	Prueba Desarrollador	Fecha creación: 20-09-2022
		Versión: 2.0
		Pág: 2

Prueba Desarrollador Back end

Contexto

En la presente prueba se busca medir las habilidades técnicas, experiencia y conocimientos adquiridos por el próximo desarrollador back end.

Persona clave que hará parte de un equipo de trabajo y del cual se espera un gran desempeño.

Problema

Se posee la información de un api externa en la cual se encuentran los datos de usuarios, álbumes, publicaciones, comentarios y fotos. <https://jsonplaceholder.typicode.com/>

(NOTA: En caso de que el api no se encuentre disponible por ser algo público se debe poder descargar el proyecto y subirlo en un ambiente local <https://github.com/typicode/jsonplaceholder>)


El ejercicio debe tener en cuenta que el aplicativo debe estar desarrollado en NodeJS y lo ideal es que cuente con el esquema de Typescript, se deberá consultar de las apis externas y se deje crear un registro de cada petición realizada (Fecha, método consultado, datos retornados), este proyecto backend se encargará de exponer la información a un frontend, para lo cual se debe poder contar con las siguientes apis:

- Listar a los usuarios.
- Listar publicaciones.
- Consultar fotos de un usuario.
- Editar registro de una petición.
- Eliminar registro de una petición.
- Listar Registros de peticiones realizadas.
- Exportar registros en base64
- Cada endpoint debe estar protegido con autenticación por JWT o API key

Para lograr el último servicio se debe poder tener una funcionalidad que exponga un servicio el cual se usará para exportar los datos de registros, la idea es que se pueda construir un documento (excel) con los datos y enviarlos para que el frontend los pueda interpretar.

<https://base64.guru/converter/encode>

Elaborado por	Alexander Henao Moná
---------------	----------------------

	Prueba Desarrollador	Fecha creación: 20-09-2022
		Versión: 2.0
		Pág: 2

Objetivos

- Consumir API externo.
- Almacenar registros de consultas.
- Publicar servicios útiles para el frontend.
- Endpoints protegidos con autenticación.
- Pruebas unitarias.
- Documentación del api creado. (swagger, openapi)

Bases para el ejercicio

- Los servicios que se consumen y que se exponen deben de ser tipo REST.
- Los entregables se deben almacenar en un repositorio público sea Github o Gitlab.
- El almacenamiento de los logs se recomienda que sea en una base de datos (memoria, relacional, no relacional).
- Segregación de capas por responsabilidades.
- Typescript es el lenguaje de programación que se espera ser utilizado.
- README.md con la guía para ejecutar e instalar el aplicativo.

Entregables

- Rutas públicas del repositorio.
- URL del proyecto desplegado
- Scripts de base de datos en caso de ser necesario.

Elaborado por	Alexander Henao Moná
---------------	----------------------