

TRABAJO FINAL

Desarrollar un API REST para administrar usuarios, crear los siguientes endpoints:

- Para crear un usuario POST `‘/usuarios’`
- Para listar a todos lo usuarios: GET `‘/usuarios’`
- Para listar un usuario en especifico GET `‘/usuarios/:id_usuario’`
- Para actualizar los datos de un usuario: PUT `‘/usuarios/:id_usuario’`
- Para eliminar a un usuario: DELETE `‘usuarios/:id_usuario’`
- Para mostrar el promedio de edades de los usuarios: GET `‘/usuarios/promedio-edad’`
- Para mostrar la version del api rest: GET `‘/estado’`

Siguiendo las siguientes instrucciones:

- El proyecto puede ser desarrollado en Python(usar Flask) o Javascript(usar NodeJS)
- Se debe usar una base de datos para guardar los datos, se debe Postgres para esto.
- El código del proyecto debe estar dentro de un repositorio público, pueden usar GITHUB o GITLAB o BITBUCKET
- Los datos que se deben almacenar son: cedula_identidad, nombre, primer_apellido, segundo_apellido, fecha_nacimiento, deben adjuntar el script de creación de su tabla y los script para inicializar datos si se da el caso.
- Dentro del repositorio debe haber un archivo README.md que tenga las instrucciones para levantar su API REST
- Los datos que se deben mostrar para el endpoint de estado será lo siguiente:(obviamente que muestre su información)
 - `{“nameSystem”: “api-users”, “version”: “0.0.1”, “developer”:”Juan Carlos Condori Machicado”, “email”: “juanc.com@gmail.com”}`
- La respuesta del promedio debe devolver un objeto como el siguiente ejemplo:
 - `{ promedioEdad: 34.5}`
- Pueden usar como ejemplo el proyecto desarrollado en clase <https://github.com/juanca84/dip-fullstack-dev-mod-3/tree/main/api-rest-js>, solo como ejemplo.
- El trabajo final tiene un puntaje de 80 puntos, deben

hacerme llegar la dirección de su repositorio público, adjuntado también su nombre y apellidos, para que pueda revisarlo, la fecha y hora límite de entrega será el 03/09/2023 23:59:59.

- Pueden ganar puntos extras, si cumplen algunos o todas las directrices que se dan a continuación:

- Si el proyecto es desarrollado en python usando Flask.

- Si logran hacer un deploy de su proyecto en alguna plataforma que puede ser: <https://render.com/>, <https://fly.io/> (tanto la aplicación web y la base de datos), adicionar la url de su deploy dentro del README y adjuntar la documentación de sus endpoints.

Si surge alguna duda pueden escribirme a juanc.com@gmail.com procurare responderles en un tiempo prudente.

PD: Como ayuda para obtener el promedio de edades basado en fechas de nacimiento podrían usar el siguiente query:

```
SELECT AVG(EXTRACT(YEAR FROM AGE(NOW(),  
fecha_nacimiento))) AS promedio_edades FROM personas;
```

Explicación de la consulta:

- NOW() devuelve la fecha y hora actuales.
- AGE(NOW(), fecha_nacimiento) calcula la diferencia entre la fecha actual y la fecha de nacimiento de cada persona, y devuelve un intervalo.
- EXTRACT(YEAR FROM intervalo) extrae el componente de año del intervalo para obtener la edad de cada persona.
- AVG(...) calcula el promedio de las edades obtenidas.