



BACKEND CON PYTHON



AUTORIZACIÓN

- Importancia de la autorización
- Mecanismos
- Implementación en Flask



AUTORIZACIÓN

- Importancia de la autorización
- Mecanismos
- Implementación en Flask



IMPORTANCIA AUTORIZACIÓN

En la clase anterior vimos mecanismos y la importancia de autenticar al usuario. Ahora que sabemos cómo identificar quién es el usuario, podemos modificar nuestra aplicación para que ahora podamos diferenciar a los usuarios en diferentes categorías (por ejemplo, cliente, administrador, entre otros) y según sea su categoría otorgarle permisos diferentes: un administrador no realizará las mismas operaciones que un cliente.

Esta categorización de usuarios y asignación de diferentes permisos es lo que se conoce como autorización. Es un concepto fundamental cuando hablamos de seguridad.

IMPORTANCIA AUTORIZACIÓN

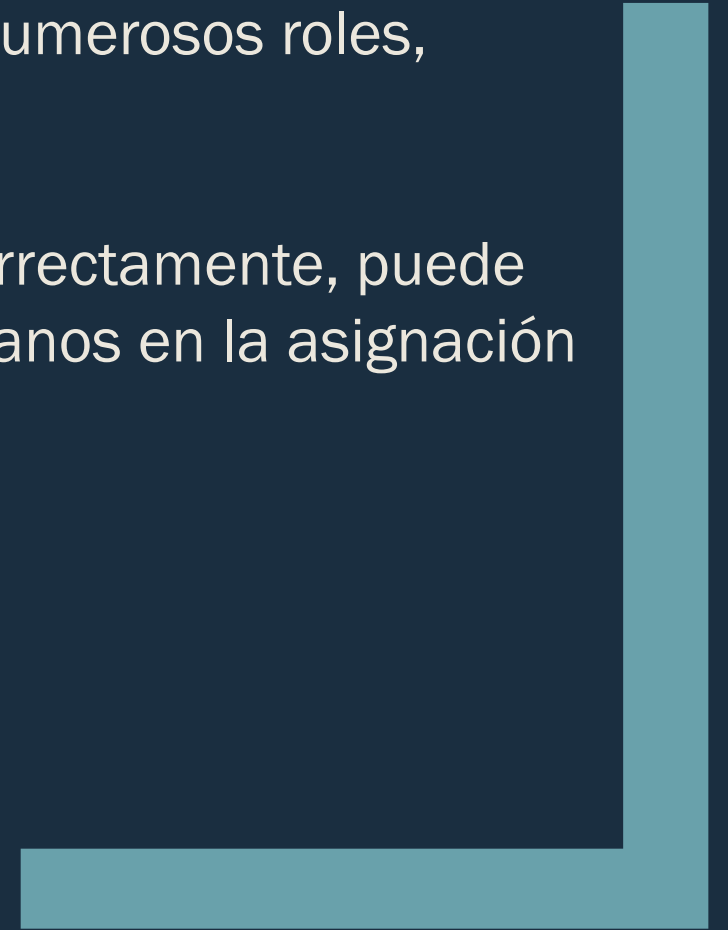
Ventajas

- Granularidad: permite asignar permisos específicos a usuarios o roles para realizar acciones específicas.
- Jerarquía de roles: permite establecer una jerarquía de roles para simplificar la gestión de permisos. Los usuarios pueden asignarse a roles y los roles pueden tener permisos asociados.
- Control de Acceso: permite asegurar que solo aquellos usuarios autorizados tengan acceso a los recursos y funciones permitidas.
- Auditoría: incluye funciones de auditoría para rastrear y registrar las actividades de los usuarios.
- Flexibilidad: permite adaptarse a las necesidades específicas de una organización.

IMPORTANCIA AUTORIZACIÓN

Desventajas

- Complejidad: Implementar y gestionar un sistema de autorización puede ser complejo, especialmente en organizaciones grandes con numerosos roles, usuarios y recursos.
- Posibilidad de errores: Si los permisos no se configuran correctamente, puede haber riesgos de seguridad. La posibilidad de errores humanos en la asignación de permisos es una desventaja.



AUTORIZACIÓN

- Importancia de la autorización
- Mecanismos
- Implementación en Flask



MECANISMOS

- Control de Acceso Basado en Listas (ACL): utiliza listas que especifican qué usuarios o sistemas tienen permisos para acceder a recursos específicos o realizar acciones particulares.
- Control de Acceso Basado en Roles (RBAC): asigna permisos a roles, y luego los usuarios se asignan a esos roles.
- Control de Acceso Basado en Políticas (ABAC): utiliza políticas que consideran atributos de usuarios, recursos y contexto para tomar decisiones de autorización.
- Token de Acceso y OAuth: utilizado para la autorización en aplicaciones web y servicios. OAuth es un protocolo estándar para la autorización.
- Control de Acceso Dinámico (DAC) y Control de Acceso Obligatorio (MAC): DAC permite a los usuarios controlar sus propios objetos y recursos, mientras que MAC asigna permisos de manera centralizada según políticas de seguridad.

AUTORIZACIÓN

- Importancia de la autorización
- Mecanismos
- Implementación en Flask



IMPLEMENTACIÓN EN FLASK

En la clase User debemos modificar la base de datos para asignar el estado de administrador o no a los usuarios:

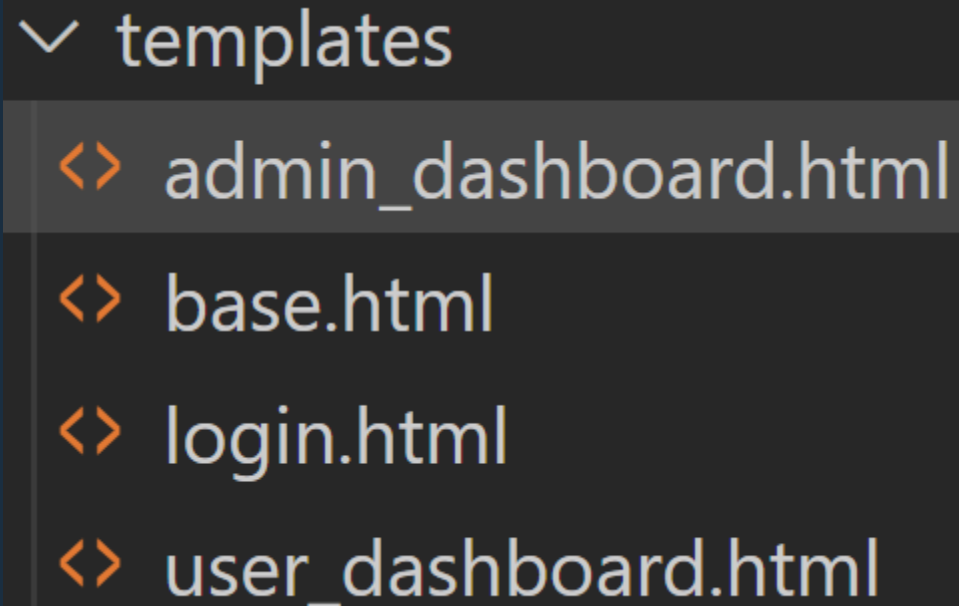
```
from flask_login import UserMixin

class User(UserMixin):
    def __init__(self, user_id, username, password, is_admin):
        self.id = user_id
        self.username = username
        self.password = password
        self.is_admin = is_admin

# Base de datos simulada para el ejemplo
users_db = {
    'zeus': User('zeus', 'Zeus', 'guau123', is_admin=True),
    'hera': User('hera', 'Hera', 'hera_password', is_admin=False),
    'poseidon': User('poseidon', 'Poseidon', 'ocean123', is_admin=False),
    'athena': User('athena', 'Athena', 'wisdom456', is_admin=False),
}
```

IMPLEMENTACIÓN EN FLASK

En el directorio templates, ahora no habrá un dashboard genérico, si no que habrá dos: uno para usuario común y otro para administrador:



```
▼ templates
  <> admin_dashboard.html
  <> base.html
  <> login.html
  <> user_dashboard.html
```

IMPLEMENTACIÓN EN FLASK

Por último, se debe modificar el en views.py la función login, para que ahora no redirija a dashboard, si no que redirija al user_dashboard o al admin_dashboard según sea el caso.

```
if user:
    login_user(user)
    flash('Inicio de sesión exitoso', 'success')
    if user.is_admin:
        return redirect(url_for('admin_dashboard'))
    else:
        return redirect(url_for('user_dashboard'))
else:
    flash('Credenciales incorrectas. Inténtalo de nuevo.', 'danger')
```

MANOS A LA OBRA

Taller 1 disponible en BloqueNeon

