



AE1 – Python y los entornos virtuales

Bootcamp Full Stack Python – Módulo 6

Hoy aprenderemos a crear, activar y gestionar entornos virtuales en Python y usarlos en proyectos con Django.

Objetivos de la clase

Comprender qué es un entorno virtual.

Crear y activar entornos virtuales con venv.

Usar entornos en proyectos Django.

Aplicar MVC y DRY en Django.



¿Qué es un entorno virtual?

Un entorno virtual es un espacio aislado que:

- ✈ **Tiene su propia instalación de Python.**
- ✈ **Guarda las librerías de un proyecto.**
- ✈ **Evita conflictos entre proyectos.**

👉 Sin entornos virtuales, todas las librerías van al Python global.

¿Por qué usarlos?


Problemas sin entornos virtuales:

 **Conflictos de versiones.**

 **Librerías innecesarias ocupando espacio.**

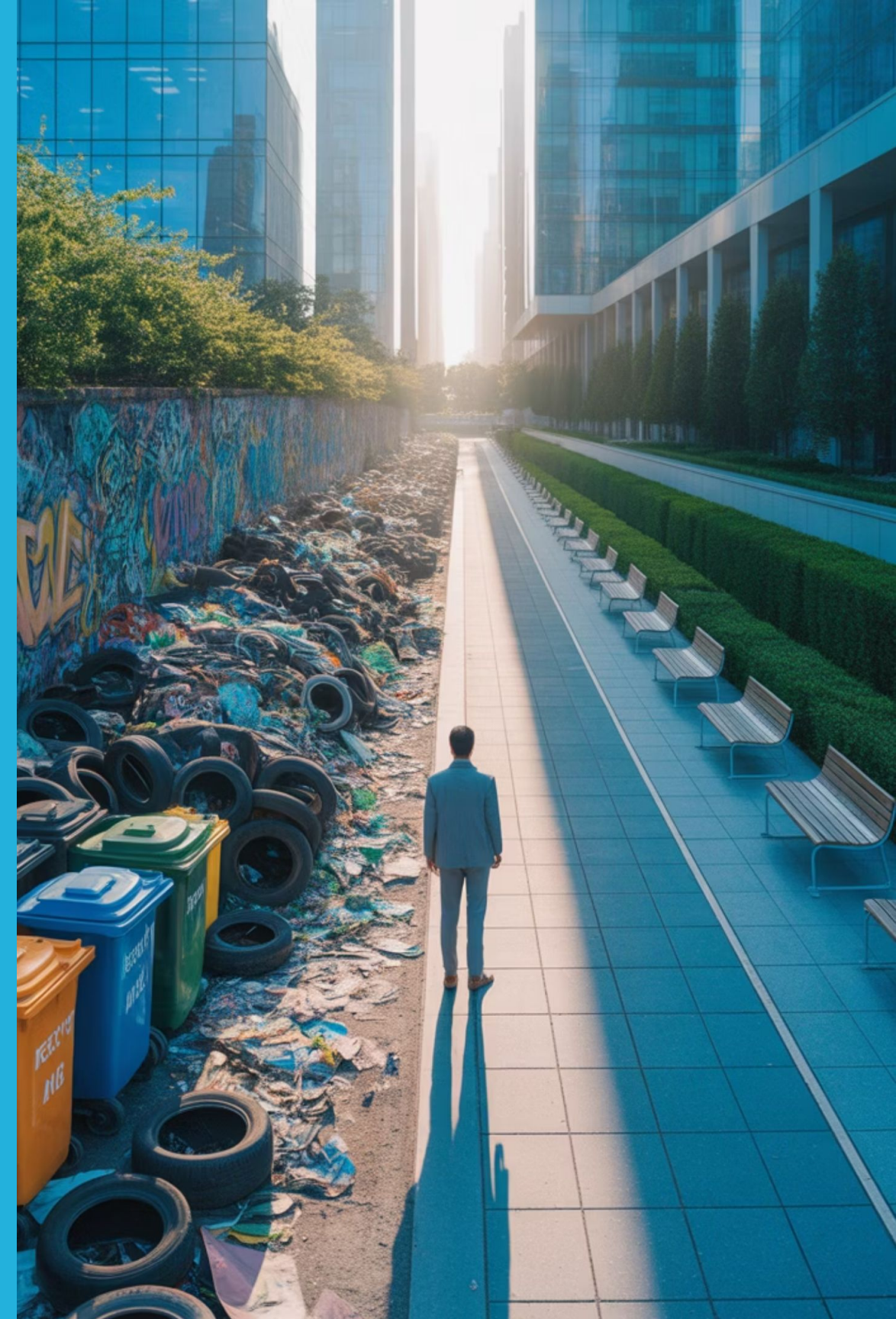
 **Dificultad para compartir**

Con entornos virtuales:

 **Cada proyecto tiene lo suyo.**

 **Reproducibilidad.**

 **Mayor velocidad de desarrollo.**



Creando un entorno virtual

Crear entorno virtual

```
python -m venv mi_entorno
```

Esto genera una carpeta con:

- 📁 **Scripts/ o bin/ (ejecutables)**
 - 📁 **lib/ (librerías del proyecto)**
 - 📁 **pyvenv.cfg (configuración)**
- Actividad rápida: Que los estudiantes creen un entorno mi_entorno.

Activando el entorno

En Windows

```
mi_entorno\Scripts\activate
```

En macOS / Linux

```
source mi_entorno/bin/activate
```

👉 Una vez activado, verás (mi_entorno) al inicio de la terminal.

Actividad: Cada estudiante activa su entorno y confirma con:


```
python --version
```



Desactivando el entorno

deactivate

Esto te devuelve al Python global.

 **Importante cuando cambias de proyecto.**

Ejemplo:

Proyecto A: usa Django 4.2

Proyecto B: usa Django 5.0

Sin desactivar, podrías instalar en el entorno incorrecto.

Diferencia Global vs Virtual

Global:

**Todo se
instala en el
sistema.**

```
pip install django
```

Virtual:

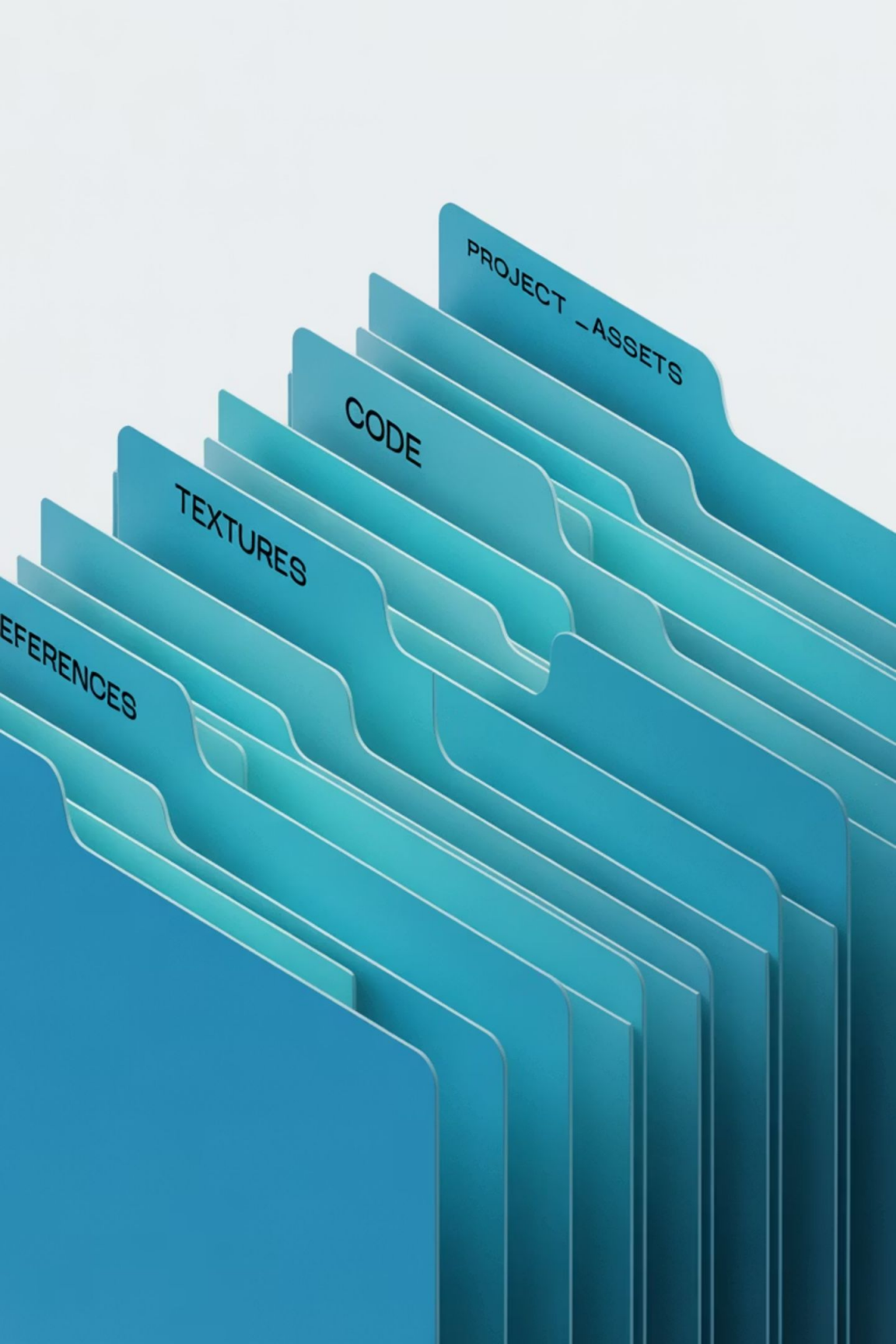
**Solo afecta
al entorno
activo.**

```
pip install django
```

**👉 Mismo comando,
distinto resultado.**

Actividad: Que un estudiante pruebe instalar requests dentro del entorno y otro fuera.





Estructura minimalista

Carpeta creada:

```
mi_entorno/|— Scripts/ o bin/|— lib/|— pyvenv.cfg
```

Es ligera y solo tiene lo básico.

👉 Permite añadir cualquier librería sin "ensuciar" el sistema.

Estructura flexible

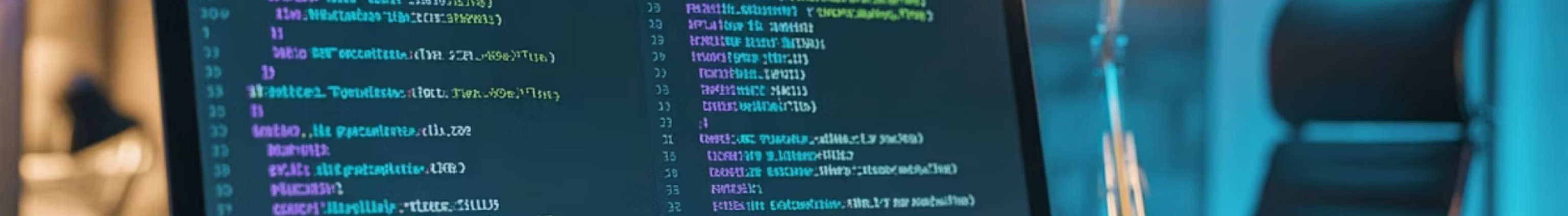
Puedes instalar cualquier librería con pip.

Cada proyecto mantiene sus propias versiones.

Ejemplo:

```
pip install django==4.2  
pip install django-rest-framework
```





Librerías por proyecto

Dentro del entorno virtual solo se instalan lo que tu proyecto necesita.

Ejemplo:

```
pip freeze > requirements.txt
```

Esto guarda las dependencias para compartir con el equipo.



Aislamiento del entorno

Ventajas del aislamiento:

- ✈ **Sin interferencias.**
- ✈ **Se pueden probar nuevas versiones.**
- ✈ **Proyectos estables y predecibles.**

Ejemplo práctico: Crear un entorno con Django 4.2 y otro con Django 5.0, correr
`django-admin --version.`

Manejo de versiones de Python

Crear un entorno con versión específica:

```
python3.9 -m venv env39python3.11 -m venv env311
```

Verificar:

```
python --version
```

 Herramienta extra: pyenv para gestionar múltiples versiones.

Conexión con Django

Cada proyecto Django debe tener su entorno:

```
pip install djangodjango-admin startproject miapp
```

Actividad: Crear un entorno, instalar Django y verificar con:

```
django-admin --version
```



Enrutador de Django

El enrutador decide qué vista ejecutar según la URL.

Ejemplo en urls.py:

```
from django.urls import pathfrom . import viewsurlpatterns = [
    path("contacto/", views.contacto),]
```

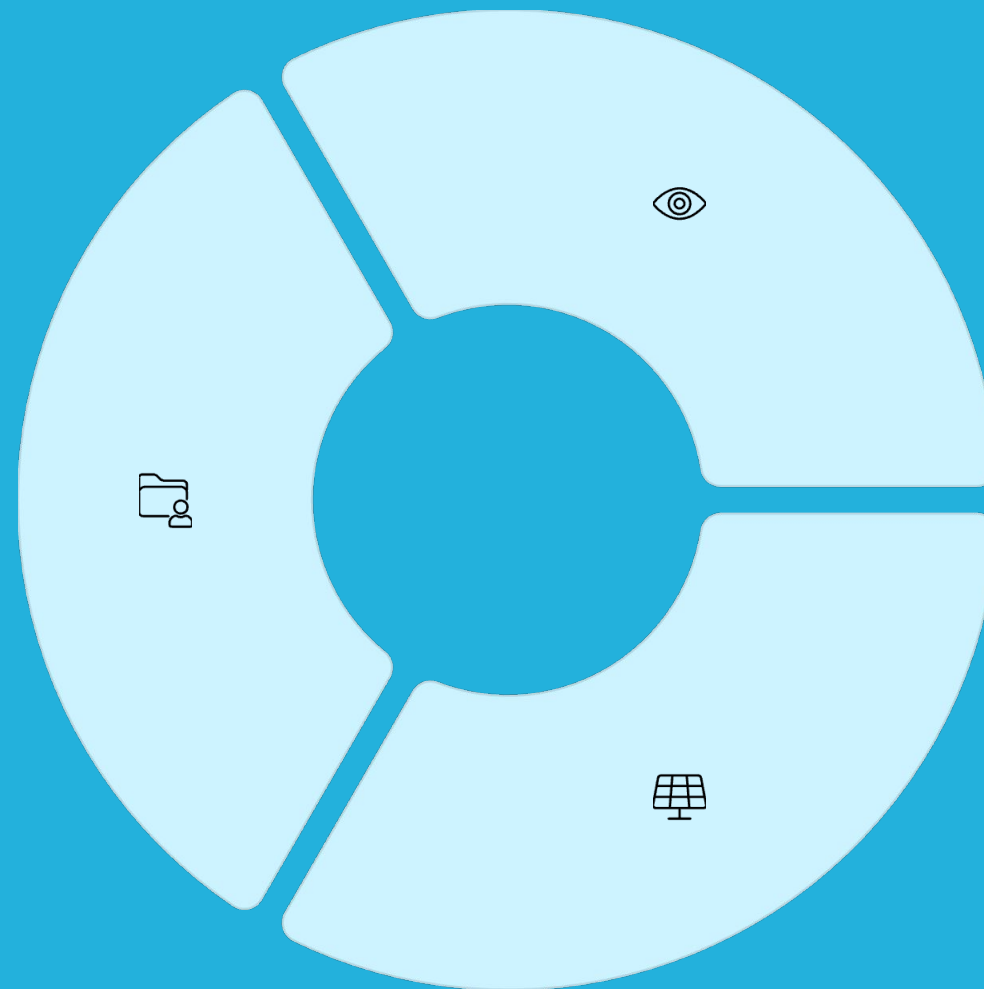
👉 Al entrar a /contacto/, Django llama a views.contacto.



MVC en Django

Django usa MVC con su propia terminología:

**Model =
Datos.**



**View = Lógica
y
presentación.
Controller =
Enrutador +
vistas.**



Principio DRY en Django

DRY = Don't Repeat Yourself

En Django se aplica con:

- ✈ **Herencia de plantillas.**
- ✈ **Modelos reutilizables.**
- ✈ **Funciones comunes en vistas.**

Ejemplo:

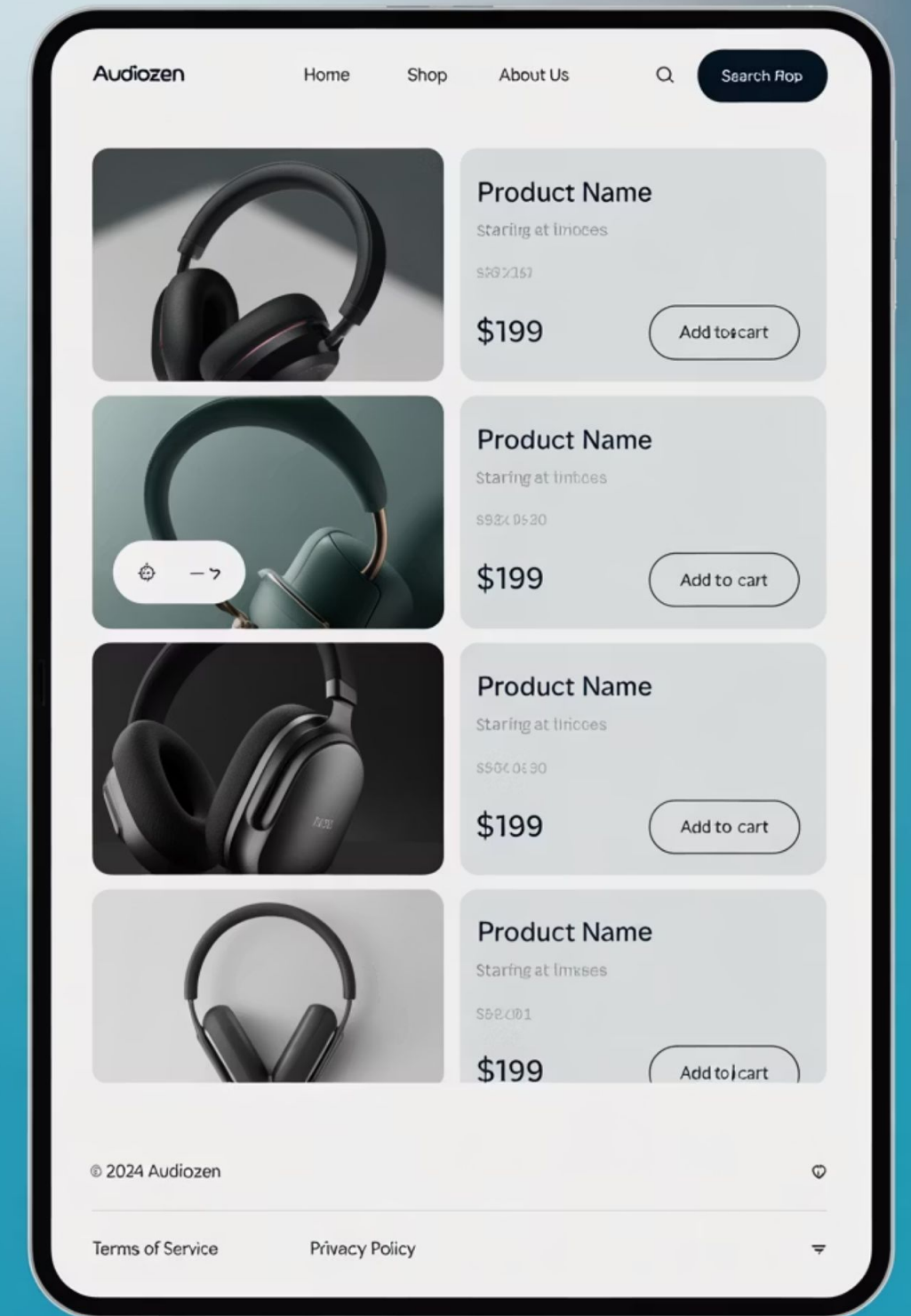
```
{% extends "base.html" %}{% block content %}# Página de Contacto{% endblock %}
```


Templates dinámicos

Ejemplo de template:

```
## Lista de Productos{% for producto in productos %}- {{  
producto.nombre }} - ${{ producto.precio }}{% endfor %}
```

👉 Django reemplaza
dinámicamente los
valores.





Actividad práctica guiada

Reto grupal:

Crear un entorno virtual.

Instalar Django.

Iniciar un proyecto.

**Crear una ruta `/hola/`
que muestre:**

`# Hola Bootcamp!`

Cierre y reflexión

Hoy aprendimos:

- ✈ **Qué es un entorno virtual y por qué usarlo.**
- ✈ **Cómo crearlo, activarlo y desactivarlo.**
- ✈ **Diferencias entre global vs virtual.**
- ✈ **Cómo conectar entornos virtuales con Django.**
- ✈ **Principios básicos de MVC y DRY.**
- 👉 **Pregunta final: ¿Qué pasa si NO usamos entornos virtuales en un proyecto?**