¡¡Hola!!

Sabemos que ha sido un camino lleno de desafíos el proceso que has llevado de adquisición de conocimientos, pero confiamos en que el proceso te ha aportado los conocimientos necesarios para abordar los diferentes retos de un ingeniero de desarrollo de software.

Ha llegado la hora de poner en práctica todo aquello que has aprendido y por ello vamos a plantearte el siguiente reto técnico que tiene la intención de enfrentarte a una situación real de proyecto.

# Estrategia de trabajo

El proyecto se realizará de manera individual por cada integrante del Bootcamp.

# Reto Backend

Para el reto backend, vamos a afrontar la solución de una necesidad que requiere exponer mediante servicios API REST capacidades para realizar el control de una agenda de citas de laboratorio.

## Data

Para la generación de información, vas a partir de la siguiente estructura de base de datos. Para fines del reto puedes decidir la utilización de una base de datos SQL.

Definición estructural

Tests: {“id”, “name”, “description”}

affiliates: {“id”, “name”, “age”, “mail”}

appoinments: {“id”, “date”, “hour”, “id\_test”, “id\_affiliate”}

Contrato de datos

id: number

name: string

description: string

mail: string

age: number

date: Date (‘dd/MM/yyyy’)

hour: Date (‘hh:mm’)

## Services

El sistema backend va a permitir exponer al front los servicios para diversas operaciones que debe de realizar que conoceremos como CRUD (Create, Remove, Update and Delete)

Para este fin creara 4 controladores, cada uno con sus respectivos métodos, los cuales vamos a describir a continuación:

### Controladores

1. tests.controller: Controlador para la tabla Tests, tiene el siguiente grupo de metodos
   1. getlist
   2. getbyid {id}
   3. post
   4. put
   5. delete {id}
2. affiliates.controller: Controlador para la tabla Affiliates, tiene el siguiente grupo de metodos
   1. getlist
   2. getbyid {id}
   3. post
   4. put
   5. delete {id}
3. appoinments.controller: Controlador para la tabla Appoinments, tiene el siguiente grupo de metodos
   1. getlist
   2. getbyid {id}
   3. post
   4. put
   5. delete {id}
   6. getbydate {date}
   7. getbyaffiliates {id\_affiliate}

### Descripción de los metodos

A continuación, vamos a proceder a describir la funcionalidad de los métodos, ten en cuenta que, según la definición de la data, se debe de conservar un contrato de datos para que los desarrollos sean uniformes y exista una comunicación entre el backend y frontend

1. getlist
   1. Entregara todo el listado de registros de la tabla, con todos sus campos
   2. Si la consulta no encuentra resultados debe entregar un status code 204
   3. Si la consulta es exitosa retorna un status code 200

1. getbyid {id}:
   1. entregará un solo registro que coincida con el id que llega por parámetro, con todos sus campos
   2. Si la consulta no encuentra resultados debe entregar un status code 404
   3. Si la consulta es exitosa retorna un status code 200
2. post:
   1. Permitirá almacenar un nuevo registro en la tabla, recibe en el body la estructura de la tabla, sin el campo id
   2. Ten en cuenta que el id para la tabla debe de ser generado o autoasignado.
   3. Si la operación de inserción es exitosa, debe retornar un status code 201
   4. Si no se pudo realizar la inserción debe retornar 404
3. put:
   1. Permite actualizar un campo de la tabla, teniendo presente que el campo id no es un campo modificable, recibe un json con todos los campos de la tabla
   2. Si la operación de actualización es exitosa, debe retornar un status code 201
   3. En caso de no poder realizar la actualización debe retornar 404
4. delete {id}
   1. Permite eliminar un registro de la tabla, el cual debe coincidir con el id recibido por parámetro
   2. Si la operación es realizada debe de retornar un status code 200
   3. Si la operación no es realizada debe de retornar un status code 204
5. getbydate {date} y getbyaffiliates {id\_affiliate}
   1. getbydate {date}: Trae todas las appoinment de una fecha, agrupadas por affiliate
   2. getbyaffiliates {id\_affiliate}: Trae todos los appoinment de un affiliate
   3. Estos métodos van a permitir retornar una consulta basada en el parámetro que reciben, la cual generara el siguiente json

# Criterio de aceptación

1. Debes de generar los endpoint con el siguiente patrón

/api/controller/ y los respectivos métodos Get, Post, Put, Delete.

1. Todos los controladores deben contar con pruebas unitarias.
2. Todos los controladores deben de ser probados en postman en la sustentación del proyecto.
3. Los fuentes terminados, deben de quedar a alojados en git en la rama master o main.

Éxitos!!