



Universidad
del Valle de México

LAUREATE INTERNATIONAL UNIVERSITIES®

SISTEMAS MULTIUSUARIOS

Mtro. Eder D, Reboloso Balderas

Actividad 8. Proyecto
integrador. Etapa 2



Alumna: Matilde Sierra Hernandez
040042609

Actividad 4. Proyecto integrador. Etapa 1

Esta actividad consiste en aplicar los conocimientos adquiridos a lo largo del curso y retomar lo aprendido en asignaturas previas, de modo que integres diferentes disciplinas. De igual forma se toman como referente actividades elaboradas previamente, lo que garantiza la transversalidad de los contenidos revisados para fortalecer el desarrollo de competencias.

El proyecto consiste en instalar la infraestructura de un sistema distribuido bajo una arquitectura cliente servidor. Para aplicar el concepto de sistema distribuido de forma clara, deberán utilizarse por lo menos dos equipos de cómputo conectados en red.

El sistema distribuido a instalar constará de:

- Servidor de base de datos (Equipo servidor)
- Servidor Web (Equipo servidor)
- Cliente (Equipo cliente)

El objetivo de la primera parte del Proyecto integrador es realizar la instalación y configuración del equipo servidor:

- servidor de base de datos
- servidor web

Explicación del proceso realizado

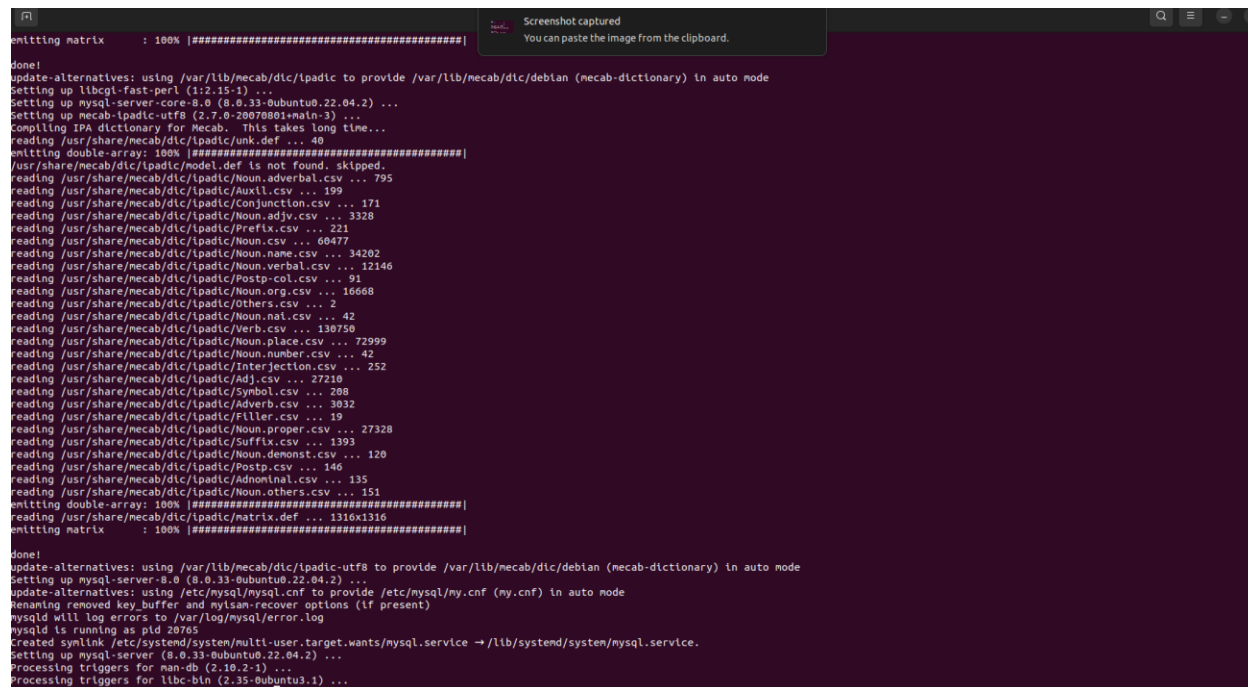
Instalación

Instala y configura el servidor de base de datos de tu preferencia en el equipo servidor. Se sugiere el servidor de MySQL Server, el cuál puedes descargar de la siguiente página:

<https://dev.mysql.com/downloads/>

La base de datos se creo en Ubuntu.

Antes de instalar cualquier paquete, es recomendable actualizar el sistema operativo.



```

enttting matrix : 100% |#####|
done!
update-alternatives: using /var/lib/mecab/dic/ipadic to provide /var/lib/mecab/dic/debian (mecab-dictionary) in auto mode
Setting up libcgl-fast-perl (1:2.15-1) ...
Setting up mysql-server-core-8.0 (8.0.33-0ubuntu0.22.04.2) ...
Setting up mecab-ipadic-utf8 (2.7.0-29070801+main-3) ...
Compiling IPA dictionary for Mecab. This takes long time...
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
enttting double-array: 100% |#####|
/usr/share/mecab/dic/ipadic/model.def is not found. skipped.
reading /usr/share/mecab/dic/ipadic/Noun.adverbal.csv ... 795
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
reading /usr/share/mecab/dic/ipadic/conjunction.csv ... 171
reading /usr/share/mecab/dic/ipadic/Noun.adjv.csv ... 3328
reading /usr/share/mecab/dic/ipadic/prefix.csv ... 221
reading /usr/share/mecab/dic/ipadic/Noun.csv ... 60477
reading /usr/share/mecab/dic/ipadic/Noun.name.csv ... 34202
reading /usr/share/mecab/dic/ipadic/Noun.verbal.csv ... 12146
reading /usr/share/mecab/dic/ipadic/Postp.col.csv ... 91
reading /usr/share/mecab/dic/ipadic/Noun.org.csv ... 16668
reading /usr/share/mecab/dic/ipadic/other.csv ... 2
reading /usr/share/mecab/dic/ipadic/Noun.nat.csv ... 42
reading /usr/share/mecab/dic/ipadic/Verb.csv ... 130750
reading /usr/share/mecab/dic/ipadic/Noun.place.csv ... 72999
reading /usr/share/mecab/dic/ipadic/Noun.number.csv ... 42
reading /usr/share/mecab/dic/ipadic/interjection.csv ... 252
reading /usr/share/mecab/dic/ipadic/Adj.csv ... 27210
reading /usr/share/mecab/dic/ipadic/symbol.csv ... 208
reading /usr/share/mecab/dic/ipadic/Adverb.csv ... 3032
reading /usr/share/mecab/dic/ipadic/filler.csv ... 19
reading /usr/share/mecab/dic/ipadic/Noun.proper.csv ... 27328
reading /usr/share/mecab/dic/ipadic/Suffix.csv ... 1393
reading /usr/share/mecab/dic/ipadic/Noun.demonst.csv ... 120
reading /usr/share/mecab/dic/ipadic/Postp.csv ... 146
reading /usr/share/mecab/dic/ipadic/Admonnal.csv ... 135
reading /usr/share/mecab/dic/ipadic/Noun.others.csv ... 151
enttting double-array: 100% |#####|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
enttting matrix : 100% |#####|
done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/mecab/dic/debian (mecab-dictionary) in auto mode
Setting up mysql-server-8.0 (8.0.33-0ubuntu0.22.04.2) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
mysql will log errors to /var/log/mysql/error.log
mysqld is running as pid 20765
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up mysql-server (8.0.33-0ubuntu0.22.04.2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...

```

[illegible]

Durante la instalación, se te solicito contraseña

```
lucero@Lucero-Inspiron-7506-2n1:~/natide$ sudo mysql
[sudo] password for lucero:
Sorry, try again.
[sudo] password for lucero:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ls
->
lucero@Lucero-Inspiron-7506-2n1:~/natide$ sudo mysql
[sudo] password for lucero:
Sorry, try again.
[sudo] password for lucero:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ls
-> CREATE DATABASE ms_multi_users_db;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'ls'
mysql> CREATE DATABASE ms_multi_users_db;
Query OK, 1 row affected (0.00 sec)

mysql> \c
mysql> \t
Outfile disabled.
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| ms_multi_users_db |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql>
```

La instalación de MySQL en Ubuntu es relativamente fácil debido a que el paquete de MySQL Server está disponible en los repositorios oficiales de Ubuntu. Esto simplifica el

proceso de instalación y actualización del software a través del sistema de gestión de paquetes APT.

Creación de Base de datos

Crea una base de datos para control de acceso e inserta datos en ella, los usuarios deberán iniciar sesión mediante nombre de usuario y contraseña, cada usuario deberá tener un rol asignado (Roles: Administrador, operativo, general).

El siguiente enlace te puede servir de apoyo: <https://evilnapsis.com/2016/09/05/3-modelos-de-base-de-datos-para-tabla-de-usuarios/>

Como primer paso, revisamos las bases de datos que tenemos actualmente:

```

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| ms_multi_users_db |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use ms_multi_users_db;
Database changed
mysql> ^C
mysql> CREATE TABLE Rol (
  -> id int,
  -> name varchar(20),
  -> created_at datetime
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_ms_multi_users_db |
+-----+
| Rol |
+-----+
1 row in set (0.01 sec)

mysql>

```

Creamos la tabla Rol

```

mysql> CREATE TABLE Rol (
  -> RolId int,
  -> name varchar(20) NOT NULL,
  -> created_at datetime,
  -> PRIMARY KEY (RolId)
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_ms_multi_users_db |
+-----+
| Rol |
+-----+
1 row in set (0.00 sec)

mysql>

```


Mostramos la tabla:

```
mysql> DESCRIBE Rol;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| RolId | int  | NO   | PRI | NULL    |       |
| name  | varchar(20) | NO   |     | NULL    |       |
| created_at | datetime | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Insertamos datos:

```
mysql> INSERT INTO Rol (name, created_at)
-> VALUES ('Administrador', NOW());
Query OK, 1 row affected (0.01 sec)

mysql> select * from Rol;
+-----+-----+-----+
| RolId | name      | created_at |
+-----+-----+-----+
| 1     | Administrador | 2023-06-04 22:02:10 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

```
mysql> CREATE TABLE User (
-> UserId int,
-> name varchar(20) NOT NULL,
-> password varchar(50) NOT NULL,
-> RolId int,
-> PRIMARY KEY (UserId),
-> FOREIGN KEY (RolId) REFERENCES Rol(RolId)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> DESCRIBE User;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| UserId | int  | NO   | PRI | NULL    |       |
| name  | varchar(20) | NO   |     | NULL    |       |
| password | varchar(50) | NO   |     | NULL    |       |
| RolId | int  | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_ms_multi_users_db |
+-----+
| Rol |
| User |
+-----+
2 rows in set (0.00 sec)

mysql>
```

Mostramos los datos en la tabla Rol:

```
mysql> INSERT INTO Rol (name, created_at)
-> VALUES ('Administrador', NOW());
Query OK, 1 row affected (0.01 sec)

mysql> select * from Rol;
+-----+-----+-----+
| RolId | name      | created_at |
+-----+-----+-----+
| 1     | Administrador | 2023-06-04 22:02:10 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO Rol (name, created_at)
-> VALUES ('Operativo', NOW());
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Rol (name, created_at)
-> VALUES ('General', NOW());
Query OK, 1 row affected (0.01 sec)

mysql> select * from Rol;
+-----+-----+-----+
| RolId | name      | created_at |
+-----+-----+-----+
| 1     | Administrador | 2023-06-04 22:02:10 |
| 2     | Operativo    | 2023-06-04 22:02:31 |
| 3     | General      | 2023-06-04 22:02:37 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

```
mysql> DESCRIBE User;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| UserId | int | NO | PRI | NULL | auto_increment |
| name | varchar(20) | NO | | NULL | |
| password | varchar(50) | NO | | NULL | |
| RolId | int | YES | MUL | NULL | |
| created_at | datetime | YES | | NULL | |
+-----+
5 rows in set (0.01 sec)

mysql> INSERT INTO User (name, password, RolId, created_at)
-> VALUES ('MatyAdmin', 'MatyAdmin', 1, NOW());
Query OK, 1 row affected (0.00 sec)

mysql> select * from User;
+-----+
| UserId | name | password | RolId | created_at |
+-----+
| 1 | MatyAdmin | MatyAdmin | 1 | 2023-06-04 22:07:13 |
+-----+
1 row in set (0.00 sec)

mysql>
```

```
mysql> DESCRIBE User;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| UserId | int | NO | PRI | NULL | auto_increment |
| name | varchar(20) | NO | | NULL | |
| password | varchar(50) | NO | | NULL | |
| RolId | int | YES | MUL | NULL | |
| created_at | datetime | YES | | NULL | |
+-----+
5 rows in set (0.01 sec)

mysql> INSERT INTO User (name, password, RolId, created_at)
-> VALUES ('MatyAdmin', 'MatyAdmin', 1, NOW());
Query OK, 1 row affected (0.00 sec)

mysql> select * from User;
+-----+
| UserId | name | password | RolId | created_at |
+-----+
| 1 | MatyAdmin | MatyAdmin | 1 | 2023-06-04 22:07:13 |
+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO User (name, password, RolId, created_at)
-> VALUES ('MatyOperativo', 'MatyOperativo', 2, NOW());
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO User (name, password, RolId, created_at)
-> VALUES ('MatyGeneral', 'MatyGeneral', 3, NOW());
Query OK, 1 row affected (0.01 sec)

mysql> select * from User;
+-----+
| UserId | name | password | RolId | created_at |
+-----+
| 1 | MatyAdmin | MatyAdmin | 1 | 2023-06-04 22:07:13 |
| 2 | MatyOperativo | MatyOperativo | 2 | 2023-06-04 22:08:03 |
| 3 | MatyGeneral | MatyGeneral | 3 | 2023-06-04 22:08:09 |
+-----+
3 rows in set (0.00 sec)

mysql>
```

Script de base de datos generada, con las instrucciones de inserción de datos

SCRIPTS

CREATE DATABASE ms_multi_users_db;

SHOW DATABASES;

use ms_multi_users_db;

CREATE TABLE Rol (
 RolId int NOT NULL AUTO_INCREMENT,
 name varchar(20) NOT NULL,

```

        created_at datetime,
        PRIMARY KEY (RolId)
    );

SHOW TABLES;

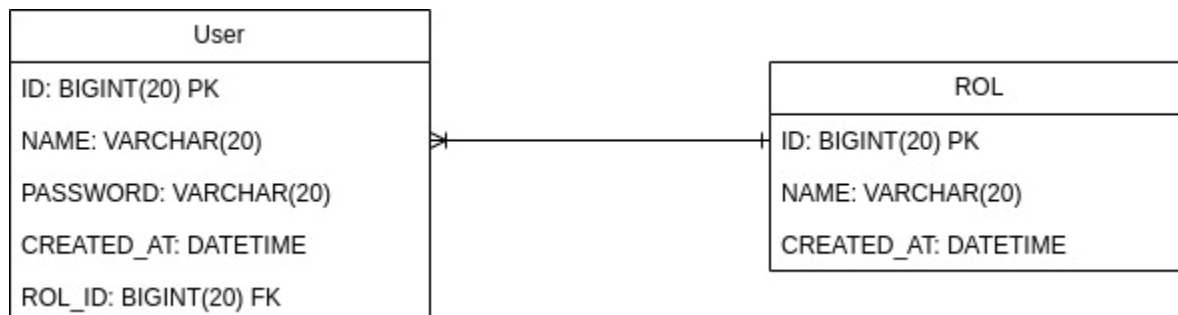
CREATE TABLE User (
    UserId int NOT NULL AUTO_INCREMENT,
    name varchar(20) NOT NULL,
    password varchar(50) NOT NULL,
    RolId int,
    created_at datetime,
    PRIMARY KEY (UserId),
    FOREIGN KEY (RolId) REFERENCES Rol(RolId)
);

INSERT INTO Rol (name, created_at)
VALUES ('Administrador', NOW());
INSERT INTO Rol (name, created_at)
VALUES ('Operativo', NOW());
INSERT INTO Rol (name, created_at)
VALUES ('General', NOW());

INSERT INTO User (name, password, RolId, created_at)
VALUES ('MatyAdmin', 'MatyAdmin', 1, NOW());
INSERT INTO User (name, password, RolId, created_at)
VALUES ('MatyOperativo', 'MatyOperativo', 2, NOW());
INSERT INTO User (name, password, RolId, created_at)
VALUES ('MatyGeneral', 'MatyGeneral', 3, NOW());

```

Diagrama entidad relación de la base de datos



Conclusión

Al realizar la instalación y creación de bases de datos en MySQL en Ubuntu, he adquirido conocimientos sobre cómo configurar y administrar un entorno de base de datos, cómo crear estructuras de datos y cómo interactuar con los datos utilizando comandos SQL básicos. Estas habilidades me permitirán comenzar a desarrollar aplicaciones y sistemas que requieran almacenamiento y manipulación de datos.

Actividad 8. Proyecto integrador. Etapa 2

Objetivo

El objetivo de la segunda parte del Proyecto integrador es establecer la conexión desde una aplicación web a la base de datos y utilizar el equipo cliente para visualizar la aplicación como un cliente Web, completando de éste modo la estructura del sistema distribuido planteado en la Etapa 1 de tu Proyecto integrador.

¿Qué hacer?

1. Utilizando el IDE de desarrollo de tu preferencia (IDE's de desarrollo sugeridos: Visual Studio, Eclipse, NetBeans), realiza una aplicación web de inicio de sesión, con los siguientes requerimientos:

Pantalla de inicio de sesión:

- a) Campo usuario (cuadro de texto para captura)
- b) Campo contraseña (cuadro de texto para captura)
- c) Campo rol (lista desplegable con los roles definidos)

La aplicación deberá conectarse a la base de datos creada en la Etapa 1 del proyecto integrador (de ser necesario realiza ajustes en tu base de datos) y realizar las validaciones correspondientes para realizar el inicio de sesión.

Envío de mensajes de error correspondientes, cuando los datos sean incorrectos.

2. Accede a la aplicación web desarrollada desde un navegador en el equipo Cliente y realiza las pruebas respectivas con los datos de tu base de datos.

3. Los entregables que formarán parte de tu trabajo son los siguientes:

- a) Explicación del proceso realizado
- b) Capturas de pantalla que evidencien el desarrollo
- c) Análisis del sistema distribuido desarrollado
- d) Esquema de Arquitectura del sistema distribuido realizado. Incluye explicación del funcionamiento de cada uno de los elementos.

Desarrollo

Iniciamos con la instalación de Flask y sus requerimientos.

```
(.venv) + proyecto2_msh pip install -r requirements.txt
Collecting Flask==0.12.2
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83 kB)
    |████████████████████████████████████████| 83 kB 1.0 MB/s
Collecting Jinja2>=2.4
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    |████████████████████████████████████████| 133 kB 4.9 MB/s
Collecting itsdangerous>=0.21
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Werkzeug>=0.7
  Downloading Werkzeug-2.3.6-py3-none-any.whl (242 kB)
    |████████████████████████████████████████| 242 kB 10.9 MB/s
Collecting click>=2.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
    |████████████████████████████████████████| 96 kB 12.8 MB/s
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.3.tar.gz (19 kB)
Using legacy 'setup.py install' for MarkupSafe, since package 'wheel' is not installed.
Installing collected packages: MarkupSafe, Jinja2, itsdangerous, Werkzeug, click, Flask
  Running setup.py install for MarkupSafe ... done
Successfully installed Flask-0.12.2 Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-2.3.6 click-8.1.3 itsdangerous-2.1.2
WARNING: You are using pip version 20.2.3; however, version 23.1.2 is available.
You should consider upgrading via the '/Users/gustavogalarreta/Projects/proyecto2_msh/.venv/bin/python3 -m pip install --upgrade pip' command.
(.venv) + proyecto2_msh
```

Instalamos Mysql

```
+ projecto2_msh brew install mysql
Running 'brew update --auto-update'...
=> Auto-updated Homebrew!
=> Updated Homebrew from 4.0.21 (8421b702c) to 4.0.24 (bb23116d3).
Updated 3 taps (homebrew/services, homebrew/core and homebrew/cask).
=> New Formulae
cargo-generate  conda-lock  gffread  gotestsum  hivex  judy  minigraph  shub  slsa-verifier  zrak
charls          getmail6  git-tools  grype  jsnm  ls-lint  ord  shush  typical
=> New Casks
audiocupcake          devpod          frappe-books          mumu-x
command-x            eset-cyber-security  graalvm-jdk          rio
Warning: Calling plist_options is deprecated! Use service.require_root instead.
Please report this issue to the elastic/tap tap (not Homebrew/brew or Homebrew/homebrew-core), or even better, submit a PR to fix it:
/opt/homebrew/Library/Taps/elastic/homebrew-tap/Formula/elasticsearch-full.rb:68

You have 26 outdated formulae and 1 outdated cask installed.

The 4.0.24 changelog can be found at:
https://github.com/Homebrew/brew/releases/tag/4.0.24
mysql 8.0.33.1 is already installed but outdated (so it will be upgraded).
=> Fetching dependencies for mysql: icu4c
=> Fetching icu4c
=> Downloading https://ghcr.io/v2/homebrew/core/icu4c/manifests/73.2
##### 100.0%
=> Downloading https://ghcr.io/v2/homebrew/core/icu4c/blobs/sha256:488714896b377fb1bb09f023e0740235730be9098a2bf70e932cd9d1386d4a4a
##### 100.0%
=> Fetching mysql
=> Downloading https://ghcr.io/v2/homebrew/core/mysql/manifests/8.0.33.2
##### 100.0%
=> Downloading https://ghcr.io/v2/homebrew/core/mysql/blobs/sha256:c50a59345c76255a1a30ff39f4890029e172375b169371f4cc28971d79902452
##### 100.0%
=> Upgrading mysql
8.0.33.1 -> 8.0.33.2

=> Installing dependencies for mysql: icu4c
=> Installing mysql dependency: icu4c
=> Pouring icu4c--73.2.arm64_monterey.bottle.tar.gz
🍺 /opt/homebrew/Cellar/icu4c/73.2: 268 files, 80.1MB
=> Installing mysql
=> Pouring mysql--8.0.33.2.arm64_monterey.bottle.tar.gz
```

El siguiente paso es crear el código usando Visual Studio Code con el lenguaje Python.

- Indicamos los requerimientos.

```
≡ requirements.txt ✕
projecto2_msh > ≡ requirements.txt
1 Flask==2.3.2
2 Jinja2==3.1.2
3 mysql-connector-python
```

- Creamos los controladores del login

```
login_controller.py X
proyecto2_msh > controllers > login_controller.py > ...
1  from db.db_connector import DbConnector
2
3  class LoginController:
4      def __init__(self, request):
5          self.request = request
6
7      # public methods
8
9      def login(self):
10         if self.__valid_login():
11             return True
12         else:
13             return False
14
15     # private methods
16
17     def __valid_login(self):
18         db = DbConnector()
19         users = db.find_user(self.__username(), self.__password(), self.__rol())
20         if users:
21             return True
22         else:
23             return False
24
25     def __username(self):
26         return self.request.form['username']
27
28     def __password(self):
29         return self.request.form['password']
30
31     def __rol(self):
32         return self.request.form['rol']
33
```

- Controladores para Flask

```

app.py 1 x
proyecto2_msh > app.py > login
1  from flask import Flask, render_template, request
2  from controllers.login_controller import LoginController
3
4  # The default folder name should be "templates" else need to mention custom folder name
5  app = Flask(__name__, template_folder='views', static_folder='static_files')
6
7  @app.route("/")
8  def main():
9      return render_template('home.html')
10
11 @app.route('/login', methods=['GET', 'POST'])
12 def login():
13     if request.method == 'POST':
14         login_controller = LoginController(request)
15         if login_controller.login():
16             return render_template('login_success.html')
17         else:
18             return render_template('login_try_again.html')
19
20     return render_template('login.html')
21

```

- Asignamos Ajax para login.

```

JS login.js x
proyecto2_msh > static_files > JS login.js > ...
1  $.ajax({
2      type: "POST",
3      contentType: 'application/json',
4      url: "/login",
5      data: {},
6      dataType: "json",
7      success: function(result) {
8          console.log("it works!!");
9          document.write(response);
10     }
11 });

```

- Asignamos Usuario, contraseña y roles.

```
login_controller.py X
proyecto2_msh > controllers > login_controller.py > LoginController > login
1  from db.db_connector import DbConnector
2
3  class LoginController:
4      def __init__(self, request):
5          self.request = request
6
7      # public methods
8
9      def login(self):
10         if self.__valid_login():
11             return True
12         else:
13             return False
14
15     # private methods
16
17     def __valid_login(self):
18         db = DbConnector()
19         users = db.find_user(self.__username(), self.__password(), self.__rol())
20         if users:
21             return True
22         else:
23             return False
24
25     def __username(self):
26         return self.request.form['username']
27
28     def __password(self):
29         return self.request.form['password']
30
31     def __rol(self):
32         return self.request.form['rol']
33
```


- Creación del servidor web:

```

db_connector.py 1 X
proyecto2_msh > db > db_connector.py > ...
1  import mysql.connector
2
3  class DbConnector:
4      DB_HOST = 'localhost'
5      DB_USER = 'root'
6      DB_PASSWORD = 'password'
7      DB_NAME = 'ms_multi_users_db'
8
9      def __init__(self):
10         self.db_manager = self.__manage_db()
11
12     def find_user(self, name, password, rol_name):
13         rol_id = self.find_rol(rol_name)
14         query = f"SELECT * FROM USER WHERE name = '{name}' AND password = '{password}' AND RolId = '{rol_id}'"
15         current_db = self.__db()
16         cursor = current_db.cursor()
17         cursor.execute(query)
18         users = cursor.fetchall()
19         if users == []:
20             return False
21         return users[0]
22
23     def find_rol(self, rol_name):
24         query = f"SELECT * FROM ROL WHERE name = '{rol_name}'"
25         current_db = self.__db()
26         cursor = current_db.cursor()
27         cursor.execute(query)
28         rol = cursor.fetchall()
29         if rol:
30             return rol[0][0]
31         else:
32             return None
33
34     # private methods
35
36     def __manage_db(self):
37         if self.__exist_db():
38             return True

```

db_connector.py 1 X

proyecto2_msh > db > db_connector.py > ...

```

37     if self.__exist_db():
38         return True
39     else:
40         self.__create_db()
41
42     def __exist_db(self):
43         current_db = self.__connector()
44         cursor = current_db.cursor()
45         cursor.execute("SHOW DATABASES")
46         databases = cursor.fetchall()
47         if databases == None:
48             return False
49         for db_name in databases:
50             if db_name[0] == self.DB_NAME:
51                 return True
52         return False
53
54     def __connector(self):
55         db = mysql.connector.connect(
56             host = self.DB_HOST,
57             user = self.DB_USER,
58             password = self.DB_PASSWORD
59         )
60
61         return db
62
63     def __create_db(self):
64         current_db = self.__connector()
65         cursor = current_db.cursor()
66         query = f"CREATE DATABASE {self.DB_NAME}"
67         cursor.execute(query)
68
69         current_db = self.__db()
70         cursor = current_db.cursor()
71         self.__create_tables(cursor)
72         self.__insert_rol(current_db, cursor)
73         self.__insert_users(current_db, cursor)
74         return True

```

db_connector.py 1 X

proyecto2_msh > db > db_connector.py > DbConnector > __connector

```

65     cursor = current_db.cursor()
66     query = f"CREATE DATABASE {self.DB_NAME}"
67     cursor.execute(query)
68
69     current_db = self.__db()
70     cursor = current_db.cursor()
71     self.__create_tables(cursor)
72     self.__insert_rol(current_db, cursor)
73     self.__insert_users(current_db, cursor)
74     return True
75
76     def __create_tables(self, cursor):
77         cursor.execute(
78             "CREATE TABLE Rol (RolId int NOT NULL AUTO_INCREMENT, name varchar(50) NOT NULL)"
79         )
80         cursor.execute(
81             "CREATE TABLE User (UserId int NOT NULL AUTO_INCREMENT, name varchar(50) NOT NULL, password varchar(50) NOT NULL)"
82         )
83
84     def __insert_rol(self, db, cursor):
85         query = "INSERT INTO Rol (name, created_at) VALUES (%s, %s)"
86         values = [
87             ('Administrador', None),
88             ('Operativo', None),
89             ('General', None)
90         ]
91         cursor.executemany(query, values)
92         db.commit()
93
94     def __insert_users(self, db, cursor):
95         query = "INSERT INTO User (name, password, RolId, created_at) VALUES (%s, %s, %s, %s)"
96         values = [
97             ('MatyAdmin', 'MatyAdmin', self.find_rol('Administrador'), None),
98             ('MatyOperativo', 'MatyOperativo', self.find_rol('Operativo'), None),
99             ('MatyGeneral', 'MatyGeneral', self.find_rol('General'), None)
100         ]
101         cursor.executemany(query, values)
102         db.commit()
103
104     def __db(self):
105         db = mysql.connector.connect(
106             host = self.DB_HOST,
107             user = self.DB_USER,
108             password = self.DB_PASSWORD,
109             database = self.DB_NAME
110         )
111         return db
112
113

```

Agregamos las vistas para:

Home

```

home.html X
proyecto2_msh > views > home.html > html > head > style
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
7 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
8 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
9 <style>
10 body,h1,h2,h3,h4,h5,h6 {font-family: "Lato", sans-serif;}
11 body, html {
12   height: 100%;
13   color: #777;
14   line-height: 1.8;
15 }
16
17 /* Create a Parallax Effect */
18 .bgimg-1, .bgimg-2, .bgimg-3 {
19   background-attachment: fixed;
20   background-position: center;
21   background-repeat: no-repeat;
22   background-size: cover;
23 }
24
25 /* First image (Logo. Full height) */
26 .bgimg-1 {
27   background-image: url('/w3images/parallax1.jpg');
28   min-height: 100%;
29 }
30
31 .w3-wide {letter-spacing: 10px;}
32 .w3-hover-opacity {cursor: pointer;}
33
34 /* Turn off parallax scrolling for tablets and phones */
35 @media only screen and (max-device-width: 1600px) {
36   .bgimg-1, .bgimg-2, .bgimg-3 {
37     background-attachment: scroll;
38     min-height: 400px;
39   }
40 }
41 </style>
42
43 </head>
44 <body>
45
46 <div class="bgimg-1 w3-display-container w3-opacity-min" id="home">
47   <div class="w3-display-middle" style="white-space:nowrap;">
48     <a href="login" target="_blank" class="w3-center w3-padding-large w3-black w3-xlarge w3-wide w3-animate-opacity"> LOGIN</a>
49   </div>
50 </div>
51
52 <!-- Footer -->
53 <footer class="w3-center w3-black w3-padding-64 w3-opacity w3-hover-opacity-off">
54   <div class="w3-xlarge w3-section">
55     <i class="fa fa-twitter w3-hover-opacity"></i>
56     <i class="fa fa-linkedin w3-hover-opacity"></i>
57   </div>
58 </footer>
59
60 </body>
61 </html>

```

Inicio de pantalla exitoso:

```

home.html login_success.html X
proyecto2_msh > views > login_success.html > ...
1 <!DOCTYPE html>
2 <html>
3 <meta name="viewport" content="width=device-width, initial-scale=1">
4 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
5 <body class="">
6 <h2>Tu Login ha sido exitoso</h2>
7 <div class="w3-panel w3-pale-green w3-border">
8 <h3>Felicitades!</h3>
9 <p>Tus datos de inicio de session son correctos.</p>
10 </div>
11 </body>
12 </html>
13
14 <a href="login" target="_blank">Volver al Login</a>

```

Inicio de pantalla error:

```

home.html login_try_again.html X
proyecto2_msh > views > login_try_again.html > ...
1 <!DOCTYPE html>
2 <html>
3 <meta name="viewport" content="width=device-width, initial-scale=1">
4 <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
5 <body class="">
6 <h2>Tu Login no ha sido exitoso</h2>
7 <div class="w3-panel w3-pale-red w3-border">
8 <h3>Error!</h3>
9 <p>Tus datos de inicio de session no correctos.</p>
10 </div>
11 </body>
12 </html>
13
14 <a href="login" target="_blank">Volver al Login</a>

```

Inicio de sesión:

```

home.html login.html X
proyecto2_msh > views > login.html > ...
1 <!-- how to use css in python_ flask-->
2 <!DOCTYPE html>
3 <html lang="en">
4   <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
5   <head>
6     <!-- Load CSS in Flask with html-->
7     <link rel="stylesheet" href="/static_files/main.css" />
8     <!-- Load JavaScript in Flask with html-->
9   </head>
10  <body>
11    <div class="logo"></div>
12    <div class="login-block">
13      <form action="login" method="post">
14        <h1>Login</h1>
15        <input type="text" value="" placeholder="user@gmail.com" name="username" id="username" />
16        <input type="password" value="" placeholder="password" name="password" id="password" />
17        <select name="rol" id="rol">
18          <option value="Administrador">Administrador</option>
19          <option value="Operativo">Operativo</option>
20          <option value="General">General</option>
21        </select>
22        <input type="submit" value="Submit">
23      </form>
24    </div>
25  </body>
26 </html>
27

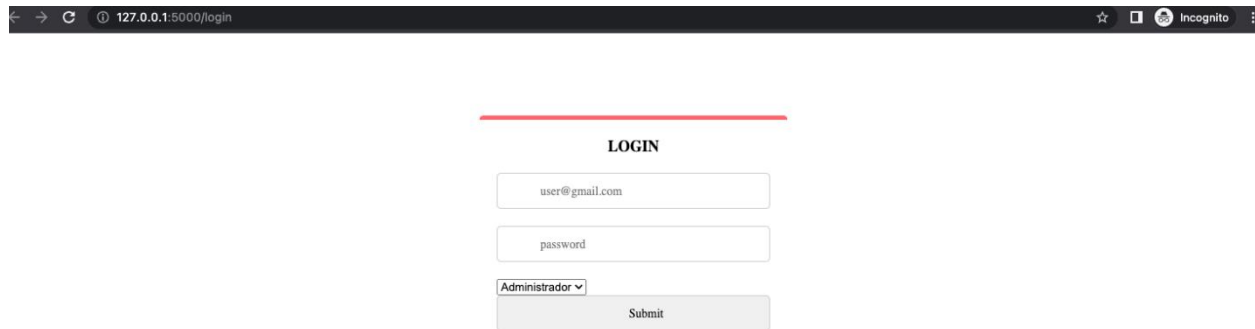
```

Pantalla de inicio de sesión:



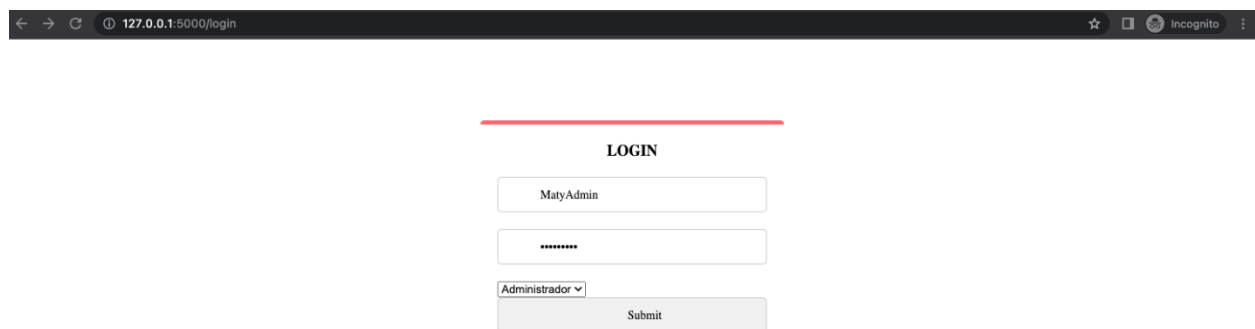
LOGIN

Ingresamos al inicio de sesión:



A screenshot of a web browser window. The address bar shows the URL "127.0.0.1:5000/login". The page title is "LOGIN". The form contains three input fields: the first contains "user@gmail.com", the second contains "password", and the third is a dropdown menu with "Administrador" selected. A "Submit" button is located below the dropdown menu.

Ingresamos con el usuario y contraseña asignados en la base de datos:

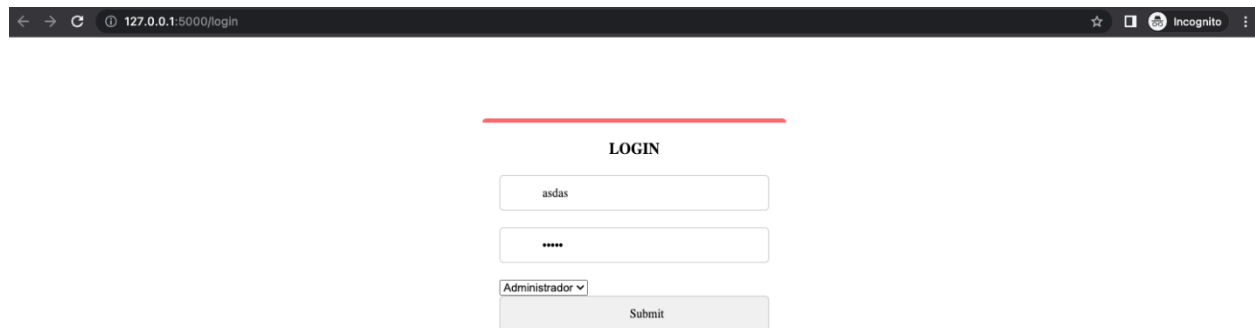


A screenshot of a web browser window, identical to the one above. The address bar shows the URL "127.0.0.1:5000/login". The page title is "LOGIN". The form contains three input fields: the first contains "MatyAdmin", the second contains "*****", and the third is a dropdown menu with "Administrador" selected. A "Submit" button is located below the dropdown menu.

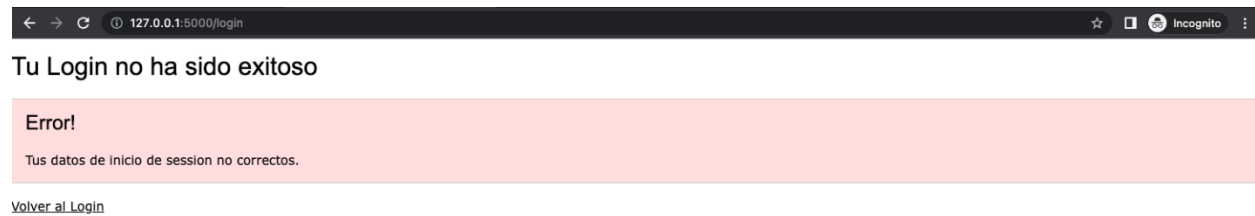
Comprobamos que es correcto y regresamos al inicio:



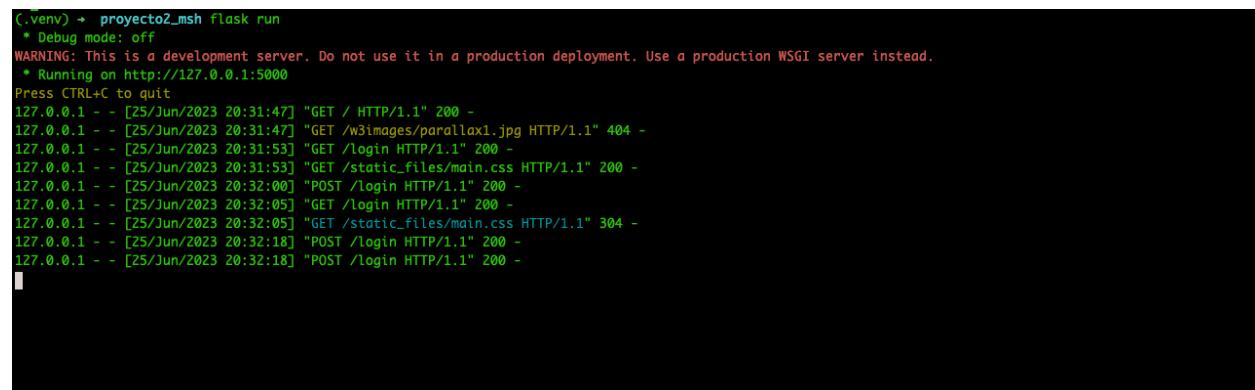
Intentamos ingresar con un usuario no existente en la base de datos:



Recibimos un error y regresamos al inicio:



Corroboramos de que no existen errores realizados en el inicio de sesión:



Durante el proyecto tuve varios errores, como lazy load el cual estaba ejecutando en segundo plano el cursor ya que al principio lo había duplicado, otro error fue callable, no se ejecutaba el cursor.

La definición de los roles en la creación de las tablas también me encontré con errores al igual que al agregar a los usuarios.

La creación del front fue en parte con ayuda de páginas externas, ya que no estoy muy familiarizada con HTML.

Conclusión

El proyecto fue un reto, tenía conocimiento de bases de datos lo que me facilitó realizar la primera parte del proyecto, pero la segunda parte fue más difícil, porque fue un código largo en el cual tuve que enlazar la base de datos con el servidor para que el cliente tenga visibilidad. El usar Flask y realizar las conexiones entre MySQL y Python fueron todo un reto, también se tiene que asegurar de que el usuario tenga los permisos adecuados para acceder a la base de datos y ejecutar las consultas necesarias.

Es importante revisar y depurar cuidadosamente el código para identificar y corregir posibles errores.

Referencias

School, W. (s.f.). W3 School. Obtenido de Python MySQL Create Database: https://www.w3schools.com/python/python_mysql_create_db.asp

Microsoft Azure (2017). Procesamiento por lotes Haga clic para ver más opciones [sitio web.] Recuperado de <https://docs.microsoft.com/es-es/azure/architecture/data-guide/big-data/batch-processing>

Yuson, P. (2017). Types of Mainframe Processing: Batch and Online Haga clic para ver más opciones [sitio web]. Recuperado de <http://conceptsolutionsbc.com/wordpress/types-of-mainframe-processing-batch-and-online/>