



MANUAL TECNICO

SISTEMA DE GESTIÓN DE PIZZAS

VERSIÓN 1.0

Enlace:

<https://desafio-junior-fullstack.vercel.app/>

REALIZADO POR: LUIS FERNANDO CUJI LLUCO

RIOBAMBA – ECUADOR

2025

1. Introducción

Este proyecto consiste en una aplicación de pizzeria con un backend en Node.js y un frontend en React. El objetivo de este manual es proporcionar una guía técnica para la instalación, ejecución y pruebas de la aplicación.

2. Requisitos del Sistema

Backend:

Node.js (versión 14 o superior)

npm o yarn

Express.js

Express-validator (para validación de datos)

Jest y Supertest (para pruebas)

Frontend:

React.js

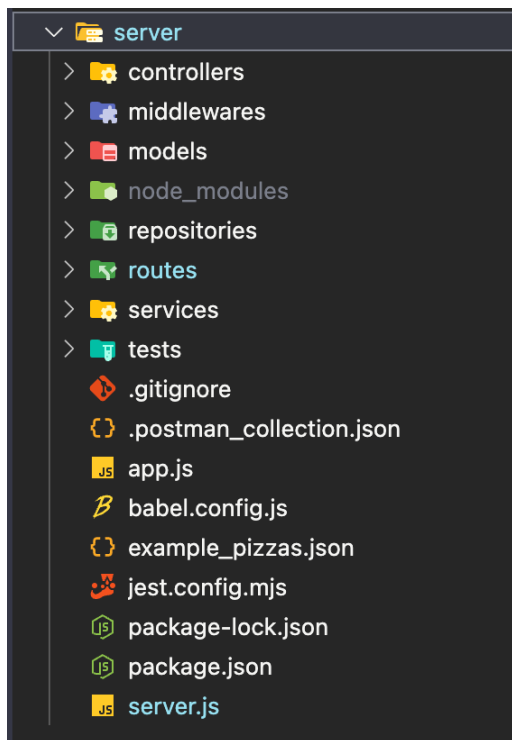
Axios (para realizar peticiones HTTP)

React Testing Library y Jest (para pruebas)

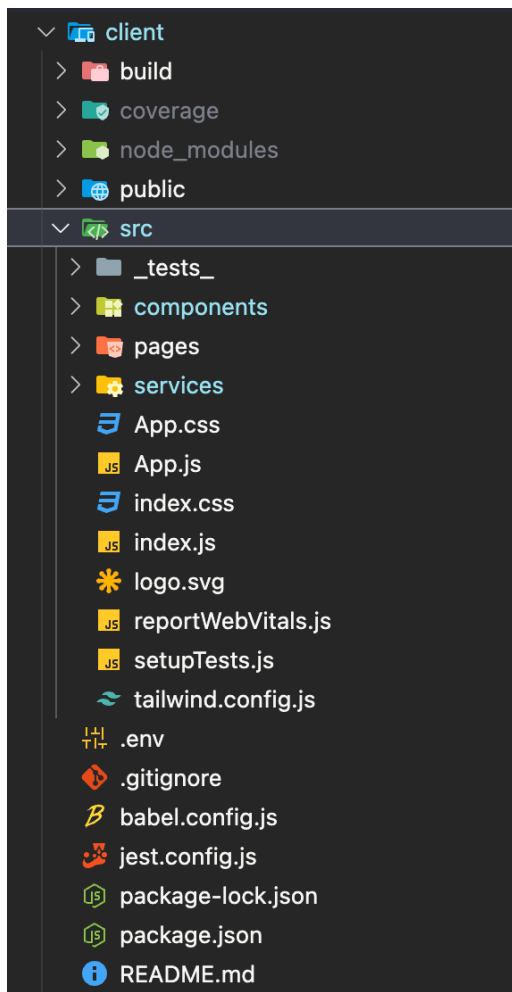
3. Estructura del Proyecto

La estructura del proyecto está organizada de la siguiente manera:

Backend:



Frontend:



4. Instalación

Backend:

1. **Clona el repositorio:**

- `git clone https://github.com/Luisfer593/desaf-o-jr_fullstack_Luis_C`
- `cd backend`

2. **Instala las dependencias:**

- `npm install`

3. **Corre el servidor:**

- `npm start`

El servidor debería estar corriendo en `http://localhost:4000`.

Frontend:

4. **Clona el repositorio:**

- `git clone https://github.com/Luisfer593/desaf-o-jr_fullstack_Luis_C`
- `cd frontend`

5. **Instala las dependencias:**

- `npm install`

6. **Ejecuta la aplicación:**

- `npm start`

La aplicación debería abrirse en `http://localhost:3000`.

5. Pruebas

Pruebas Backend:

7. Asegurarse de que el servidor esté en funcionamiento.
8. Correr las pruebas unitarias con:
 - `npm test`

Pruebas Frontend:

1. Asegúrate de que la aplicación frontend esté en funcionamiento.
2. Corre las pruebas con:
 - `npm test`

6. Endpoints de la API

GET /api/pizzas: Obtiene la lista de pizzas disponibles.

GET /api/orders: Obtiene la lista de pedidos realizados.

POST /api/orders: Realiza un nuevo pedido. El cuerpo de la solicitud debe contener:

```
{
  "items": [
    {
      "pizzaName": "Margherita",
      "quantity": 2
    },
    {
      "pizzaName": "Diavola",
      "quantity": 1
    }
  ]
}
```

7. Validaciones

POST /api/orders: Se valida que el campo items esté presente y que cada elemento tenga un pizzaName y quantity > 0.

8. Tecnologías Usadas

Backend: Node.js, Express.js, Express-validator, Jest, Supertest.

Frontend: React.js, Axios, React Testing Library, Jest.

9 Despliegue

Despliegue en Vercel (Frontend)

Conectar tu repositorio a Vercel:

- Inicia sesión en Vercel o crea una cuenta.
- Desde el dashboard, haz clic en "New Project" y selecciona tu repositorio de GitHub (o conecta tu cuenta si no lo has hecho aún).

Configurar el despliegue:

- Vercel detectará automáticamente que estás usando React y configurará el entorno de manera predeterminada.
- Verifica que tu archivo package.json esté correctamente configurado con los scripts de build y start.

Desplegar:

- Hacer clic en "Deploy" para iniciar el proceso de despliegue.
- Una vez completado, obtendremos una URL pública para el frontend: <https://desafio-junior-fullstack.vercel.app/>

Despliegue en Railway (Backend)

Conectar tu repositorio a Railway:

- Iniciar sesión en Railway o crear una cuenta.
- Hacer clic en "New Project" y seleccionar "Deploy from GitHub".
- Conectar el repositorio de backend con Railway.

Configurar el entorno:

- Railway detectará automáticamente que el proyecto es un backend de Node.js.
- Si necesitamos agregar variables de entorno, podemos hacerlo desde la sección Settings > Variables. Agregar la variable REACT_APP_API_URL con el valor de la API desplegada (<https://serverpizza.up.railway.app/api>).

Desplegar:

- Hacer clic en "Deploy" para iniciar el proceso de despliegue.
- Una vez completado, obtendremos una URL pública para el backend: <https://serverpizza.up.railway.app/api>

Conexión entre el Frontend y el Backend

- En Vercel, asegurarse de que el archivo .env del frontend contenga la URL correcta del backend. La variable `REACT_APP_API_URL` debe estar configurada con la URL de Railway:

`REACT_APP_API_URL=https://serverpizza.up.railway.app/api`

- Después de hacer estos ajustes, el frontend (en Vercel) se comunicará correctamente con el backend desplegado en Railway.