

Prueba Técnica: API Activos Fijos.

Los activos de 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, fecha de baja, estado actual (activo, dado de baja, en reparación, disponible, asignado), color.

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 RESTFul que permite administrar los activos, esta debería permitir:

- Buscar por todos los activos.
- Buscar los activos por: tipo, fecha de compra, serial.
- Crear nuevos activos.
- Actualizar activos.
 - Permite cambiar el serial interno y fecha de baja

Plataforma:

Es libre de seleccionar como base de su solución alguna de las siguientes tecnologías:

- Spring
- Spring + JEE
- JEE

Tecnologías Extras

Puede utilizar alguna de las siguientes tecnologías

- JAVA 7 o superior
- Spring 4
- Tomcat / Glassfish / Payara
- Maven, Gradle

Base de datos:

Es libre de utilizar cualquier motor de base de datos relacional, pero se recomienda

Postgres o MYSQL si decide utilizar tecnologías **NoSQL** se requiere justificar la selección de motor y ventajas.

PLUS

Los siguientes puntos no son requeridos en su solución, pero si los decide incluir serán tenidos muy en cuenta.

- Documentación de la API (RAML, Swagger)
- Documentación Java (JAVADOC)
- Internacionalización (i18n)
- Respuestas múltiples de la API (JSON, XML).
- Geolocalización.
- Log (log4j)
- Pruebas Unitarias (JUnit).
- Cliente javascript (Angular, React, etc), que consuma algún servicio.

Reglas

- El código debe estar versionado en GIT o Subversion, siguiendo algún modelo de entrega. (se sugiere GIT, basado en un flujo de rama por característica).
- se deben validar las fechas (nunca una fecha de baja debe ser superior a una fecha de compra)
- no debe haber excepciones sin capturar.
- Las respuestas 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 que paso.

Entregables:

- Repositorio con el código donde se puedan ver los commit realizados.
- WAR, EAR, JAR, con las instrucciones de generación.
- Instrucciones de despliegue.
- Archivo de pruebas en POSTMAN.

Tiempo

- lo define el aplicando, pero se sugiere que no sea superior a 5 días.