

Plan de pruebas

El presente documento contiene el Plan de Testing para la aplicación web bajo prueba. El objetivo principal de este plan es garantizar la calidad y el correcto funcionamiento de los endpoints proporcionados. El plan incluye una serie de pruebas unitarias que abarcan diferentes escenarios para evaluar la funcionalidad de la aplicación.

Utilizaremos Thunder Client como cliente HTTP para ejecutar las pruebas.

Guía de configuración del ambiente de pruebas

Instalación de Node.js

Descarga e instala Node.js en tu sistema operativo siguiendo las instrucciones correspondientes a tu plataforma.

Clonar el repositorio de la aplicación

Clona el repositorio de la aplicación desde el repositorio remoto.

Navega hasta el directorio del proyecto en tu sistema local.

Instalación de dependencias

Abre una terminal o línea de comandos y navega hasta el directorio del proyecto.

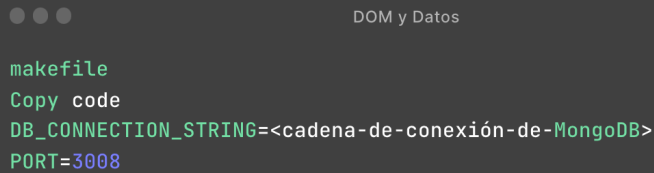
Ejecuta el comando `npm install` para instalar todas las dependencias especificadas en el archivo `package.json`.

Configuración del archivo de entorno

Crea un archivo `.env` en el directorio raíz del proyecto.

En este archivo, define las variables de entorno necesarias para la conexión a la base de datos MongoDB y otros valores de configuración según sea necesario.

Contenido del archivo .env



```
makefile
Copy code
DB_CONNECTION_STRING=<cadena-de-conexión-de-MongoDB>
PORT=3008
```

Configuración de la conexión a la bb.dd. remota

Accede a tu clúster de MongoDB remoto y obtén la cadena de conexión necesaria para conectar la aplicación con la base de datos remota.

Copia la cadena de conexión y reemplaza **<cadena-de-conexión-de-MongoDB>** en el archivo .env con la cadena de conexión correspondiente.

Ejecución de la aplicación en modo de prueba

En la terminal, ejecuta el comando `npm start` o `node app.js` para iniciar la aplicación en modo de prueba.

Verifica que la aplicación se esté ejecutando correctamente sin errores.

Configuración de Thunder Client

Instala Thunder Client, una extensión de cliente HTTP, en tu navegador web preferido.

Abre Thunder Client y crea una nueva carpeta para el proyecto de pruebas.

Agrega una nueva solicitud para cada prueba unitaria, siguiendo los detalles y las instrucciones proporcionadas en el Plan de Testing.

Ejecución de las pruebas unitarias

Ejecuta cada prueba unitaria en Thunder Client según las instrucciones proporcionadas en el Plan de Testing.

Verifica los resultados de cada prueba y asegúrate de que se cumplan las expectativas especificadas.

A continuación, se detallan los pasos a seguir para cada una de las pruebas, que abarcan desde la conexión al servidor hasta la manipulación de datos relacionados con las frutas.

Cada prueba especifica el método HTTP, el endpoint correspondiente y las instrucciones detalladas para ejecutarla. El plan de testing se centra en verificar la correcta respuesta de la aplicación a diferentes solicitudes, asegurando así su robustez y rendimiento.

Pasos

1. Prueba de conexión al servidor:

- Método: GET
- Endpoint: **http://localhost:3008/**
- Instrucciones: En Thunder Client, ejecuta una solicitud GET a la URL mencionada para verificar si se establece correctamente la conexión con el servidor. Deberías recibir una respuesta exitosa (código de estado 200) junto con un mensaje de bienvenida u otra información relevante.

2. Prueba de obtener todas las frutas:

- Método: GET
- Endpoint: **http://localhost:3008/frutas**
- Instrucciones: Envía una solicitud GET a la URL proporcionada para obtener todas las frutas disponibles en la aplicación. Verifica si recibes una respuesta exitosa (código de estado 200) y asegúrate de que la respuesta incluya un listado de frutas.

3. Prueba de obtener una fruta por ID:

- Método: GET
- Endpoint: **http://localhost:3008/frutas/:id** (reemplaza ":id" con un ID válido)
- Instrucciones: Realiza una solicitud GET a la URL especificada, reemplazando ":id" con el ID de una fruta existente. Comprueba si recibes una respuesta exitosa (código de estado 200) y verifica que la respuesta incluya los detalles de la fruta correspondiente.

4. Prueba de obtener una fruta por nombre:

- Método: GET
- Endpoint: **http://localhost:3008/frutas/nombre/:nombre** (reemplaza ":nombre" con un nombre válido)
- Instrucciones: Envía una solicitud GET a la URL mencionada, reemplazando ":nombre" con el nombre de una fruta existente. Verifica si recibes una respuesta exitosa (código de estado 200) y

asegúrate de que la respuesta incluya los detalles de la fruta correspondiente al nombre especificado.

5. Prueba de obtener frutas por importe:

- Método: GET
- Endpoint: **http://localhost:3008/frutas/importe/:precio** (reemplaza ":precio" con un precio válido)
- Instrucciones: Realiza una solicitud GET a la URL proporcionada, reemplazando ":precio" con un precio válido. Comprueba si recibes una respuesta exitosa (código de estado 200) y verifica que la respuesta incluya las frutas que cumplan con el precio especificado.

6. Prueba de crear una nueva fruta:

- Método: POST
- Endpoint: **http://localhost:3008/frutas/**
- Instrucciones: Envía una solicitud POST a la URL mencionada para crear una nueva fruta. Incluye los datos necesarios en el cuerpo de la solicitud, como el nombre, el precio, etc. Verifica si recibes una respuesta exitosa (código de estado 200) y asegúrate de que la respuesta incluya los detalles de la fruta recién creada.

7. Prueba de actualizar una fruta existente:

- Método: PUT
- Endpoint: **http://localhost:3008/frutas/:id** (reemplaza ":id" con un ID válido)
- Instrucciones: Realiza una solicitud PUT a la URL especificada, reemplazando ":id" con el ID de una fruta existente. Incluye los datos actualizados en el cuerpo de la solicitud. Verifica si recibes una respuesta exitosa (código de estado 200) y asegúrate de que la