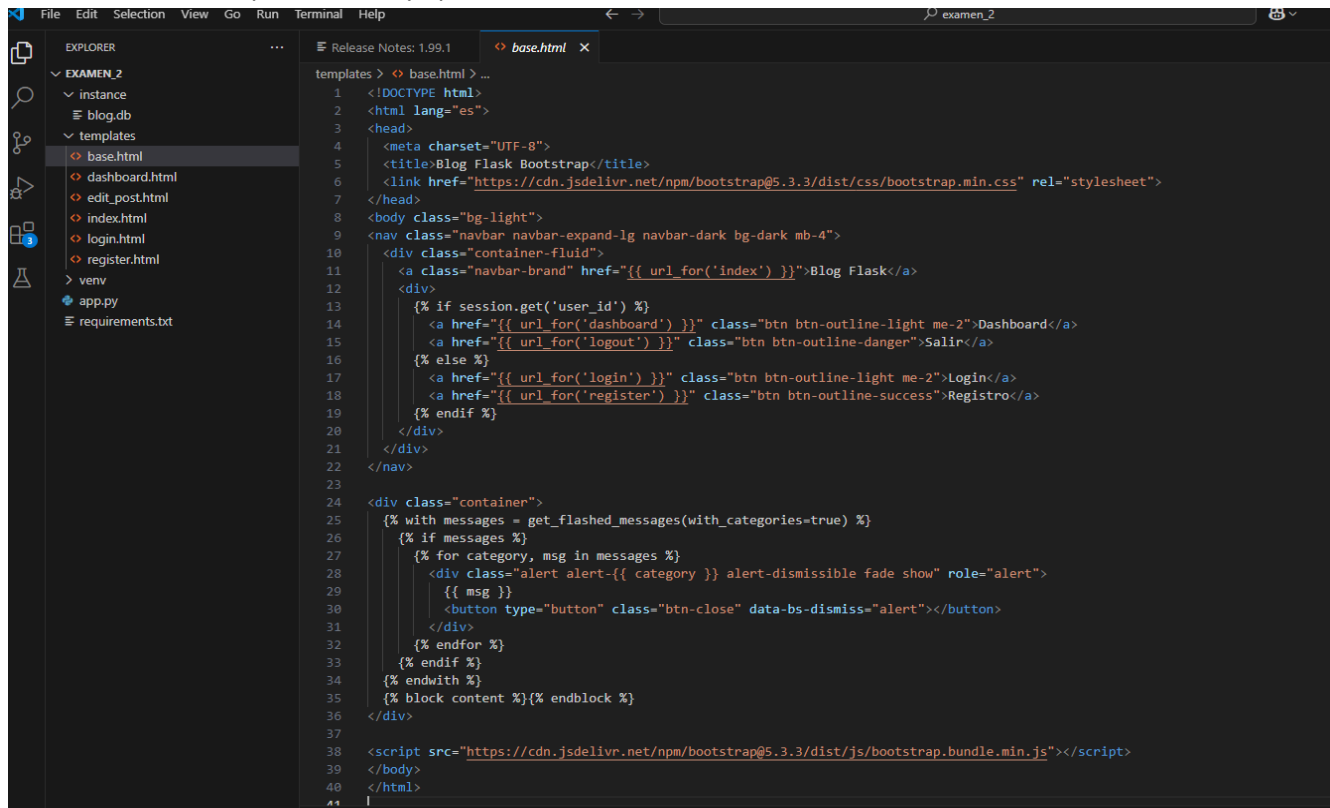


Parcial 2

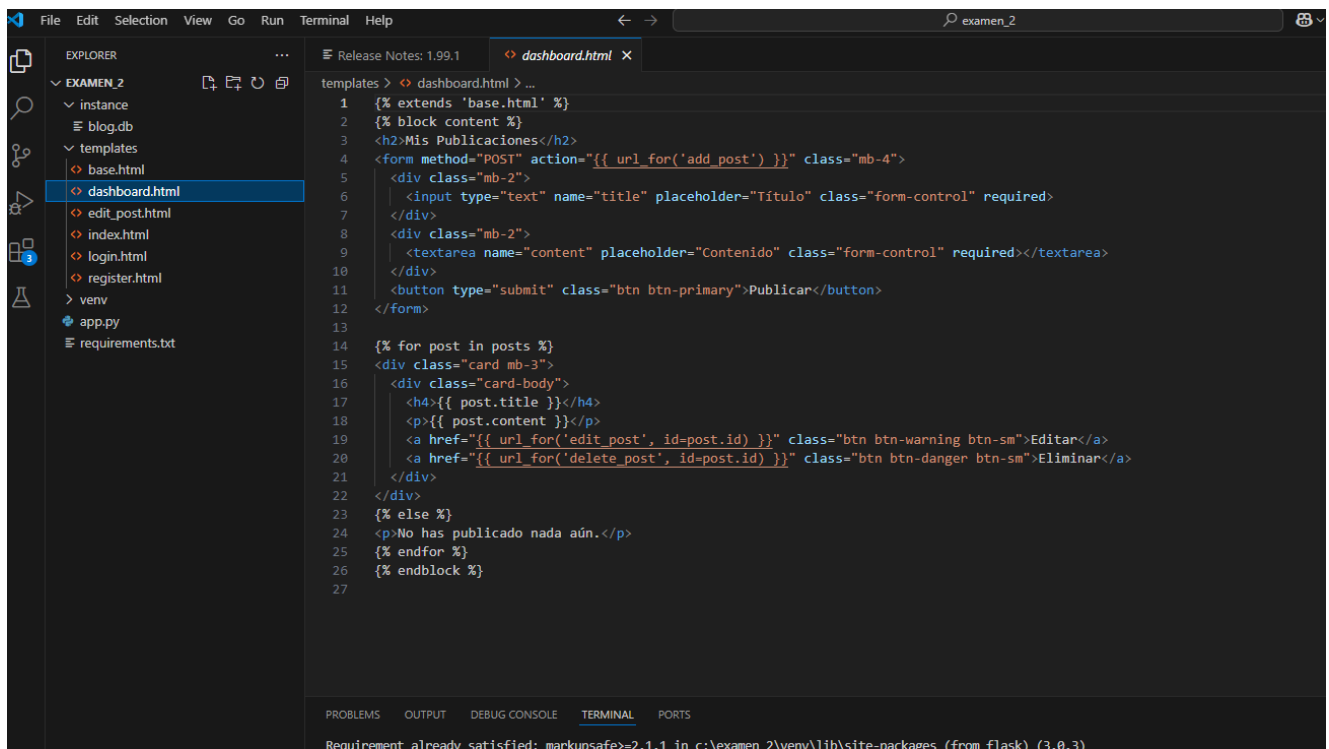
Nombre: linette primavera ilaya paco ci.9101166LP.

28/10/20225



The screenshot shows the VS Code editor with the 'base.html' file open. The Explorer panel on the left shows the project structure: EXAMEN_2, instance, blog.db, templates, and venv. The 'base.html' file is selected in the templates folder. The code in the editor is a Jinja2 template for a Flask application, featuring a Bootstrap navbar and a container for messages and content.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <title>Blog Flask Bootstrap</title>
6 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
7 </head>
8 <body class="bg-light">
9 <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
10 <div class="container-fluid">
11 <a class="navbar-brand" href="{{ url_for('index') }}">Blog Flask</a>
12 <div>
13 {% if session.get('user_id') %}
14 <a href="{{ url_for('dashboard') }}" class="btn btn-outline-light me-2">Dashboard</a>
15 <a href="{{ url_for('logout') }}" class="btn btn-outline-danger">Salir</a>
16 {% else %}
17 <a href="{{ url_for('login') }}" class="btn btn-outline-light me-2">Login</a>
18 <a href="{{ url_for('register') }}" class="btn btn-outline-success">Registro</a>
19 {% endif %}
20 </div>
21 </div>
22 </nav>
23
24 <div class="container">
25 {% with messages = get_flashed_messages(with_categories=true) %}
26 {% if messages %}
27 {% for category, msg in messages %}
28 <div class="alert alert-{{ category }}" alert-dismissible fade show" role="alert">
29 {{ msg }}
30 <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
31 </div>
32 {% endfor %}
33 {% endif %}
34 {% endif %}
35 {% block content %}{% endblock %}
36 </div>
37
38 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
39 </body>
40 </html>
41
```



The screenshot shows the VS Code editor with the 'dashboard.html' file open. The Explorer panel on the left shows the project structure: EXAMEN_2, instance, blog.db, templates, and venv. The 'dashboard.html' file is selected in the templates folder. The code in the editor is a Jinja2 template for a Flask application, featuring a form for adding posts and a list of posts.

```
1 {% extends 'base.html' %}
2 {% block content %}
3 <h2>Mis Publicaciones</h2>
4 <form method="POST" action="{{ url_for('add_post') }}" class="mb-4">
5 <div class="mb-2">
6 <input type="text" name="title" placeholder="Titulo" class="form-control" required>
7 </div>
8 <div class="mb-2">
9 <textarea name="content" placeholder="Contenido" class="form-control" required></textarea>
10 </div>
11 <button type="submit" class="btn btn-primary">Publicar</button>
12 </form>
13
14 {% for post in posts %}
15 <div class="card mb-3">
16 <div class="card-body">
17 <h4>{{ post.title }}</h4>
18 <p>{{ post.content }}</p>
19 <a href="{{ url_for('edit_post', id=post.id) }}" class="btn btn-warning btn-sm">Editar</a>
20 <a href="{{ url_for('delete_post', id=post.id) }}" class="btn btn-danger btn-sm">Eliminar</a>
21 </div>
22 </div>
23 {% else %}
24 <p>No has publicado nada aún.</p>
25 {% endfor %}
26 {% endblock %}
27
```

Requirement already satisfied: markupsafe>=2.1.1 in c:\examen 2\venv\lib\site-packages (from flask) (3.0.3)

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The 'EXAMEN_2' project is expanded, showing a 'templates' folder. The 'edit_post.html' file is selected and open in the editor. The code is a Django template that extends 'base.html'. It contains a form for editing a post, with fields for 'title' and 'content', and a 'Guardar cambios' button. The form is styled with Bootstrap classes.

```
1 {% extends 'base.html' %}
2 {% block content %}
3 <div class="card p-4 mx-auto" style="max-width: 600px;">
4   <h3>Editar publicación</h3>
5   <form method="POST">
6     <div class="mb-3">
7       <label>Título</label>
8       <input type="text" name="title" value="{{ post.title }}" class="form-control" required>
9     </div>
10    <div class="mb-3">
11      <label>Contenido</label>
12      <textarea name="content" class="form-control" required>{{ post.content }}</textarea>
13    </div>
14    <button type="submit" class="btn btn-success">Guardar cambios</button>
15  </form>
16 </div>
17 {% endblock %}
18
```

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The 'EXAMEN_2' project is expanded, showing a 'templates' folder. The 'index.html' file is selected and open in the editor. The code is a Django template that extends 'base.html'. It displays a list of recent publications, with each item shown in a card. The cards include the title, content, and author information. If there are no publications, a message is displayed.

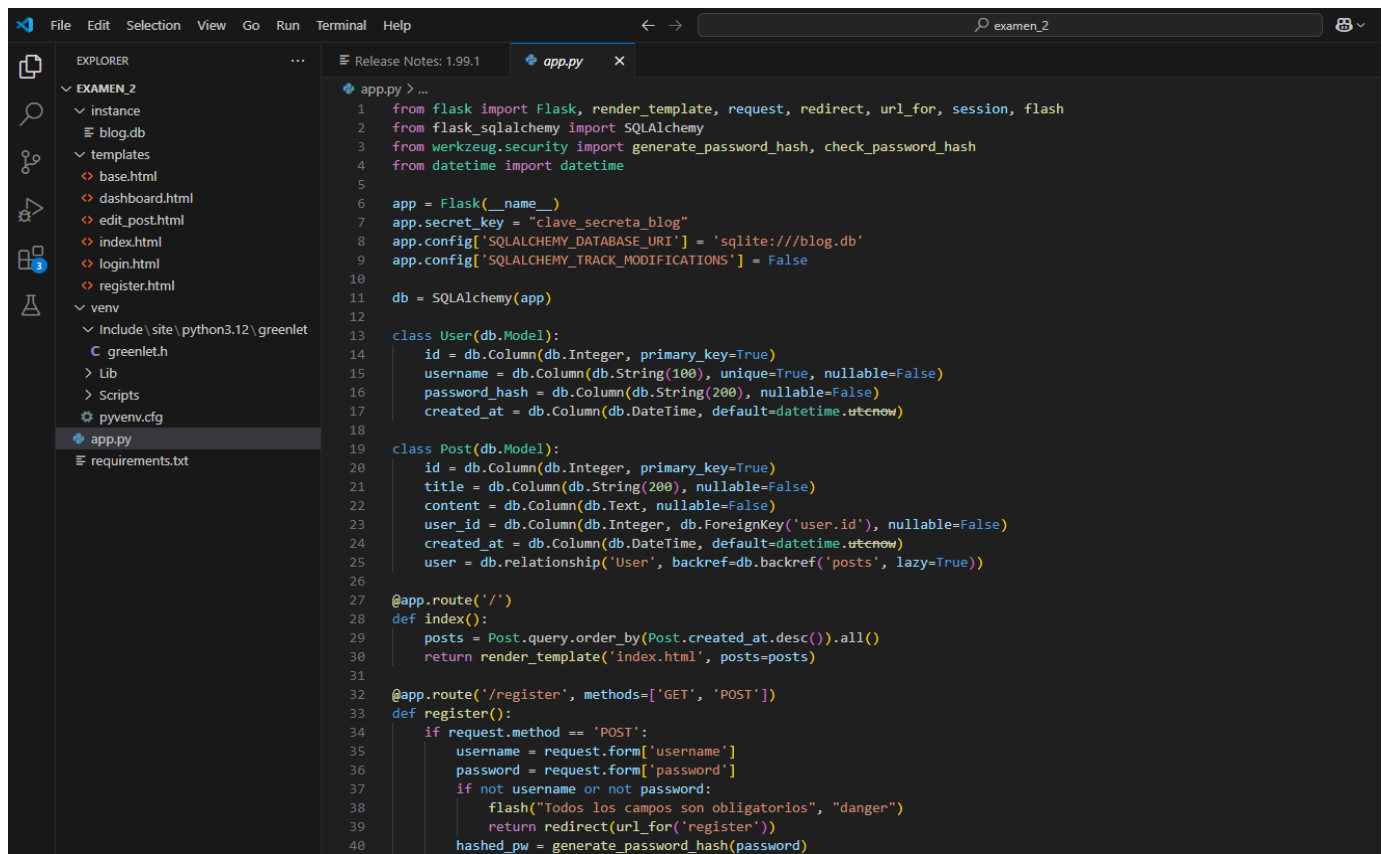
```
1 {% extends 'base.html' %}
2 {% block content %}
3 <h2 class="mb-4">Publicaciones Recientes</h2>
4 {% for post in posts %}
5 <div class="card mb-3 shadow-sm">
6   <div class="card-body">
7     <h4 class="card-title">{{ post.title }}</h4>
8     <p class="card-text">{{ post.content }}</p>
9     <p class="text-muted">Autor: {{ post.user.username }} | {{ post.created_at.strftime('%d/%m/%Y') }}</p>
10   </div>
11 </div>
12 {% else %}
13 <p>No hay publicaciones disponibles.</p>
14 {% endfor %}
15 {% endblock %}
16
```

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The project is named 'EXAMEN_2'. Under the 'templates' folder, 'login.html' is selected. The main editor displays the content of 'login.html', which is a Django template. It extends 'base.html' and contains a form for logging in. The form has two input fields: 'username' and 'password', both required. A 'Entrar' button is at the bottom of the form.

```
1 {% extends 'base.html' %}
2 {% block content %}
3 <div class="card p-4 mx-auto" style="max-width: 400px;">
4   <h3>Iniciar Sesión</h3>
5   <form method="POST">
6     <div class="mb-3">
7       <label>Usuario</label>
8       <input type="text" name="username" class="form-control" required>
9     </div>
10    <div class="mb-3">
11      <label>Contraseña</label>
12      <input type="password" name="password" class="form-control" required>
13    </div>
14    <button type="submit" class="btn btn-primary w-100">Entrar</button>
15  </form>
16 </div>
17 {% endblock %}
18
```

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The project is named 'EXAMEN_2'. Under the 'templates' folder, 'register.html' is selected. The main editor displays the content of 'register.html', which is a Django template. It extends 'base.html' and contains a form for registering a new user. The form has two input fields: 'username' and 'password', both required. A 'Registrar' button is at the bottom of the form.

```
1 {% extends 'base.html' %}
2 {% block content %}
3 <div class="card p-4 mx-auto" style="max-width: 400px;">
4   <h3>Registro de Usuario</h3>
5   <form method="POST">
6     <div class="mb-3">
7       <label>Usuario</label>
8       <input type="text" name="username" class="form-control" required>
9     </div>
10    <div class="mb-3">
11      <label>Contraseña</label>
12      <input type="password" name="password" class="form-control" required>
13    </div>
14    <button type="submit" class="btn btn-success w-100">Registrar</button>
15  </form>
16 </div>
17 {% endblock %}
18
```



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left displays a project structure for 'EXAMEN_2', including files like 'blog.db', 'base.html', 'dashboard.html', 'edit_post.html', 'index.html', 'login.html', 'register.html', and a 'venv' directory. The main editor area shows the 'app.py' file, which contains the following Python code:

```
1 from flask import Flask, render_template, request, redirect, url_for, session, flash
2 from flask_sqlalchemy import SQLAlchemy
3 from werkzeug.security import generate_password_hash, check_password_hash
4 from datetime import datetime
5
6 app = Flask(__name__)
7 app.secret_key = "clave_secreta_blog"
8 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///blog.db'
9 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
10
11 db = SQLAlchemy(app)
12
13 class User(db.Model):
14     id = db.Column(db.Integer, primary_key=True)
15     username = db.Column(db.String(100), unique=True, nullable=False)
16     password_hash = db.Column(db.String(200), nullable=False)
17     created_at = db.Column(db.DateTime, default=datetime.utcnow)
18
19 class Post(db.Model):
20     id = db.Column(db.Integer, primary_key=True)
21     title = db.Column(db.String(200), nullable=False)
22     content = db.Column(db.Text, nullable=False)
23     user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
24     created_at = db.Column(db.DateTime, default=datetime.utcnow)
25     user = db.relationship('User', backref=db.backref('posts', lazy=True))
26
27 @app.route('/')
28 def index():
29     posts = Post.query.order_by(Post.created_at.desc()).all()
30     return render_template('index.html', posts=posts)
31
32 @app.route('/register', methods=['GET', 'POST'])
33 def register():
34     if request.method == 'POST':
35         username = request.form['username']
36         password = request.form['password']
37         if not username or not password:
38             flash("Todos los campos son obligatorios", "danger")
39             return redirect(url_for('register'))
40         hashed_pw = generate_password_hash(password)
```

```
File Edit Selection View Go Run Terminal Help
EXAMEN_2
instance
blog.db
templates
base.html
dashboard.html
edit_post.html
index.html
login.html
register.html
venv
Include\site\python3.12\greenlet
greenlet.h
Lib
Scripts
pyvenv.cfg
app.py
requirements.txt

app.py > ...
33 def register():
34     return redirect(url_for('register'))
35     hashed_pw = generate_password_hash(password)
36     new_user = User(username=username, password_hash=hashed_pw)
37     try:
38         db.session.add(new_user)
39         db.session.commit()
40         flash("Usuario registrado correctamente", "success")
41         return redirect(url_for('login'))
42     except:
43         flash("El nombre de usuario ya existe", "danger")
44         return redirect(url_for('register'))
45     return render_template('register.html')
46
47 @app.route('/login', methods=['GET', 'POST'])
48 def login():
49     if request.method == 'POST':
50         username = request.form['username']
51         password = request.form['password']
52         user = User.query.filter_by(username=username).first()
53         if user and check_password_hash(user.password_hash, password):
54             session['user_id'] = user.id
55             session['username'] = user.username
56             flash("Inicio de sesión exitoso", "success")
57             return redirect(url_for('dashboard'))
58         else:
59             flash("Usuario o contraseña incorrectos", "danger")
60             return render_template('login.html')
61
62 @app.route('/logout')
63 def logout():
64     session.clear()
65     flash("Sesión cerrada correctamente", "info")
66     return redirect(url_for('index'))
67
68 @app.route('/dashboard')
69 def dashboard():
70     if 'user_id' not in session:
71         flash("Debes iniciar sesión", "danger")
72         return redirect(url_for('login'))
```

```
File Edit Selection View Go Run Terminal Help
EXAMEN_2
instance
blog.db
templates
base.html
dashboard.html
edit_post.html
index.html
login.html
register.html
venv
Include\site\python3.12\greenlet
greenlet.h
Lib
Scripts
pyvenv.cfg
app.py
requirements.txt

app.py > ...
74 def dashboard():
75     if 'user_id' not in session:
76         flash("Debes iniciar sesión", "danger")
77         return redirect(url_for('login'))
78     posts = Post.query.filter_by(user_id=session['user_id']).all()
79     return render_template('dashboard.html', posts=posts)
80
81 @app.route('/add_post', methods=['POST'])
82 def add_post():
83     if 'user_id' not in session:
84         return redirect(url_for('login'))
85     title = request.form['title']
86     content = request.form['content']
87     new_post = Post(title=title, content=content, user_id=session['user_id'])
88     db.session.add(new_post)
89     db.session.commit()
90     flash("Post creado exitosamente", "success")
91     return redirect(url_for('dashboard'))
92
93 @app.route('/edit_post/<int:id>', methods=['GET', 'POST'])
94 def edit_post(id):
95     post = Post.query.get_or_404(id)
96     if request.method == 'POST':
97         post.title = request.form['title']
98         post.content = request.form['content']
99         db.session.commit()
100         flash("Post actualizado", "success")
101         return redirect(url_for('dashboard'))
102     return render_template('edit_post.html', post=post)
103
104 @app.route('/delete_post/<int:id>')
105 def delete_post(id):
106     post = Post.query.get_or_404(id)
107     db.session.delete(post)
108     db.session.commit()
109     flash("Post eliminado", "info")
110     return redirect(url_for('dashboard'))
111
112 if __name__ == '__main__':
113     with app.app_context():
114         db.create_all()
115     app.run(debug=True)
116
```

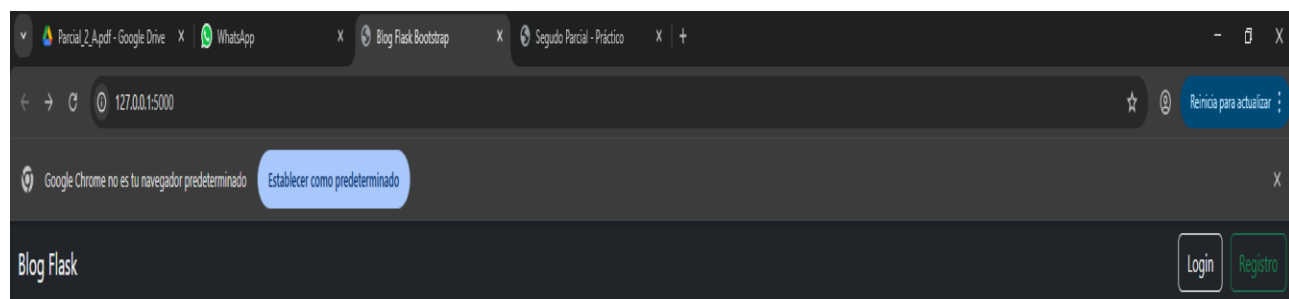
```
... Release Notes: 1.99.1 app.py x
app.py > ...
74 def dashboard():
75     if 'user_id' not in session:
76         flash("Debes iniciar sesión", "danger")
77         return redirect(url_for('login'))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS python +v ... x x

Collecting typing-extensions>=4.6.0 (from sqlalchemy>=2.0.16->flask_sqlalchemy->-r requirements.txt (line 2))
  Downloading typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
  Downloading flask-3.1.2-py3-none-any.whl (103 kB)
    103.3/103.3 kB 450.5 kB/s eta 0:00:00
Using cached flask_sqlalchemy-3.1.1-py3-none-any.whl (25 kB)
Using cached werkzeug-3.1.3-py3-none-any.whl (224 kB)
Using cached blinker-1.9.0-py3-none-any.whl (8.5 kB)
  Downloading click-8.3.0-py3-none-any.whl (107 kB)
    107.3/107.3 kB 344.2 kB/s eta 0:00:00
Using cached itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Using cached Jinja2-3.1.6-py2-none-any.whl (134 kB)
  Downloading markupsafe-3.0.3-cp312-cp312-win_amd64.whl (15 kB)
    2.1/2.1 kB 217.5 kB/s eta 0:00:00
  Downloading greenlet-3.2.4-cp312-cp312-win_amd64.whl (290 kB)
    290.9/290.9 kB 152.0 kB/s eta 0:00:00
  Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
    44.6/44.6 kB 182.0 kB/s eta 0:00:00
Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: typing-extensions, markupsafe, itsdangerous, greenlet, colorama, blinker, werkzeug, sqlalchemy, Jinja2, click, flask, flask_sqlalchemy
Successfully installed blinker-1.9.0 click-8.3.0 colorama-0.4.6 flask-3.1.2 flask_sqlalchemy-3.1.1 greenlet-3.2.4 itsdangerous-2.2.0 Jinja2-3.1.6 markupsafe-3.0.3 sqlalchemy-2.0.44 typing-extensions-4.15.0 werk

[notice] A new release of pip is available: 24.0 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: flask in c:\examen_2\venv\lib\site-packages (3.1.2)
Requirement already satisfied: blinker>=1.9.0 in c:\examen_2\venv\lib\site-packages (from flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in c:\examen_2\venv\lib\site-packages (from flask) (8.3.0)
Requirement already satisfied: itsdangerous>=2.2.0 in c:\examen_2\venv\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: Jinja2>=3.1.2 in c:\examen_2\venv\lib\site-packages (from flask) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in c:\examen_2\venv\lib\site-packages (from flask) (3.0.3)
Requirement already satisfied: werkzeug>=3.1.0 in c:\examen_2\venv\lib\site-packages (from flask) (3.1.3)
Requirement already satisfied: colorama in c:\examen_2\venv\lib\site-packages (from click>=8.1.3->flask) (0.4.6)

[notice] A new release of pip is available: 24.0 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS C:\examen_2> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Restarting with stat
* Restarting with stat
* Debugger is active!
* Debugger is active!
* Debugger PIN: 378-246-943
* Debugger PIN: 378-246-943
127.0.0.1 - - [28/Oct/2025 19:25:20] "GET / HTTP/1.1" 200 -
```



Publicaciones Recientes

No hay publicaciones disponibles.

