**1) Crear entorno y proyecto**

**# 1. Crear carpeta de trabajo**

mkdir mercado && cd mercado

**# 2. (Opcional pero recomendado) Crear entorno virtual**

python -m venv .venv

# Windows:

.venv\Scripts\activate

# Mac/Linux:

source .venv/bin/activate

**# 3. Instalar dependencias base**

pip install "Django>=5.0,<6.0" "django-allauth>=0.63" django-environ

**# 4. Crear proyecto y app base**

django-admin startproject config .

python manage.py startapp core

python manage.py makemigration

python manage.py migrate

python manage.py createsuperuser comandos para usar por consola

# 2) Configurar settings (allauth + templates)

Editá config/settings.py:

from pathlib import Path

**import environ**

**env = environ.Env(DEBUG=(bool, True))**

**environ.Env.read\_env() # lee .env si existe**

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

**SECRET\_KEY = env("SECRET\_KEY", default="dev-secret-no-usar-en-prod")**

**DEBUG = env("DEBUG", default=True)**

**ALLOWED\_HOSTS = ["\*"]**

INSTALLED\_APPS = [

# Django

"django.contrib.admin",

"django.contrib.auth",

"django.contrib.contenttypes",

"django.contrib.sessions",

"django.contrib.messages",

"django.contrib.staticfiles",

**"django.contrib.sites", # <-- requerido por allauth**

**# Terceros**

**"allauth", # núcleo**

**"allauth.account", # cuentas locales (si querés)**

**"allauth.socialaccount", # social login**

**"allauth.socialaccount.providers.google",**

**"allauth.socialaccount.providers.github",**

**# Apps propias**

**"core",**

]

**SITE\_ID = 1**

**AUTHENTICATION\_BACKENDS = [**

**"django.contrib.auth.backends.ModelBackend",**

**"allauth.account.auth\_backends.AuthenticationBackend",**

**]**

**LOGIN\_REDIRECT\_URL = "home"**

**LOGOUT\_REDIRECT\_URL = "home"**

**# (opcional) Config de allauth**

**ACCOUNT\_LOGIN\_METHODS = {"email", "username"}**

**ACCOUNT\_SIGNUP\_FIELDS = ["email\*", "username\*", "password1\*", "password2\*"]**

MIDDLEWARE = [

"django.middleware.security.SecurityMiddleware",

"django.contrib.sessions.middleware.SessionMiddleware",

"django.middleware.common.CommonMiddleware",

"django.middleware.csrf.CsrfViewMiddleware",

"django.contrib.auth.middleware.AuthenticationMiddleware",

"django.contrib.messages.middleware.MessageMiddleware",

"django.middleware.clickjacking.XFrameOptionsMiddleware",

**"allauth.account.middleware.AccountMiddleware", 🡨necesario para auth**

]

ROOT\_URLCONF = "config.urls"

TEMPLATES = [

{

"BACKEND": "django.template.backends.django.DjangoTemplates",

**"DIRS": [BASE\_DIR / "templates"], # <-- carpeta de** templates

"APP\_DIRS": True,

"OPTIONS": {

"context\_processors": [

"django.template.context\_processors.debug",

**"django.template.context\_processors.request", # <-- requerido por allauth**

"django.contrib.auth.context\_processors.auth",

"django.contrib.messages.context\_processors.messages",

],

},

},

]

WSGI\_APPLICATION = "config.wsgi.application"

DATABASES = {

"default": {

"ENGINE": "django.db.backends.sqlite3",

"NAME": BASE\_DIR / "db.sqlite3",

}

}

STATIC\_URL = "static/"

# 3) URLs

config/urls.py:

from django.contrib import admin

from django.urls import path, include

from core.views import home

urlpatterns = [

path("admin/", admin.site.urls),

**path("accounts/", include("allauth.urls")), # rutas de allauth**

**path("", home, name="home"),**

]

# 4) Vista y templates base

core/views.py:

from django.shortcuts import render

def home(request):

return render(request, "home.html")

Estructura de carpetas:

templates/

base.html

home.html

**templates/base.html (navbar + botones de login):**

<!doctype html>

<html lang="es">

<head>

<meta charset="utf-8">

<title>{% block title %}Mercadito{% endblock %}</title>

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/@picocss/pico@2/css/pico.min.css">

</head>

<body>

<nav class="container-fluid">

<ul>

<li><strong>Mercadito</strong></li>

</ul>

<ul>

{% if user.is\_authenticated %}

<li>Hola, {{ user.first\_name|default:user.username }}</li>

<li><a href="{% url 'account\_logout' %}">Salir</a></li>

{% else %}

<li><a href="{% url 'account\_login' %}">Entrar</a></li>

{% endif %}

</ul>

</nav>

<main class="container">

{% block content %}{% endblock %}

</main>

</body>

</html>

templates/home.html:

{% extends "base.html" %}

{% load socialaccount %}

{% block title %}Inicio · Mercadito{% endblock %}

{% block content %}

<h1>Inicio</h1>

{% if not user.is\_authenticated %}

<p>Ingresá para publicar o comprar.</p>

<p>

<a href="{% provider\_login\_url 'google' %}" role="button">Entrar con Google</a>

<a href="{% provider\_login\_url 'github' %}" role="button" class="secondary">Entrar con GitHub</a>

</p>

{% else %}

<p>Ya estás logueado ✅. Próximo paso: crear productos y compras.</p>

{% endif %}

{% endblock %}

# 5) Migraciones + superusuario

python manage.py migrate

python manage.py createsuperuser

python manage.py runserver

Agregamos bootstrap

# 6) Alta de credenciales OAuth

## A) Google

https://console.cloud.google.com/apis/credentials?authuser=1&project=rock-idiom-470221-j2&supportedpurview=project

1. Ir a **Google Cloud Console** → APIs & Services → **Credentials** → **Create Credentials** → OAuth client ID.
2. Tipo: **Web application**.
3. **Authorized redirect URI** (local):
   * http://localhost:8000
   * http://localhost:8000/accounts/google/login/callback/
4. Copiá **Client ID** y **Client Secret**.

## B) GitHub

1. Ir a **Settings** (perfil) → **Developer settings** → **OAuth Apps** → New OAuth App.
2. **Homepage URL**: http://localhost:8000/
3. **Authorization callback URL**:
   * http://localhost:8000/accounts/github/login/callback/
4. Guardá **Client ID** y **Client Secret**.

## C) Registrar en Django (django-allauth)

Entrá a /admin:

* **Sites** → editá el example.com a localhost:8000.
* **Social applications** → Add social application:
  + Elegí **Google** o **GitHub**.
  + Pegá **Client id** y **Secret key**.
  + En **Sites**, agregá el sitio localhost:8000.
  + Guardá.

(Desde ese momento los botones “Entrar con Google/GitHub” funcionan.)

# 7) (Opcional) Ajustar permisos y scopes

Podés afinar qué datos pedís a cada proveedor, creando SOCIALACCOUNT\_PROVIDERS en settings.py:

SOCIALACCOUNT\_PROVIDERS = {

"google": {

"APP": {

**"client\_id": "xxxxxxxxxxxxxxxxxxx", # claves google**

**"secret": "yyyyyyyyyyyyyyyyyyyyyyyyy",**

},

"SCOPE": ["profile", "email"],

"AUTH\_PARAMS": {"access\_type": "online"},

},

"github": {

"APP": {

**"client\_id": "ccccccccccc", # claves github**

**"secret": "ccccccccccccccccccccccccccc",**

},

"SCOPE": ["user:email"],

},

}

Configurad desde localhost:8000/admin (con la clave de super usuario)

**En SITES**

Agregar la ip o las ip a utilizar (dominios)

### 📝 SOCIALMEDIA

1. **Provider**
   * Seleccionás el proveedor de autenticación (en tu caso **Google**).
2. **Provider ID**
   * En la mayoría de los casos **se deja vacío**.
   * Solo se usa si configurás múltiples aplicaciones distintas del mismo proveedor (ejemplo: una para localhost y otra para producción).
   * Para tu caso: **dejalo en blanco** o poné algo descriptivo como google si te lo pide.
3. **Name**
   * Es un nombre descriptivo que vos quieras (ejemplo: "Google Login").
4. **Client ID**
   * Lo copiaste bien desde la consola de Google (8573...apps.googleusercontent.com).
5. **Secret key**
   * También lo copiaste bien (el GOCSPX-...).
6. **Key**
   * Este campo casi nunca se usa con Google. Se puede dejar vacío.
7. **Sites**
   * Muy importante ⚠️: tenés que asociar esta aplicación a tu **Site** definido en Django.
   * Por defecto, cuando instalás Django, viene creado un **Site** con:
     + **ID 1**
     + **Domain name:** example.com
     + **Display name:** example.com
   * Lo que tenés que hacer es entrar en **Sites** (en el admin, en la sección **Sites**) y modificarlo:
     + Domain name: localhost:8000
     + Display name: Localhost
   * Luego, cuando estés creando el **Social Application**, seleccionás ese site (localhost:8000) en la parte de “Available sites” y lo movés a “Chosen sites”.

✅ Con eso, tu configuración quedaría así:

* **Provider**: Google
* **Provider ID**: (vacío)
* **Name**: Google Login
* **Client ID**: tu client id
* **Secret key**: tu secret key
* **Key**: (vacío)
* **Sites**: Localhost

¿Querés que te arme el **paso a paso de cómo probar el login con Google** ahora (entrando a http://localhost:8000/accounts/login/), o preferís que avancemos a configurar también **GitHub login** para que tengas los dos listos?

# 8) Mini-esqueleto de “marketplace”

Así dejás listo lo mínimo para avanzar:

**python manage.py startapp market**

market/models.py:

from django.conf import settings

from django.db import models

class Product(models.Model):

    seller = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE, related\_name="products")

    title = models.CharField(max\_length=200)

    description = models.TextField(blank=True)

    marca = models.CharField(max\_length=100, blank=True, default="Generico")

    price = models.DecimalField(max\_digits=12, decimal\_places=2)

    created\_at = models.DateTimeField(auto\_now\_add=True)

    active = models.BooleanField(default=True)

    def \_\_str\_\_(self):

        return self.title

market/views.py (listado simple):

from django.shortcuts import render

from .models import Product

def product\_list(request):

    products = Product.objects.filter(active=True).order\_by("-created\_at")

    return render(request, "product\_list.html", {"products": products})

myclase/urls.py (agregar ruta):

from django.contrib import admin

from django.urls import path, include

from core.views import home

urlpatterns = [

    path('admin/', admin.site.urls),

    path("", home, name="home"),

    path("accounts/", include("allauth.urls")),  # allauth

    path("productos/", include("market.urls")),  # users

]

templates/ product\_list.html:

{% extends "base.html" %}

{% block content %}

<h2>Productos</h2>

<ul>

  {% for p in products %}

    <li>

      <strong>{{ p.title }}</strong> — $ {{ p.price }}  <p>Marca: {{ p.marca }}</p>  <br>

      <small>Publicado por {{ p.seller.get\_full\_name|default:p.seller.username }}</small>

    </li>

  {% empty %}

    <li>No hay productos aún.</li>

  {% endfor %}

</ul>

{% endblock %}

market/admin.py

from django.contrib import admin

from .models import Product

@admin.register(Product)

class ProductAdmin(admin.ModelAdmin):

    list\_display = ("title", "seller", "marca", "price", "active", "created\_at")  # columnas que ves en la lista

    search\_fields = ("title", "description", "marca", "seller\_\_username")          # campos por los que podés buscar

    list\_filter = ("active", "created\_at", "seller")                       # filtros en la barra lateral

Eso es para cargar datos por el administración de Django -> entrar con super usuario

Activá la app en settings.py:

Luego:

python manage.py makemigrations market

python manage.py migrate

Proxima clase -> Link para entrar con vinculo y con con la url

* + - Formulario para CRUD productos, agregar imagen
    - **users/serializers.py**
    - perfil usuario
    - terminar login