**Evaluación Java Spring Boot**

Preguntas.

1.Cuál es la diferencia entre un controlador y un DAO en el contexto de la aplicación Spring.

Un controlador se usa para exponer endpoints

Un dao para persistencia de información en bases de datos

2. Como se configura SpringBoot para que pueda devolver datos en formatos XML o JSON desde una misma API REST.

Con anotaciones en los endpoints

Y las siguientes propiedades:

spring.mvc.contentnegotiation.favor-path-extension=false

spring.mvc.contentnegotiation.favor-parameter=true

spring.mvc.contentnegotiation.parameter-name=format

spring.mvc.contentnegotiation.media-types.json=application/json

spring.mvc.contentnegotiation.media-types.xml=application/xml

3. Como cargar configuraciones desde archivos externos

Usando archivos .properties ya vienen soportados por springboot y el te genera un application.properties ´por defecto.

4. Cuál es el propósito de los repositorios y como los usaría en un proyecto.

Enviar y recibir información al mecanismo de persistencia,ya sea base de datos relacional,no relacional,archivos etc.

5. Que son las interfaces y cuál es su importancia.

Permiten definir contratos entre lo que debería hacer una clase, una clase puede implementar varias interfaces.

6. Ejemplo Practico: Desarrolla una API REST utilizando Spring Boot que permita la gestión de productos. Cada producto debe tener un ID único, un nombre, una descripción y un precio. La API debe permitir realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los productos, utilizando tanto JSON como XML para el intercambio de datos. a. Requerimientos técnicos. i. Usar Spring boot para la configuración del proyecto.

* + 1. ii. Debe contener una capa de persistencia usando Spring Data JPA
    2. iii. La Configuración de la base de datos debe cargarse desde un archivo externo.
    3. iv. Se deben regresar datos en JSON o en XML según se requiera.
    4. v. Manejar configuración HTTP y HTTPS.
    5. vi. Debe contener una capa de servicios.
    6. vii. Se debe implementar el repositorio que use los métodos definidos por defecto, además de contar con dos métodos adicionales donde se implemente un SQL Nativo que regrese datos de una consulta SQL definida por el aspirante.
    7. viii. Implementar validaciones/manejo de excepciones personalizadas.
    8. ix. Entregable: código fuente documentado.

Aspectos para evaluar:

1. Cumplimiento de requerimientos funcionales.

2. Comentarios en el código.

3. Limpieza de código fuente: Nombre de variables, indentación, Funciones.

4. Principio SOLID y DRY.

5. Estructura de directorio.

Aspectos que no son obligatorios, pero agregan un plus:

1. Uso de ORM

2. Manejo de excepciones.

3. Manejo de validaciones.

Modo de entrega:

* Se debe entregar código fuente en formato zip sin las librerías de los framworks en la fecha límite establecida por correo.
* La sustentación se hará con base en el código fuente entregado.
* La sustentación se hará en vivo, utilizando la plataforma en un ambiente local.
* (Opcional genera puntos): tener la aplicación en un servidor web.