# Parcial #3

# Punto 2

# Jhojan Espinosa y Brayan Maca

#clonar el repositorio APIrestflaskmysqlubunto, propiedad de Oscar Mondragon.

https://github.com/omondragon/APIRestFlaskMySQLUbuntu.git

#Crear una nueva box con el vagrantfile que contiene el repo

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

if Vagrant.has\_plugin? "vagrant-vbguest"

config.vbguest.no\_install = true

config.vbguest.auto\_update = false

config.vbguest.no\_remote = true

end

config.vm.define :servidorRest do |servidorRest|

servidorRest.vm.box = "bento/ubuntu-22.04"

servidorRest.vm.network :private\_network, ip: "192.168.60.3"

servidorRest.vm.provision "file", source: "apirest\_mysql.py", destination: "/home/vagrant/apirest\_mysql.py"

servidorRest.vm.provision "file", source: "init.sql", destination: "/home/vagrant/init.sql"

servidorRest.vm.provision "shell", path: "script.sh"

servidorRest.vm.hostname = "servidorRest"

end

end

#El vagrantfile viene aprovicionado con todos los recursos necesarios

#para su correcto funcionamiento.

#Realizar la instalación de node, como es ubuntu usar el siguiente comando

sudo apt-get install npm

#Crear un archivo con la extencion js, en este caso:

sudo vim nodeclient.js

#El archivo deberá contener el siguiente codigo:

const axios = require('axios');

const baseUrl = 'http://localhost:5000';

// GET, obtener todos los libros

async function getBooks() {

try {

const respuesta = await axios.get(`${baseUrl}/books`);

const books = response.data.books;

console.log("Api client node realizó GET");

console.log(books);

} catch (error) {

console.error(error);

}

}

// GET Buscar por ID

async function getBook(bookId) {

try {

const respuesta = await axios.get(`${baseUrl}/books/${bookId}`);

const book = response.data.book;

console.log("Api client node realizó GET by ID");

console.log(book);

} catch (error) {

console.error(error);

}

}

// Añadir nuevo libro

async function createBook(title, description, author) {

try {

const data = {

title,

description,

author,

};

const response = await axios.post(`${baseUrl}/books`, data);

const book = response.data.book;

console.log("Api client node añadió un libro");

console.log(book);

} catch (error) {

console.error(error);

}

}

// Editar información del libro

async function updateBook(bookId, title, description, author) {

try {

const data = {

title,

description,

author,

};

const response = await axios.put(`${baseUrl}/books/${bookId}`, data);

const book = response.data.book;

console.log("Api client node Editó un libro");

console.log(book);

} catch (error) {

console.error(error);

}

}

// Eliminar libro

async function deleteBook(bookId) {

try {

await axios.delete(`${baseUrl}/books/${bookId}`);

console.log('API client node Eliminó un libro');

} catch (error) {

console.error(error);

}

}

(async () => {

await getBooks();

await updateBook(7, 'El libro actualizado', 'Description actualizada', 'Jorgito');

await getBooks();

})();

#Como el codigo trabaja con una Axios, una libreria de peticiones http,

#debemos realizar su instalación:

sudo npm install axios

#Para ejercutar el codigo, usar el comando:

node nodeclient.js