad de estas y permite que el usuario pueda guardarlas.

### 2.3.1.5. Soysuper

Es una plataforma online que permite comparar los precios entre los principales supermercados y enviar la compra [58], utilizando la ubicación actual en la que se encuentra el usuario para localizar los supermercados próximos, de modo que permite visualizar cualquier producto y realizar la comparativa del mismo entre los supermercados análogos.

Tiene una gran variedad de supermercados notorios a nivel nacional donde poder comparar los diferentes precios de los productos.

The screenshot shows the Soysuper website interface. At the top, there is a search bar with 'chocolate' entered and a 'Buscar' button. To the right, there are links for 'Crear cuenta', 'Entrar', and a shopping cart icon showing '12,32 €'. Below the search bar, the main content area is titled 'Tu cesta en Alcampo' and contains a list of four chocolate products, each with a quantity selector (minus, plus, and a central number box). The products are: Valor Chocolate puro (2,83 € / 300g), Intermón Oxfam Chocolate blanco (1,85 € / 100g), Lacasa Chocolate negro 70% cacao (1,73 € / 100g), and Lacasitos Lacasa Grajeas de chocolate blanco (1,97 € / 150g). At the bottom of the cart, it says 'Total sin gastos de envío 12,32 €'. On the left side, there is a 'Buscar por categorías' section with a list of categories and their respective product counts. On the right side, there is a 'Tu cesta en otros supers' section showing similar offers from Coprin, Condis, and Condis.

Figura 2.9: Web Soysuper

## 2.3.2. Software de Web Scraping

Una alternativa para obtener los precios de los productos es utilizar herramientas de Web Scraping [70]. Estas permiten la obtención de datos mediante la ejecución de acciones que el usuario define para obtener unos resultados.

Estos son exportados a ficheros de cualquier formato para ser tratados por el cliente.

Es por ello que entender cómo funcionan este tipo de herramientas facilita la comprensión de cómo se realizan extracciones de datos dentro de las páginas webs y además cómo manipularlos. Estas funcionalidades pueden ser de gran utilidad para el desarrollo del proyecto.

Los software más populares son:

- webscraper.io
- data-miner.io
- parsehub

### 2.3.2.1. Webscraper.io y Data-miner

Webscraper.io [68] y Data-miner [9] son extensiones de Google que permiten realizar scraping en cualquier entorno web. Son sencillas de utilizar puesto que lo único que tenemos que hacer es instalarlas en el navegador y seguir su documentación.

Ofrecen páginas de prueba donde aprender las nociones básicas de la herramienta e iniciarnos en los conceptos básicos del scraping.



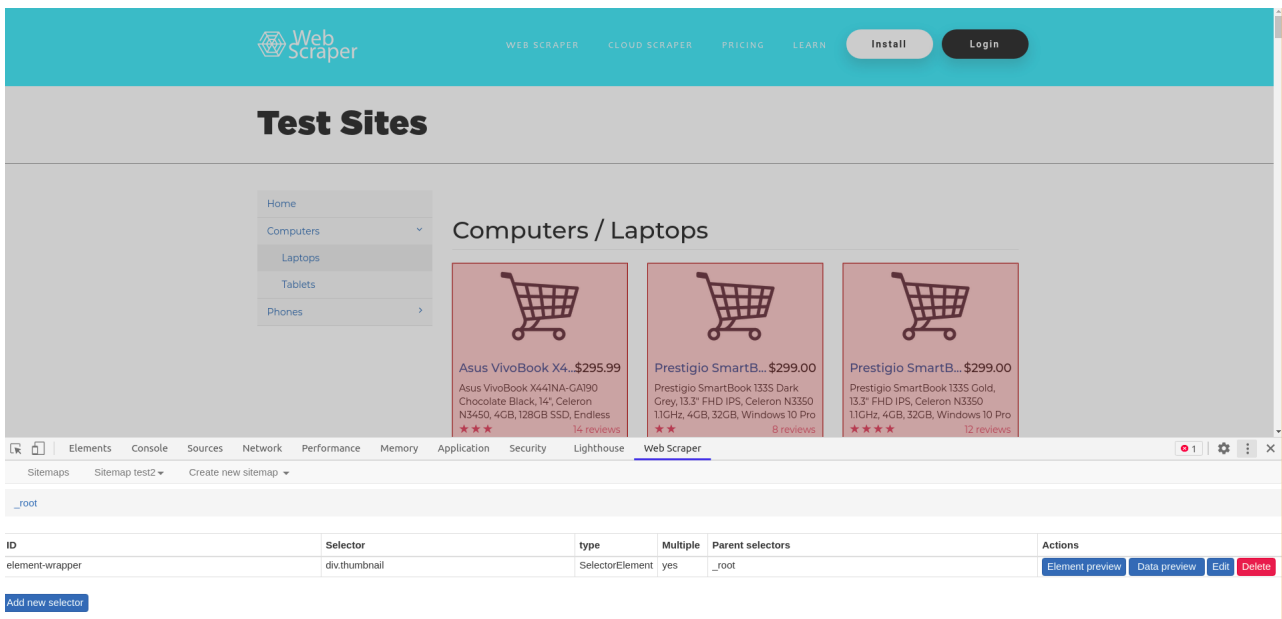


Figura 2.10: Web Scraper

### 2.3.2.2. Parsehub

Es una aplicación de escritorio que nos permite realizar scraping de una manera más completa en comparación con las opciones anteriores.

Permite implementar funcionalidades lógicas como condicionales, lanzar varias instancias al mismo tiempo, acceder a los datos a través de APIs, etc [45].

Es Click and Point, por lo que simplemente se seleccionan los campos que se desean obtener y el software retorna la información. Para el procesamiento de datos más complejos, ofrece una serie de herramientas más próximas a un lenguaje de programación para realizar estas tareas.

En la siguiente figura se muestra un ejemplo de uso de esta herramienta donde se obtienen los nombres de unas localizaciones en una tabla.

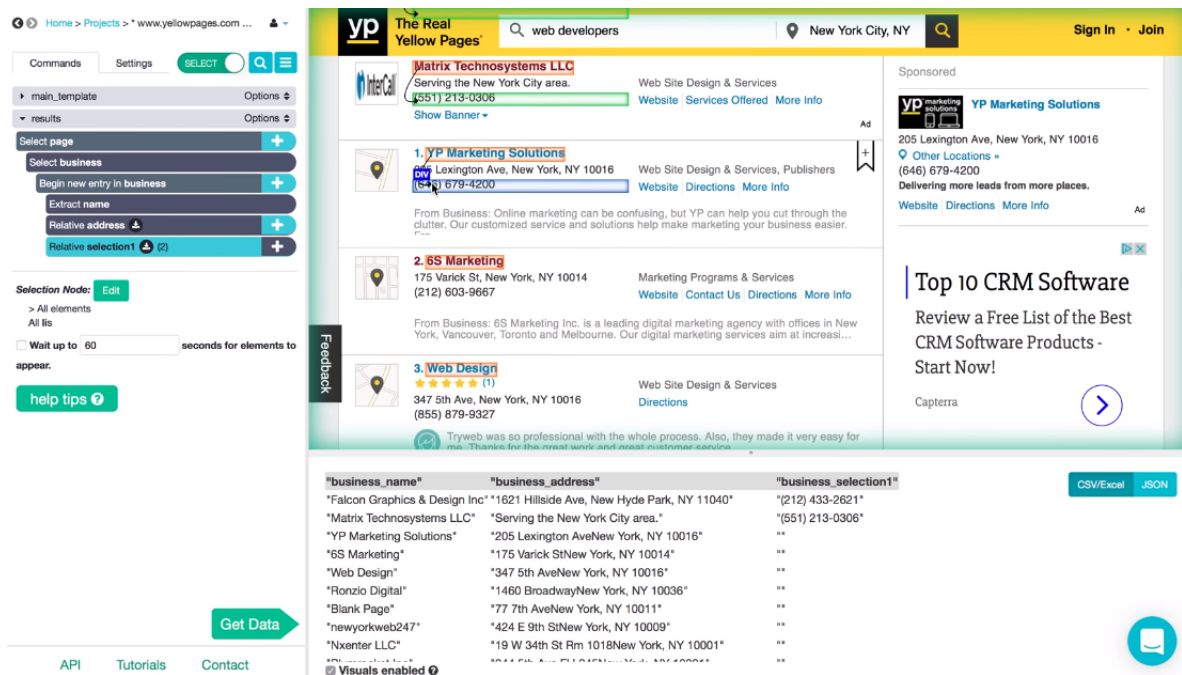


Figura 2.11: Ejemplo de uso ParseHub

## 2.4. Comparativa de datos

A continuación se mostrará a modo de resumen, una tabla comparativa entre las bases de datos, aplicaciones nutricionales y los comparadores de precio más interesantes y en los que se ha inspirado la aplicación:

Tabla 2.1: Comparativa entre las BDCAs más completas

	Precio	Marca	País	Idioma	Productos procesados y ecológico	Añadir productos	Micronutrientes y Macronutrientes	Registro	Gratuita
Open Food Facts	No	Si	Internacional	Internacional	Procesados (y algunos ecológicos)	Sí	Ambos	No	Sí
BEDCA	No	No	ES	ES	Ambos	No	Ambos	No	Sí
Food Data Central	No	No	EEUU	EN-US	Ambos	No	Sí	No	Sí

Tabla 2.2: Comparativa entre las Aplicaciones analizadas

	Precio	Marca	País	Idioma	Productos procesados y ecológico	Añadir productos	Micronutrientes y Macronutrientes	Planes de nutrición	Registro	Gratuita	Recetas
Calorie King	No	No	AUS UK EEUU	Internacional	Ambos	No	Ambos	Sí	Opcional	No	Sí
Edaman	No	No	UK	EN	Ambos	No	Ambos	-	Obligatorio	Sí	Sí
Oorenji	No	No	ES	ES	Ambos	Sí	Ambos	Sí	Obligatorio	Sí	Sí

Tabla 2.3: Comparativa entre los comparadores de precio

	Precio	Marca	Productos procesados y ecológico	Añadir productos	Micronutrientes y Macronutrientes	Registro	Gratuita	Lista de compra
SoySuper	Sí	Sí	Procesados	Sí	No	Opcional	No	Sí
Carritus	Sí	Sí	Procesados	No	No	Opcional	Sí	Sí
Oferia	Sí	Sí	-	Sí	No	No	Sí	No
RadarPrice	Sí	Sí	Procesados	Sí	No	No	Sí	No

Se ha seleccionado Open Food Facts como principal proveedor de datos para desarrollar este proyecto debido a que la mayor parte de sus productos presentan información nutricional muy completa con respecto a los macronutrientes y micronutrientes, y a su vez, presenta gran variedad de productos que se pueden encontrar en los mercados españoles cuyos datos provienen de la colaboración de los usuarios y de aplicaciones como Yuka.

Además, a modo de soporte, se ha utilizado BEDCA por su minuciosidad con los valores nutricionales de los productos y debido a que ofrece datos de alimentos naturales que no están presentes en OPF.

Con respecto a la obtención de los precios, la mayor parte de los comparadores de precio analizados utilizan modelos de web scraping para obtener el precio de los productos. Esto implica que para adquirir los precios se ha de utilizar las mismas técnicas que ellos.

# Capítulo 3

## Tecnologías y Arquitectura

En este capítulo se hablará del análisis de las distintas tecnologías y la arquitectura que se puede utilizar en el proyecto.

### 3.1. Análisis de Tecnologías

Se ha realizado un estudio de las distintas tecnologías, las cuales, no se han utilizado todas para la implementación del mismo. Es por ello que a continuación se detallan las distintas tecnologías.

#### 3.1.1. Backend

- **Express [14]:** es un framework web, escrito en JavaScript y alojado dentro del entorno de ejecución NodeJS [36]. Utiliza un modelo orientado a eventos y emplea como gestor de paquetes NPM [37].
- **Django [10]:** es un framework Open Source, escrito en Python, que respeta el patrón de diseño conocido como modelo vista controlador (MVC) [21].
- **Micronaut [28]:** es un framework JVM [63] que nos permite crear aplicaciones basadas en microservicios de una manera sencilla y rápida. Además de Java, Micronaut permite el uso de otros lenguajes como Groovy [17] y Kotlin [23].
- **Spring [59]:** es un framework de código abierto para la creación de aplicaciones empresariales en Java, con soporte para Groovy y Kotlin. Tiene una estructura modular y una gran flexibilidad para implementar diferentes tipos de arquitectura según las necesidades de la aplicación.
- **Quarkus [49]:** es un framework nativo de Java para Kubernetes [24] diseñado para GraalVM [16] y HotSpot [42], creado a partir de las mejores librerías y estándares Java.

#### 3.1.2. Frontend

- **JavaScript [26]:** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript [69]. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza del lado del cliente.

- **Vue [67]:** es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single Page Applications (SPA) cuando se utiliza en combinación con herramientas modernas y librerías de apoyo.
- **React [52]:** es una librería Javascript desarrollada por Facebook focalizada en el desarrollo de interfaces de usuario. Así se define la propia librería y evidentemente, esa es su principal área de trabajo. Sin embargo, lo cierto es que en React encontramos un excelente aliado para hacer todo tipo de aplicaciones web, SPA o incluso aplicaciones para móviles. Para ello, alrededor de React existe un completo ecosistema de módulos, herramientas y componentes capaces de ayudar al desarrollador a cubrir objetivos avanzados con relativamente poco esfuerzo.
- **Angular [1]:** es un framework opensource desarrollado por Google para facilitar la creación y programación de SPAs.
- **Quasar [50]:** es un framework basado en Vue que permite a los desarrolladores web crear de forma rápida sitios web responsivos y aplicaciones de diferentes tipos: SPAs, SSR, PWAs, aplicaciones móviles y multiplataforma.
- **ElementUI [64]:** es una librería de componentes basada en Vue 3.0. Se basa en los principios de coherencia retroalimentación tangible, eficiencia y capacidad de control.
- **SASS [54]:** es un preprocesador de css que reduce el tiempo de crear y mantener el CSS. Permite tener una organización modular de los estilos y proporciona estructuras avanzadas propias de lenguajes de programación entre otros.

### 3.1.3. Web scraping

- **Puppeteer [48]:** es una herramienta, desarrollada principalmente por Google, que nos permite utilizar un navegador Chromium/Chrome headless a través del protocolo DevTools. Se ejecuta sobre NodeJS y se instala a través de npm. Un navegador headless se ejecuta dentro de un proceso del terminal.
- **BeautifulSoup [57]:** es una librería de Python para analizar el HTML y XML. Esta librería crea un árbol con todos los elementos del documento y puede ser utilizado para extraer información.
- **Selenium [56]:** es un entorno de pruebas de software para aplicaciones basadas en la web. Incluye también un lenguaje específico de dominio para pruebas (Selenese [4]) para escribir pruebas en un amplio número de lenguajes de programación.
- **Scrapy [55]:** es una plataforma colaborativa de código libre que corre en Python para extraer datos de páginas web usado para una serie de aplicaciones como minería de datos, procesamiento de información o registro histórico.

- **Postman [47]:** es una herramienta que se utiliza, sobre todo, para el testing de API REST [18], aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas. Permite testear, consumir y depurar APIs REST, escribir pruebas automatizadas para ellas, documentarlas, mockearlas, simularlas, etc.
- **Curl [8]:** Es un comando disponible en la mayoría de los sistemas basados en Unix [65]. Los comandos de Curl están diseñados para funcionar como una forma de verificar la conectividad a las URL y como una gran herramienta para transferir datos.

### 3.1.4. Bases de datos

- **MongoDB [30]:** es un sistema de base de datos NoSQL orientado a documentos de código abierto y escrito en C++, que en lugar de guardar los datos en tablas lo hace en estructuras de datos BSON (similar a JSON) con un esquema dinámico.
- **MySQL [31]:** es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Se trata de una aplicación que permite gestionar archivos llamados de bases de datos.
- **PostgreSQL [46]:** es un sistema de código abierto de administración de bases de datos del tipo relacional, y extiende el lenguaje de SQL combinando muchas características de almacenamiento y escalabilidad de manera que asegura las cargas de trabajo de datos más complicadas.

## 3.2. Tecnologías utilizadas

Tras el previo análisis de las diferentes tecnologías, se detallan cuales fueron seleccionadas para el desarrollo de la aplicación.

### 3.2.1. Lenguajes de programación

Para el desarrollo del proyecto, se han seleccionado los lenguajes de programación de Python y JavaScript. El primero, se va a utilizar para el web scraping debido al gran número de librerías que dan soporte a estas herramientas y por su facilidad a la hora de manipular ficheros con un gran volumen de datos.

Por otro lado, JavaScript se utilizará principalmente para el desarrollo del cliente y del servidor utilizando la versión de ES6 debido a su fácil integración con herramientas webs y su gran número de librerías a disposición.

### 3.2.2. Frameworks

Se han utilizado los frameworks de Quasar para el desarrollo del cliente y el framework de Express para el servidor.

Al ser un proyecto que requiere de un desarrollo frontend amplio, es necesario utilizar herramientas que permitan escalabilidad, portabilidad, diseños responsivos y flexibilidad de diseños. Es por ello que Quasar es perfecto pues unifica todos estos en una sola

plataforma. Además ofrece una gran extensión de pluggins que facilitan y agilizan el desarrollo del proyecto y soporte de preprocesadores de CSS, de los cuales se seleccionó SASS por su organización modular a la hora mantener y crear nuevos estilos.

Al ser un framework que trabaja con JavaScript, Express es fácil de utilizar y tiene una gran comunidad que lo respalda. Permite crear cualquier tipo de aplicaciones con un periodo corto de respuesta al lado del cliente. Además proporciona lo que se llama middleware [19] que se encarga de tomar decisiones para dar las respuestas más adecuadas a las solicitudes de los clientes.

### **3.2.3. Base de datos**

Con respecto al diseño e implementación de la base de datos, se ha utilizado MongoDB como proveedor de estos servicios. Esto es debido a que los datos no tienen por que seguir un esquema relacional al tratarse de productos de alimentación que simplemente deseamos almacenar. Permite una alta flexibilidad y disponibilidad, alta escalabilidad, tanto vertical como horizontal y tras la versión 4.2 permite realizar transacciones como las bases de datos relacionales.

### **3.2.4. Web Scraper**

Se trabajaron con la gran mayoría de las tecnologías del apartado 3.1.3 puesto que cada una tiene un rendimiento diferente en función del diseño de la web a analizar.

Para recolectar los datos de los precios, se utilizó como principal herramienta BeautifulSoup por su facilidad a la hora de extraer los documentos y procesarlos con las herramientas de soporte Curl y Postman. El primero se utilizó para extraer el formato de las peticiones con sus cabeceras de los distintos endpoints de las webs, mientras que Postman sirvió para formatear esta petición y hacerla viable para obtener los datos.

### **3.2.5. Librerías**

Tanto para la parte del desarrollo Full Stack [33] como para el web scraping, se han utilizado una serie de librerías importantes a mencionar:

- Python: Pandas, Tqdm, Request, BeautifulSoup y Requests
- JavaScript: Lodash, Axios, Vuelidate, Material icons, Nodemon, JWT y Bcrypt.

## **3.3. Diseño de la arquitectura**

Tras el previo análisis de las distintas tecnologías que se han utilizado dentro del proyecto, se ha diseñado una arquitectura acorde con el objetivo, permitiendo una gran escalabilidad y facilidad a la hora de desarrollarlo. En la Figura 3.1 se muestra el resultado de la arquitectura.

# Spanish Open Food Network (SOFN)

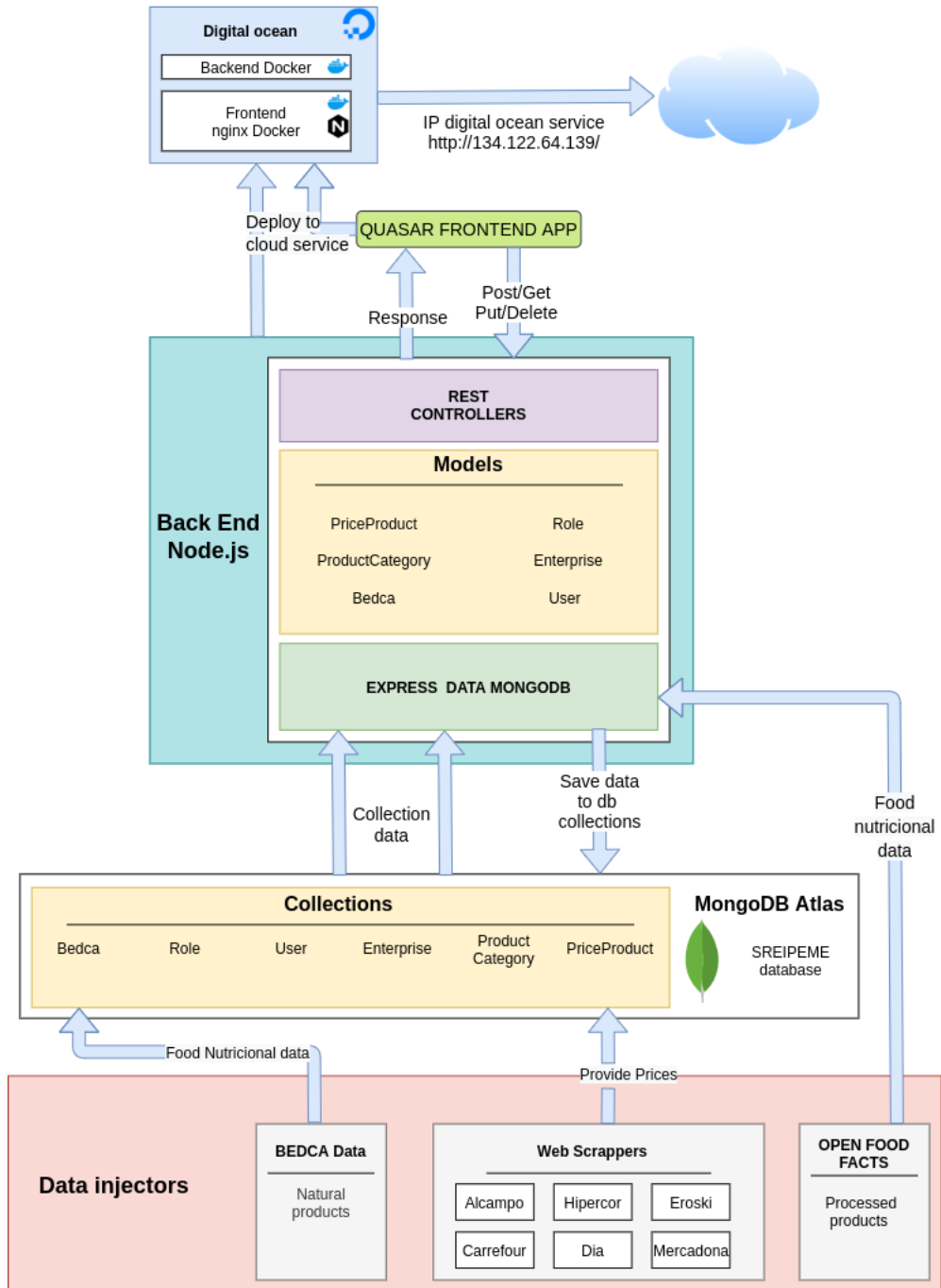


Figura 3.1: Arquitectura del sistema



Como se ve en la figura, existen cuatro grupos diferenciados dentro del sistema:

- Inyectores de datos
- MongoDB Atlas
- Backend y Frontend
- Despliegue

Es por ello que en esta sección se explicarán las diferentes características y funciones de estos.

### **3.3.1. Inyectores de datos**

Los inyectores de datos consisten en herramientas que provean de datos de manera externa a la base de datos. Es decir, cualquier información que no pueda ser obtenida mediante la aplicación web, se define como un inyector de dato. Para el proyecto, se han desarrollado 3 inyectores: BEDCA, web scrapers y OPF.

#### **3.3.1.1. BEDCA**

Mediante el uso de un web scraper implementado en python [60], se obtienen en formato CSV todos los productos con su información nutricional presentes en BEDCA. Este fichero, permitirá guardar dentro de del sistema toda su base de datos para su uso en la plataforma.

#### **3.3.1.2. Scrappers**

Debido a que los precios de los productos no están presentes en una base de datos pública, se ha implementado una herramienta de scraping para obtener los precios de los productos de los supermercados más notorios de OPF. Estos proporcionan los precios y se almacenan en una colección [7].

#### **3.3.1.3. OPF**

Al nutrirse el sistema principalmente por los productos que provienen de Open Food Facts y estos no almacenarse dentro de la base de datos del proyecto, se le considera un inyector de datos. Por ello y mediante el uso de la API pública que estos proporcionan, se obtienen los datos para presentarlos en SOFN (nombre de la aplicación) para los usuarios.

### **3.3.2. MongoDB Atlas**

Se ha utilizado MongoDB Atlas como servicio de Cloud Database que permite crear y administrar la base de datos de MongoDB desde cualquier lugar a través de su plataforma.

Dentro de este, se configura la base de datos y todas las colecciones que se necesita para la herramienta. Para el proyecto se escogió la versión gratuita ofreciendo hasta un máximo de 512 MB de almacenamiento.

La base de datos se comunica con el backend de la aplicación, permitiendo el registro, modificaciones y obtención de la información de las 6 colecciones de la aplicación. A continuación se detallan las principales características de estas y su funcionalidad.

### 3.3.2.1. User

Almacena la información personal de los usuarios cuando se registran en el sistema y controla el acceso mediante el token de autenticación.

Los datos del documento de esta colección son los siguientes: `_id` (identificador del elemento por defecto de MongoDB), `email` (correo del usuario (este campo ha de ser único)), `password` (contraseña encriptada), `token` (token de autenticación) y `role` (rol del usuario)

### 3.3.2.2. Role

Guarda todos los roles que están disponibles para la aplicación. Estos definen las diferentes partes a las que puede acceder un usuario dentro de la plataforma.

Esta información solo es accesible y editable para el administrador.

Los datos del documento de esta colección son los siguientes: `_id` (identificador del elemento por defecto de MongoDB) y `name` (nombre del rol)

### 3.3.2.3. Enterprise

Se refiere a todos los datos provenientes de E-Commerce. Contiene 2 subcolecciones: `Address` y `Product`. La primera se refiere a todas las direcciones del establecimiento, y en la segunda se almacenan todos los alimentos que la tienda tiene registrados.

Los datos del documento de la colección son los siguientes: `_id` (identificador del elemento por defecto de MongoDB), `name` (nombre de la empresa/tienda), `logo` (logo de la empresa/tienda), `banner` (banner de la empresa/tienda), `website` (página oficial), `facebook` (cuenta de facebook), `twitter` (cuenta de twitter), `instagram` (cuenta de instagram), `linkedin` (cuenta de linkedin), `email` (correo de contacto), `phone` (telefono de contacto), `owner` (dueño de la empresa), `company_number` (código de empresa), `short_description` (descripción corta), `long_description` (descripción larga) y `is_producer` (si es productor o proveedor)

Además, esta contiene dos subcolecciones:

- **address:** `first_address` (primera dirección), `second_address` (segunda dirección), `province` (provincia), `country` (país) y `zip_code` (código postal)
- **product:** `_id` (identificador del elemento por defecto de MongoDB), `name` (nombre del producto), `amount` (cantidad del producto), `unit_type` (tipo de unidad, si gramos, kilogramos, kilolitros...), `unit_size` (cantidad de una unidad), `price` (precio por unidad), `description` (descripción del producto) y `category` (categoría del producto)

### 3.3.2.4. ProductCategory

La colección `ProductCategory` guarda las distintas categorías de los productos de las tiendas, por lo que, al añadir un nuevo alimento al stock, se le ha de añadir una categoría registrada dentro de esta colección.

Los datos del documento de la colección son los siguientes: `_id` (identificador del elemento por defecto de MongoDB) y `name` (nombre de la categoría)

### 3.3.2.5. BEDCA

La colección Bedca almacena todos los productos provistos por esta Base de Datos junto con su información nutricional para su futura visualización. Estos datos son provistos por el inyector de datos de BEDCA 3.3.1.1

Los datos del documento de la colección son los siguientes: `_id` (identificador del elemento por defecto de MongoDB), `f_id` (identificador del producto de la Base de Datos original), `f_ori_name` (nombre del producto), `edible_portion` (medida por cada 100g), `carbohidratos`, `colesterol`, `riboflavina`, `tiamina`, `calcio`, `magnesio`, `sodio`, `ioduro`, `alcohol`, `grasa`, `proteína`, `fósforo`, `fibra`, `vitamina_A`, `vitamina_D`, `vitamina_E`, `folato`, `niacina`, `vitamina_B12`, `vitamina_B6`, `selenio` y `zinc`

### 3.3.2.6. PriceProduct

La colección PriceProduct agrega los precios de los productos provistos por OPF a la base de datos. Esta información es provista por el inyector de datos de scraping 3.3.1.2

Los datos del documento de la colección son los siguientes: `_id` (identificador del elemento por defecto de MongoDB), `product_name` (nombre del producto de OPEN FOOD FACTS), `product_name_es` (nombre del producto en español de OPEN FOOD FACTS) y `price` (precio del producto)

## 3.3.3. Backend y frontend

La aplicación ha sido desarrollada con una API REST para intercomunicar los datos de la base de datos con la aplicación web.

El backend se encarga de gestionar todas las peticiones entre el cliente y el servidor, permitiendo así la intercomunicación entre ambos servicios. Además se encarga de gestionar los datos provenientes de la API de OPF

El front envía y recibe los datos de los usuarios para mostrarlos de una forma agradable, haciendo atractiva la aplicación para atraer mayor público.

Tenemos 2 tipos de despliegues: El primero se corresponde al desarrollo en local donde simplemente mediante un script se levanta el front e indicaremos que puertos debe escuchar para conectarse al backend; mientras que el segundo se corresponde al despliegue de producción donde utilizaremos Docker [11], yNginx [35].

## 3.3.4. Despliegue en la nube

Para realizar el despliegue de la aplicación en producción, se ha seleccionado a Digital Ocean [39] como proveedor de hospedaje en la nube debido a su bajo coste y la facilidad.

Se ha seleccionado un Droplet [38] con Ubuntu 20.04 donde hacer el despliegue de la aplicación.

El despliegue de la herramienta se ha realizado mediante dos dockers uno para el cliente y otro para el servidor. Estos se encuentran intercomunicados entre sí mediante el uso de un Nginx presente en el Docker del front. La aplicación esta expuesta por la URL `http://134.122.64.139`

# Capítulo 4

## SOFN (Spanish Open Food Network)

En este capítulo se detalla el proceso de desarrollo y funcionamiento de la aplicación.

### 4.1. Descripción

Al existir la falta de una plataforma que sea capaz de unificar la mayor parte de las funcionalidades de aplicaciones orientadas en la alimentación, se ha diseñado una aplicación web orientada en la recopilación y centralización de productos de alimentación mejorando a su vez los métodos de visualización de los productos y de E-Commerce. Esta aplicación recibe el nombre de SOFN (Spanish Open Food Network) y permite visualizar productos del mercado español y obtener los datos nutricionales junto con su precio. Además, provee de un sistema de comercio donde los pequeños productores podrán promocionarse para vender sus productos a los clientes de la plataforma.

El objetivo principal de esta aplicación ha sido el proveer a los usuarios del sistema donde obtener información nutricional de un producto específico junto con su precio de manera rápida, sencilla y eficaz, y dotar a los productores y distribuidores una nueva forma de captar clientes y poder comercializar sus productos sin necesidad de recurrir a un tercero.

SOFN se ha diseñado con un modelo sencillo y visual para que cualquier usuario pueda utilizarla.



Figura 4.1: Logotipo de SOFN

### 4.2. Implementación de la herramienta

En este apartado se detallan las distintas partes de la aplicación como el desarrollo de SOFN.

### 4.2.1. Bases del proyecto

La primera parte del desarrollo fue diseñar la estructura básica del proyecto, creando así los repositorios para el frontend, backend y el web scraping.

Se realizó la configuración de los 3 repositorios y se instalaron las dependencias necesarias para cada uno de ellos. Para el frontend, se inicializó toda la configuración del CLI de Quasar, indicando los pluggins y herramientas que se desean para el proyecto.

Por último se preparó el Droplet de Digital Ocean junto con los Dockers para tener listo el entorno de producción.

### 4.2.2. Módulo de recopilación y centralización de alimentos

Acorde con lo indicado en la planificación (Véase el punto 1.1), este módulo se ha centrado en la implementación de un sistema de autenticación, obtención de los productos de OPF mediante el uso de su API y la obtención de los datos de BEDCA para que los usuarios de SOFN accedan a esta información. Además, se ha implementado una mejora visual en la presentación de los productos con respecto al resto de aplicaciones.

#### 4.2.2.1. Autenticación de usuarios

La autenticación de usuarios abarca el registro, acceso, modificación de datos y el control de acceso a las rutas de la página.

El primer paso fue habilitar que los usuarios puedan registrarse en el sistema. Para ello se ha creado un formulario en el front en una vista específica. Si el registro es exitoso, se redirecciona a la página de login donde iniciar sesión.

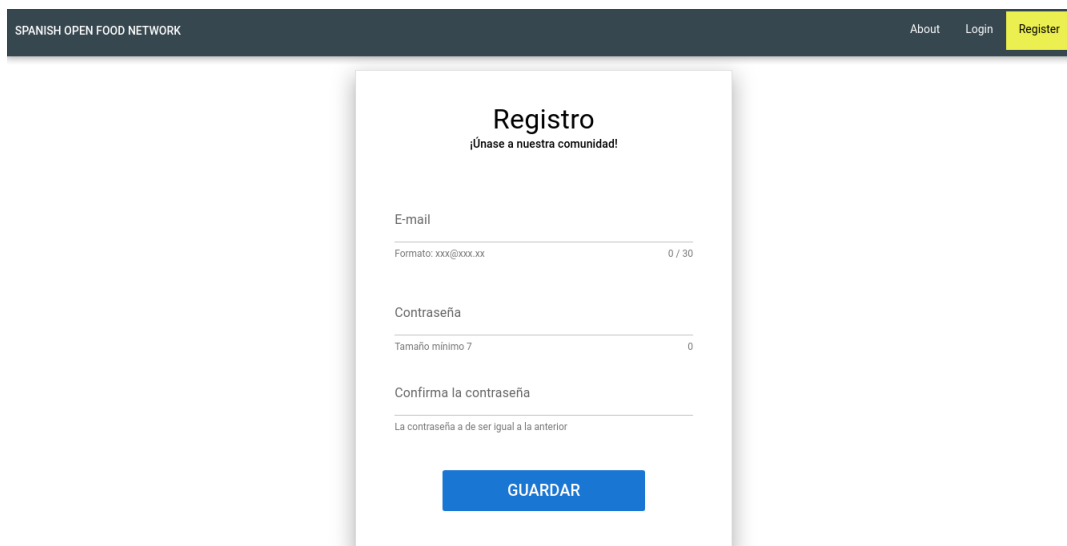


Figura 4.2: Página de registro de SOFN

Para iniciar sesión en el sistema el usuario rellena el formulario de acceso y tras ser verificado por el backend, este podrá acceder a todos los recursos de la aplicación.

Cuando un usuario inicia sesión en la aplicación, se genera un **Json Web Tokens (JWT)** [61] que se guarda, tanto en el front, como en el backend. Esto permite que si un usuario tiene registrado un JWT válido, no necesita de aportar nuevamente sus credenciales para acceder a la página hasta un plazo de 7 días. Cuando este token caduque, se deberá iniciar sesión nuevamente con sus credenciales para refrescar el token.

Además, JWT permite verificar los accesos a las diferentes páginas de la aplicación. Esto se debe a que cada vez que realizamos una petición a las distintas vista, si este usuario tiene un token desactualizado o no válido, se le redirecciona al login. Esto se debe a un middleware de autenticación que comprueba si en la petición se envía un token válido. Si estos datos no concuerdan, el sistema redirecciona al usuario nuevamente al login.

En la siguiente imagen se muestra el código del middleware y cómo comprobar si el usuario tiene acceso a la plataforma.

```
1  const jwt = require('jsonwebtoken')
2  const config = require('../services/config')
3  const ErrorRequest = require('../errorHandling/requestError')
4
5  module.exports = (req, res, next) => {
6    try {
7      const token = req.headers.authorization.split(' ')[1]
8      const decodedToken = jwt.verify(token, config.TOKEN_SECRET)
9      const userId = decodedToken._id
10     if (req.params.id && req.params.id !== userId) throw new ErrorRequest('[ERROR] User ID not authorized', 401)
11     else next()
12   } catch (error) {
13     if (error.status !== undefined) {
14       return res.status(error.status).send({
15         success: false,
16         error: `${error.message}`
17       })
18     } else {
19       return res.status(401).json({
20         success: false,
21         error: `${error}`
22       })
23     }
24   }
25 }
```

Figura 4.3: Middleware de autenticación

#### 4.2.2.2. Datos de OPEN FOOD FACTS

Al tener OPF una API pública, se ha creado un controlador dentro del backend para gestionarlo.

El primer paso para obtener unos datos legibles, fue analizar el objeto que se obtiene. OPF devuelve un array de jsons con un número de claves considerables, llegando a tener en algunos productos más de cien claves. Es por ello que se ha de procesar esa información y descartar los datos no relevantes

Como se explica en el apartado 2.1, se han recogido todos los macronutrientes y micronutrientes y también las tiendas en las que se comercializa el producto con su nombre e identificador (el id representa el código de barras bajo el estándar **ean 13 [ean13]**).

Para llevar a cabo esta selección, primero se obtuvieron los datos de OPF mediante **axios [2]** y se ajusta el array que devuelve en el formato deseado.

Finalmente el objeto de OPF es retornado al front y este es presentado en una vista mediante una tabla que permite el filtrado, ocultar columnas (puesto que son muchos elementos) y paginación.

Además dentro de esta vista, el usuario puede realizar búsquedas específicas de productos de interés mediante el buscador situado encima de la tabla.

```

const openFoodFactsProducts = await axios(config)
const maxItems = openFoodFactsProducts.data.count
const formattedProducts = openFoodFactsUtils.parseOpenFoodFacts([
  openFoodFactsProducts.data.products.map(element => _.pick(element,
    ['_id',
      'product_name',
      'product_name_es',
      'allergens',
      'brands',
      'stores',
      'nutriments',
      'image_url',
      'nutrition_data_per']))
])
})
You, seconds ago • Uncommitted changes

if (formattedProducts.length !== 0) {
  return res.status(200).json({
    data: formattedProducts,
    maxItems: maxItems,
    success: true,
    msg: '[SUCCESS] Elements retrieved successfully...!'
  })
} else {
  return res.status(204).json({
    data: formattedProducts,
    success: true,
    msg: '[SUCCESS] No content available...!'
  })
}
} catch (error) {
  if (error.status !== undefined) {
    return res.status(error.status).send({
      success: false,
      error: `${error.message}`
    })
  } else {
    return res.status(400).send({
      success: false,
      error: `${error}`
    })
  }
}
}

```

Figura 4.4: Petición para obtener datos de OPF

#### 4.2.2.3. Datos de BEDCA

Para acceder a los datos de BEDCA, se utilizó un scraper de github que retornaba un csv con toda la información que contenía esta base de datos.

Al únicamente contener 955 productos y que es una base de datos sin mantenimiento, se exportó el csv y se agregó a la colección de Bedca (Véase el apartado 3.3.2.5).

Para el procesamiento del fichero, el primer paso fue arreglar las cabeceras del csv puesto que no seguían ningún tipo de estándar y los datos no eran coherentes. Es por ello, que se reajustaron las cabeceras para tener una información más consistente. En las figuras 4.5 y 4.6 se muestra el código del script donde se emplea la función picker de la librería **lodash** [25] para seleccionar los datos importantes y mediante el uso de las herramientas de ES6 se cambia el nombre de las cabeceras por unos más estandarizado y legible.

```

5  const _ = require('lodash')
6  bedcaJson = bedcaJson.map(element => {
7    return _.pick(element, [
8      'f_id',
9      'f_ori_name',
10     'edible_portion',
11     'alcohol (etanol)',
12     'energia, total',
13     'grasa, total (lipidos totales)',
14     'proteina, total',
15     'agua (humedad)',
16     'carbohidratos',
17     'fibra, dietetica total',
18     'colesterol',
19     'Vitamina A equivalentes de retinol de actividades de retinos y carotenoides',
20     'Vitamina D',
21     'Viamina E equivalentes de alfa tocoferol de actividades de vitámeros E',
22     'folato, total',
23     'equivalentes de niacina, totales',
24     'riboflavina',
25     'tiamina',
26     'Vitamina B-12',
27     'Vitamina B-6, Total',
28     'Vitamina C (ácido ascórbico)',
29     'calcio',
30     'magnesio',
31     'sodio',
32     'fósforo',
33     'ioduro',
34     'selenio, total',
35     'zinc (cinc)'
36   ])
37 })

```

Figura 4.5: Selección de los datos del CSV de BEDCA

```

39  bedcaJson.map((element) => {
40    delete Object.assign(element, { alcohol: element['alcohol (etanol)'] })['alcohol (etanol)']
41    delete Object.assign(element, { energia: element['energia, total'] })['energia, total']
42    delete Object.assign(element, { grasa: element['grasa, total (lipidos totales)'] })['grasa, total (lipidos totales)']
43    delete Object.assign(element, { proteina: element['proteina, total'] })['proteina, total']
44    delete Object.assign(element, { agua: element['agua (humedad)'] })['agua (humedad)']
45    delete Object.assign(element, { fosforo: element['fósforo'] })['fósforo']
46    delete Object.assign(element, { fibra: element['fibra, dietetica total'] })['fibra, dietetica total']
47    delete Object.assign(element, { vitamina_A: element['Vitamina A equivalentes de retinol de actividades de retinos y carotenoides'] })['Vitamina A eq
48    delete Object.assign(element, { vitamina_D: element['Vitamina D'] })['Vitamina D']
49    delete Object.assign(element, { vitamina_E: element['Viamina E equivalentes de alfa tocoferol de actividades de vitámeros E'] })['Viamina E equivale
50    delete Object.assign(element, { folato: element['folato, total'] })['folato, total']
51    delete Object.assign(element, { niacina: element['equivalentes de niacina, totales'] })['equivalentes de niacina, totales']
52    delete Object.assign(element, { vitamina_B12: element['Vitamina B-12'] })['Vitamina B-12']
53    delete Object.assign(element, { vitamina_B6: element['Vitamina B-6, Total'] })['Vitamina B-6, Total']
54    delete Object.assign(element, { vitamina_C: element['Vitamina C (ácido ascórbico)'] })['Vitamina C (ácido ascórbico)']
55    delete Object.assign(element, { selenio: element['selenio, total'] })['selenio, total']
56    delete Object.assign(element, { zinc: element['zinc (cinc)'] })['zinc (cinc)']
57  })
58
59  fs.writeFile('./data/parsedJson.json', JSON.stringify(bedcaJson), (error) => {
60    if (error) { console.error(error); return };
61    console.log('File has been created')
62  })

```

Figura 4.6: Renombrado de las columnas del CSV de BEDCA

Tras realizar este proceso, se extraen los datos en un nuevo csv y se almacena dentro de la colección de BEDCA.

Por último el controlador envía al front los datos y este los muestra a modo de tabla como en OPF con las mismas características.



### 4.2.3. Módulo de E-Commerce

El módulo de E-Commerce se refiere a la creación de un sistema de comercio dentro de nuestra aplicación para que los usuarios puedan abrir sus tiendas y dar de alta sus productos.

Cada usuario podrá tener tantas tiendas como desee, donde se especificarán diferentes métodos de contacto para que los clientes puedan notificar que están interesados en los productos.

Para el desarrollo de este módulo, se han configurado 2 colecciones con sus respectivos controladores y 3 vistas en la aplicación.

#### 4.2.3.1. Controladores

Al requerir de un registro de las tiendas dentro del sistema, se ha realizado un CRUD con diferentes opciones para la colección Enterprise (Véase el apartado 3.3.2.3).

A continuación se muestra una imagen con los distintos endpoints disponibles junto con los controladores asignados a estos.

```
1  const express = require('express')
2  const enterpriseController = require('../controller/enterpriseController')
3  const auth = require('../middleware/auth')
4
5  const router = express.Router()
6
7  router.post('/', auth, enterpriseController.createEnterprise)
8
9  router.put('/details', auth, enterpriseController.updateEnterpriseDetails)
10
11 router.put('/logo', auth, enterpriseController.updateLogo)
12 router.put('/banner', auth, enterpriseController.updateBanner)
13 router.put('/social', auth, enterpriseController.updateSocialMedia)
14 router.put('/', auth, enterpriseController.updateEnterprise)
15
16 router.get('/:id', auth, enterpriseController.getEnterpriseProfile)
17 router.post('/getSelected', auth, enterpriseController.getEnterpriseById)
18 router.post('/newProduct', auth, enterpriseController.updateProduct)
19
20 router.get('/', auth, enterpriseController.getAllEnterprise)
21
22 module.exports = router
```

Figura 4.7: Fichero de rutas de la colección Enterprise

Como se puede observar en la Figura 4.7, se han implementado distintas formas para actualizar o crear una tienda. Esto se debe a que el formulario de registro de la tienda, tiene 10 campos obligatorios, por lo que el resto podrán ser actualizados en cualquier momento desde la vista del perfil.

Para el registro de los nuevos productos de las tiendas se ha implementado el endpoint newProduct para recoger estos datos y almacenarlos en la subcolección de productos de Enterprise.

Todos estos endpoint requieren de un usuario registrado con una sesión activa en el sistema, por lo que ha sido necesario agregar el middleware de autenticación mencionado previamente.

#### 4.2.3.2. Dashboard de la Tienda

El dashboard es una herramienta que todos los usuarios de SOFN poseen para acceder a todas las tiendas registradas bajo su nombre, pudiendo modificar los datos del perfil y también añadir productos de alimentación.

Para editar o visualizar los datos, se deberá seleccionar que tienda es la que desea visualizar en el selector y tras esta acción, se recogen los datos del perfil del backend y se rellenan los campos ya creados, después, el usuario podrá navegar entre las distintas secciones y actualizar o eliminar las opciones.

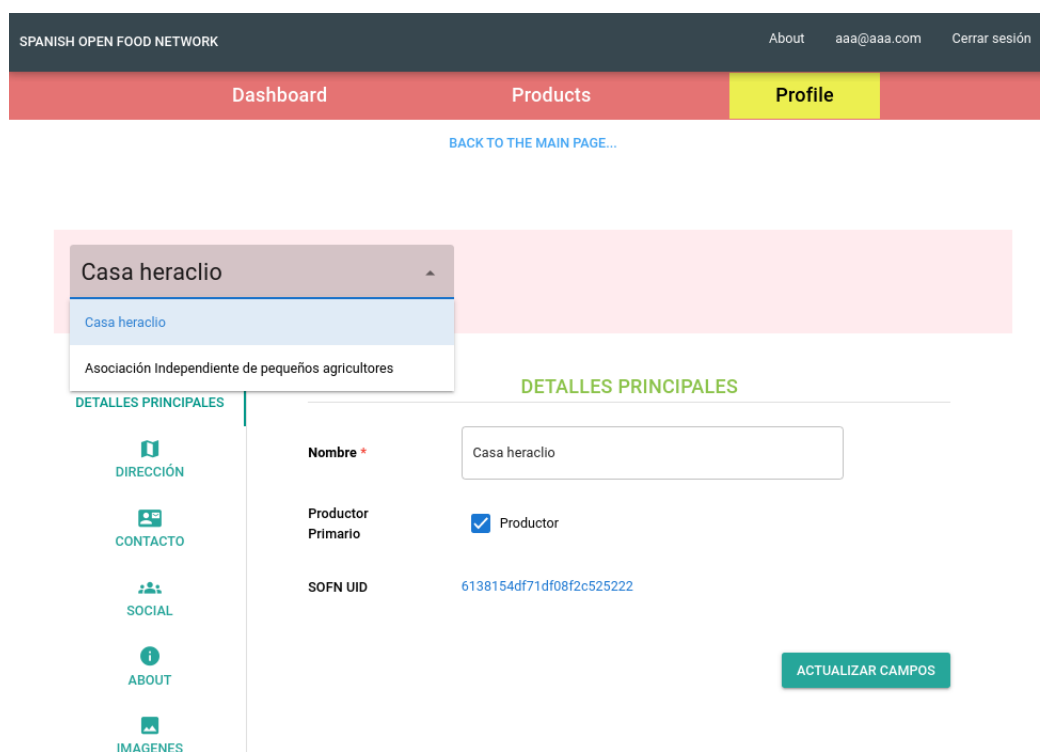


Figura 4.8: Perfil de la tienda

Respecto al menú de los productos (Véase la figura 4.9), tiene una funcionalidad similar a la del perfil. El usuario nuevamente deberá seleccionar que tienda desea visualizar o editar y se recogen los datos de los productos asociados a esta. Los alimentos se muestran en una tabla donde filtrar, ocultar columnas y modificar, además de un formulario para añadir nuevos productos en base a los atributos de la subcolección de Enterprise Product (Véase el apartado 3.3.2.3) al que se accede haciendo click en *añadir un producto*.

Asociación Independiente de pequeños agricultores ▾

Seleccione su empresa

+ AÑADIR UN PRODUCTO

Busque datos en la tabla 🔍

OCULTAR COLUMNAS ▾ 15 FILAS ▾

1 - 1 de 1

Nombre	Descripción	Categoría del producto	Tipo de unidad	Cantidad por unidad	Precio por unidad	Cantidad total del producto
Kiwi		Frutas y verduras	g	100	3.22	10000

Figura 4.9: Productos de la tienda

#### 4.2.3.3. Acceso a Tiendas

Para visualizar las distintas tiendas junto con los productos que comercializa, se ha implementado una vista que permita estas funcionalidades.

En la Figura 4.10 se muestra la vista de las tiendas. En esta, se traen del backend todas las tiendas existentes en la plataforma y se muestran en un listado, junto con su nombre, descripción corta y logotipo.

Los usuarios tienen disponible la opción de realizar un filtrado y buscar tiendas en específico.

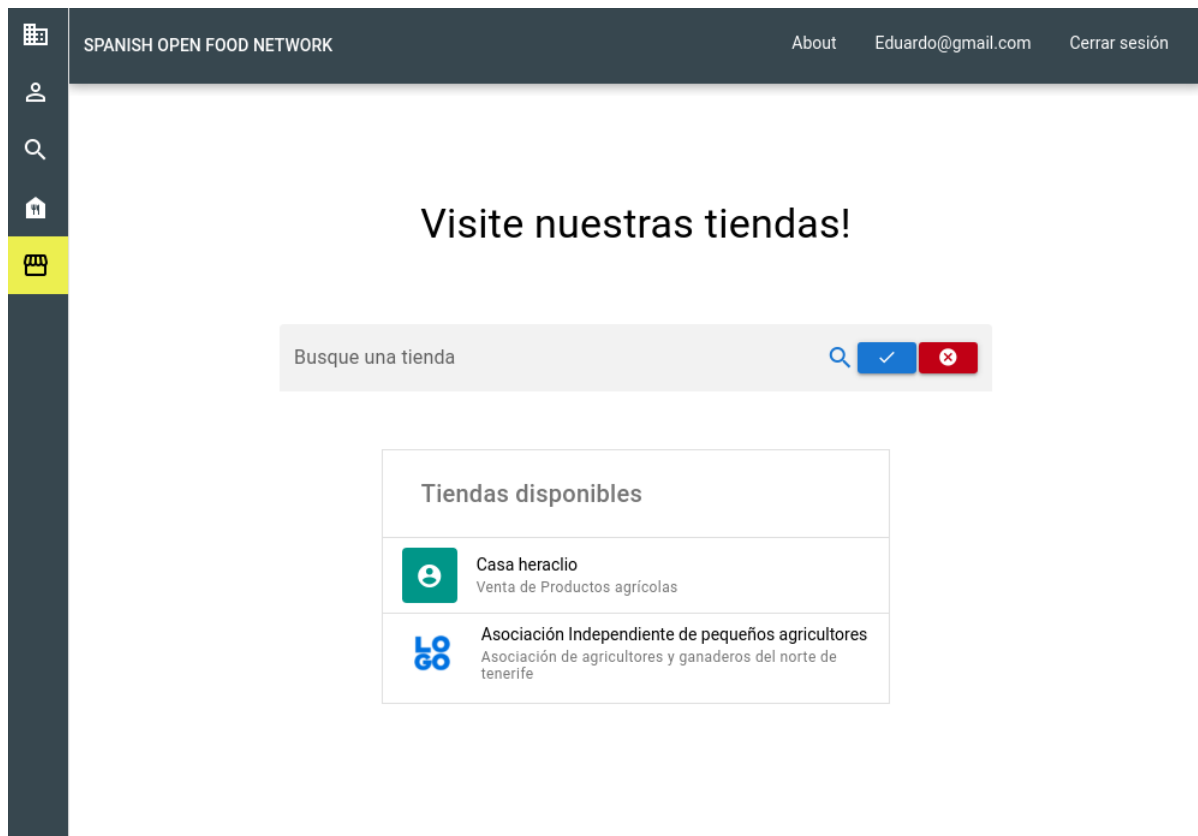


Figura 4.10: Lista de las tiendas disponibles

Para visualizar los productos y las distintas formas de contacto, el usuario deberá hacer click en la tienda para acceder a su información. Esta acción ejecuta un modal donde se muestran todos los datos de contacto, el banner de la empresa, la descripción larga y una tabla con el stock que tiene a su disposición.

Esto permite que cualquier cliente de la tienda, pueda ponerse en contacto con el gestor y solicitar los productos que este considere.

#### 4.2.4. Módulo de Web Scraping

Como último módulo principal, queda por tratar la obtención de los precios de los productos de OPF. Es por ello que en este apartado, se explica el proceso de obtención de precios mediante el uso de las tecnologías de Web Scraping mencionadas en el apartado 3.2.4.

##### 4.2.4.1. Obtención de datos

El primer problema que se ha abordado es la sincronización de todos los productos de la Base de Datos de OPF con sus precios de manera dinámica, por lo que se evaluó si los comparadores de precios del apartado 2.3.1 ofrecían los alimentos de OPF. . Tras una serie de pruebas, prácticamente el 80 % de los productos no estaban presentes en la plataforma por lo que se descartó inmediatamente esta vía y se decidió investigar otra líneas, por lo que se prosiguió investigando la API de OPEN FOOD FACTS y ver qué datos se podían utilizar para realizar esta tarea. Finalmente, la API provee del número total de productos que tiene un supermercado o marca. Esto permitió tener una selección de los supermercados que más productos proveen al sistema.

Al tener acceso a los productos de una manera ordenada, se abordó el problema seleccionando los productos de los supermercados más notorios e implementando un modelo de scraping específico para cada web de compra online.

Los supermercados o marcas seleccionadas para realizar el scraping son los siguientes: Alcampo y Auchan, Alteza, Aldi, Bonarea, Carrefour, El Corte Inglés e Hipercor, Día, Mercadona y Lidl.

Al no ofrecer todos los comercios un servicio de compra online, primero se investigó cuáles proveían de estos y cuáles no, Lidl, Bonarea, Aldi y Alteza fueron descartados al no existir una forma de obtener sus precios utilizando esta estrategia.

Tras la selección, se implementó un script dentro del backend que retorna todos los productos de OPF asignados al comercio. Para ejecutarlo simplemente se deberá indicar el nombre del fichero de salida, el número máximo de elementos que se van a retornar y el endpoint de los datos y este retorna un archivo en formato json

Además, a la hora de seleccionar los datos que se necesitan para identificar los productos, se seleccionaron los campos de `product_name` (nombre genérico del producto), `product_name_es` (nombre en español del producto) y el código de barras. Como los alimentos no tienen siempre el nombre en español, para identificar al producto cuando este no aparece se utilizó el nombre genérico.

#### **4.2.4.2. Análisis de los comercios**

El siguiente paso para obtener los precios es analizar las páginas de compra online de los supermercados seleccionados.

Para ello, se ha utilizado el inspector de elementos del navegador Chrome, donde se han analizado los distintos endpoints que utilizan los supermercados. La mayoría de las páginas no emplean métodos de protección ante el web scraping salvo Carrefour y Mercadona que para ocultar los datos, utilizan ficheros en formato XHR [71] que transfieren datos. Además Mercadona tiene una restricción extra y es que solicitan de un registro para utilizar su página.

Para solventar el registro de Mercadona se utilizó el inspector de elementos del navegador. Probando a eliminar partes del código HTML, se pudo saltar este registro y acceder a la página donde se encontraban las rutas de los productos. Curiosamente, esta página solicita un registro previo pero no comprueba si el usuario que accede a los endpoints se encuentra logueado en el sistema.

Es por ello que tras obtener todos los endpoints y realizar pruebas de concepto con postman, se configuraron todas las cabeceras, cookies y parámetros necesarios para poder obtener los en estas páginas.

#### **4.2.4.3. Implementación de scrapers**

Tras realizar este primer acercamiento, se prosiguió con la implementación de los métodos de web scraping específicos para los diferentes supermercados. Para esto, se creó un nuevo módulo de Python que consta de 6 clases específicas para cada comercio.

Para la implementación de estas, el código se ha implementado según el tipo de acceso a los datos las cuáles son las siguientes:

- Acceso a los datos mediante las peticiones XHR.
- Acceso a los datos mediante librerías de Web Scraping.

El primero modelo que se llevo a cabo fue para las páginas de Mercadona y Carrefour donde los datos están en formato XHR.

La ventaja de que estas plataformas empleen este formato, es que los datos vienen directamente en formato JSON, por lo que procesarlos es relativamente sencillo. Simplemente se recogen todos los resultados que se han obtenido de la búsqueda, se realiza una media y se guardan dentro del **dataframe de Pandas [44]** para exportarlo a un nuevo fichero csv.

Para calcular el tiempo medio que tardan en procesarse los datos, se utilizó la librería **Tqdm [62]**, que mediante una barra de carga, muestra el porcentaje que lleva completado el proceso y además, una estimación de cuánto tarda por iteración en procesar una petición. La estimación media para obtener los precios de 1000 productos ronda los 15 minutos.

El segundo modelo se llevo a cabo mediante la librería de BeautifulSoup por lo que será necesario conocer cuales son las clases y etiquetas que contienen los precios dentro del HTML. Para obtener esta información, se utilizó la herramienta de desarrollador de Chrome y se buscaron en el documento de la página.

Tras obtener estos datos, se le indicó a la librería BeautifulSoup donde tenía que buscar y se realizó la llamada al endpoint de compra online del supermercado para obtener el HTML. Finalmente y tras el procesamiento del HTML por parte del scraper, se obtienen los precios del producto que se guardan en un dataframe para exportarlos a un fichero en formato csv

En esta parte, también se utilizó Tqdm para realizar una comparativa con respecto a los 2 modelos utilizados, viéndose así una clara diferencia. El método de BeautifulSoup era considerablemente más lento puesto que para procesar 1000 productos, tardó una media de 25 minutos. Esto implica que siempre que se puedan obtener los datos mediante ficheros XHR, será la elección prioritaria.

Tras el procesamiento de los datos, se obtienen 6 ficheros en formato csv con los precios de los productos de los supermercados seleccionados. Además si se produce algún error al procesar los datos, se guardan en ficheros csv de errores.

#### **4.2.4.4. Precios de OPF**

Al únicamente conseguir un valor aproximado de 20.000 productos con sus precios, es una cantidad notablemente inferior a los más de 200.000 productos que tiene OPF en su base de datos.

Es por ello que en este apartado, se va a realizar el procesamiento de un fichero que contiene una enorme gran cantidad de datos con sus precios en base a uno de los scrapers expuestos en el paso anterior.

Como se realizó en el apartado 4.2.4.1, indicaremos el endpoint que vamos a utilizar para obtener 200.000 productos de OPF y el nombre del fichero json. Este proceso tardó 2 horas en completarse.

Tras obtener el fichero csv con todos los productos, el siguiente paso es seleccionar qué scraper se va a utilizar para realizar esta prueba.

Analizando los resultados obtenidos y la información que ofrece Tqdm, se decidió escoger a Carrefour por su velocidad en comparación a los otros para procesar estos datos. A la hora de trabajar con un fichero con una gran cantidad de datos, surgieron una gran variedad de dificultades que obstaculizaron esta tarea. A continuación se listan los problemas encontrados::

- Tiempos de Procesamiento de 24 horas.
- Peticiones repetidas y rápidas a un endpoint.
- El proceso se quedaba sin memoria.
- Manejo de errores mediante try catch.

Para los bloqueos de IP, la solución fue implementar en los headers proxys y user-agents [27] aleatorios por cada petición. Investigando y agrupando diferentes ficheros de proxys válidos en internet, se obtuvo una batería de más de 60.000 para ser utilizados en las distintas peticiones y evitar estos bloqueos. Además se seleccionaron diferentes user-agents para simular búsquedas en otros dispositivos.

Se implementó una secuencia de control de excepciones para que si en algún momento había un problema con los datos o alguna caída y continuase al ejecución del proceso.

Para disminuir el tiempo de procesamiento y solventar el problema de la memoria, se crearon 6 procesos que se ejecutan para rangos de entre 30.000 y 35.000 productos. Esto permitió que los procesos no se quedaran colgados y que el tiempo de ejecución fuera de 10 horas.

Finalmente se obtuvieron 6 ficheros con todos los datos de OPF asignados a los precios de Carrefour.

# Capítulo 5

## Conclusiones y futuro del proyecto

En este capítulo se presentan las conclusiones obtenidas a partir del desarrollo del proyecto SOFN y sus líneas futuras.

### 5.1. Resultados finales

Con el paso de los años, las personas han comenzado a cuidar más su salud y a seguir estilos de vida más saludable. Es por ello que se ha ido incrementando el número de herramientas dedicadas a este propósito.

SOFN ha mejorado la visualización de la información nutricional de los productos de alimentación, permitiendo al usuario acceder a los datos que necesita de una manera directa y sencilla, además de mostrar la mayor parte de los productos que se encuentran disponibles en el mercado español.

Actualmente la obtención de los precios se ha plasmado en un programa que extrae los precios de los productos almacenados en un fichero en formato csv. Por un lado, esto permite que un desarrollador externo pueda reutilizar este módulo para sus propios proyectos, y por el otro, facilita que un usuario no experto pueda realizar una simulación de su cesta de la compra para organizarla de una manera más óptima e incluso obtener el precio final.

Además SOFN alberga un espacio donde cualquier persona puede dar de alta sus productos de alimentación y poder comercializarlos. Gracias a ello, cualquier usuario que produzca alimentos puede obtener beneficios de sus ventas.

Como resultado, se ha obtenido una plataforma que ha integrado las diferentes funcionalidades y servicios que ofrecen la mayoría de aplicaciones de alimentación en una sola e innovando dentro de este sector.

### 5.2. Líneas futuras

Las líneas futuras del desarrollo se han orientado principalmente en la mejora de las características de la misma como a la unificación de los de recopilación y centralización de alimentos, E-Commerce y web scraping. Las propuestas para mejorar SOFN son las siguientes:

- **Implementación un sistema de sugerencias a las tiendas:** cuando un gestor de un comercio agrega un nuevo producto a su stock, este añade el precio que más le parece. Es por ello que implementar sistema de sugerencias donde indique cuál



es el precio promedio al que se está vendiendo ese producto en el mercado podría resultar interesante.

- **API para la obtención de precios en tiempo real:** implementar una nueva API que ofrezca en tiempo real los precios de los productos de un script que se ejecute manualmente.
- **Mejora en los modelos de scraping:** actualmente, los modelos creados para la obtención de los precios es excesivamente lento. Es por ello que sería interesante implementar un sistema más optimizado y que permita exportarse a obtener los precios de cualquier tipo de entorno web.
- **Módulo de recetas:** además de la visualización de productos, implementar un sistema donde las empresas puedan agregar sus propias recetas junto su información nutricional y comercializar los platos.
- **Sistema de compra dentro de la aplicación:** para mejorar el sistema actual de compra, sería interesante implementar un módulo donde no haga falta comunicación entre el cliente y el vendedor, además de un sistema de envíos de los pedidos a los domicilios.
- **Realizar una aplicación móvil:** exportar la aplicación web a un entorno móvil podría resultar muy positivo y atractivo para los usuarios.
- **Registrar productos y precios mediante un modelo similar al de Yuka:** sería puede valorar la obtención de los precios y de nuevos productos mediante la participación de los usuarios dentro de la plataforma utilizando una metodología similar a la de Yuka.

# Capítulo 6

## conclusions and future of the project

This chapter presents the conclusions obtained from the development of the SOFN project and its future lines of action.

### 6.1. Final results

Over the years, people have started to take more care of their health and follow healthier lifestyles. This is why the number of tools dedicated to this purpose has been increasing.

SOFN has improved the visualisation of the nutritional information of food products, allowing users to access the data they need in a direct and simple way, as well as showing most of the products that are available on the Spanish market.

Currently, prices are obtained through a software that extracts the prices of the products stored in a file in csv format. On the one hand, this allows an external developer to reuse this module for their own projects, and on the other hand, it allows a non-expert user to carry out a simulation of their shopping basket in order to organise it in a more optimal way and even obtain the final price.

SOFN also hosts a space where anyone can register their food products and market them. Thanks to this, any user who produces food can make a profit from their sales.

As a result, a platform has been obtained that has integrated the different functionalities and services offered by the majority of food applications into a single one, innovating within this sector.

### 6.2. Future lines

The future lines of development have been mainly oriented towards the improvement of its features, such as the unification of food collection and centralisation, E-Commerce and web scraping. The proposals for improving SOFN are as follows:

- **Implementation of a suggestion system for shops:** when a shop manager adds a new product to his stock, he adds the price he thinks is the best. This is why implementing a suggestion system that indicates the average price at which that product is being sold in the market could be interesting.
- **API for obtaining prices in real time:** implement a new API that offers in real time the prices of the products of a script that is executed manually.

- **Improvement in scraping models:** currently, the models created to obtain prices are excessively slow. That is why it would be interesting to implement a more optimised system that allows exporting to obtain prices from any type of web environment.
- **Recipes module:** in addition to the visualisation of products, implement a system where companies can add their own recipes together with their nutritional information and market the dishes.
- **Purchase system within the application:** to improve the current purchase system, it would be interesting to implement a module where there is no need for communication between the customer and the seller, as well as a system for sending orders to homes.
- **Create a mobile application:** exporting the web application to a mobile environment could be very positive and attractive for users.
- **RRegister products and prices using a model similar to Yuka's:** it would be possible to evaluate the collection of prices and new products through the participation of users within the platform using a methodology similar to Yuka's.

# Capítulo 7

## Presupuesto

Para dar una estimación real del coste total de lo que ha sido el proyecto, deberemos tener en cuenta las distintas tareas realizadas al igual que las herramientas implicadas en estas. Las principales tareas realizadas fueron las siguientes:

- **Análisis del estado del arte**, consiste en una investigación sobre las distintas BDCA (Bases de datos de composición de alimentos) y aplicaciones que existen en el mercado actual y de las cuáles pudieramos extraer datos para la obtención de productos alimeticios y sus precios.
- **Análisis de las herramientas de desarrollo full stack actuales**, donde se investigó sobre todas las alternativas que podrías ser interesantes y que se adecuaran más a las necesidades del proyecto.
- **Estudio de Web Scraping y sus aplicaciones**, implicando todo lo que es el aprendizaje sobre este tipo de herramientas y el cómo aplicarlas en la obtención de los precios de los productos.
- **Diseño e implementación de la herramienta**, refiriéndose a todo el desarrollo de la aplicación, desde el diseño de la arquitectura, hasta todas las funcionalidades que esta incorpora.
- **Despliegue de la aplicación en un servicio de la nube**, esto es, la investigación de los diferentes servicios existentes y el despliegue de producción dentro del mismo.
- **Documentación**, haciendo alusión a todo el trabajo en el desarrollo de la memoria, utilizando la herramienta de Latex por medio de Overleaf

En primer lugar, nombraré las distintas tecnologías usadas y el coste de cada una de ellas:

Tecnología	Precio
Github	0€
API OPEN FOOD FACTS	0€
Quasar Framework	0€
Python	0€
MongoDB	0€ (puede variar)
Digital Ocean	5€ Mensuales (puede variar)

Tabla 7.1: Presupuesto tecnológico

En la siguiente tabla, se estimará el coste de la realización de cada tarea mencionada anteriormente.

	<b>Horas</b>	<b>Precio</b>	<b>€/h</b>
Análisis del estado del arte	45	540 €	12
Análisis de las herramientas de desarrollo full stack actuales	10	80 €	8
Estudio de Web Scraping y sus aplicaciones	80	2000 €	25
Diseño e implementación de la herramienta	80	1200 €	15
Despliegue de la aplicación en un servicio nube	10	150 €	15
Documentación	30	240 €	8
<b>TOTAL</b>	<b>255</b>	<b>4210</b>	<b>16,5</b>

En la última tabla, se calcula el coste presupuesto total por desarrollar el proyecto.

	<b>Total</b>
Tabla1	5 €
Tabla2	4210 €
	4215 €

Tabla 7.2: Presupuesto Total

# Bibliografía

- [1] Angular. *Angular Documentation*. <https://angular.io/guide/what-is-angular>. (Visitado 18-07-2021).
- [2] Axios. *Documentación de Axios*. <https://axios-http.com/docs/intro>. (Visitado 29-08-2021).
- [3] BEDCA. *Base de Datos Española de Composición de Alimentos (BEDCA)*. <https://www.bedca.net/>. (Visitado 05-06-2021).
- [4] BrowserStack. *Selenium IDE Tutorial : Definition, Features, Benefits*. <https://www.browserstack.com/guide/what-is-selenium-ide>. (Visitado 13-08-2021).
- [5] CalorieKing. *CalorieKing Home*. <https://www.calorieking.com/us/en/>. (Visitado 29-06-2021).
- [6] Carritus. *Comparar productos de Carrefour*. <https://www.carritus.com/supermercado/carrefour>. (Visitado 05-07-2021).
- [7] Colecciones en MongoDB. <https://docs.mongodb.com/manual/core/databases-and-collections/>. (Visitado 27-07-2021).
- [8] Comando Curl. <https://curl.se/>. (Visitado 22-08-2021).
- [9] DataMiner. *web scraping tool for professional data miners*. <https://dataminer.io/>. (Visitado 08-07-2021).
- [10] Django. *Django Documentación*. <https://www.djangoproject.com/>. (Visitado 15-07-2021).
- [11] Docker. *Empowering App Development for Developers*. <https://www.docker.com/>. (Visitado 01-09-2021).
- [12] Edaman. *Edaman Home*. <https://www.edamam.com/>. (Visitado 30-06-2021).
- [13] EuroFIR. *Welcome to EuroFIR AISBL Comments*. <https://www.eurofir.org/>. (Visitado 20-06-2021).
- [14] Express. *Express Documentation*. <https://expressjs.com/>. (Visitado 15-07-2021).
- [15] Open Food Facts. *Página oficial de Open Food Facts*. <https://es.openfoodfacts.org/>. (Visitado 03-09-2021).
- [16] GraaLVM. *GraaLVM*. <https://www.graalvm.org/>. (Visitado 30-08-2021).
- [17] Apache Groovy. *Groovy Documentation*. <https://groovy-lang.org/>. (Visitado 29-08-2021).
- [18] Red Hat. *What is a REST API?* <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. (Visitado 02-08-2021).
- [19] IBM. *What is Middleware?* <https://www.ibm.com/cloud/learn/middleware>. (Visitado 31-07-2021).

- [20] INFOODS. *FAO/INFOODS Food Composition Databases*. <http://www.fao.org/infoods/infoods/tables-and-databases/faoinfoods-databases/en/>. (Visitado 05-06-2021).
- [21] Servicio de Informática ASP.NET MVC 3. *Modelo vista controlador (MVC)*. <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. (Visitado 01-09-2021).
- [22] Agriculture Food Systems Institute. *Scientific Insights for Healthier Outcomes*. <https://foodsystems.org/>. (Visitado 20-06-2021).
- [23] Kotlin. *Kotlin Documentation*. <https://kotlinlang.org/>. (Visitado 30-08-2021).
- [24] Kubernetes. *Kubernetes Documentation*. <https://kubernetes.io/>. (Visitado 02-09-2021).
- [25] Lodash. *Lodash*. <https://lodash.com/>. (Visitado 02-09-2021).
- [26] MDN. *JavaScript*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. (Visitado 18-07-2021).
- [27] HTTP | MDN. *User-Agent - HTTP: MDN*. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>. (Visitado 10-08-2021).
- [28] Micronaut. *Micronaut Documentation*. <https://micronaut.io/>. (Visitado 29-08-2021).
- [29] Out of Milk. *The Grocery Shopping List App*. <https://www.outofmilk.com/>. (Visitado 07-07-2021).
- [30] MongoDB. *MongoDB Documentation*. <https://docs.mongodb.com/manual/reference/program/mongod/>. (Visitado 20-07-2021).
- [31] MySQL. *MySQL Documentation*. <https://dev.mysql.com/doc/>. (Visitado 20-07-2021).
- [32] Organización de las Naciones Unidas para la Alimentación y la Agricultura. *Bases de datos de composición de alimentos (BDCA)*. <http://www.fao.org/3/y4705s/y4705s01.pdf>. (Visitado 04-06-2021).
- [33] NEOLAND. *¿Qué es Full Stack?* <https://www.neoland.es/blog/que-es-desarrollador-full-stack>. (Visitado 03-09-2021).
- [34] Nestle. *Que són los nutrientes: macronutrientes y micronutrientes*. <https://nestlefamilyclub.es/articulo/que-son-los-nutrientes-macronutrientes-y-micronutrientes#>. (Visitado 02-09-2021).
- [35] NGINX. *High Performance Load Balancer, Web Server, and Reverse Proxy*. <https://www.nginx.com/>. (Visitado 27-08-2021).
- [36] NodeJS. *NodeJS Documentation*. <https://nodejs.org/en/docs/>. (Visitado 15-07-2021).
- [37] NPM. *Gestor de paquetes para JavaScript*. <https://www.npmjs.com/>. (Visitado 15-07-2021).
- [38] Digital Ocean. *Droplets: DigitalOcean's Scalable Virtual Machines*. <https://www.digitalocean.com/products/droplets/>. (Visitado 26-08-2021).
- [39] Digital Ocean. *The developer cloud*. <https://www.digitalocean.com/>. (Visitado 26-08-2021).
- [40] Ofertia. *Ofertas, catálogos online y tiendas*. <https://www.ofertia.com/>. (Visitado 05-07-2021).

- [41] Oorenji. *Nutrición de precisión*. <https://oorenji.com/>. (Visitado 30-06-2021).
- [42] Oracle. *Java HotSpot Virtual Machine*. <https://www.oracle.com/java/technologies/javase/vmoptions-jsp.html>. (Visitado 30-08-2021).
- [43] Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO). <http://www.fao.org/home/es/>. (Visitado 09-05-2021).
- [44] Pandas. *Documentación de Dataframes*. <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>. (Visitado 11-08-2021).
- [45] ParseHub. *The most powerful web scraper*. <https://www.parsehub.com/>. (Visitado 09-07-2021).
- [46] PostgreSQL. *PostgreSQL Documentation*. <https://www.postgresql.org/about/>. (Visitado 21-07-2021).
- [47] Postman. *Postman Documentation*. <https://learning.postman.com/docs/getting-started/introduction/>. (Visitado 20-08-2021).
- [48] Puppeteer. *Puppeteer Documentation*. <https://pptr.dev/>. (Visitado 18-08-2021).
- [49] Quarkus. *Quarkus Documentation*. <https://quarkus.io/>. (Visitado 29-08-2021).
- [50] Quasar. *Why Quasar?* <https://quasar.dev/introduction-to-quasar>. (Visitado 18-07-2021).
- [51] Radarprice. *Aplicación móvil Radarprice*. <https://play.google.com/store/apps/details?id=onyougo.price.scanner&hl=en&gl=US>. (Visitado 05-07-2021).
- [52] React. *React Documentation*. <https://reactjs.org/docs/getting-started.html>. (Visitado 18-07-2021).
- [53] Organización Mundial de la Salud (OMS). *Alimentación sana*. <https://www.who.int/es/news-room/fact-sheets/detail/healthy-diet>. (Visitado 18-04-2021).
- [54] SASS. *CSS with superpowers*. <https://sass-lang.com/>. (Visitado 20-07-2021).
- [55] Scrapy. *Scrapy 2.5 documentation*. <https://docs.scrapy.org/en/latest/>. (Visitado 13-08-2021).
- [56] Selenium. <https://www.selenium.dev/>. (Visitado 13-08-2021).
- [57] Beautiful Soup. *Beautiful Soup Documentation*. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. (Visitado 17-08-2021).
- [58] SoySuper. *Soysuper, tu supermercado online*. <https://soysuper.com/>. (Visitado 08-07-2021).
- [59] Spring. *Spring Documentation*. <https://spring.io/>. (Visitado 29-08-2021).
- [60] statickidz. *PHP API wrapper to http://www.bedca.net/bdpub/public foods*. <https://github.com/statickidz/bedca-api>. (Visitado 06-08-2021).
- [61] JSON Web Tokens. *JWT.IO*. <https://jwt.io/>. (Visitado 09-08-2021).
- [62] Tqdm. *Repositorio de la librería Tqdm*. <https://tqdm.github.io/>. (Visitado 12-08-2021).
- [63] Matthew Tyson. *Java Virtual Machine*. <https://www.infoworld.com/article/3272244-what-is-the-jvm-introducing-the-java-virtual-machine.html>. (Visitado 30-08-2021).



- [64] Element UI. *Element Ui Documentation*. <https://element.eleme.io/#/en-US>. (Visitado 21-07-2021).
- [65] *Unix y Linux*. [https://bioinf.comav.upv.es/courses/unix/unix\\_intro.html](https://bioinf.comav.upv.es/courses/unix/unix_intro.html). (Visitado 22-08-2021).
- [66] USDA. *U.S. Department of Agriculture Home*. <https://www.usda.gov/>. (Visitado 20-06-2021).
- [67] Vue. *Vue Documentation*. <https://vuejs.org/v2/guide/>. (Visitado 18-07-2021).
- [68] Webscraper.io. *Making web data extraction easy and accessible for everyone*. <https://webscraper.io/>. (Visitado 08-07-2021).
- [69] Wikipedia. *ECMAScript*. <https://es.wikipedia.org/wiki/ECMAScript>. (Visitado 18-07-2021).
- [70] Wikipedia. *Web scraping*. [https://en.wikipedia.org/wiki/Web\\_scraping](https://en.wikipedia.org/wiki/Web_scraping). (Visitado 02-09-2021).
- [71] Wikipedia. *XMLHttpRequest*. <https://es.wikipedia.org/wiki/XMLHttpRequest>. (Visitado 02-09-2021).
- [72] Xataka. *API: que és y para qué sirve*. <https://www.xataka.com/basics/api-que-sirve>. (Visitado 10-08-2021).
- [73] Yuka. *The mobile app that scans your diet and cosmetics*. <https://yuka.io/en/>. (Visitado 09-07-2021).