# Estrutuctura del backend

La aplicación está compuesta por un directorio principal “CESTA-VERDE-SERVER” en el que se encuentran varios subdirectorios.

# Subdirectorios:

## Src

## apiServices

## auth

## bin

## config

## microServices

## middleware

## routes

## services

## testes

# Funcionamiento del backend

Para poner en marcha el backend, desde la terminal de comandos, accedemos a la raíz del proyecto y utilizamos el comando ***npm start***.

Durante el desarrollo del backend, hemos utilizado la plataforma API para construir y usar una API. Postman incluye un conjunto integral de herramientas que ayudan a acelerar el ciclo de vida de la API, desde el diseño, las pruebas, la documentación y la simulación hasta el uso compartido y la detección de sus API. En nuestro caso nos fue bastante útil para probar el correcto funcionamiento de los métodos implementados en nuestro backend.

## Registro del usuario

Para dar de alta un usuario en la plataforma, se necesitan las informaciones las informaciones como: nombre y apellidos, email, nombre de usuario y la contraseña.

Desde la plataforma Postman, hacemos una petición de tipo POST al backend mediante la ruta ***http://localhost:3000/api/v1/user****.* El cuerpo de esta petición debe ser un documento json que contenga los datos del usuario mencionados anteriormente tal como se puede observar en la siguiente imagen.

![Texto

Descripción generada automáticamente]()

**Ilustración 1:** Registro del usuario utilizando el entorno de desarrollo Postman. Fuente: Elaboración propia

Si todos los datos han sido introducidos correctamente, tras haber envido la petición al backend (haciendo clic en botón *Send*), se registra el usuario y la plataforma nos devuelve una respuesta de tipo json que contiene el estatus, cuerpo con el mensaje de creación y el error que puede ser true o false.

![Interfaz de usuario gráfica, Texto

Descripción generada automáticamente]()

**Ilustración 2:** Respuesta de la creación del usuario. Fuente: Elaboración propia

El email y el nombre del usuario deben ser únicos para cada usuario. Veamos un caso de uso en donde intentamos realizar el registro de un usuario con el nombre de usuario o email existentes.

![Captura de pantalla de un celular

Descripción generada automáticamente]()

**Ilustración 3:** Creación del usuario con el email existente. Fuente: Elaboración propia

![Captura de pantalla de un celular

Descripción generada automáticamente]()

**Ilustración 4:** Creación del usuario con el nombre de usuario existente. Fuente: Elaboración propia

## Login

Todos los usuarios registrados en la plataforma pueden hacer el login para realización de diversas actividades en función del rol que se les ha otorgado.

Para realizar el login, enviamos una petición de tipo POST al backend mediante la ruta ***http://localhost:3000/api/v1/login*** con los datos de autenticación del usuario (en este caso el usuario e la contraseña). Si el usuario existe en la base de datos de la plataforma devolverá los datos de este y el token de autenticación que se podrá usar desde el frontend para comprar la seguridad y el tiempo de duración que el usuario estará autenticado. Si el usuario no existe, la plataforma mostrará un mensaje de error especificando que los datos son incorrectos. Veamos un ejemplo de cada caso.

![Texto

Descripción generada automáticamente]()

**Ilustración 5:** Autenticación con los datos existentes. Fuente: Elaboración propia

![Captura de pantalla de un celular

Descripción generada automáticamente]()

**Ilustración 6:** Autenticación con los datos de usuario incorrectos. Fuente: Elaboración propia

![Captura de pantalla de un celular

Descripción generada automáticamente]()

**Ilustración 7:** Autenticación sin datos del usuario. Fuente: Elaboración propia

## Registro del negocio

Tal como se ha mostrado anteriormente en el diseño de la base de datos, un usuario puede tener uno o muchos negocios. Para efectuar el registro de un negocio a la plataforma, enviamos una petición de tipo POST al backend a través de la ruta ***http://localhost:3000/api/v1/business.*** El cuerpo de esta petición debe ser un json que contenga los datos del negocio (nombre del negocio, descripción, dirección, coordenadas para la localización en el mapa y el identificador del usuario). Si todos los campos han sido introducidos de forma correcta y se comprueba que existe el identificador del usuario introducido, se creará el negocio y el sistema devolverá una respuesta con el estatus 201 y el mensaje de creación satisfactoria del negocio.

![Interfaz de usuario gráfica, Texto

Descripción generada automáticamente]()

**Ilustración 8:** Creación del negocio. Fuente: Elaboración propia

Podemos obtener un listado con los negocios registrados en la plataforma enviando una petición de tipo GET al backend a través de la ruta ***http://localhost:3000/api/v1/business.*** La repuesta incluye tanto los datos del negocio como los datos del usuario dueño del negocio.

![Texto

Descripción generada automáticamente]()

**Ilustración 9:** Listado de los negocios registrados en la plataforma. Fuente: Elaboración propia

Si hacemos la misma petición al backend, pero a la ruta le añadimos el identificador del usuario, obtendríamos todos los negocios correspondientes al usuario con dicho identificador.

![Texto

Descripción generada automáticamente]()

**Ilustración 10:** Negocios registrado a un usuario especifico. Fuente: Elaboración propia

## Registro de categorías

En nuestra plataforma, las categorías son importantes a la hora de registrar los productos. Esto permitirá a los negocios tener una agrupación mas controlada de sus productos. Las categorías podrían ser: frutas, verduras, vegetales etc.

Para registrar una categoría enviaremos una petición de tipo POST al backend a través de la ruta ***http://localhost:3000/api/v1/category.*** En este caso se requiere menos información, dado que solo necesitamos conocer el nombre de la categoría, el cuerpo de esta petición debe ser un json que contenga esta información.

![Captura de pantalla de un celular

Descripción generada automáticamente]()

**Ilustración 11:** Registrar categoria. Fuente: Elaboraci]on propia

Podemos obtener una categoría específica haciendo una petición de tipo GET al servidor a través de la ruta añadiendo a la ruta el identificador de la categoría que queramos.

![Interfaz de usuario gráfica, Texto

Descripción generada automáticamente]()

**Ilustración 12:** Categoría obtenida a través del id. Fuente: Elaboración propia

Podemos obtener el listado de todas las categorías registradas en la plataforma enviando una solicitud de tipo GET al servidor a través de la ruta ***http://localhost:3000/api/v1/category.***

![Texto

Descripción generada automáticamente]()

**Ilustración 13:** Listado de categorías. Fuente: Elaboración propia

# Registro de productos

Una vez creada la categoría y el negocio, el usuario puede proceder con la creación del producto. Para ello hacemos una llamada de tipo POST a nuestro servidor a través de la ruta ***http://localhost:3000/api/v1/product/*** y en el cuerpo de esta petición especificamos los siguientes atributos del producto: nombre del producto, precio, moneda, unidad de venta, descripción del producto, categoría y el negocio.

Si los datos han sido introducidos correctamente y se valida la existencia de los que sean necesarios, el sistema devolverá la respuesta con el estatus 201 y el mensaje de creación del producto.

![Texto

Descripción generada automáticamente]()

**Ilustración 14:** Creación del producto. Fuente: Elaboración propia

Para obtener el listado de todos los productos registrados en la plataforma, hacemos una petición de tipo GET a nuestro servidor a través de la ruta [***http://localhost:3000/api/v1/product/***](http://localhost:3000/api/v1/product/)***.*** En la respuesta se podrá observar los datos del producto y los datos del negocio en donde se registró cada producto.

![Texto

Descripción generada automáticamente]()

**Ilustración 15:** Obtener listado de los productos registrados en la plataforma. Fuente: Elaboración propia

De la misma forma, se puede obtener la información de un determinado producto enviando una petición de tipo GET a nuestro servidor a través de la ruta [***http://localhost:3000/api/v1/product/product\_id***](http://localhost:3000/api/v1/product/product_id)***,*** en este caso añadimos a la ruta el identificador del producto que queremos obtener su información, tal como se observa en la siguiente imagen.

![Texto

Descripción generada automáticamente]()

**Ilustración 16:** Método getProduct para obtener la información de un producto específico. Fuente: Elaboración propia.

# Obtención de la información nutricional del producto

Para proporcionar información nutricional de un producto, hemos utilizado la API Nutrition API de API Ninjas. Esta API extrae información nutricional del texto (en nuestro caso, el nombre del producto) mediante procesamiento de lenguaje natural. Desde blogs de comida hasta menús y recetas, puede leer cualquier texto y calcular los datos nutricionales correspondientes. Una característica inteligente de esta API es el porcionado personalizado: si el texto especifica cantidades de alimentos o ingredientes individuales, el algoritmo escalará automáticamente los datos nutricionales en el resultado en consecuencia.

Para utilizar la API en referencia, es necesario la Api-Key de la misma q que se obtiene al registrarse en la aplicación. A continuación, la codificación para la conexión de nuestro servidor backend con la API.

![Texto

Descripción generada automáticamente]()

**Ilustración 17:** Conexión con la API para obtener la información nutricional de cada producto. Fuente: Elaboración propia

Dado que no todos los nombres de alimentos en español son reconocidos por la API, hemos implementado un método traductor, que dado el nombre de un producto en español lo traduce en inglés y, la llamada al método GET de la API se hace mediante la traducción en inglés.

El método GET devuelve información nutricional correspondiente a cada 100 gramas del producto, tales como la cantidad de calorías, grasas, colesterol, carbohidratos, proteínas, vitaminas y minerales en la porción indicada del alimento.

Para obtener la información nutricional de un determinado producto desde nuestro servidor, enviamos una petición de tipo GET a través de la ruta http://localhost:3000/api/v1/product/nutritionalInfo y en el cuerpo de la petición, colocamos el nombre del producto.

![Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente]()

**Ilustración 18:** Información nutricional del producto. Fuente: Elaboración propia.