**Documentación de la API AbmApi**

La API AbmApi proporciona servicios para la gestión de recursos relacionados con el sistema de ABM (Alta, Baja y Modificación).

**Ejecución del proyecto**

Para ejecutar el proyecto, sigue los siguientes pasos:

Asegúrate de tener instalado Java Development Kit (JDK) en tu máquina.

Descarga o clona el proyecto AbmApi desde este enlace <https://github.com/Aldoarevalo/AbmApi.git>.

Abre el proyecto con un editor de codigo por ejemplo Spring Boot tool Suite: Click derecho sobre el proyecto luego Run as Maven install

Para ejecutar el proyecto desde la linea de comandos, utiliza el siguiente comando:

java -jar target/Abm-1.0.0.jar Esto iniciará la aplicación y estará disponible en la URL <http://localhost:8080/swagger-ui.html>.

**Ejecución de pruebas unitarias**

Las pruebas unitarias se pueden ejecutar utilizando la herramienta Spring Boot Tool 4 en Spring Tool Suite. Sigue estos pasos:

Abre el proyecto en Spring Tool Suite.

Navega hasta la carpeta "src/test/java" en la vista del proyecto.

Haz clic derecho sobre la clase de prueba que deseas ejecutar y selecciona "Run As" > "JUnit Test".

Esto ejecutará la prueba unitaria y mostrará los resultados en la consola.

**Mejores prácticas en la construcción de servicios**

En la construcción de los servicios de la API AbmApi, se han seguido las siguientes mejores prácticas:

Se han utilizado patrones de diseño para garantizar la modularidad y la separación de responsabilidades en el código.

Los recursos se han definido claramente, especificando sus propiedades, relaciones y acciones asociadas.

Se ha implementado un sistema de versionamiento para permitir una evolución controlada de la API sin afectar a los clientes existentes.

Se han utilizado los verbos HTTP adecuados para cada acción sobre los recursos.

Se ha implementado funcionalidad de filtrado, ordenamiento y paginación para facilitar la consulta eficiente de los recursos.

El manejo de excepciones se ha realizado de manera adecuada, proporcionando respuestas claras y coherentes en caso de errores.

Los códigos de estado HTTP se han utilizado correctamente para indicar el resultado de cada operación realizada en la API.

Se ha implementado un mecanismo de autenticación seguro para proteger los recursos de la API.

Se han diseñado y construido pruebas unitarias para verificar el correcto funcionamiento de la API.

El código ha sido desarrollado siguiendo buenas prácticas de programación, con énfasis en la legibilidad, coherencia en la nomenclatura y mantenibilidad.

Las interfaces expuestas por la API han sido diseñadas de manera intuitiva y fácil de entender para los clientes.

El código es usable, mantenible y extensible, permitiendo realizar cambios y agregados sin generar efectos no deseados.

Se ha buscado la eficiencia y la baja complejidad en el diseño y la implementación de la solución.

Se han seguido estándares ampliamente aceptados, como RESTful, para el diseño y la implementación de la API.

Se ha buscado el uso mínimo de dependencias externas para mantener la simplicidad y evitar posibles problemas de compatibilidad.

Se han implementado pruebas unitarias para garantizar la calidad y confiabilidad del código.

**Test y ejemplos**

El proyecto AbmApi incluye pruebas unitarias para verificar el correcto funcionamiento de los diferentes componentes de la API. Estas pruebas se encuentran en la carpeta "src/test/java" y se pueden ejecutar siguiendo las instrucciones mencionadas anteriormente.

Además, se recomienda revisar los ejemplos de uso de la API en la carpeta "examples". Estos ejemplos proporcionan casos de uso comunes y muestran cómo interactuar con los diferentes recursos de la API.

**Documentación y código claro, auto explicado**

Se ha puesto especial énfasis en proporcionar una documentación clara y concisa para el proyecto AbmApi. Además, se ha trabajado en mantener un código limpio y autoexplicativo, siguiendo buenas prácticas de programación.

Cada clase, método y variable ha sido nombrado de manera significativa, facilitando su comprensión y manteniendo una estructura coherente en todo el proyecto.

Además, se han agregado comentarios descriptivos en el código para brindar una explicación adicional sobre la funcionalidad y el propósito de cada componente.

**Conclusiones**

La API AbmApi ha sido desarrollada siguiendo las mejores prácticas en la construcción de servicios, con una clara definición de recursos, versionamiento adecuado, uso de verbos HTTP correctos, implementación de funcionalidades como filtrado y paginación, manejo adecuado de excepciones, uso correcto de los códigos de estado HTTP, autenticación segura, pruebas unitarias, calidad de programación y diseño e implementación clara de la solución.

Además, se ha prestado atención a la documentación y el código claro, auto explicado, para facilitar su comprensión y uso por parte de los desarrolladores.