KRUGER CHALLENGE

Jorge Luis Ariosa

PROYECTO INVENTARIO DE VACUNACIÓN

Perfil Back-End

ghp\_QIztAqFMVIYQWAX7BGs26xNABR8gFz12fMjW

**Introducción**

Se requiere una aplicación para llevar el registro de vacunación de los empleados de una “x” empresa

El sistema estará segmentado en 2 tipos de usuarios, el USER y el ADMIN, este ultimo con mas privilegios

La aplicación para su funcionamiento iniciara con un usuario ADMIN ficticio cullas credenciales serán user admin y password root

la URL base de la API sera api / v1 / , luego seguiría el tipo de usuario y luego la accion

# End Point

Los end point están segmentados por el tipo de usuario y solo un end point sera accesible de manera publica

## End Point públicos

* / login tipo POST: mediante esta URL se realizara la autenticación de los usuarios , esta devolverá un código 201 mas una cabecera acces\_token con el token de la sesión que es obligatorio enviar a todas las URL de la API , en este se encuentra el rol del usuarios

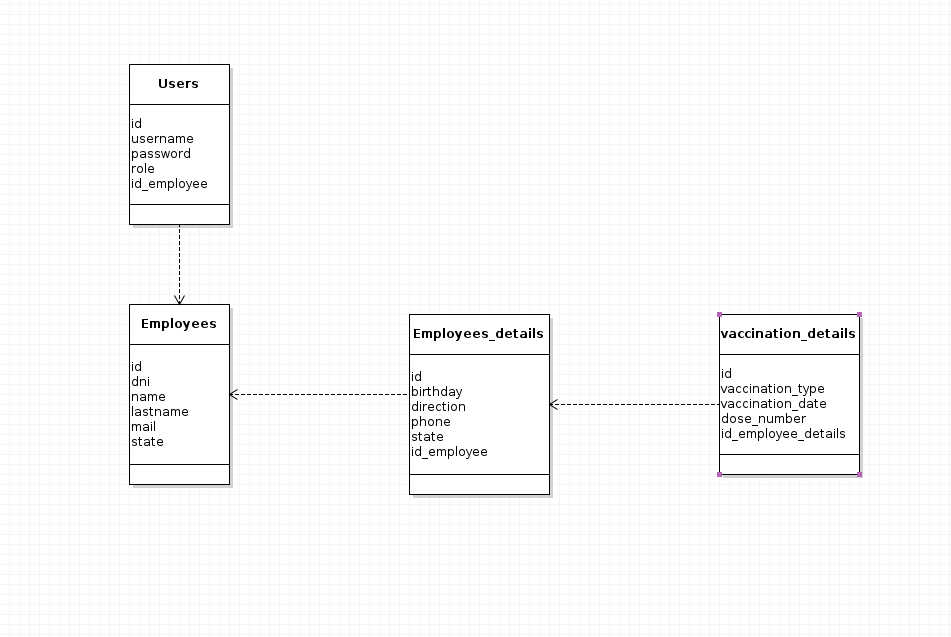
## **End Point privados del Admin**

* / api / v1 / admin / employees tipo GET: devuelve una lista de los datos básicos de todos los empleados, esta URL solo requiere el header authorization con el access\_token
* / api / v1 / admin / add-employee tipo POST: esta url agrega un empleado, requiere un json con los siguientes datos: dni, name , lastname y mail. Caso de ser exitosa devolverá un 201 y el empleado creado, caso fallido un error 400 y la respuesta nula
* / api / v1 / admin / update-employee tipo PUT: esta url actualiza un empleado, requiere un json con los siguientes datos: id, dni, name , lastname y mail. Caso de ser exitosa devolverá un 201 y el empleado, caso fallido un error 400 y la respuesta nula
* / api / v1 / admin / change-employee-status tipo PUT: esta url da de alta a un empleado y crear su cuenta con username igual a su apellido y password igual a su dni, requiere un json con los siguientes datos: id. Caso de ser exitosa devolverá un 201 y el empleado, caso fallido un error 400 y la respuesta nula
* / api / v1 / admin / delete-employee tipo DELETE: esta url elimina un empleado, requiere un json con los siguientes datos: id. Caso de ser exitosa devolverá un 201, caso fallido un error 400 y la respuesta nula
* / api / v1 / admin / filter-vaccination-state tipo POST: esta url filtra los empleado según el estado de su vacuna, requiere un json con los siguientes datos: state. Caso de ser exitosa devolverá un 201 y lista de empleados, caso fallido un error 400 y la respuesta nula
* / api / v1 / admin / filter-vaccination-type tipo POST: esta url filtra los empleado según su vacuna, requiere un json con los siguientes datos: type. Caso de ser exitosa devolverá un 201 y la lista de empleados, caso fallido un error 400 y la respuesta nula
* / api / v1 / admin / filter-vaccination-date-range tipo POST: esta url filtra los empleado según las fechas de su vacuna y el rango enviado, requiere un json con los siguientes datos: first , end. Caso de ser exitosa devolverá un 201 y la lista de empleados, caso fallido un error 400 y la respuesta nula

## End **P**oint privados de todos los usuarios

* + - * / api / v1 / user / update-account tipo PUT : esta url permite al usuario actualizar sus datos de su cuenta de usuario, requiere un json con los siguientes datos: id. Caso de ser exitosa devolverá un 201 y los datos del usuario, caso fallido un error 404 y la respuesta nula
      * / api / v1 / user / my-account tipo POST : esta url permite al usuario visualizar los datos de su cuenta de usuario, requiere un json con los siguientes datos: id. Caso de ser exitosa devolverá un 201 y los datos del usuario, caso fallido un error 404 y la respuesta nula
      * / api / v1 / user / add-detail tipo POST : esta url permite al usuario agregar los datos extra de su cuenta de usuario, requiere un json con los siguientes datos: id , state(vacunado o no),bd(fecha de nacimiento), direction, phone. Caso de ser exitosa devolverá un 201 y los datos extra agregados, caso fallido un error 400 y la respuesta nula
      * / api / v1 / user / update-detail tipo POST : esta url permite al usuario actualizar los datos extra de su cuenta de usuario, requiere un json con los siguientes datos: id(usuario) , state(vacunado o no),bd(fecha de nacimiento), direction, phone. Caso de ser exitosa devolverá un 201 y los datos extra actualizados, caso fallido un error 400 y la respuesta nula
      * / api / v1 / user / details tipo POST : esta url permite al usuario visualizar los datos extra de su cuenta de usuario, requiere un json con los siguientes datos: id(usuario). Caso de ser exitosa devolverá un 201 y los datos extra, caso fallido un error 400 y la respuesta nula
      * / api / v1 / user / add-vaccination-detail tipo POST : esta url permite al usuario agregar los datos de su vacuna, requiere un json con los siguientes datos: id(usuario) , date, type(tipo de vacuna), number(numero de dosis). Caso de ser exitosa devolverá un 201 y los datos de su vacuna, caso fallido un error 400 y la respuesta nula
      * / api / v1 / user / update-vaccination-detail tipo PUT : esta url permite al usuario actualizar los datos de su vacuna, requiere un json con los siguientes datos: id(usuario) , date, type(tipo de vacuna), number(numero de dosis). Caso de ser exitosa devolverá un 201 y los datos de su vacuna, caso fallido un error 400 y la respuesta nula

**Modelo Entidad relación**

****