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

# vi: set ft=ruby :

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

config.vm.define :generic do |generic|

generic.vm.box = "generic/centos8"

generic.vm.network :private\_network, ip: "192.168.50.3"

generic.vm.hostname = "generic"

end

end

**DESACTIVAR SELINUX**

vim /etc/selinux/config

**¡¡¡PARAR EL SERVICIO FIREWALLD SIEMPRE QUE PRENDAMOS LAS MAQUINAS!!!**

service firewalld stop **To Install Flask using pip3:**

sudo yum install epel-release

sudo yum install python3-pip

pip3 –V

sudo pip3 install Flask

pip3 freeze | grep Flask

**To Install MySQL and libraries:**

sudo wget https://dev.mysql.com/get/mysql80-community-release-el7-7.noarch.rpm

sudo rpm -Uvh mysql80-community-release-el7-7.noarch.rpm

sudo yum update **(GPG FAILED)**

sudo yum install mysql-server

sudo systemctl start mysqld

sudo systemctl status mysqld

sudo yum install epel-release

sudo yum install python3-pip

pip3 -V

**MySQL + Flask Support**

sudo yum install mysql-devel **(GPG FAILED)**

sudo yum install python3-devel

sudo yum install gcc (en el video no lo hace) **(GPG FAILED)**

sudo yum groupinstall ‘Development Tools’

sudo pip3 install flask-mysqldb **(ERROR DE INSTALACIÓN)**

mysql -u root -p

Contraseña vacia

show databases;

CREATE DATABASE myflaskapp;

USE myflaskapp;

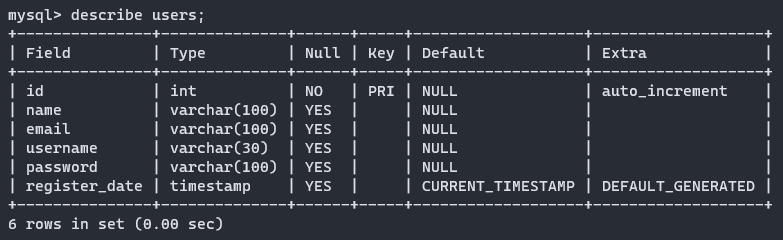
CREATE TABLE users(id INT(11) AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100), email VARCHAR(100), username VARCHAR(30), password VARCHAR(100), register\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP);

Show TABLES;

Texto

Descripción generada automáticamente

DESCRIBE users;



Nos salimos de mySQL (ctrl + D) y seguimos:

**Install Form Tools**

sudo pip3 install Flask-WTF

sudo pip3 install passlib

**FORMULARIO REGISTRO USUARIOS PARTE 1**

**VIEWS.PY**

from flask import render\_template, flash, redirect, url\_for, session, request, logging

from flask\_mysqldb import MySQL

from wtforms import Form, StringField, TextAreaField, PasswordField, validators

from passlib.hash import sha256\_crypt

from functools import wraps

# Config MySQL

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = ''

app.config['MYSQL\_DB'] = 'myflaskapp'

app.config['MYSQL\_CURSORCLASS'] = 'DictCursor'

#INIT MYSQL

mysql = MySQL(app)

class RegisterForm(Form):

name = StringField('Name', [validators.Length(min=1, max=50)])

username = StringField ('Username', [validators.Length(min=4, max=25)])

email = StringField('Email', [validators.Length(min=6, max=50)])

password = PasswordField('Password', [

validators.DataRequired(),

validators.EqualTo('confirm', message='Password do not match')

])

confirm = PasswordField ('Confirm Password')

@app.route('/register', methods=['GET', 'POST'])

def register():

form = RegisterForm(request.form)

if request.method == 'POST' and form.validate():

name = form.name.data

email = form.email.data

username = form.username.data

password = sha256\_crypt.encrypt(str(form.password.data))

# Create Cursor

cur = mysql.connection.cursor()

#Execute Query

cur.execute("INSERT INTO users (name, email, username, password) VALUES (%s, %s, %s, %s)", (name, email, username, password))

# Commit to DB

mysql.connection.commit()

# Close connection

cur.close()

flash('You are now registered and can log in', 'success')

return redirect(url\_for('index'))

return render\_template('register.html', form=form)

**EN EL DIRECTORIO TEMPLATES**

mkdir includes

cd includes

vim \_formhelpers.html

{

{% macro render\_field(field) %}

{{ field.label }}

{{ field(\*\*kwargs) |safe }}

{% if field.errors %}

{% for error in field.errors %}

<span class=”help.inline”>{{ error }}</span>

{% endfor %}

{% endif %}

{% endmacro %}

}

vim \_messages.html

{

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

{% for category, message in messages %}

<div class=”alert alert-{{ category }}”>{{ message }}</div>

{% endfor %}

{% endif %}

{% endwith %}

{% if error %}

<div class=”alert alert-danger”>{{error}}</div>

{% endif %}

{% if msg %}

<div class=”alert alert-success”>{{msg}}</div>

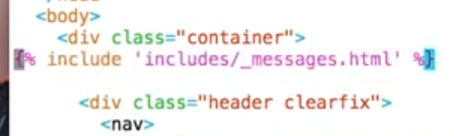
{% endif %}

}

**EN EL DIRECTORIO TEMPLATES NUEVAMENTE:**

**BASE.HTML**

{% include 'includes/\_messages.html' %}



<li role=”presentation”><a href=”/register”>Register</a></li>



**REGISTER.HTML**

<!--register.html-->

{% extends "base.html" %}

{% block title %} Register{% endblock %}

{% block body %}

<h1>Register</h1>

{% from "includes/\_formhelpers.html" import render\_field %}

<form method="POST" action="">

<div class="form-group">

{{render\_field(form.name, class\_="form-control")}}

</div>

<div class="form-group">

{{render\_field(form.email, class\_="form-control")}}

</div>

<div class="form-group">

{{render\_field(form.username, class\_="form-control")}}

</div>

<div class="form-group">

{{render\_field(form.password, class\_="form-control")}}

</div>

<div class="form-group">

{{render\_field(form.confirm, class\_="form-control")}}

</div>

<p><input type="submit" class="btn btn-primary" value="Submit"> </p>

</form>

{% endblock %}

**VIEWS.PY**

app.secret\_key=’secrect123’

****

**PROBAMOS:**

Recordar tener prendido el servicio mysqld:

sudo systemctl start mysqld

sudo systemctl status mysqld

export FLASK\_ENV=development

python3 -m flask run --host=0.0.0.0

Interfaz de usuario gráfica, Aplicación

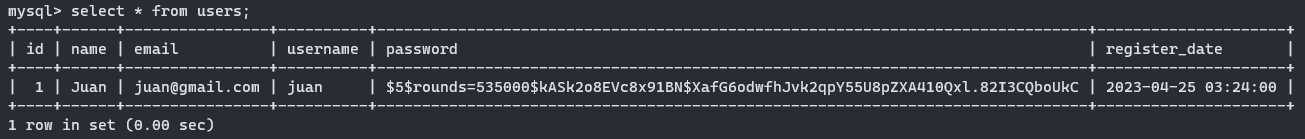
Descripción generada automáticamente



mysql -u root -p

USE myflaskapp

SELECT \* FROM users;



**FORMULARIO PARA LOGIN USUARIOS**

**views.py**

# User Login

@app.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

# Get Form Fields

username= request.form['username']

password\_candidate= request.form['password']

# Create cursor

cur = mysql.connection.cursor()

# Get user by username

result = cur.execute("SELECT \* FROM users WHERE username=%s", [username])

if result> 0:

# Get stored hash

data = cur.fetchone()

password = data['password']

# Compare Passwords

if sha256\_crypt.verify(password\_candidate, password):

# Passed

session['logged\_in'] = True

session ['username'] = username

flash('You are logged in', 'success')

app.logger.info('PASSWORD MATCHED')

return redirect(url\_for('dashboard'))

else:

app.logger.info('PASSWORD NO MATCHED')

error = 'Invalid login'

return render\_template('login.html', error=error)

# Close connection

cur.close()

else:

app.logger.info('NO USER')

error = 'Username not found'

return render\_template('login.html', error=error)

return render\_template('login.html')

# Check if user logged in

def is\_logged\_in(f):

@wraps(f)

def wrap(\*args, \*\*kwargs):

if 'logged\_in' in session:

return f(\*args, \*\*kwargs)

else:

flash('Unauthorized, Plese login', 'danger')

return redirect(url\_for('login'))

return wrap

# Dashboard

@app.route('/dashboard')

@is\_logged\_in

def dashboard():

return render\_template('dashboard.html')

# Logout

@app.route('/logout')

def logout():

session.clear()

flash('You are now logged out', 'success')

return redirect(url\_for('login'))

**En templates base.html modificamos:** <nav>

<ul class="nav nav-pills pull-right">

<li role="presentation"><a href="/">Home</a></li>

<li role="presentation"><a href="/about">About</a></li>

<li role="presentation"><a href="/articles">Articles</a></li>

{% if session.logged\_in %}

<li role="presentation"><a href="/dashboard">Dashboard</a><li>

<li role="presentation"><a href="/logout">Logout</a><li>

{% else %}

<li role="presentation"><a href="/register">Register</a></li>

<li role="presentation"><a href="/login">Login</a></li>

{% endif %}

<li role="presentation"><a href="/http://flask.pocoo.org" target="\_blank">More About Flask</a></li>

</ul>

</nav>



**En templates creamos login.html:**

{% extends 'base.html' %}

{% block body %}

<h1>Login</h1>

<form action="" method="post">

<div class="form-group">

<label>Username</label>

<input type="text" name="username" class="form-control" value="{{request.form.username}}">

</div>

<div class="form-group">

<label>Password</label>

<input type="password" name="password" class="form-control" value="{{request.form.password}}">

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

{% endblock %}

**En templates creamos dashboard.html:**

{% extends 'base.html' %}

{% block body %}

<h1>Dashboard <small> Welcome {{session.username}} </small></h1>

<a class="btn btn-success" href="/add\_article">Add Article</a>

<hr>

<table class="table table-striped">

<tr>

<th>ID</th>

<th>Title</th>

<th>Author</th>

<th>Date</th>

<th></th>

<th></th>

</tr>

{% for article in articles %}

<tr>

<td>{{article.id}}</td> <!-- id -->

<td>{{article.title}}</td> <!-- title -->

<td>{{article.author}}</td> <!-- author -->

<td>{{article.create\_date}}</td> <!-- create\_date -->

<td><a href="edit\_article/{{article.id}}" class="btn btn-default pull-right">Edit</a>

<td>

<form action="{{url\_for('delete\_article', id=article.id)}}" method="post">

<input type="hidden" name="\_method" value="DELETE">

<input type="submit" name="" value="Delete" class="btn btn-danger">

</form>

</td>

<td></td>

</tr>

{% endfor %}

</table> {% endblock %}

**PROBAMOS:**

Recordar tener prendido el servicio mysqld:

sudo systemctl start mysqld

sudo systemctl status mysqld

export FLASK\_ENV=development

python3 -m flask run --host=0.0.0.0

mysql -u root -p

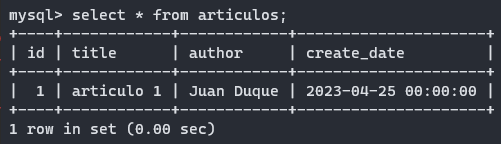
USE myflaskapp

CREATE TABLE articulos(id INT(11) PRIMARY KEY, title VARCHAR(100), author VARCHAR(100), create\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP);

Show TABLES;

DESCRIBE articulos;

INSERT INTO articulos VALUES (1,'articulo 1','Juan Duque', '2023-04-25');



INCOMPLETO!!!

**LOS POSIBLES ERRORES PARA CORREGIR:**

* **IDENTACIÓN O SANGRIA, REVISAR BIEN EL CODIGO!**
* **COMILLAS SIMPLES O DOBLES, EN EL WORD SE PEGAN DIFERENTE QUE EN LA LINEA DE COMANDOS OJO**
* **LO QUE ESTA IMCOMPLETO AQUI EN EL WORD ESTA COMPLETO EN NOTA!!!.TXT (README) Y EN TODOS LOS ARCHIVOS.**
* **DISFRUTEN 😊**

2. Administración de bases de datos con MySQLWorkbench

* Instale MySQLWorkbench

Disponible en <https://www.mysql.com/products/workbench/>

* Conéctese al servidor mysql instalado. Para esto necesitara crear un nuevo usuario en el servidor con privilegios

mysql -u root -p

mysql> CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';

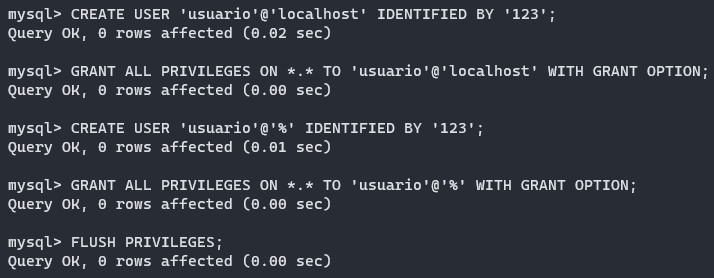
mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'username'@'localhost' WITH GRANT OPTION;

mysql> CREATE USER 'username'@'%' IDENTIFIED BY 'password';

mysql> GRANT ALL PRIVILEGES ON \*.\* TO 'username'@'%' WITH GRANT OPTION;

mysql> FLUSH PRIVILEGES;

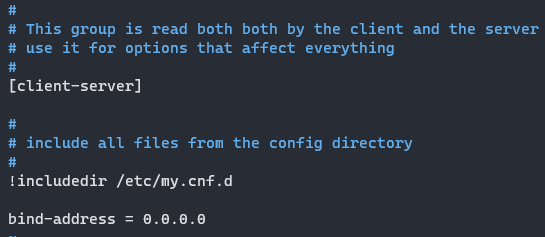
service mysqld restart

****

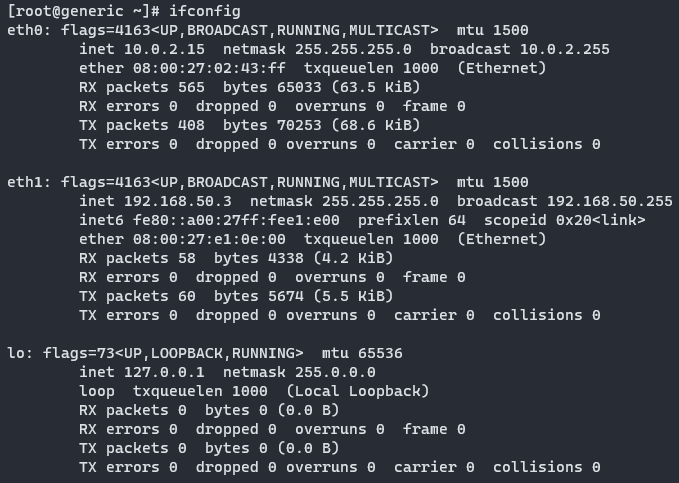
**Para esto habilitamos las conexiones remotas en la siguiente ruta:**

vim /etc/my.cnf

bind-address = 0.0.0.0

****

**Ifconfig para ver la IP que es la que usaremos en workbench para conectarnos**

****

**Probamos en workbench:**

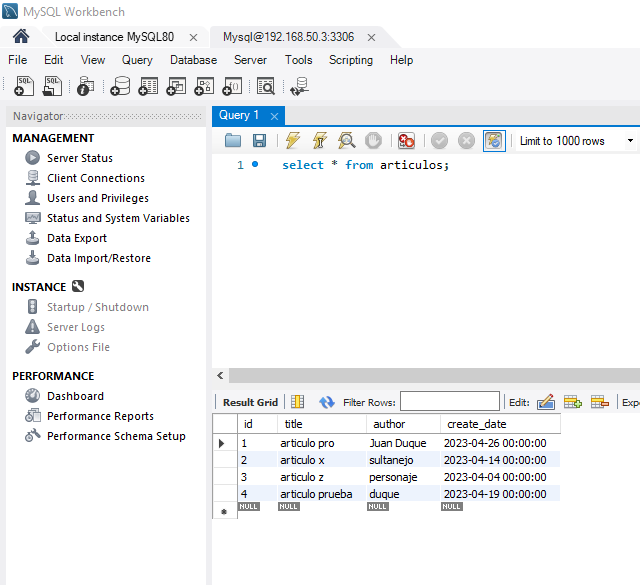


**Hostname:** IP del servidor  
**Username:** El usuario que creamos anteriormente  
**Default Schema:** El nombre de la BD que estamos utilizando

Al presionar OK, pide la contraseña, se coloca la del usuario y listo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

****