

Challenge MercadoLibre: Clima para Pokémons



Los Pokémon son una clase de criaturas inspiradas en animales reales, insectos, objetos, plantas o criaturas mitológicas. Cada uno de ellos tiene uno o varios tipos (fuego, agua, hielo, entre otros) y según su tipo son capaces de realizar diferentes habilidades. En esta ocasión se requiere lo siguiente:

Desarrollar una API en [Python](#) (utilizando el framework [Flask](#) idealmente, pero queda a tu elección) que tenga los siguientes endpoints:

- Obtener el tipo de un Pokémon (fuego, agua, tierra, aire, etc...) según su nombre.
- Obtener un Pokémon al azar de un tipo en específico.
- Obtener el Pokémon con nombre más largo de cierto tipo.
- Obtener un Pokémon al azar que contenga alguna de las letras 'l','A','M' en su nombre y que sea del tipo específico más fuerte en base al clima actual de tu ciudad (ver *referencia 1 y 2*)

Temperatura	Tipo más fuerte
>= 30°C	Fuego
>= 20°C y < 30°C	Tierra
>= 10°C y < 20°C	Normal
>= 0°C y < 10°C	Agua
< 0°C	Hielo

Referencia 1. Umbrales de temperatura a utilizar como referencia para decidir tipo de pokemon

Referencia 2: Un ejemplo puede resumirse en que si vivo en Buenos Aires y mi temperatura actual es de 29°, un pokémon que puede responder la API es *onix* (por ser de tierra y tener la letra "l"). En el siguiente request, de forma aleatoria me podría tocar *golem*

Además de estas funcionalidades, la API deberá ser segura. Para lograr esto tendrá que -como mínimo- contar con un esquema de autenticación. Solicitamos que pienses en este diseño de AuthN para tu API, el cual será abordado y conversado en la siguiente etapa del proceso de selección.

Recomendación: Puedes utilizar las siguientes APIs para lograr las funcionalidades de tu app:

- [Poké API](#)
- [Open-Meteo.com](#)

¿Qué vamos a evaluar?

- Apropiado manejo de llaves secretas, credenciales y cualquier información sensible
- Implementación de buenas prácticas de programación
 - Modularización
 - Nomenclatura de variables
 - Manejo de excepciones
 - Uso de comentarios
- [Trazabilidad](#)
- Documentación

Puntos bonus:

- [Dockerizar](#) la aplicación para facilitar la reproducción de la misma
- Utilizar una plataforma de control de versiones. Ej: [GitHub](#)
- Implementar el método de autenticación pensado para la API desarrollada.