

Fecha:	
Nombre:	
Cédula:	
Calificación:	

Prueba Técnica: API Activos Fijos

Los activos fijos de una empresa son los bienes materiales, son necesarios para el funcionamiento de la empresa y no se destinan para la venta, ejemplos de estos son: bienes inmuebles, maquinaria, material de oficina, etc.

Estos activos fijos tienen características comunes entre todas como son: nombre, descripción, tipo, serial, numero interno de inventario, peso, alto, ancho, largo, valor compra, fecha de compra.

Los activos se pueden asignar a un área o persona, si es a un área esta se encuentra en una ciudad (diferentes áreas se encuentran en diferentes ciudades).

Diseñar una API RESTFul¹ que permite administrar los activos, esta debería permitir:

- Buscar los activos por: tipo, fecha de compra, serial.
- Crear nuevos activos.
- Actualizar activos.

¹ Servicios WEB REST

Tecnologías

Debe utilizar las siguientes tecnologías:

- Plataforma Java 8 o superior.
- Framework Spring Boot.

Tecnologías extras

Puede utilizar alguna de las siguientes tecnologías:

- Servidor de aplicaciones Tomcat, Wildfly o Glassfish.
- Maven, Gradle u otro.

Motor de base de datos

Es libre de utilizar cualquier motor de base de datos relacional, pero se recomienda MYSQL o Postgres. Si decide utilizar tecnologías NoSQL se requiere justificar la selección de motor y ventajas.

Reglas

- No debe haber excepciones sin capturar.
- Las respuestas HTTP de la API deben estar estandarizadas a:
 - 200 Por consultas exitosas
 - 400 Para reportar datos faltantes al crear o actualizar un activo.
 - 404 para búsquedas sin resultados.
 - 500 Para errores que pasen en la capa de backend.
 - Todas las respuestas deben contener una descripción o resumen de lo ocurrido.



Entregables

- Archivo de despliegue o ejecutable (WAR, EAR o JAR).
- Documento en formato PDF con los siguientes items:
 - Modelo de datos: puede ser un modelo de clases con las entidades o un modelo relacional.
 - Instrucciones para la construcción y generación del archivo de despliegue o ejecutable.
 - Instrucciones para el despliegue y ejecución de la aplicación.
- Archivo de pruebas en POSTMAN.

Puntos adicionales

Los siguientes puntos no son requeridos en su solución, pero si los decide incluir serán tenidos en cuenta para mejorar el escalafón.

- Documentación de la API: esta puede ser realizada manualmente o utilizando una herramienta como RAML, Swagger u otra.
- Documentación Java (JAVADOC)
- Geolocalización.
- Soporte de autenticación por token de alguno de los servicios REST.
- Demostrar el manejo de internacionalización (i18n).
- Demostrar el manejo de librería de Log (log4j, logback u otra).
- Pruebas Unitarias (JUnit, TestNG u otra).
- Frontend que consuma uno de los servicios utilizando alguna tecnología web como por ejemplo Angular, React u otro.
- Crear archivos Dockerfile para el proyecto.

