



**DIVISION DE INGENIERIA EN
SISTEMAS
COMPUTACIONALES**

TECNOLOGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

MANUAL DE USUARIO "LOGIN"

Base de datos para dispositivos
móviles

Grupo: 5851

Profesora: Griselda Cortes Barrera

INTEGRANTES:
AGUILAR PÉREZ AQUETZALIN
PÉREZ ROMERO ALAN ANTONY
PINEDA GARCÍA JESÚS DANIEL

Índice

<i>Análisis del proyecto</i>	3
<i>Modulo Login/Recuperar/Registro</i>	3
<i>Requerimientos y necesidades</i>	3
<i>Base de Datos</i>	5
<i>Especificaciones</i>	¡Error! Marcador no definido.
<i>Enlace GitHub</i>	10

Análisis

Modulo Login/Recuperar/Registro

Requerimientos y necesidades

- La página permitirá la visualización del módulo Login mediante un botón o sección el menú.
- La página permitirá el ingreso a los usuarios registrados en el sistema, de acuerdo con su perfil.
- La página permitirá el cambio de contraseña por medio un correo registrado.
- La página permitirá el registro de una cuenta con datos obligatorios completos.

Requerimientos Login

- Se requiere llenar el formulario para el inicio de sesión.
- El usuario ingresara al sistema por medio de un correo y contraseña registrados en la base de datos.

Requerimientos recuperar contraseña

- Se requiere de un correo registrado en la base de datos para poder recuperar la contraseña.

Requerimientos y necesidades registró

- Se requiere llenar el formulario completo para registrar al empleado.
- Se requiere de un correo electrónico.
- Se requiere de una contraseña de al menos 8 caracteres.

Problemática

El programa login se encarga de la autenticación del usuario (comprobando que el nombre de usuario y contraseña sean correctos) dado que la información es sensible y confidencial ya que se cuenta con datos personales de cada usuario e información relacionada con la empresa.

Objetivos generales

1. Desarrollar un componente que autorice el acceso a los usuarios registrados.
2. Elaborar un componente que apruebe la creación de una cuenta.
3. Llevar a cabo un componente que permita recuperar la cuenta.

Objetivos específicos

- Analizar el proceso de inicio de sesión y registro de usuarios.
- Determinar los requerimientos y necesidades.
- Diseñar la interfaz gráfica.
- Implementar la interfaz gráfica.

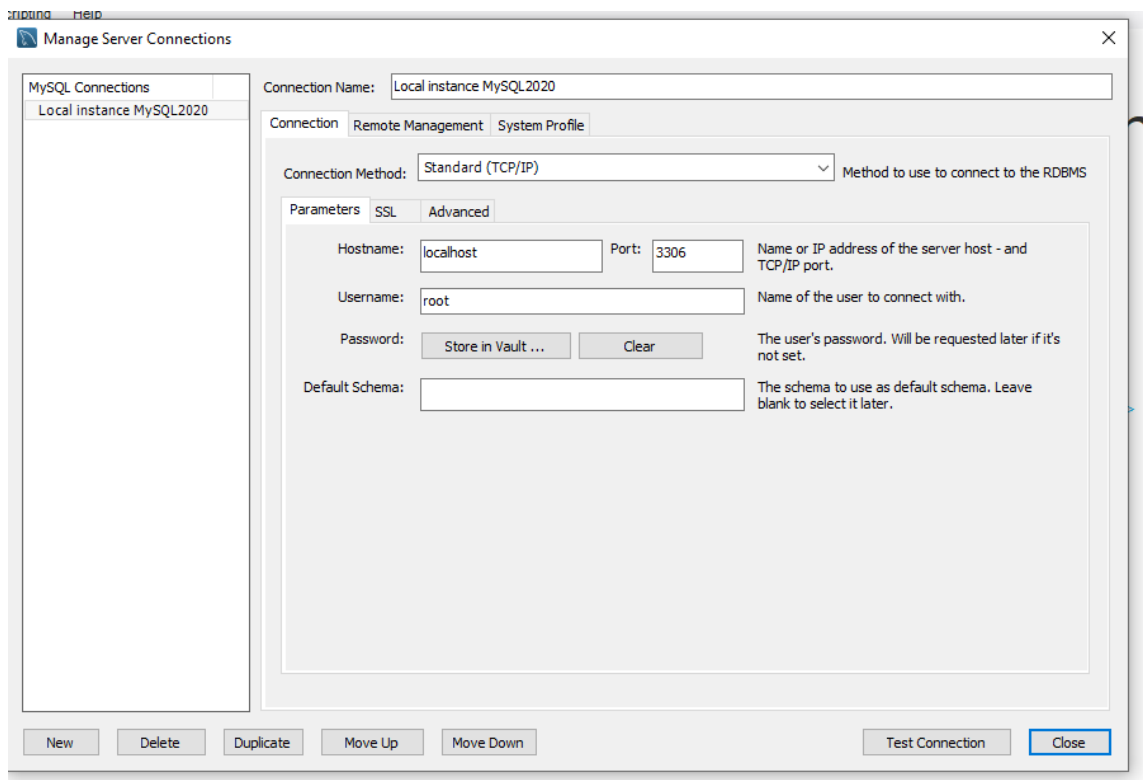
Base de Datos

Para el desarrollo e implementación de la base de datos para el funcionamiento de nuestro sistema login, nos apoyaremos del gestor de base de datos MYSQL, con el frame de Workbench, la base de datos llevara por nombre **model** la cual contara con una tabla llamada **users** en la cual se alojaran los datos necesarios para realizar el login, estos son;

- Id
- Username
- Password
- Rol

Proceso de creación de la conexión.

Parámetros y configuración de la conexión que alojara la base de datos a utilizar en el sistema.



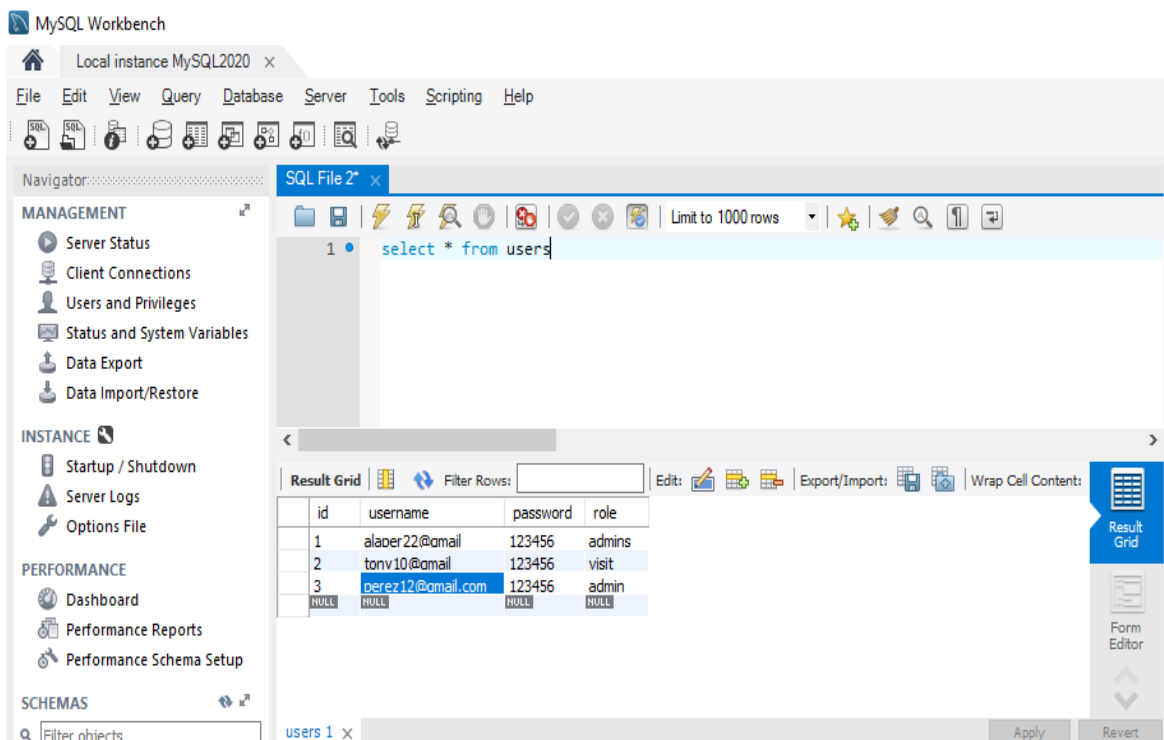
Comandos para el desarrollo de la base datos a implementar:

create database model

use model

create table users(id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY NOT NULL, username VARCHAR (50) NOT NULL, password VARCHAR (50) NOT NULL,, role VARCHAR (50) NOT NULL)

insert into users(id, username, password, role) values('3', 'perez12@gmail.com', '123456', 'admin')



Especificaciones

Instrucciones:

Para iniciar sesión se debe ingresar el correo electrónico y la contraseña, en caso de no tener estos datos, deberá hacer clic en Crear cuenta; en caso de no recordar la contraseña puede hacer clic en ¿Has olvidado tu contraseña?

En la sección de Crear cuenta, se debe de elegir si es para invitado o empleado, una vez que se elige una opción se muestra un formulario donde se tiene que llenar toda la información que se requiere, finalmente dar clic en el botón de registrarse.

En la sección de Recuperar cuenta, se debe ingresar el correo electrónico y dar clic al botón para recuperar la contraseña.

Una vez teniendo acceso al repositorio lo único que debemos hacer es descargar las carpetas, y abrirla desde visual code, un vez descargada y abierta desde angular, debemos proceder a instalar todas las dependencias e instrucciones necesarias para su funcionamiento, para lo cual vamos a proceder a clonar el repo.

Clonar Repo.

En la terminación de angular, en la ruta raíz (no en la ruta donde se encuentra ubicada la carpeta del proyecto) debemos iniciar la clonación del repo haciendo uso del comando:

git clone + “ruta o link del repo”

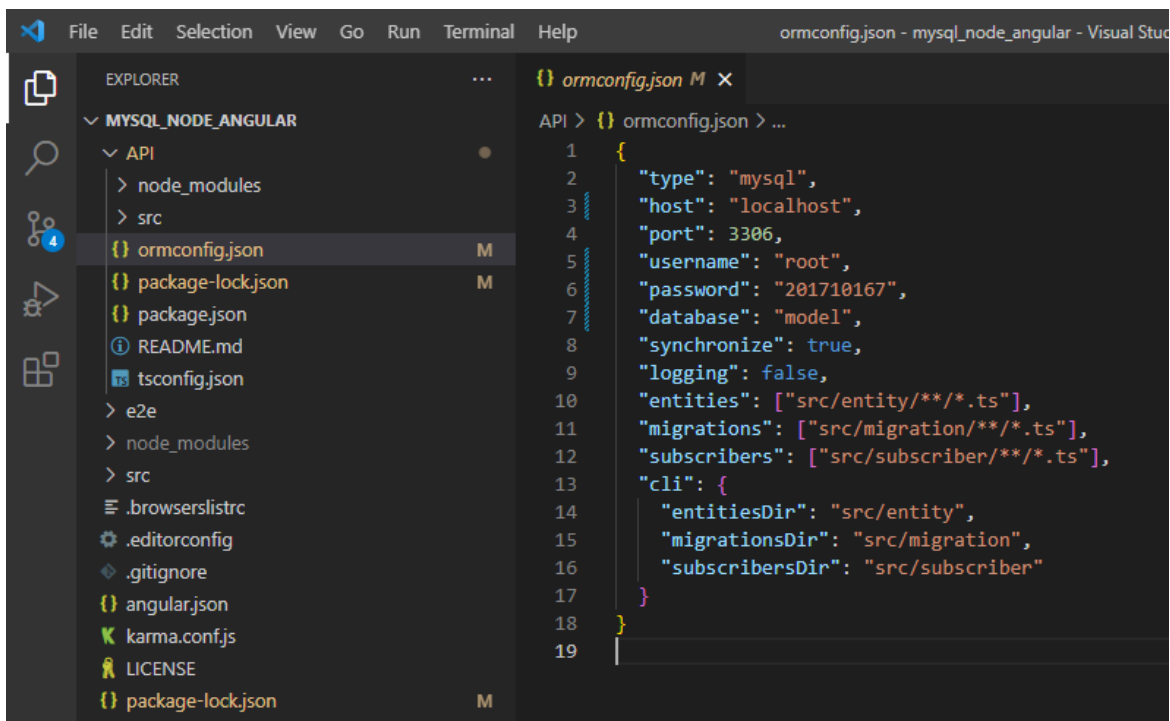
```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
```

Una vez echo esto, debemos posicionarnos desde la terminal en la ruta de nuestro proyecto, y ahí debemos hacer un npm install, para actualizar todos los paquetes y parámetros necesarios.

Npm install

Después debemos posicionarnos en la carpeta “API”, la cual es el servidor de nuestra base de datos, servicio que va consumir nuestra API, por lo cual igual debemos realizar un **npm i** dentro de la ruta de la carpeta API.

Finalmente no debemos olvidar que debemos crear la configuración para la conexión a la base de datos, con los siguientes parámetros, o bien, configurar en base a tu base de datos desde el archivo ormconfig.json.

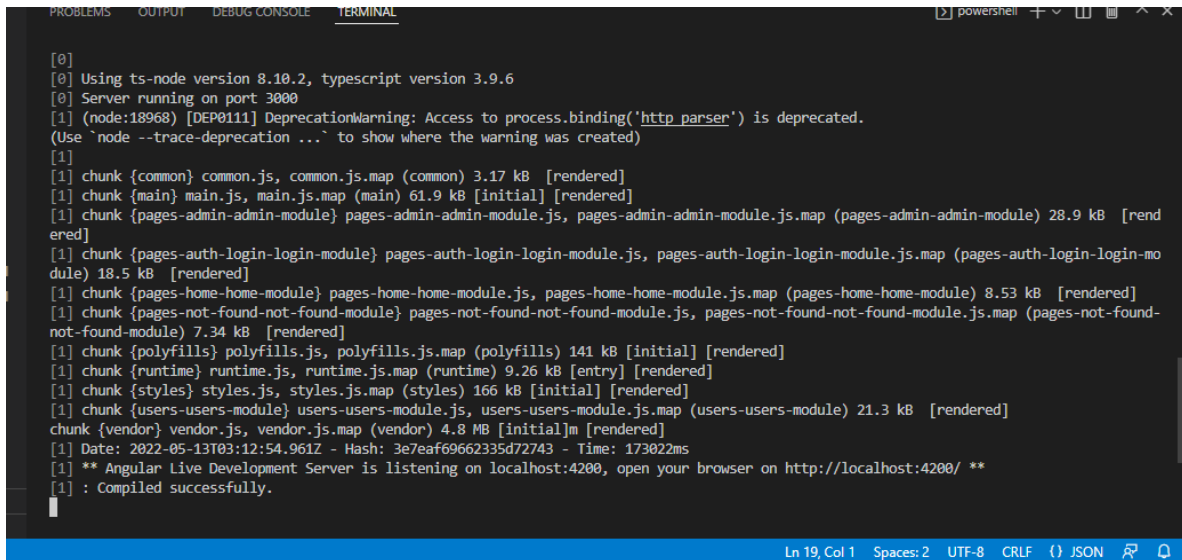


```
1 {
2   "type": "mysql",
3   "host": "localhost",
4   "port": 3306,
5   "username": "root",
6   "password": "201710167",
7   "database": "model",
8   "synchronize": true,
9   "logging": false,
10  "entities": ["src/entity/**/*.ts"],
11  "migrations": ["src/migration/**/*.ts"],
12  "subscribers": ["src/subscriber/**/*.ts"],
13  "cli": {
14    "entitiesDir": "src/entity",
15    "migrationsDir": "src/migration",
16    "subscribersDir": "src/subscriber"
17  }
18 }
19
```

Una vez finalizada esta configuración e instalado todos los parámetros, debemos correr el servidor y la api, con el comando:

Npm run dev

Y esperar a que realice la conexión a la base y compile de manera exitosa nuestro api y el servidor, así como la conexión a la base de datos.



```
[0]
[0] Using ts-node version 8.10.2, typescript version 3.9.6
[0] Server running on port 3000
[1] (node:18968) [DEP0111] DeprecationWarning: Access to process.binding('http_parser') is deprecated.
(Use `node --trace-deprecation ...` to show where the warning was created)
[1]
[1] chunk {common} common.js, common.js.map (common) 3.17 kB [rendered]
[1] chunk {main} main.js, main.js.map (main) 61.9 kB [initial] [rendered]
[1] chunk {pages-admin-admin-module} pages-admin-admin-module.js, pages-admin-admin-module.js.map (pages-admin-admin-module) 28.9 kB [rendered]
[1] chunk {pages-auth-login-login-module} pages-auth-login-login-module.js, pages-auth-login-login-module.js.map (pages-auth-login-login-module) 18.5 kB [rendered]
[1] chunk {pages-home-home-module} pages-home-home-module.js, pages-home-home-module.js.map (pages-home-home-module) 8.53 kB [rendered]
[1] chunk {pages-not-found-not-found-module} pages-not-found-not-found-module.js, pages-not-found-not-found-module.js.map (pages-not-found-not-found-module) 7.34 kB [rendered]
[1] chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
[1] chunk {runtime} runtime.js, runtime.js.map (runtime) 9.26 kB [entry] [rendered]
[1] chunk {styles} styles.js, styles.js.map (styles) 166 kB [initial] [rendered]
[1] chunk {users-users-module} users-users-module.js, users-users-module.js.map (users-users-module) 21.3 kB [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.8 MB [initial]m [rendered]
[1] Date: 2022-05-13T03:12:54.961Z - Hash: 3e7eaf69662335d72743 - Time: 173022ms
[1] ** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
[1] : Compiled successfully.
```

Enlace GitHub

<https://github.com/alananpr/modulo-acceso.git>