

## PARTICIPACION

Nombre: Linette Primavera ilaya paco

CI: 9101166 Lp.

RU: 14000699

The screenshot shows a Visual Studio Code editor with a project named 'participacionrodrigo'. The Explorer sidebar on the left shows a file structure with 'templates' containing 'base.html', 'inicio.html', 'login.html', and 'registro.html', and a 'venv' directory with 'app.py' and 'requirements.txt'. The main editor window displays the 'base.html' file, which is a Jinja2 template for an HTML page. It includes a DOCTYPE declaration, a meta charset, a title, and a style block for error and success messages. The body contains a Jinja2 loop for displaying flashed messages. The bottom panel shows the 'TERMINAL' output, which indicates that the Flask app is running on http://127.0.0.1:5000. It also shows several HTTP requests from a browser, including GET requests for the root, login, and favicon.ico, and a POST request for the login endpoint.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>{% block title %}{% endblock %}</title>
6   <style>
7     .error { color: red; }
8     .success { color: green; }
9   </style>
10 </head>
11 <body>
12   {% with messages = get_flashed_messages(with_categories=true) %}
13   {% if messages %}
14     {% for category, message in messages %}
15       <p class="{{ category }}">{{ message }}</p>
16     {% endfor %}
17   {% endif %}
18   {% endwith %}
19   {% block content %}{% endblock %}
20 </body>
```

```
* Serving Flask app 'app'
deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 238-178-650
127.0.0.1 - - [24/Sep/2025 22:35:19] "GET / HTTP/1.1" 302 -
127.0.0.1 - - [24/Sep/2025 22:35:19] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [24/Sep/2025 22:35:19] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [24/Sep/2025 22:35:28] "POST /login HTTP/1.1" 200 -
```

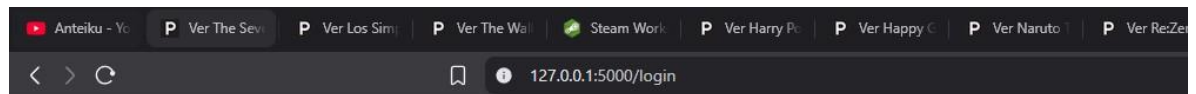
The screenshot shows the 'app.py' file in the Visual Studio Code editor. The code defines a Flask application with a secret key, a dictionary for users, and a route for the index page. The index page logic checks if a user is logged in and redirects them to the 'inicio' or 'login' page accordingly.

```
1 from flask import Flask, render_template, request, redirect, url_for, session, flash
2
3 app = Flask(__name__)
4 app.secret_key = "clave_secreta"
5
6 usuarios = {}
7
8 @app.route("/")
9 def index():
10     if "usuario" in session:
11         return redirect(url_for("inicio"))
12     return redirect(url_for("login"))
13
```

```
app.py x <> base.html <> login.html <> registro.html <> inicio.html ≡ Untitled-1
app.py > ...
14
15 @app.route("/registro", methods=["GET", "POST"])
16 def registro():
17     if request.method == "POST":
18         username = request.form["username"]
19         password = request.form["password"]
20         color = request.form["color"]
21
22         if username in usuarios:
23             flash("El usuario ya existe", "error")
24         else:
25             usuarios[username] = {"password": password, "color": color}
26             flash("Usuario registrado correctamente", "success")
27             return redirect(url_for("login"))
28     return render_template("registro.html")
29
30 @app.route("/login", methods=["GET", "POST"])
31 def login():
32     if request.method == "POST":
33         username = request.form["username"]
34         password = request.form["password"]
35
36
```

```
participacionrodrigo
app.py x <> base.html <> login.html <> registro.html <> inicio.html ≡ Untitled-1
app.py > ...
29
30 @app.route("/login", methods=["GET", "POST"])
31 def login():
32     if request.method == "POST":
33         username = request.form["username"]
34         password = request.form["password"]
35
36
37         if username == "admin" and password == "12345678":
38             session["usuario"] = "admin"
39             session["color"] = "#87CEEB"
40             return redirect(url_for("inicio"))
41
42
43         if username in usuarios and usuarios[username]["password"] == password:
44             session["usuario"] = username
45             session["color"] = usuarios[username]["color"]
46             return redirect(url_for("inicio"))
47
48         flash("Credenciales inválidas", "error")
49     return render_template("login.html")
50
51
```

```
50
51
52 @app.route("/inicio")
53 def inicio():
54     if "usuario" not in session:
55         return redirect(url_for("login"))
56     return render_template("inicio.html", usuario=session["usuario"], color=session["color"])
57
58
59 @app.route("/logout")
60 def logout():
61     session.clear()
62     return redirect(url_for("login"))
63
64 if __name__ == "__main__":
65     app.run(debug=True)
66
```



## Iniciar sesión

¿No tienes cuenta? [Regístrate aquí](#)

## Registro de usuario

Color favorito:

¿Ya tienes cuenta? [Inicia sesión](#)

Usuario registrado correctamente

## Iniciar sesión

¿No tienes cuenta? [Regístrate aquí](#)



127.0.0.1:5000/inicio

# Bienvenido, admin

[Cerrar sesión](#)