

Requisitos

Para la prueba es necesario subir tu código a un repositorio en github, envía el enlace de tu repositorio

como respuesta al correo

Realizar un proyecto con NodeJS y Framework de preferencia (Express, Nestjs, etc) si es posible manejarlo

con TS se tomará en cuenta, pero con JS puro está correcto, para el ejercicio debes desarrollar dos

servicios utilizando la información de la pokeapi <https://pokeapi.co>

Primer endpoint:

- Tipo GET y recibe parámetros en el query
- El método deberá consultar la Pokeapi para obtener todos los pokemons, al hacer uso de este endpoint la respuesta debe ser un array ordenado de manera alfabética los pokemons de la lista. Si no se agrega ningún parámetro mas al query de la petición deberá responder con todos los pokemons, los parámetros que se pueden enviar en el query son limit y page, si se envían ambos hará el filtrado solo de la página solicitada, siendo page el número de la página y limit el tamaño de las páginas, al hacerlo por página se sigue teniendo en cuenta el orden alfabético.
- En caso de mandar datos que no sean paginables debe responder con un error que se manejara en el servidor.
- Otro parámetro opcional será search que hará la búsqueda parcial por nombre del pokemon y deberá responder considerando lo anterior (ordenados alfabéticamente y se podrán paginar de igual manera)

Segundo endpoint

- Tipo POST con parámetros en el body
- En el body recibirá el nombre de un pokemon que deberá ser validado si existe o no, en caso de no serlo responderá con un error el servidor, en caso de ser un nombre válido dentro de la pokeapi se procederá a generar un archivo pdf con la información básica del pokémon y se deberá poder descargar mas no guardar en el servidor

Los datos necesarios para poder ejecutar el proyecto deben estar mencionados en el read.me del repositorio.

Se tomará en cuenta la organización y lectura del código, así como buenas prácticas dentro del mismo

Resultado

- Documentación del **readme** es correcta.
- Manejo de TypeScript en el backend, es un buen agregado.
- Al inicio se tenía el error *has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.*
- El diseño del frontend cumple con los requerimientos pero hubiera sido un buen agregado mostrar imágenes.
- Si se envía una página fuera del rango, por ejemplo page: 200, no devuelve un error.
- Si se busca un nombre que no existe, no se devuelve un mensaje adecuado.
- en el backend no se valida el caso de que el pokémon no existe cuando se intenta generar el pdf
- En el front se utilizó una forma deprecada de subscribe
- En general el código es legible y funcional.

Proyecto funcional muy apegado a los requisitos