

# Tienda en Línea

...

Integrante: Arocutipa Gutierrez, Luis Edgar

# Introducción

Se eligió una tienda en línea ya que tiene una estructura clara en la cual se puede ver la relación de muchos a muchos. Para esto se tomaron en cuenta dos modelos principales que son: Usuario y Producto.

Se tomaron estos modelos ya que cumple con la relación de muchos a muchos. Por un lado, un usuario puede comprar varios productos; así mismo, un producto puede ser comprado por varios usuarios.

Para la realización de la tienda, se usará Django como back-end y Angular como front-end.

# ¿Por qué Django y Angular?

Se utilizará Django en el back-end ya que es un framework web de alto nivel en Python, que ofrece una estructura organizada para el back-end. Además es seguro y escalable, permitiéndonos construir una tienda online sólida y confiable.

Mientras que en el front-end se utilizará Angular debido a que es un framework de desarrollo front-end potente y flexible. Así mismo, nos permite crear interfaces de usuario modernas y dinámicas con componentes reutilizables.

The Django logo, which consists of the word "django" in a lowercase, bold, sans-serif font. The letter 'j' has a distinctive hook that extends downwards.

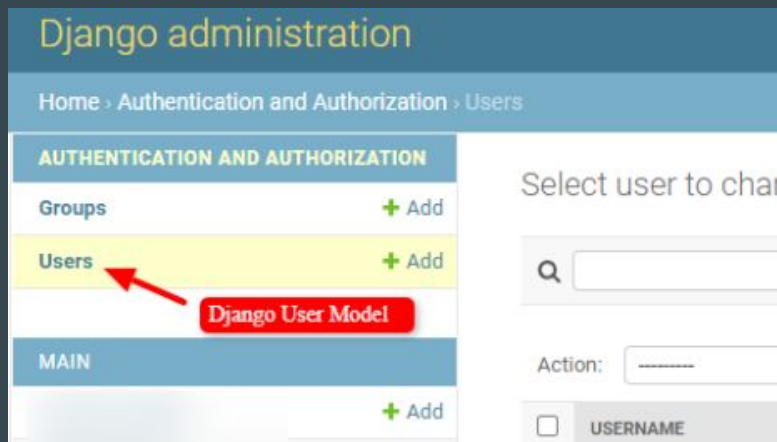
# Modelos en Django: Usuario

Django proporciona un modelo de usuario incorporado para la autenticación y la gestión de usuarios. Permite almacenar información relacionada con los usuarios, como nombre, correo electrónico y contraseña de forma segura. Además, nos permitirá guardar relación con el modelo Producto.

```
empresa > tienda > models.py
1  from django.db import models
2  from django.contrib.auth.models import User
3
4  class Producto(models.Model):
5      nombre = models.CharField(max_length=100)
6      descripcion = models.TextField()
7      precio = models.DecimalField(max_digits=10, decimal_places=2)
8      oferta = models.BooleanField(default=False)
9      imagen = models.ImageField(upload_to='imagenes/', blank=True, null=True)
10
11     vendido = models.ManyToManyField(User, blank=True)
12
13     def __str__(self):
14         return self.nombre
15
```

# Modelos en Django: Producto

Representa los productos que se venden en nuestra tienda online. Contiene detalles como nombre, descripción, precio, imagen y un campo booleano que nos indicará si el producto está en oferta. Adicionalmente, se añadió la relación ManyToMany con el modelo Usuario.



La relación entre estas dos tablas se establece mediante el campo vendido en el modelo Producto, que es una clave foránea. El campo vendido hace referencia al modelo User, lo que significa que cada registro en la tabla "productos" está vinculado a un registro en la tabla "usuarios".

Por lo tanto, mediante esta relación:

- Podemos acceder a los detalles del comprador de un producto específico.
- Podemos encontrar todos los productos comprados por un usuario en particular.
- Podemos realizar consultas que involucren datos de ambas tablas, como encontrar todos los productos comprados por un usuario con un nombre específico.

# Desarrollo del Front-end con Angular

- Configuramos el proyecto de Angular y las dependencias necesarias.
- Creamos componentes para representar la página de inicio, detalles del producto, formulario para agregar productos, formularios para el registro e inicio de sesión de un usuario, etc.
- Implementamos servicios para realizar solicitudes HTTP al backend y obtener datos dinámicos.
- Configuramos las URL de la API para que Angular pueda enviar y recibir datos del backend.
- Implementamos funciones en el frontend para procesar respuestas y mostrar datos en la interfaz de usuario.

# Seguridad, Autenticación y Diseño

La seguridad es esencial en una tienda online. Implementamos autenticación y autorización en Django.

Configuramos Angular para manejar tokens de autenticación y proteger rutas sensibles.

Además, se le añadió un diseño llamativo e interactivo a través de los archivos correspondientes a cada sección de la página.



# Conclusiones

En esta presentación, hemos aprendido cómo crear una tienda online utilizando Django y Angular.

Django nos brindó un backend sólido y seguro, mientras que Angular nos permitió construir una interfaz de usuario moderna e interactiva.

La combinación de estos dos frameworks es ideal para el desarrollo de aplicaciones web complejas y escalables.