

README: Prueba Técnica en Spring Boot y MongoDB

Descripción del Proyecto

Este proyecto es una aplicación web construida con Spring Boot que gestiona productos a través de una base de datos MongoDB. El controlador `ProductoControlador` expone varios endpoints para realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre los productos. La aplicación está configurada para conectarse a MongoDB en el puerto 9080.

Endpoints

1. Crear un Nuevo Producto

Endpoint: `POST /Productos/crear`

Este endpoint permite crear un nuevo producto. Envía los detalles del producto en el cuerpo de la solicitud en formato JSON.

Ejemplo de Solicitud:

```
{
  "nombre": "Producto 1",
  "precio": 100.0,
  "descripcion": "Descripción del Producto 1"
}
```

Respuesta Exitosa:

- **Código:** 201 Created
- **Cuerpo:** Contiene el producto creado con su ID generado automáticamente.

2. Obtener Todos los Productos (con Paginación)

Endpoint: `GET /Productos`

Este endpoint devuelve una lista paginada de productos. Si la cantidad de productos supera los 10, se puede usar paginación mediante parámetros en la URL.

Parámetros de Consulta:

- **page:** Número de página (por defecto es 0).
- **size:** Tamaño de la página (por defecto es 10).

Ejemplo de Solicitud:

```
GET /Productos?page=1&size=10
```

Respuesta Exitosa:

- **Código:** 200 OK
- **Cuerpo:** Contiene una lista de productos para la página solicitada.

Respuesta cuando no hay productos:

- **Código:** 204 No Content

3. Obtener un Producto por ID

Endpoint: GET /Productos/leer/{id}

Este endpoint permite obtener los detalles de un producto específico mediante su ID.

Ejemplo de Solicitud:

```
GET /Productos/leer/64c89fb8f793de0c23a8e723
```

Respuesta Exitosa:

- **Código:** 200 OK
- **Cuerpo:** Contiene los detalles del producto.

Respuesta cuando el producto no se encuentra:

- **Código:** 404 Not Found

4. Actualizar un Producto

Endpoint: PUT /Productos/actualizar/{id}

Este endpoint permite actualizar los detalles de un producto existente. El ID del producto se especifica en la URL y los nuevos datos del producto en el cuerpo de la solicitud en formato JSON.

Ejemplo de Solicitud:

```
{  
  "nombre": "Producto Actualizado",  
  "precio": 150.0,  
  "descripcion": "Nueva descripción"  
}
```

Respuesta Exitosa:

- **Código:** 200 OK
- **Cuerpo:** Contiene los detalles del producto actualizado.

Respuesta cuando el producto no se encuentra:

- **Código:** 404 Not Found

5. Eliminar un Producto por ID

Endpoint: DELETE /Productos/eliminar/{id}

Este endpoint permite eliminar un producto específico mediante su ID. La eliminación requiere una confirmación que se pasa como parámetro en la URL.

Parámetro de Consulta:

- **confirm:** Debe ser true para proceder con la eliminación.

Ejemplo de Solicitud:

```
DELETE /Productos/eliminar/64c89fb8f793de0c23a8e723?confirm=true
```

Respuesta Exitosa:

- **Código:** 204 No Content

Respuesta cuando no se confirma:

- **Código:** 400 Bad Request

Respuesta cuando el producto no se encuentra:

- **Código:** 500 Internal Server Error

6. Eliminar Todos los Productos

Endpoint: DELETE /Productos/eliminar/todo

Este endpoint permite eliminar todos los productos en la base de datos. La eliminación también requiere una confirmación que se pasa como parámetro en la URL.

Parámetro de Consulta:

- **confirm:** Debe ser true para proceder con la eliminación.

Ejemplo de Solicitud:

```
DELETE /Productos/eliminar/todo?confirm=true
```

Respuesta Exitosa:

- **Código:** 204 No Content

Respuesta cuando no se confirma:

- **Código:** 400 Bad Request

Respuesta cuando ocurre un error:

- **Código:** 500 Internal Server Error

7. Manejo de Rutas No Existentes

Endpoint: *

Este endpoint captura cualquier solicitud a rutas no definidas en la aplicación y devuelve un error.

Respuesta:

- **Código:** 404 Not Found
- **Cuerpo:** "Error: Endpoint no encontrado."

Cómo Probar la Aplicación

1. **Ejecución:** Asegúrate de que MongoDB esté corriendo en el puerto 9080.
2. **Acceso a Endpoints:** Puedes probar los endpoints usando herramientas como Postman o Swagger.
3. **Paginación:** Para listas grandes de productos, usa los parámetros page y size en la URL para obtener diferentes páginas de resultados.
4. **Confirmación al Eliminar:** Recuerda agregar el parámetro confirm=true en la URL para eliminar productos, de lo contrario, recibirás un error 400 Bad Request.

