

Actividad de la Lección

En esta actividad, los estudiantes aplicarán los conceptos aprendidos para diseñar e implementar una funcionalidad de autenticación para su proyecto Capstone. La actividad incluye la configuración de rutas seguras y la creación de sesiones para usuarios autenticados.

Actividad

1. Implementar autenticación básica:

- Crear un servidor Flask con Flask-Login para manejar usuarios autenticados, utiliza el código de ejemplo discutido en clase o el código en el repositorio del curso (<https://github.com/javierdastas/comp2052>).
- Configurar rutas restringidas para usuarios autenticados.

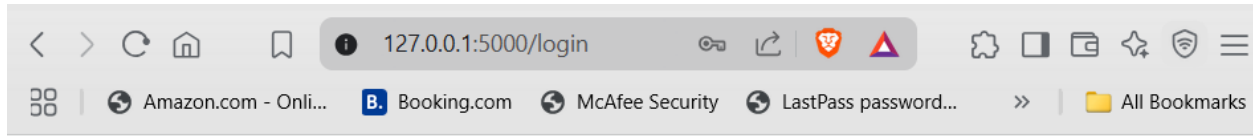
1. Diseño de la funcionalidad de autenticación para la Aplicación:

- Diseñar el flujo de autenticación del proyecto.
- Puedes ver los siguientes ejemplos:
- [Flowchart diagram tutorial \(user login example\)](#)
- [Flow chart || Login process flow chart](#)
- [Create Login Flowchart using Mindmap](#)
- Crear un esquema de usuario que incluya:
- Nombre de usuario.
- Contraseña (almacenada como hash).
- Rol o permisos.

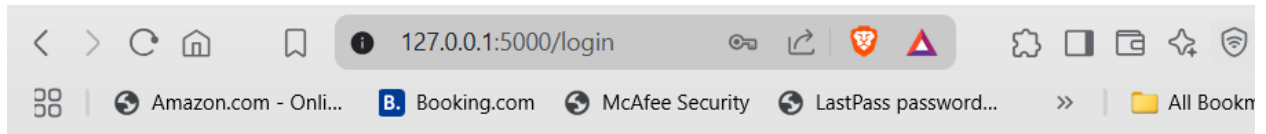
1. Entrega:

- Subir el código del servidor Flask con la funcionalidad implementada a tu repositorio de GitHub.
- Desarrolla un documento en formato PDF que incluya:
- Incluir un diagrama que muestre el flujo de autenticación del proyecto.
- Captura de pantallas de las pruebas de la aplicación en Flask que desarrollaste.

- Incluye la dirección de tu repositorio en GitHub que incluye el código de esta actividad

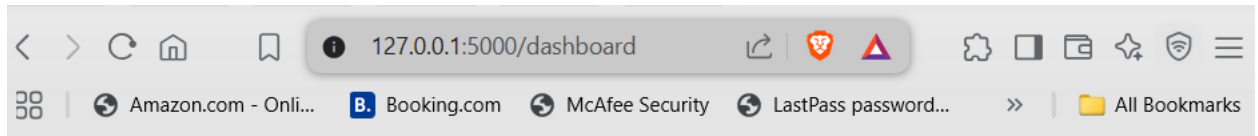


Bienvenido Zelda con rol user!

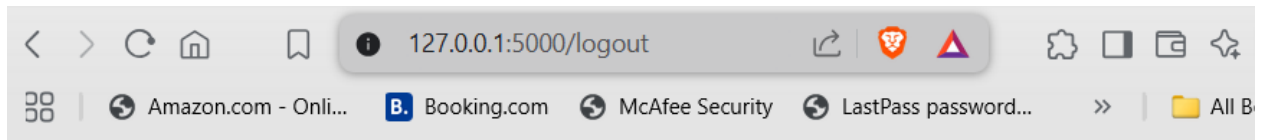


Usuario:

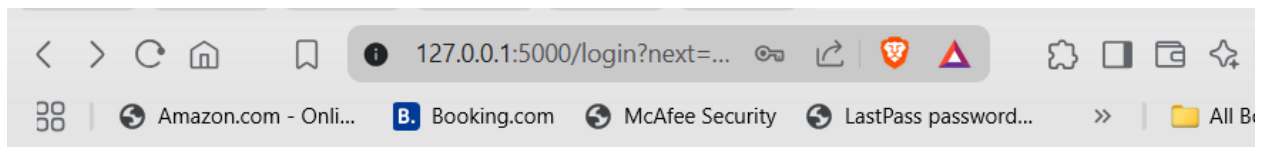
Contraseña:



Hola, Zelda! Bienvenido a tu panel de user.

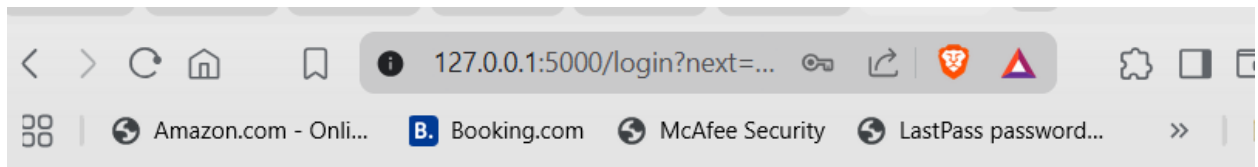


Sesión cerrada

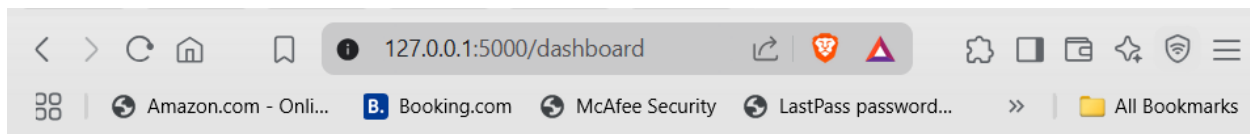


Usuario:

Contraseña:



Bienvenido juan con rol admin!



Hola, juan! Bienvenido a tu panel de admin.

```
Welcome hash.py
m3l1 > hash.py >...
1 from flask import Flask, request, redirect, url_for
2 from flask_login import LoginManager, UserMixin, login_user, logout_user, login_required, current_user
3 from werkzeug.security import generate_password_hash, check_password_hash
4
5 app = Flask(__name__)
6 app.secret_key = "clave_secreta"
7
8 login_manager = LoginManager(app)
9 login_manager.login_view = "login"
10
11 # Modelo de usuario
12 class User(UserMixin):
13     def __init__(self, id, username, password_hash, role):
14         self.id = id
15         self.username = username
16         self.password_hash = password_hash
17         self.role = role
18
19 # Crear hashes de contraseñas
20 hashed_password_juan = generate_password_hash("secreto")
21 hashed_password_maria = generate_password_hash("secreto")
22 hashed_password_link = generate_password_hash("ocarina")
23 hashed_password_zelda = generate_password_hash("trifuerza")
24
25
26 # Usuarios
27 users = {
28     "juan": User(1, "juan", hashed_password_juan, "admin"),
29     "maria": User(2, "maria", hashed_password_maria, "user"),
30     "link": User(3, "link", hashed_password_link, "user"),
31     "zelda": User(4, "zelda", hashed_password_zelda, "user")
32 }
```



```

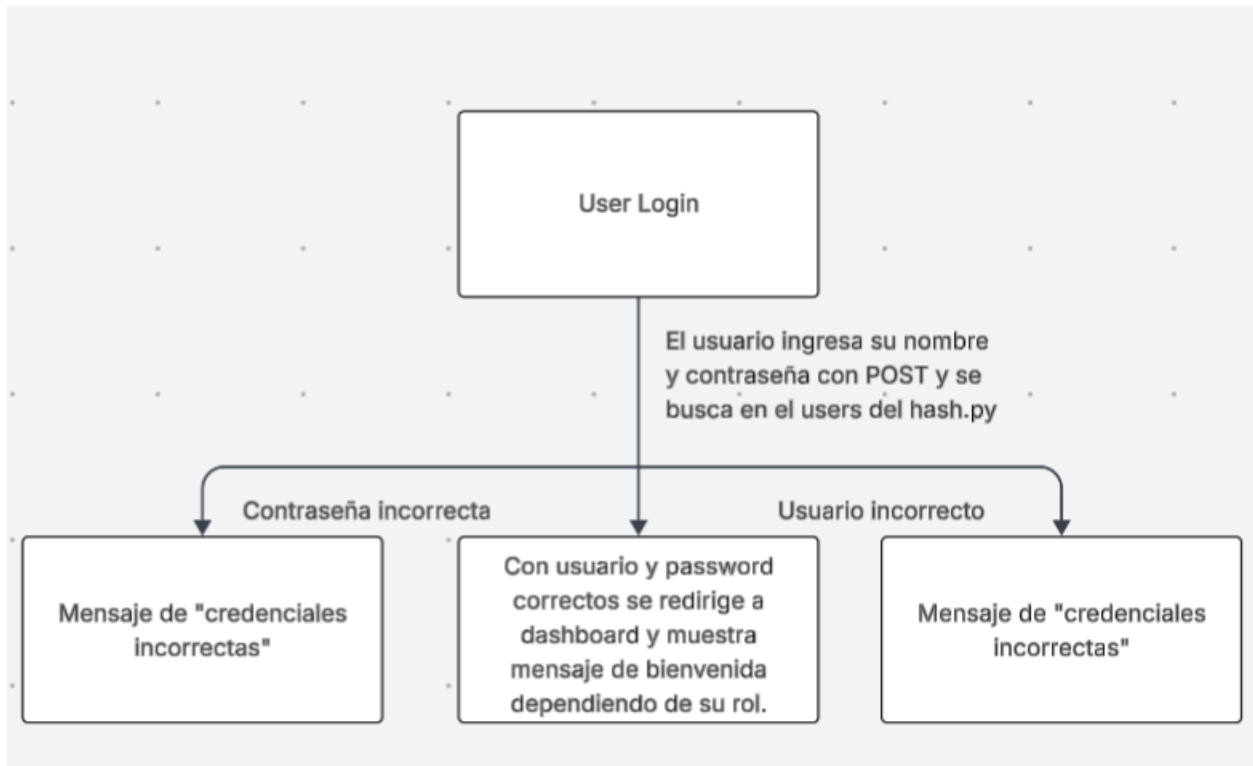
33
34 @login_manager.user_loader
35 def load_user(user_id):
36     for user in users.values():
37         if str(user.id) == user_id:
38             return user
39     return None
40
41 # Ruta para login
42 @app.route("/login", methods=["GET", "POST"])
43 def login():
44     if request.method == "POST":
45         username = request.form["username"]
46         password = request.form["password"]
47         user = users.get(username)
48         if user and check_password_hash(user.password_hash, password):
49             login_user(user)
50             return f"Bienvenido {current_user.username} con rol {current_user.role}!"
51         return "Credenciales incorrectas", 401
52     return '''
53         <form method="post">
54             Usuario: <input type="text" name="username"><br>
55             Contraseña: <input type="password" name="password"><br>
56             <input type="submit" value="Iniciar sesión">
57         </form>
58     '''
59
60 # Ruta para cerrar sesión
61 @app.route("/logout")
62 @login_required
63 def logout():
64     logout_user()
65     return "Sesión cerrada"
66
67 # Ruta protegida
68 @app.route("/dashboard")
69 @login_required

```

```

70 @login_required
71 def dashboard():
72     return f"Hola, {current_user.username}! Bienvenido a tu panel de {current_user.role}."
73
74 if __name__ == "__main__":
75     app.run(debug=True)

```



Link al GitHub:

<https://github.com/Drewster64/microserver/tree/master/Modulo3%20Apps/m3l1>