Prueba Técnica

Pokemon Battle:

Se requiere hacer un servicio web que tenga un endpoint para simular una batalla Pokémon. Para esto utilizaremos la API publica de Pokemon que nos permitirá obtener los datos. La idea es que el servicio retorne al ganador de la batalla, sin embargo. Para que en el caso de que alguien solicite simular de nuevo una batalla, no tengamos que simularla vamos a almacenar en una base de datos los resultados de quien gana entre un Pokémon y otro, para así retornar el ganador.

Para lograr esto debemos tener cuenta lo siguiente:

1. Crear un servicio WEB que tenga un endpoint POST /pokemon-battle que reciba lo siguiente:

   {
   “nombre-pokemon-1”:”xxxx”,
   “nombre-pokemon-2”:”xxxx”
   }

2. Controlar los errores que puedan surgir durante el proceso y traducirlos a errores HTTP.
3. Crear una tabla para almacenar los resultados de las batallas.
4. Validar si existe en la BD una batalla entre esos dos Pokémones. Sino la debe simular.

5. Simular una batalla Pokemon en función del atributo stats que retorna el API de Pokemón. En este caso necesitamos los siguientes stats dentro de la respuesta:

   a. hp: Puntos de salud
   b. speed: Velocidad para atacar primero que otro
   c. attack: Puntos de ataque
   d. defense: Puntos de defensa

   La batalla se debe realizar por turnos y comienza primero el Pokémon que tenga más velocidad. En cada turno el Pokémon atacante resta del defensor los puntos de salud en una cantidad de: puntos de ataque del atacante menos los puntos de defensa del defensor. Gana el Pokémon que primero lleve a cero el otro.

6. Almacenar en una base de datos el resultado para que en un próximo request busque si ya tengo un resultado de dicha batalla y así poder retornar.

API Pokemon:
API Doc: https://pokeapi.co/docs/v2

Duración: 60

Preguntas conceptuales:

Preguntas sobre GIT

¿Qué es un Commit?
¿Qué es y para qué sirve un Git ignore?
¿Para qué sirve un git stash?
¿Para qué sirven los tags en git?
¿Para qué sirve un Cherry pick?

Queries y ORM

¿Cuál es la diferencia entre un Inner un Outter join?
¿Qué tipos de relaciones hay? /// Como implementa la relación de muchos a muchos?
¿Que es mejor para administrar las entidades y queries a una BD usar un ORM o escribir directamente los Queries?
¿Qué son las migraciones y para que usan?

Diseño de sistemas

Teniendo el siguiente escenario:

En dos computadores diferentes que se pueden comunicar entre sí, se ejecutan dos programas respectivamente que realizan lo siguiente:

Programa 1: Envía correos electrónicos a un destinatario con un contenido en particular.

Programa 2: Carga una lista de mensajes por enviar y le solicita al programa 1 que envíe cada uno de estos.

Condiciones:

- El programa 1 es capaz de solo procesar un mensaje a la vez.
- El programa 1 puede presentar fallas temporales al tratar de enviar un mensaje. Sin embargo, en una segunda solicitud podría procesarlo correctamente.

Problema:

De qué manera podemos adaptar el programa 1 y el programa 2 para garantizar el envío de todos los mensajes una sola vez.

Duración: 30