

UNIVERSIDAD DE GUADALAJARA
Universidad de Guadalajara Centro Universitario
de los Valles



Título: Heriberto Buenrostro Hernández

Materia: Programación para Dispositivos Móviles

Por: Heriberto Buenrostro Hernández

Docente: Jorge Ernesto Castillo Rizo

Fecha de entrega: 24-10-2025

U2 - A3 Autenticación y Sesiones en el mini-sistema de Videojuegos

Durante este proyecto, he profundizado en el desarrollo de aplicaciones web con Flask, un framework ligero y flexible de Python. Uno de los aspectos más importantes que aprendí fue el manejo de usuarios y sesiones, utilizando herramientas como Flask-Login, Werkzeug y WTForms.

Con Flask-Login, entendí cómo se gestionan las sesiones de usuario, permitiendo que una persona inicie sesión, permanezca autenticada durante su interacción con la aplicación y pueda cerrar sesión de forma segura. Aprendí que para esto es fundamental definir un modelo de usuario en la base de datos y vincularlo con Flask-Login mediante la función `user_loader`.

Por otro lado, Werkzeug me permitió asegurar las contraseñas de los usuarios mediante hashing, evitando almacenar contraseñas en texto plano y aumentando la seguridad de la aplicación. Además, WTForms me enseñó a crear formularios de registro e inicio de sesión de forma sencilla y segura, con validaciones integradas que evitan errores y datos maliciosos.

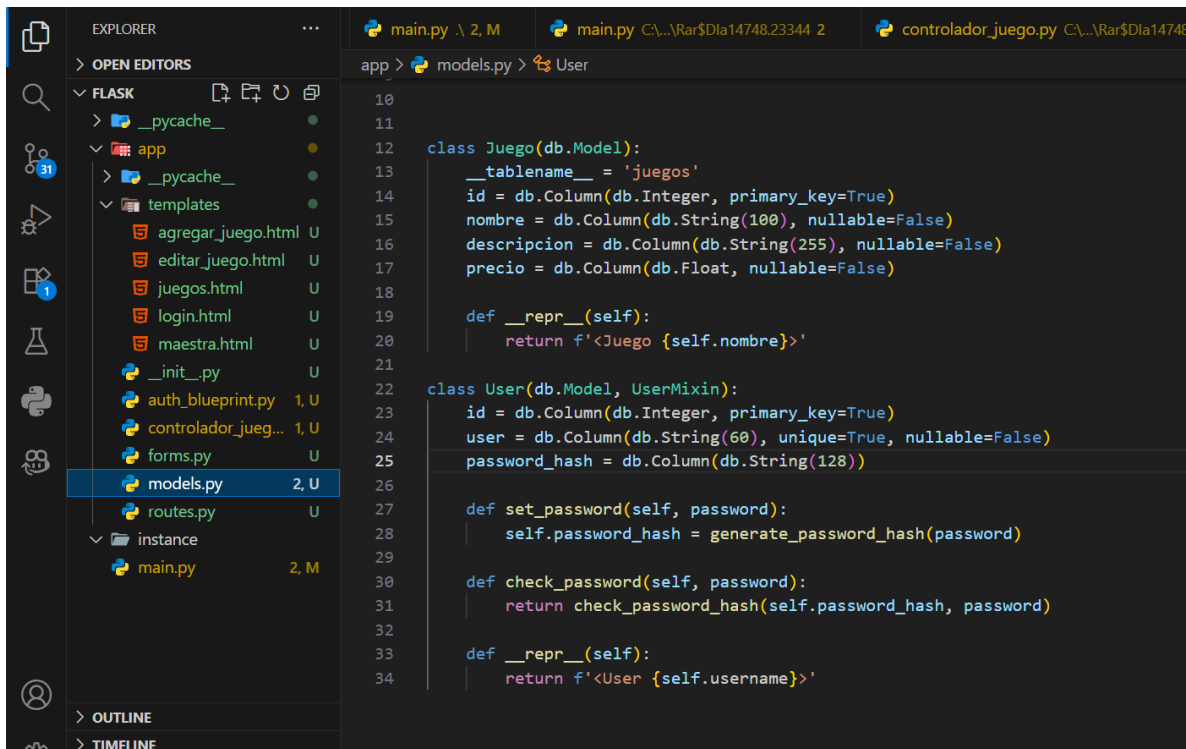
Un aprendizaje clave fue la organización del proyecto mediante blueprints, lo que facilita separar la lógica de la aplicación en módulos claros, como `auth` para autenticación y `main` para rutas principales. También comprendí la importancia de evitar importaciones circulares, asegurando que cada módulo importe solo lo necesario y manteniendo una estructura de carpetas coherente.

Finalmente, este proyecto me permitió entender la importancia de la estructura de carpetas en Flask y cómo trabajar con bases de datos, ya sea SQLite o MySQL, asegurando que la aplicación sea escalable y mantenible. En resumen, he adquirido conocimientos prácticos sobre cómo construir un sistema de usuarios seguro, modular y funcional en Flask, combinando buenas prácticas de programación, seguridad y arquitectura de proyectos web.

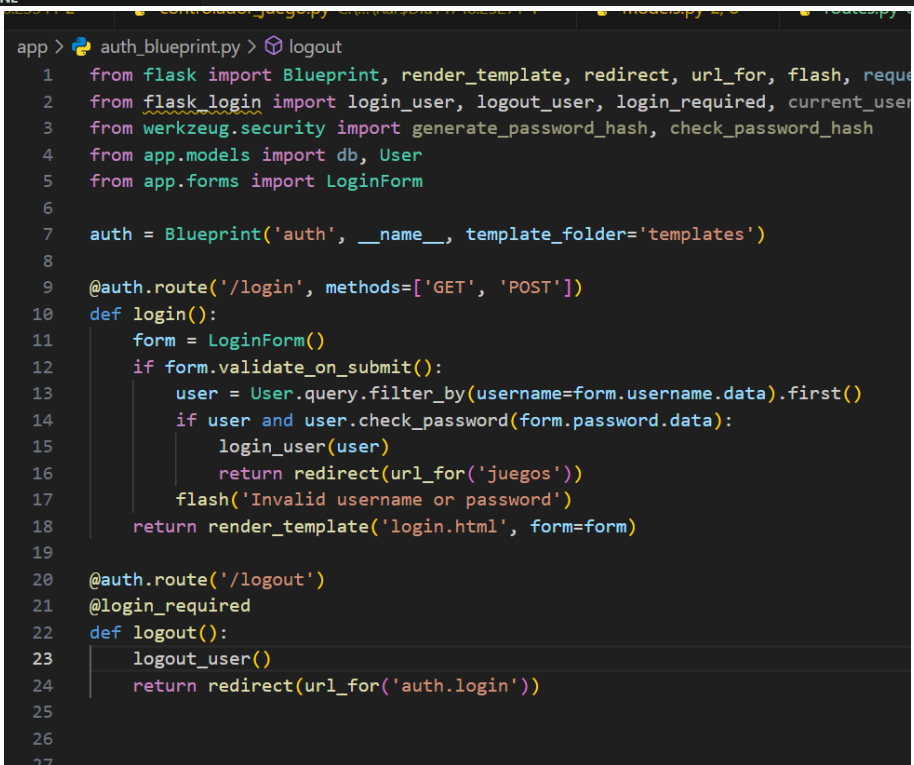
```
mysql> show tables;
+-----+
| Tables_in_juegos |
+-----+
| juegos            |
| user              |
+-----+
2 rows in set (0.011 sec)

mysql> INSERT INTO user (nombre, password) VALUES ('beto', '1234
');
Query OK, 1 row affected (0.080 sec)

mysql> |
```



```
10
11
12 class Juego(db.Model):
13     __tablename__ = 'juegos'
14     id = db.Column(db.Integer, primary_key=True)
15     nombre = db.Column(db.String(100), nullable=False)
16     descripcion = db.Column(db.String(255), nullable=False)
17     precio = db.Column(db.Float, nullable=False)
18
19     def __repr__(self):
20         return f'<Juego {self.nombre}>'
21
22 class User(db.Model, UserMixin):
23     id = db.Column(db.Integer, primary_key=True)
24     user = db.Column(db.String(60), unique=True, nullable=False)
25     password_hash = db.Column(db.String(128))
26
27     def set_password(self, password):
28         self.password_hash = generate_password_hash(password)
29
30     def check_password(self, password):
31         return check_password_hash(self.password_hash, password)
32
33     def __repr__(self):
34         return f'<User {self.username}>'
```



```
app > auth_blueprint.py > logout
1 from flask import Blueprint, render_template, redirect, url_for, flash, request
2 from flask_login import login_user, logout_user, login_required, current_user
3 from werkzeug.security import generate_password_hash, check_password_hash
4 from app.models import db, User
5 from app.forms import LoginForm
6
7 auth = Blueprint('auth', __name__, template_folder='templates')
8
9 @auth.route('/login', methods=['GET', 'POST'])
10 def login():
11     form = LoginForm()
12     if form.validate_on_submit():
13         user = User.query.filter_by(username=form.username.data).first()
14         if user and user.check_password(form.password.data):
15             login_user(user)
16             return redirect(url_for('juegos'))
17         flash('Invalid username or password')
18     return render_template('login.html', form=form)
19
20 @auth.route('/logout')
21 @login_required
22 def logout():
23     logout_user()
24     return redirect(url_for('auth.login'))
25
26
27
```