

# Actividad 3

Pablo Sanchez Galdamez (21001135)

## Contenidos

<b>Vulnerabilidad</b>	<b>1</b>
<b>Solución</b>	<b>2</b>
<b>Justificación</b>	<b>3</b>

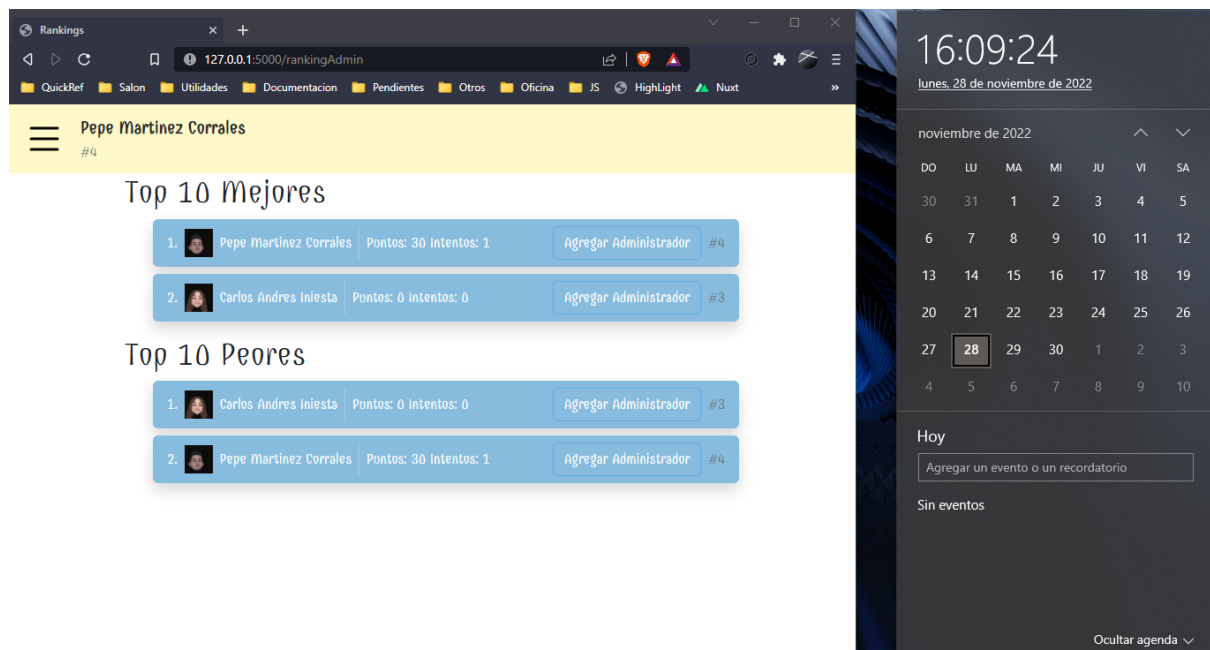
Tomaré la aplicación que desarrollamos en el curso “Práctica del Desarrollo de software II”.  
Puede ver la aplicación original en este repositorio:

<https://github.com/Polo123456789/Practica-del-desarrollo-de-software-II>

La buena práctica que se aplicará es “Control de acceso basado en roles”, o RBAC por sus siglas en inglés.

## Vulnerabilidad

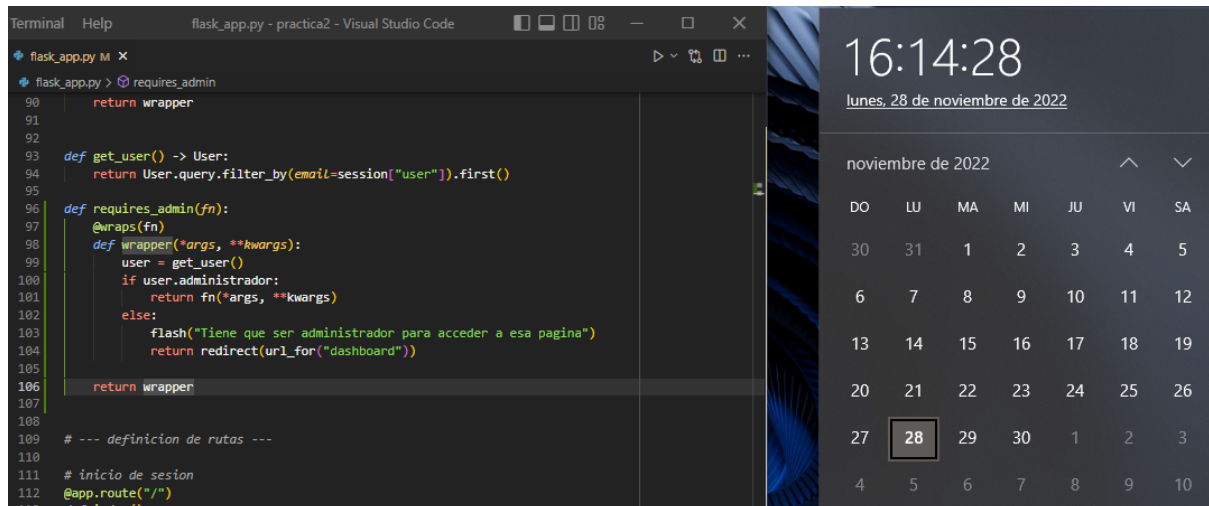
En su estado inicial, cualquier usuario puede cambiar la url inicial de **/dashboard** a **/rankingAdmin** y obtener acceso al panel de administrador.



# Solución

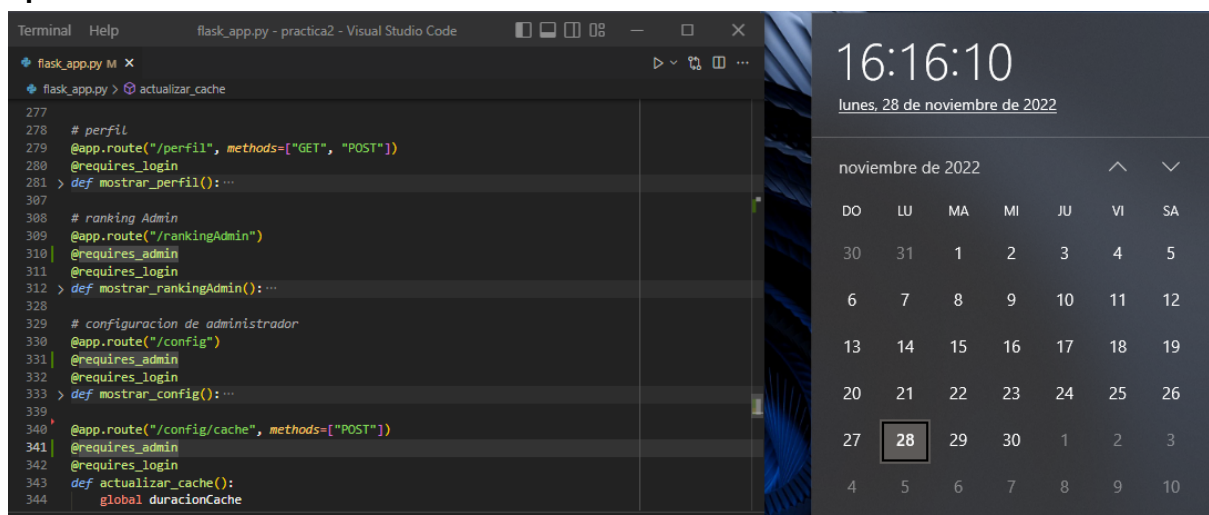
Para solucionarlo utilizaremos un decorador llamado **requires\_admin** y lo aplicaremos a las rutas que lo requieran.

## Decorador:



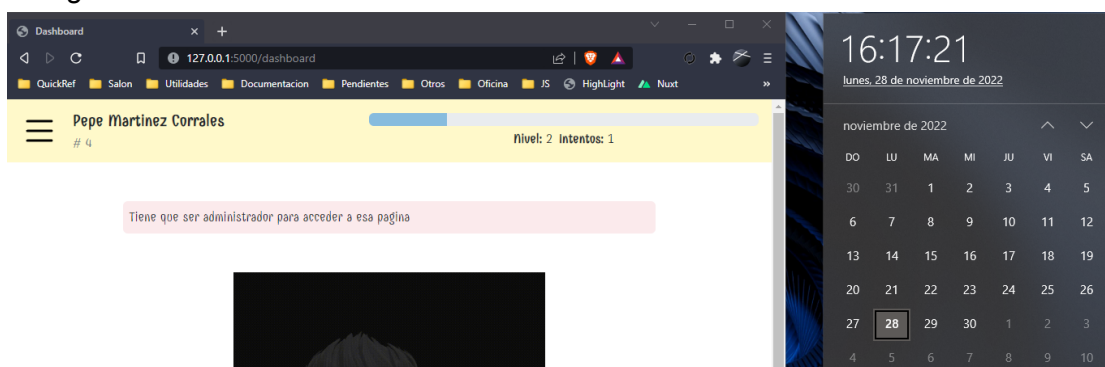
```
flask_app.py M X
flask_app.py > requires_admin
90     return wrapper
91
92
93 def get_user() -> User:
94     return User.query.filter_by(email=session["user"]).first()
95
96 def requires_admin(fn):
97     @wraps(fn)
98     def wrapper(*args, **kwargs):
99         user = get_user()
100         if user.administrador:
101             return fn(*args, **kwargs)
102         else:
103             flash("Tiene que ser administrador para acceder a esa pagina")
104             return redirect(url_for("dashboard"))
105     return wrapper
106
107
108 # --- definicion de rutas ---
109
110 # inicio de sesion
111 @app.route("/")
112 ...
```

## Aplicando el decorador a las rutas del administrador:



```
flask_app.py M X
flask_app.py > actualizar_cache
277
278 # perfil
279 @app.route("/perfil", methods=["GET", "POST"])
280 @requires_login
281 > def mostrar_perfil(): ...
307
308 # ranking Admin
309 @app.route("/rankingAdmin")
310 @requires_admin
311 @requires_login
312 > def mostrar_rankingAdmin(): ...
328
329 # configuracion de administrador
330 @app.route("/config")
331 @requires_admin
332 @requires_login
333 > def mostrar_config(): ...
339
340 @app.route("/config/cache", methods=["POST"])
341 @requires_admin
342 @requires_login
343 def actualizar_cache():
344     global duracionCache
```

Ahora si un usuario que no es administrador intenta acceder a alguna de esas rutas se topa con el siguiente error:



## Justificación

Elegí solucionar este problema ya que cualquier usuario que tenga un poco de conocimiento sobre cómo funcionan los internos de la aplicación puede obtener un nivel de acceso bastante peligroso.

Elegí solucionarlo con un decorador ya que es la forma más sencilla de implementar la verificación en varias rutas a la vez.