Estructura del Proyecto /proyecto |— app.py |— models.py |— /templates

— dashboard.html

| |--- login.html

```
mod03Lecc01 > templates > 5 dashboard.html > ...
       <!DOCTYPE html>
  1
       <html lang="es">
  2
       <head>
  3
  4
           <meta charset="UTF-8">
  5
           <title>Dashboard</title>
      </head>
  6
  7
       <body>
           <h2>Bienvenido {{ user.username }}</h2>
  8
  9
           Rol: {{ user.role }}
           <a href="{{ url_for('logout') }}">Cerrar sesión</a>
 10
 11
       </body>
       </html>
 12
```

```
mod03Lecc01 > templates > ■ login.html > ...
      <!DOCTYPE html>
  2
      <html lang="es">
      <head>
  4
          <meta charset="UTF-8">
  5
          <title>Iniciar Sesión</title>
  6
      </head>
  7
      <body>
          <h2>Iniciar Sesión</h2>
  8
  9
           {% with messages = get_flashed_messages() %}
            {% if messages %}
 10
 11
 12
                 {% for message in messages %}
                 {li>{{ message }}
 13
                {% endfor %}
 14
 15
              16
            {% endif %}
 17
           {% endwith %}
 18
 19
          <form method="POST">
              <input type="text" name="username" placeholder="Nombre de usuario" required><br>
 20
              <input type="password" name="password" placeholder="Contraseña" required><br>
 21
 22
              <button type="submit">Entrar</button>
 23
          </form>
 24
       </body>
 25
```

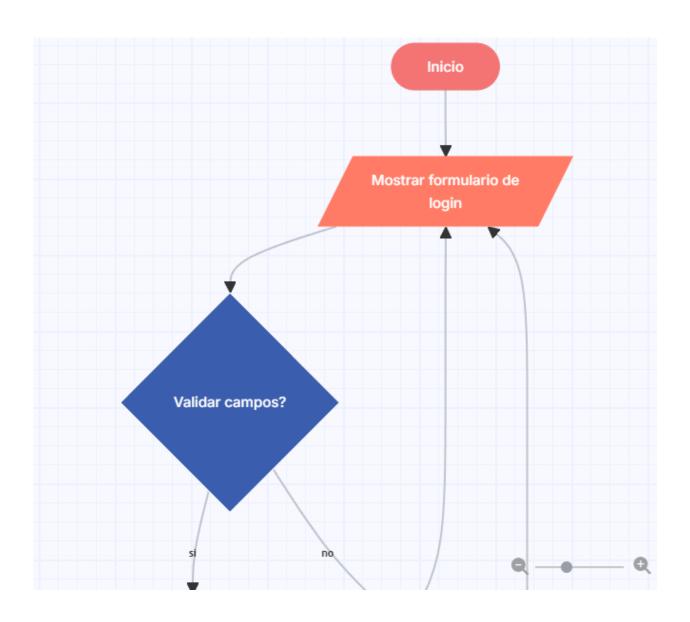
```
mod03Lecc01 > 👶 app.py > ...
       from flask import Flask, render_template, redirect, url_for, request, flash
       from flask_login import LoginManager, login_user, login_required, logout_user, current_user
      from models import users, User
      app = Flask(__name__)
      app.secret_key = 'clave_secreta'
  8
      login_manager = LoginManager()
      login_manager.init_app(app)
      login_manager.login_view = 'login'
 10
 11
 12
      @login manager.user loader
 13
      def load user(user id):
 14
           for user in users.values():
 15
               if str(user.id) == user_id:
 16
                  return user
 17
           return None
 18
 19
      @app.route('/')
      def home():
 20
 21
           return redirect(url_for('login'))
 23
      @app.route('/login', methods=['GET', 'POST'])
      def login():
 24
 25
           if request.method == 'POST':
               username = request.form['username']
 26
 27
               password = request.form['password']
 28
               user = users.get(username)
 29
```

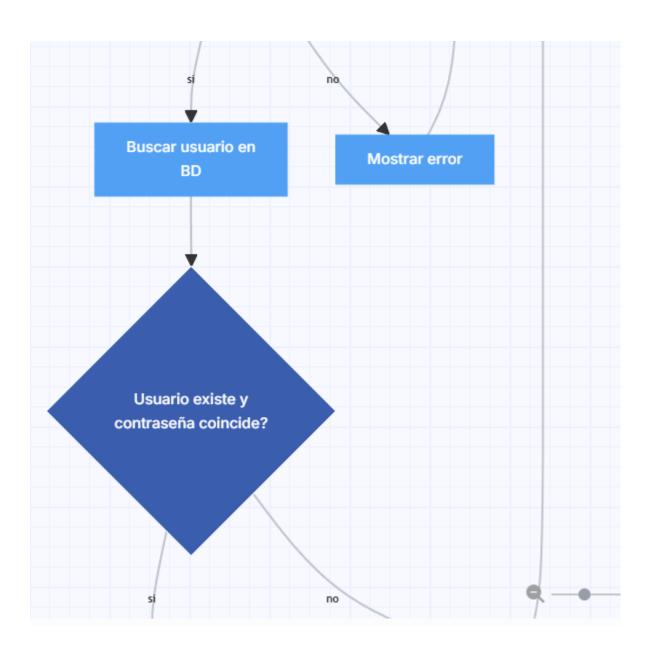
```
mod03Lecc01 > 👶 app.py > ...
      def login():
 24
               if user and user.check password(password):
 30
                   login user(user)
 31
                   flash('Inicio de sesión exitoso.')
 32
                   return redirect(url for('dashboard'))
 33
 34
               else:
 35
                   flash('Credenciales inválidas.')
           return render template('login.html')
 36
 37
 38
       @app.route('/dashboard')
 39
       @login required
 40
       def dashboard():
           return render template('dashboard.html', user=current user)
 41
 42
 43
       @app.route('/logout')
 44
       @login required
 45
      def logout():
 46
           logout user()
 47
           flash('Sesión cerrada.')
 48
           return redirect(url for('login'))
 49
       if __name__ == '__main_ ':
 50
 51
           app.run(debug=True)
 52
```

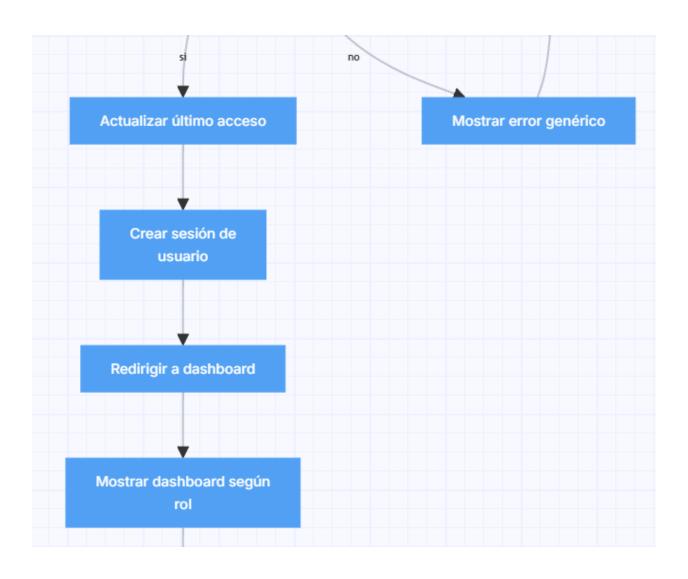
```
mod03Lecc01 > 🔷 models.py > ...
      from flask login import UserMixin
                                           Import "flask login" could not be resolved
      from werkzeug.security import generate password hash, check password hash
  3
  4
      class User(UserMixin):
          def __init__(self, id, username, password, role):
  5
  6
              self.id = id
               self.username = username
  7
               self.password hash = generate password hash(password)
  8
  9
               self.role = role
 10
          def check_password(self, password):
 11
               return check password hash(self.password hash, password)
 12
 13
 14
      # Simulación de base de datos
 15
      users = {
           "admin": User(1, "admin", "secreto", "admin"),
 16
           "user1": User(2, "user1", "clave123", "usuario")
 17
 18
```

Diagrama de Flujo de Autenticación

gleek_ Free Version → Upgrade **User login flowchart** Basic Inicio:queue Mostrar formulario de login:rectangleR Inicio→Mostrar formulario de login Validar campos?:diamond←Mostrar formulario de login Validar campos?-si→Buscar usuario en BD→Usuario existe y contraseña coincide? :diamond-si→Actualizar último acceso Actualizar último acceso→Crear sesión de usuario Crear sesión de usuario→Redirigir a dashboard Redirigir a dashboard → Mostrar dashboard según rol-→Opción de logout-→Destruir sesión→Redirigir a login→Fin:queue Validar campos?-no→Mostrar error Mostrar error→Mostrar formulario de login Usuario existe y contraseña coincide? :diamond-no→Mostrar error genérico→Mostrar formulario de login







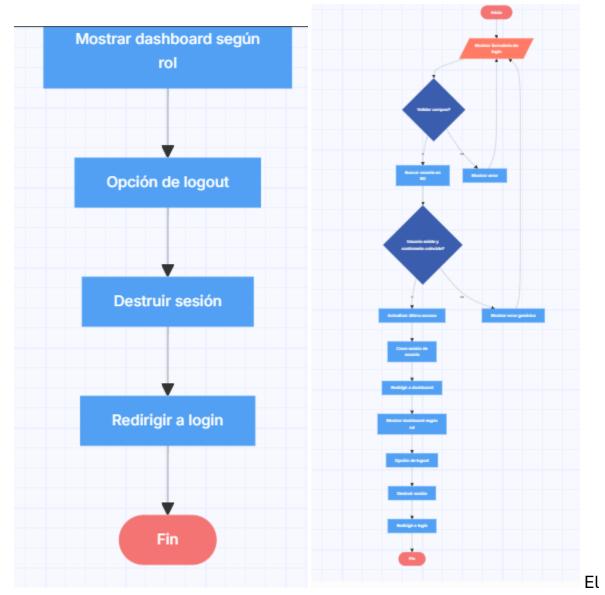
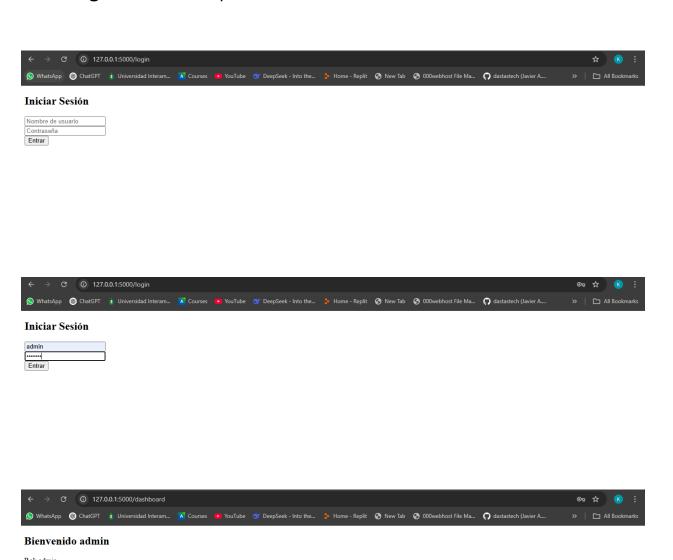


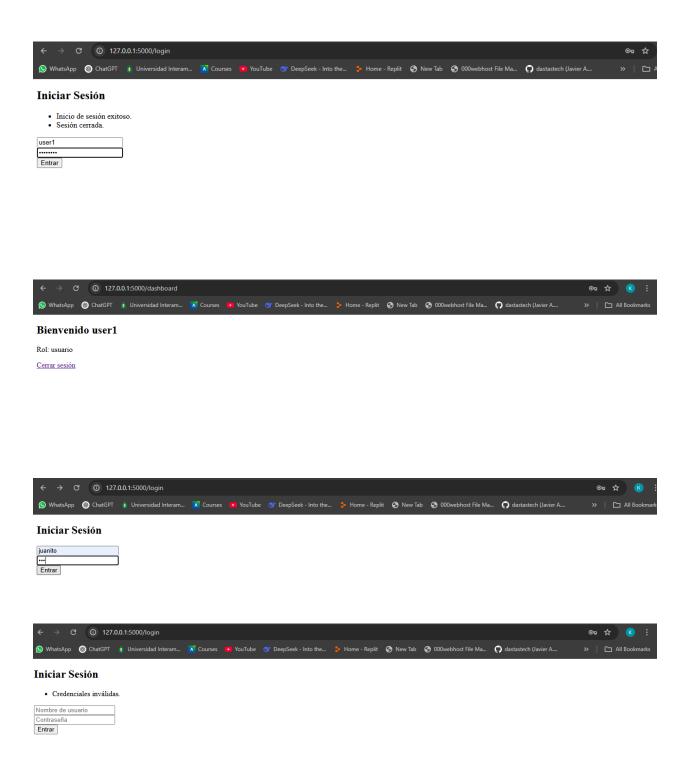
diagrama muestra el proceso completo de autenticación:

- 1. Inicio: El usuario accede a la aplicación
- 2. **Mostrar formulario de login**: Se presenta el formulario de inicio de sesión
- 3. **Validar campos**: Se verifica que se hayan completado los campos requeridos
- 4. **Buscar usuario en BD**: Se busca el usuario en la "base de datos" simulada

- 5. **Verificar contraseña**: Se compara la contraseña proporcionada con el hash almacenado
- 6. Crear sesión: Si las credenciales son válidas, se crea una sesión
- 7. Redirigir a dashboard: El usuario accede al área privada
- 8. Logout: El usuario puede cerrar sesión cuando lo desee



Cerrar sesión



Pruebas de la Aplicación

1. Pantalla de Login:

- Muestra el formulario de inicio de sesión
- o Valida campos obligatorios
- o Muestra mensajes de error si las credenciales son incorrectas

2. Dashboard:

- Solo accesible para usuarios autenticados
- o Muestra información del usuario (nombre y rol)
- o Incluye opción para cerrar sesión

3. Protección de Rutas:

- Intento de acceso directo a /dashboard sin autenticarse redirige al login
- o Sesión persistente durante la navegación
- o Cierre de sesión correcto que impide acceso no autorizado

https://github.com/Seliz05/COMP-2052/tree/main/mod03Lecc01