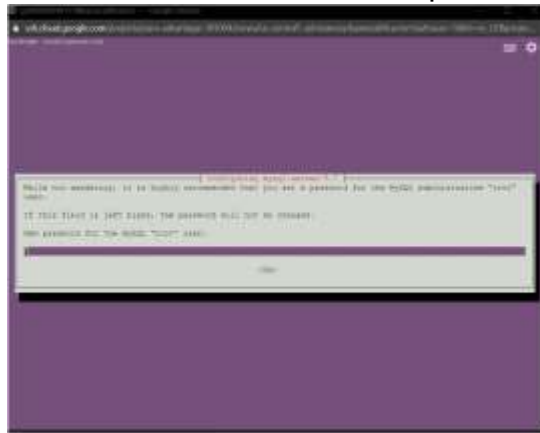


Manual de Usuario

Instalación de Mysql

- Accedemos por medio del SSH a la maquina virtual
Ingresamos los siguientes comandos para instalar Mysql
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
 - `sudo wget https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb`
 -
 - `dpkg -i mysql-apt-config_0.8.10-1_all.deb`
 -
 - `apt-get update`
 -
 - `apt-get install mysql-server`
 -
 - Introducimos una contraseña para el usuario root



- Ingresamos a la base de datos de la siguiente manera `"mysql -uroot -p"`
 - Creamos la base de datos con el archivo DDL que se encuentra en el repositorio
-
- Creamos un usuario `"CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';"`
 - Le damos permisos `"GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';"`
 - Recargamos los privilegios `"FLUSH PRIVILEGES;"`

Para realizar la instalación de la aplicación:

- Como primer paso clonamos el repositorio https://github.com/Stevensishernandez/API_RES_BASES.git

```

j3050560340117@pruebasbasura:~$ git clone https://github.com/Stevensishernandez/API_RES_BASES.git
Cloning into 'API_RES_BASES'...
remote: Enumerating objects: 3375, done.
remote: Counting objects: 100% (3375/3375), done.
remote: Compressing objects: 100% (2133/2133), done.
remote: Total 3375 (delta 1215), reused 3315 (delta 1155), pack-reused 0
Receiving objects: 100% (3375/3375), 9.83 MiB | 16.94 MiB/s, done.
Resolving deltas: 100% (1215/1215), done.
Checking connectivity... done.
j3050560340117@pruebasbasura:~$

```

- Instalamos mp2 para poder ejecutar el backend y el frontend al mismo tiempo.

```
npm --log-level verbose --log-file /home/g3050560340117/.npm/_logs/2021-05-03T10_18_51_3582-debug.log
g3050560340117@pruebasbasea:~$ sudo npm install -g pm2
[...]
```

- Nos dirigimos a la carpeta de backend para ejecutarlo

[illegible]

- Con esto ya tenemos levantado el backend y ahora levantaremos el frontend, nos dirigimos a su carpeta y lo levantamos con pm2

Antes de esto asignaremos la ip de nuestra api

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Injectable({
  providedIn: 'root'
})
export class GamesService {

  API_URL = 'http://10.0.0.1:3000/';
  constructor(private http: HttpClient) {
  }

  getTempi() {
    return this.http.get(`${this.API_URL}buscarAlbum`);
  }

  getPokemon() {
    return this.http.get('https://pokeapi.co/api/v2/pokemon?offset=0');
  }

  getC1() {
    return this.http.get(`${this.API_URL}c1`);
  }

  getC2() {
    return this.http.get(`${this.API_URL}c2`);
  }

  getC3() {
    return this.http.get(`${this.API_URL}c3`);
  }

  getC4() {
    return this.http.get(`${this.API_URL}c4`);
  }

  getC5() {
    return this.http.get(`${this.API_URL}c5`);
  }
}

```

- Con el siguiente comando podemos ejecutar nuestro angular pm2 start /usr/bin/ng -- serve --port 4200 --host 0.0.0.0 --disableHostCheck true
- Ahora lo tenemos listo

```

root@pruebasbasea:/home/g3050560340117/API_RES_BASES# pm2 ls

```

id	name	mode	U	status	cpu	memory
0	index	fork	3	online	0%	51.8mb
1	ng	fork	1	online	0%	376.2mb

```

root@pruebasbasea:/home/g3050560340117/API_RES_BASES# git reset --hard

```

- Ahora estamos listos para usar la aplicación

