

## Desarrollo de Aplicaciones Web con Django y Templates

### Objetivo del día:

Implementar una aplicación web en Django utilizando templates y contenido estático.

Integrar Bootstrap para mejorar el diseño.

Crear páginas dinámicas y responsivas.

### Metodología:

- Explicación breve + código en vivo + ejercicios prácticos.



# ¿Qué aprenderás hoy?



**Configuración de archivos estáticos en Django**



**Uso de Bootstrap (navbar, jumbotron, footer, grid)**



**Páginas responsivas**



**Navegación interna con etiquetas URL**



**MVC vs MTV en Django**



**Uso de templates dinámicos (variables, condicionales, bucles)**



**Herencia e inclusión de templates**



**Manejo básico de errores**



**Creación de un modelo simple**

# Contenido Estático en Django

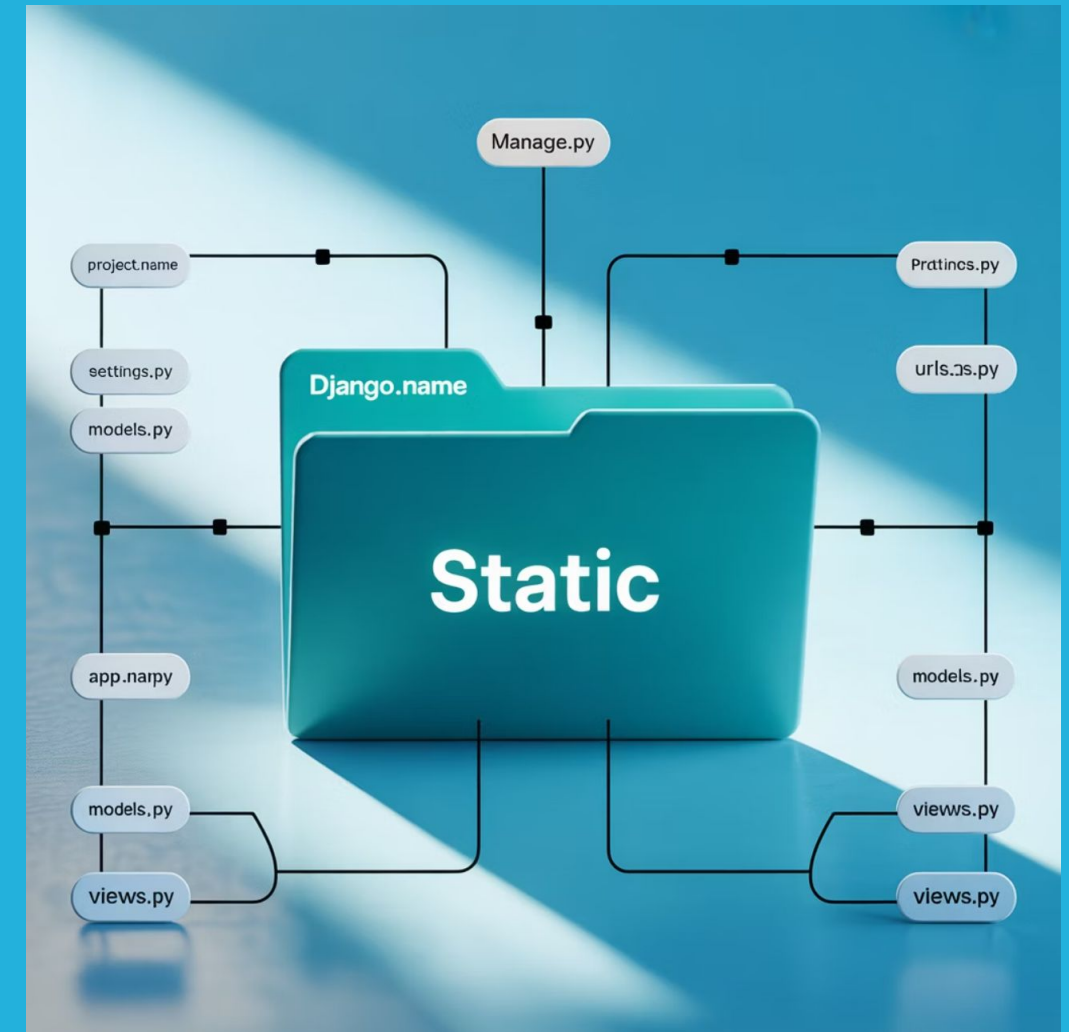
## Definición:

Archivos como imágenes, CSS o JavaScript que no cambian con la interacción del usuario.

## Ejemplo práctico:

📁 Estructura recomendada:

```
mi_proyecto/  
├── mi_app/  
├── static/  
│   ├── css/  
│   ├── js/  
│   └── images/  
├── templates/  
└── manage.py
```



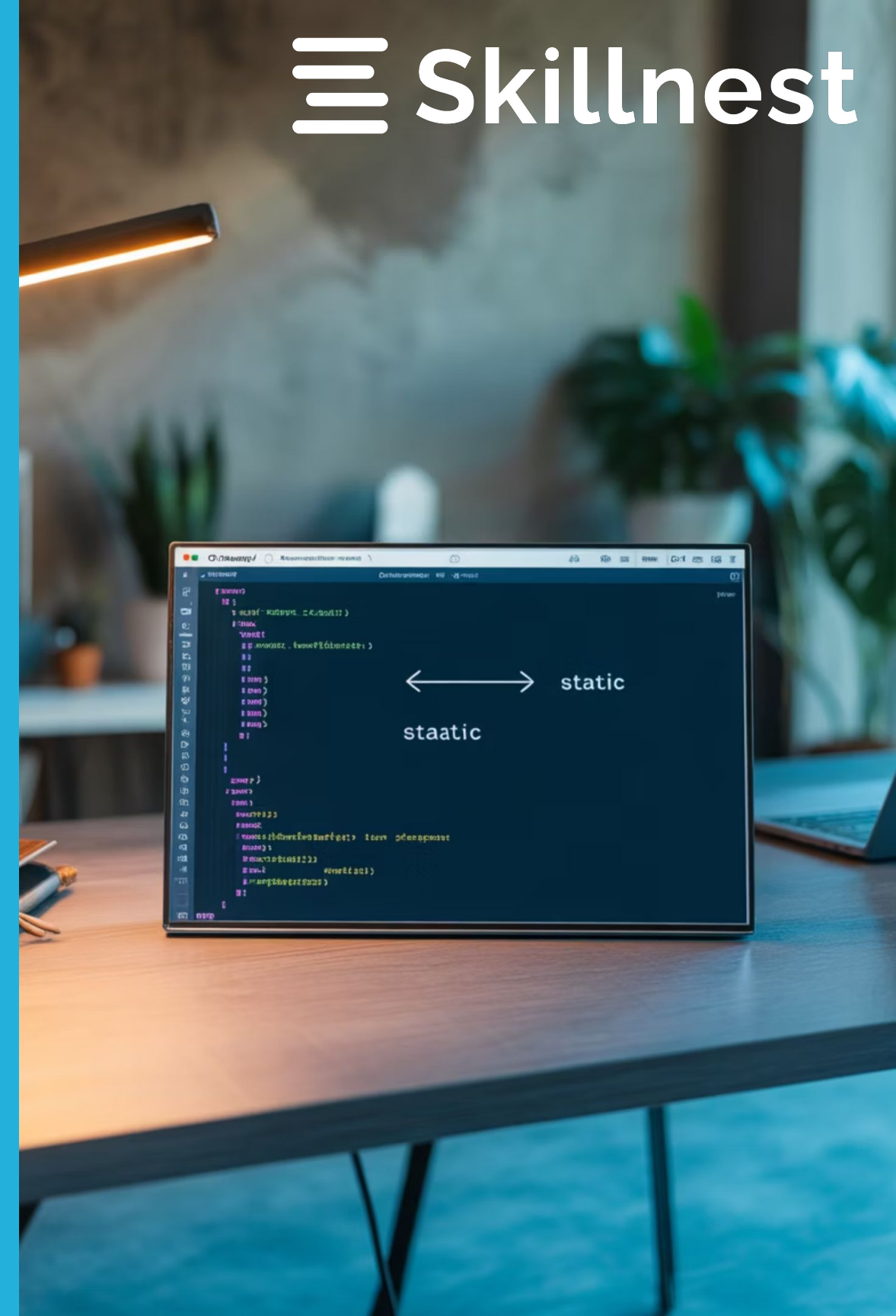
# Configuración de Archivos Estáticos

En settings.py:

```
STATIC_URL = '/static/'  
from pathlib import Path  
BASE_DIR = Path(__file__).resolve().parent.parent  
STATICFILES_DIRS = [  
    BASE_DIR / "static",  
]
```

En el template:

```
{% load static %}  
<link rel="stylesheet" href="{% static 'css/estilos.css' %}">
```





# Servir una Página Estática

## views.py

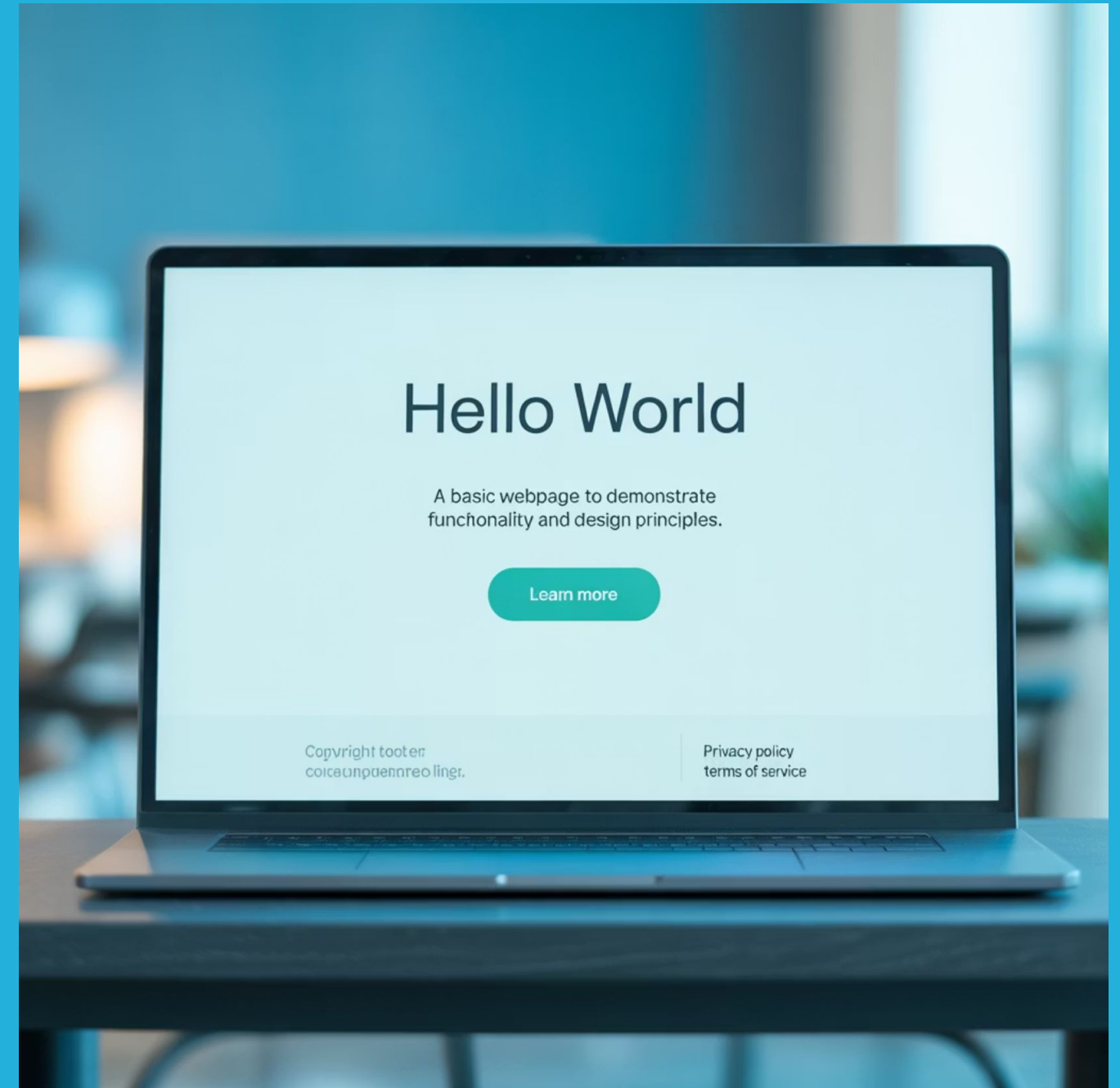
```
from django.shortcuts import render

def pagina_estatica(request):
    return render(request, 'index.html')
```

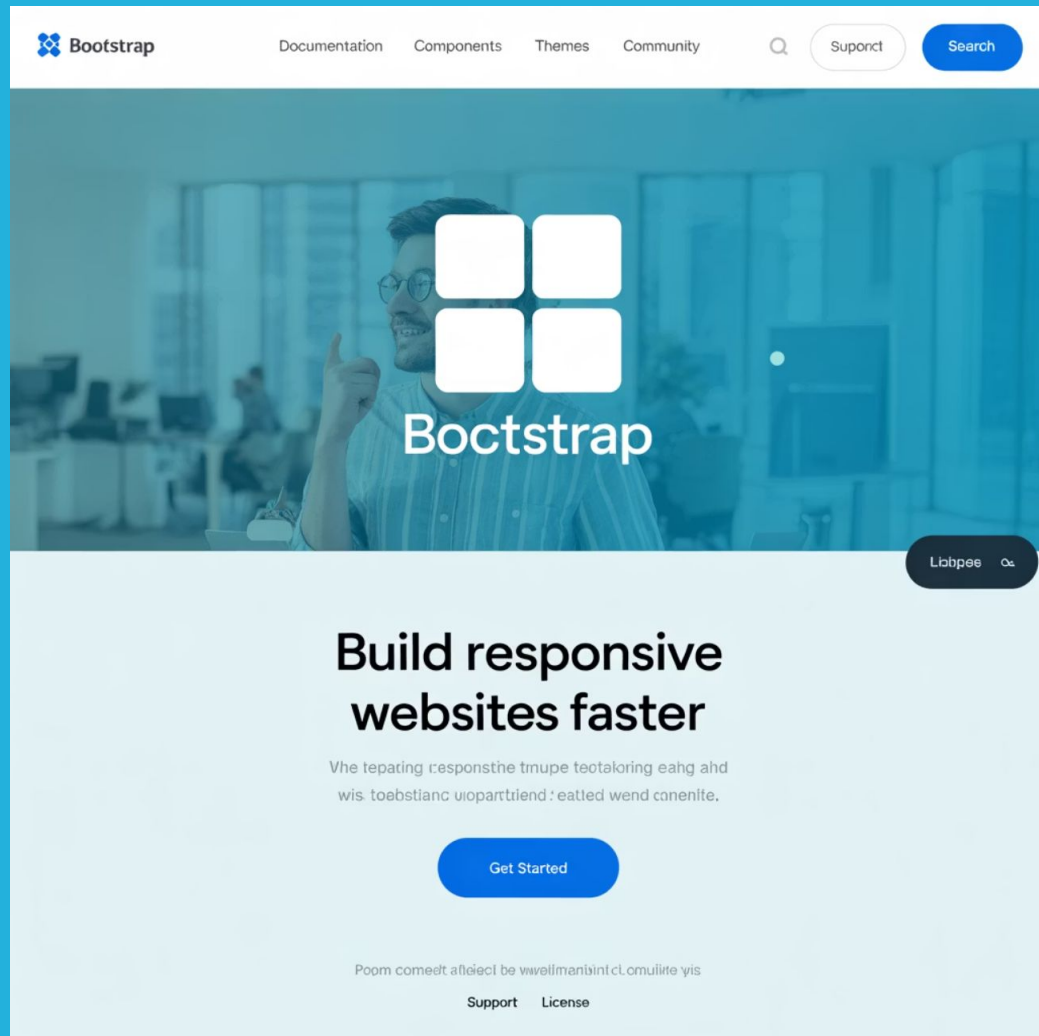
## urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.pagina_estatica, name='pagina_estatica'),
]
```



# Integrando Bootstrap

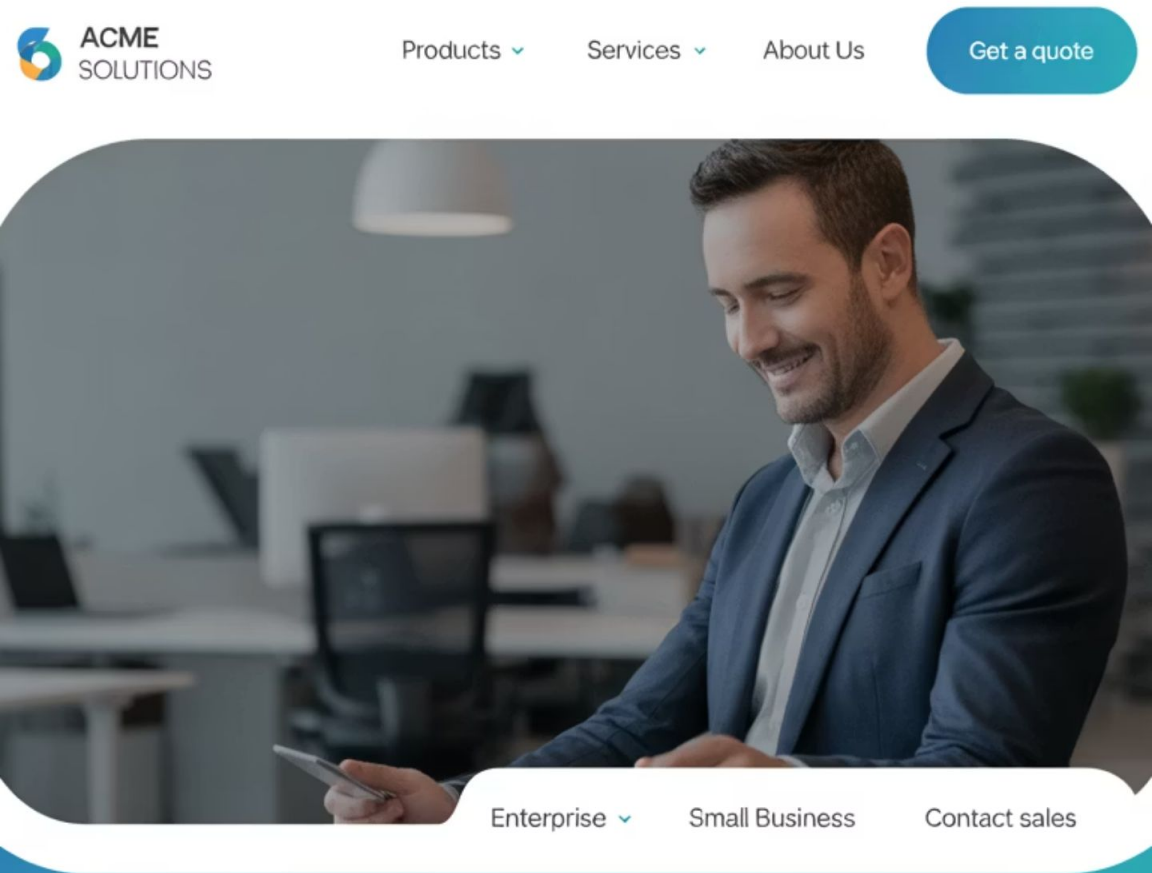


## ¿Por qué Bootstrap?

- Plantillas listas para usar.
- Diseño responsivo sin esfuerzo.
- Componentes reutilizables (navbar, botones, cards).

## En index.html:

```
<link  
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstr  
ap.min.css"rel="stylesheet">
```



Your business.  
Our expertise.

Data-driven solutions for sustainable growth

Learn more

 Skillnest

# Navbar con Bootstrap

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container">
    <a class="navbar-brand" href="#">Mi Sitio</a>
    <button class="navbar-toggler" data-bs-toggle="collapse"
data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item"><a class="nav-link"
href="#">Inicio</a></li>
        <li class="nav-item"><a class="nav-link"
href="#">Servicios</a></li>
      </ul>
    </div>
  </div>
</nav>
```

[Mi Sitio](#)



• [Inicio](#)

• [Servicios](#)



**Actividad rápida: Agrega un nuevo link "Contacto".**

# Jumbotron en Bootstrap

```
<div class="container mt-4">
  <div class="p-5 mb-4 bg-light rounded-3">
    <h1 class="display-5 fw-bold">Bienvenido a Mi Sitio</h1>
    <p class="fs-4">Sitio construido con Django y Bootstrap</p>
    <a class="btn btn-primary btn-lg" href="#">Más información</a>
  </div>
</div>
```

## Bienvenido a Mi Sitio

Sitio construido con Django y Bootstrap

Más información

# Secure Your Digital Life

Unlimited storage, effortless access, and military-grade encryption.



### Sync

Un omiteestoeisortien  
iattciaoi dne siay et  
yr eceaclins,



### Share

Fon mitorndr aceatie,  
fiate aue inatice .lineeg  
in: tucraicir anis.



### Protect

Ar lia tiorecirveobmeite  
and ana, atteluctantainl  
yre caârcictveng rinén.

START FREE TRIAL



# Footer en Bootstrap

```
<footer class="bg-dark text-white text-center py-3">  
  <p>&copy; 2025 Mi Sitio. Todos los derechos  
reservados.</p>  
</footer>
```

© 2025 Mi Sitio. Todos los derechos reservados.

✓ Ejercicio: Personaliza el texto con tu nombre y curso.

© 2024 Stellartech Solutions



[Terms of Service](#) | [Privacy Policy](#) | [Contact Us](#)

## Páginas Responsivas

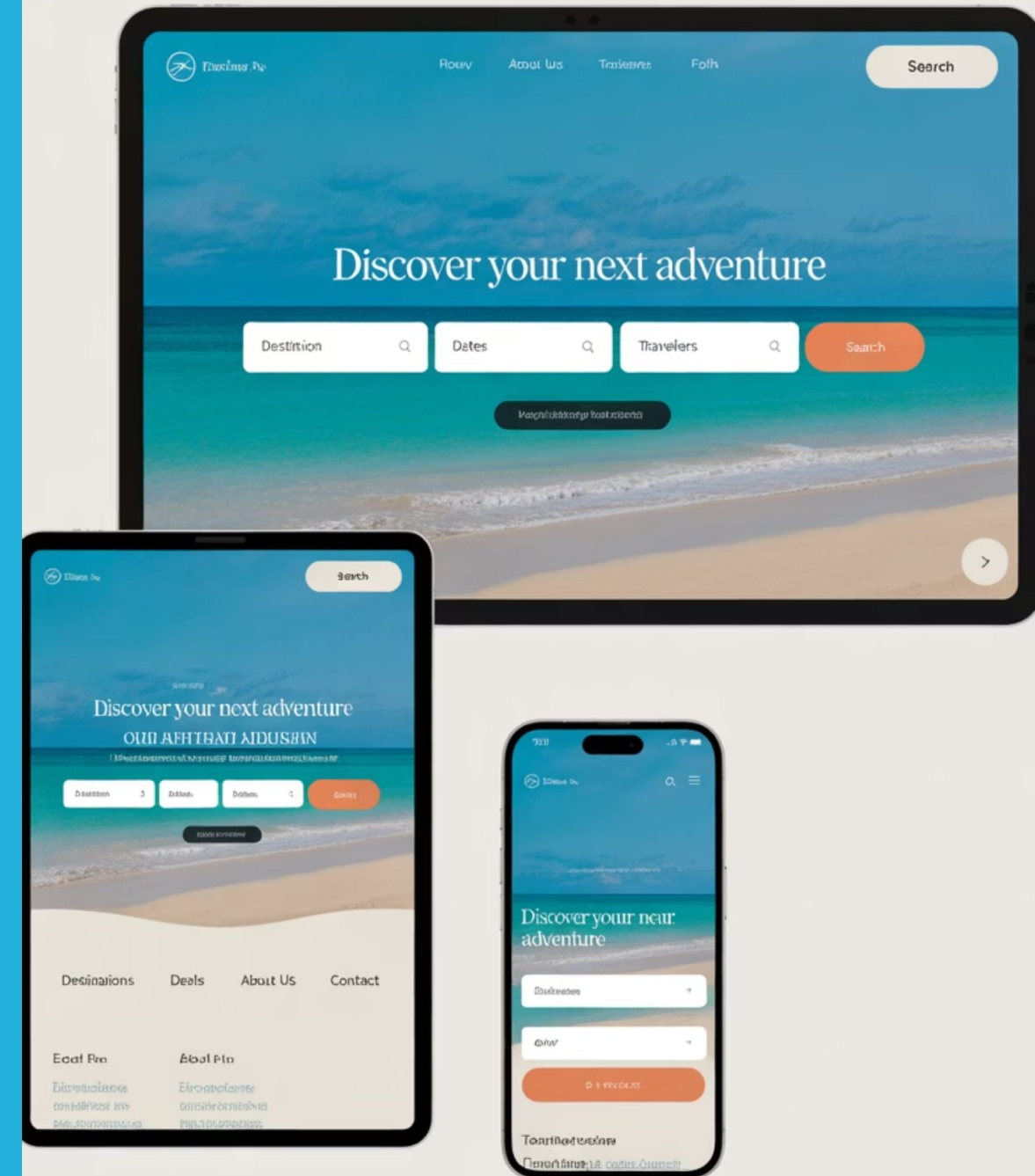
Bootstrap usa grid system

- Se adapta a pantallas chicas, medianas, grandes

```
<div class="row">
  <div class="col-md-6">Sección 1</div>
  <div class="col-md-6">Sección 2</div>
</div>
```

Sección 1

Sección 2



Connect your code,  
empower your vision

"Connect your code, empower your vision."



# Navegación con URL en Django

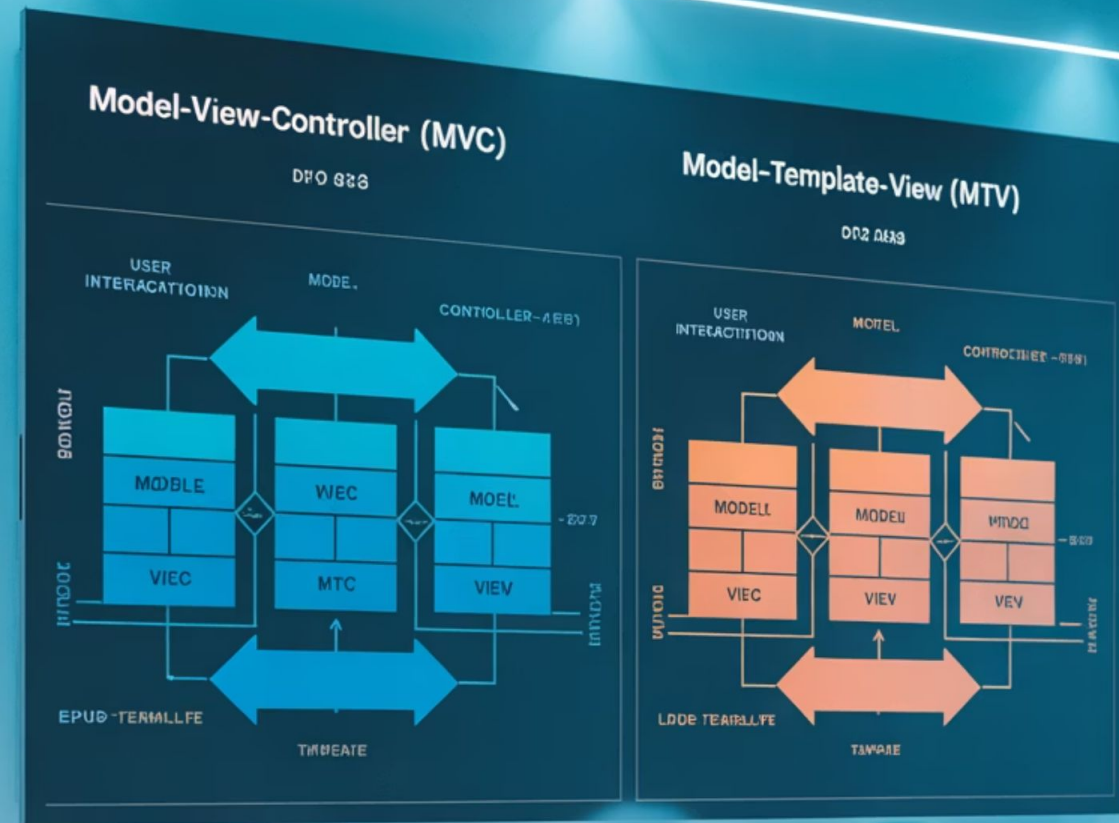
```
<a href="{% url 'pagina_estatica' %}">Ir a Inicio</a>
```

```
<li class="nav-item">  
  <a class="nav-link {% if request.path == '/' %}active{% endif %}"  
    href="{% url 'pagina_estatica' %}">Inicio</a>  
</li>
```

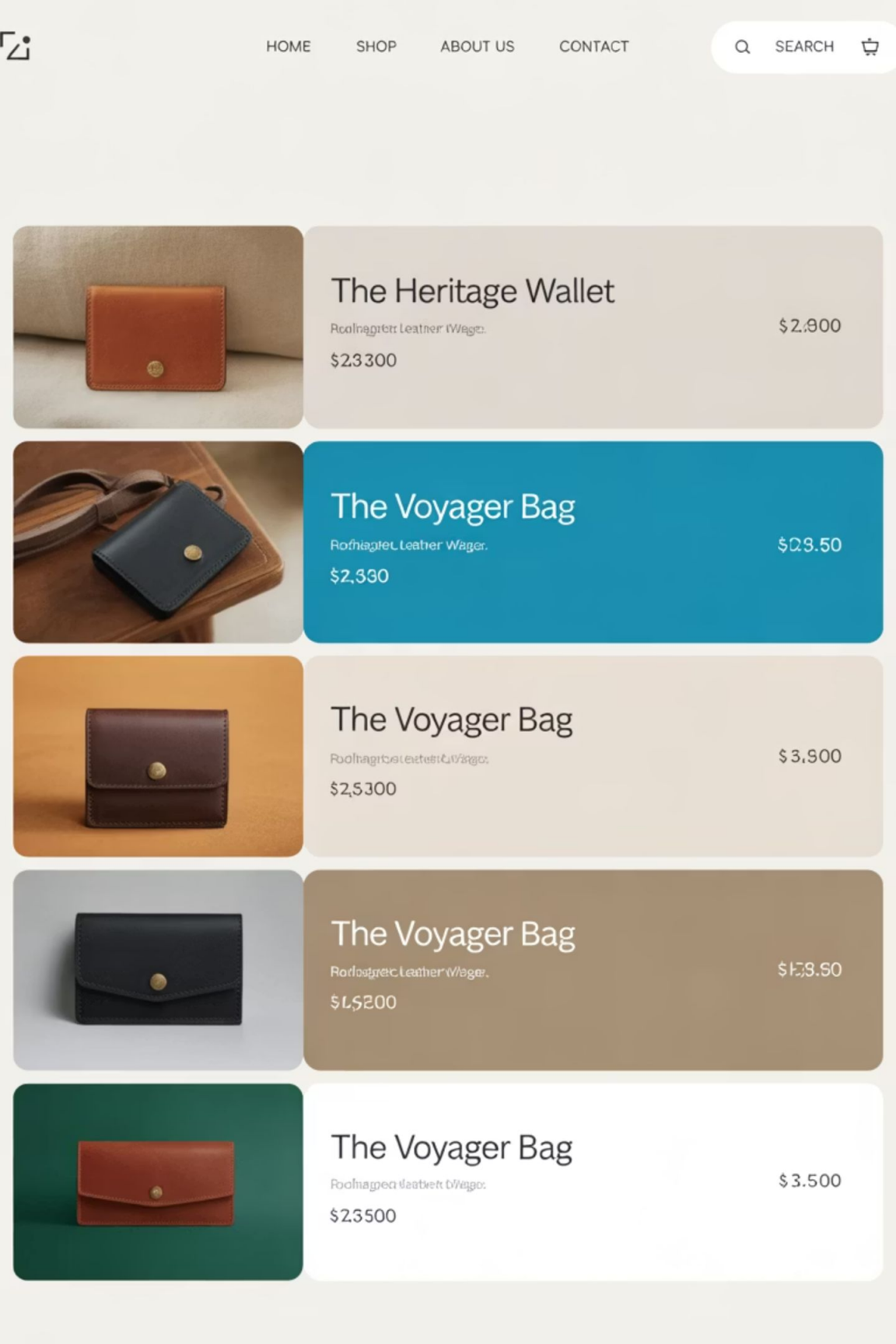
Inicio

- **Modelo** → **datos**
- **Vista** → **interfaz**
- **Controlador** → **lógica**

- **Model** → **datos**
- **Template** → **interfaz**
- **View** → **lógica**







## Ejemplo Práctico MTV

models.py

```
class Producto(models.Model):
    nombre = models.CharField(max_length=100)
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    descripcion = models.TextField()
```

views.py

```
def lista_productos(request):
    productos = Producto.objects.all()
    return render(request, 'productos.html', {'productos': productos})
```

# Template con Variables

```
{% for producto in productos %}  
  <li class="list-group-item">  
    <strong>{{ producto.nombre }}</strong> - ${{ producto.precio }}  
    <p>{{ producto.descripcion }}</p>  
  </li>  
{% endfor %}
```



## Condicionales en Templates

```
{% if producto.precio > 1000 %}  
  <span class="badge bg-success">Premium</span>  
{% else %}  
  <span class="badge bg-secondary">Económico</span>  
{% endif %}
```

 Ejercicio: Agrega un mensaje si no hay productos.





# Herencia de Templates

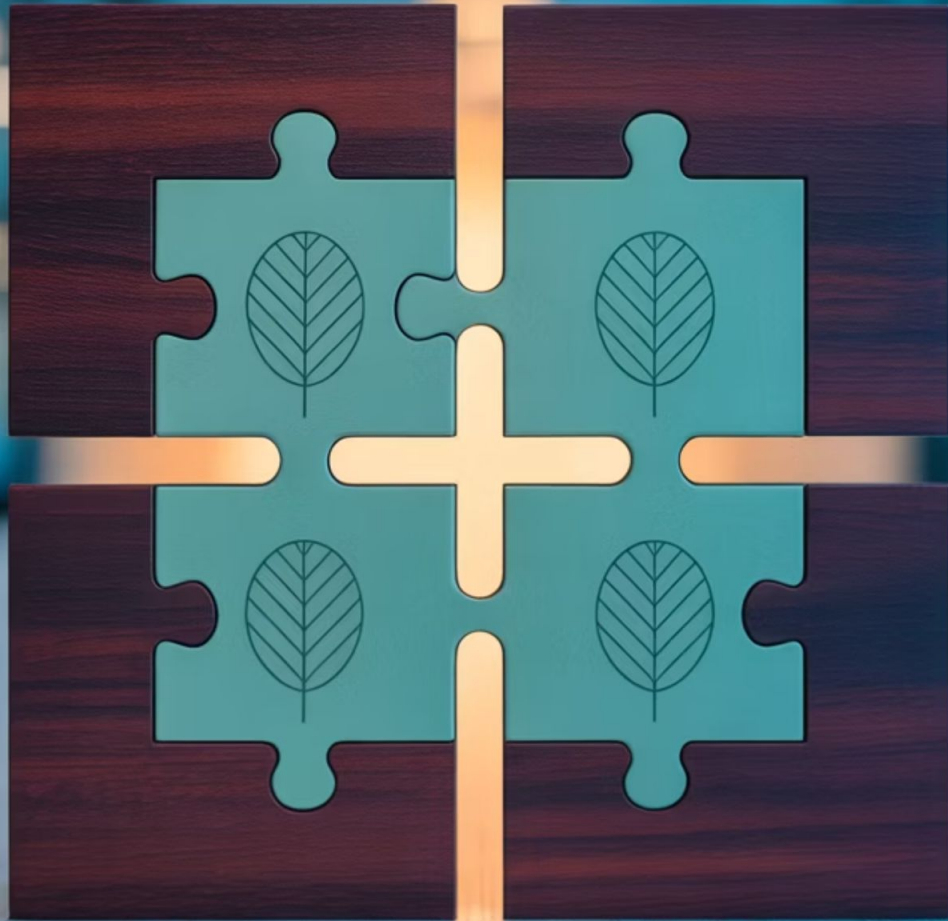
## base.html

```
<!DOCTYPE html>
<html>
  <head>{% block head %}{% endblock %}</head>
  <body>
    {% include 'navbar.html' %}
    {% block content %}{% endblock %}
    {% include 'footer.html' %}
  </body>
</html>
```

## index.html

```
{% extends 'base.html' %}
{% block content %}
  <h1>Página principal</h1>
{% endblock %}
```





# Inclusión de Templates

```
{% include 'navbar.html' %}  
{% include 'footer.html' %}
```

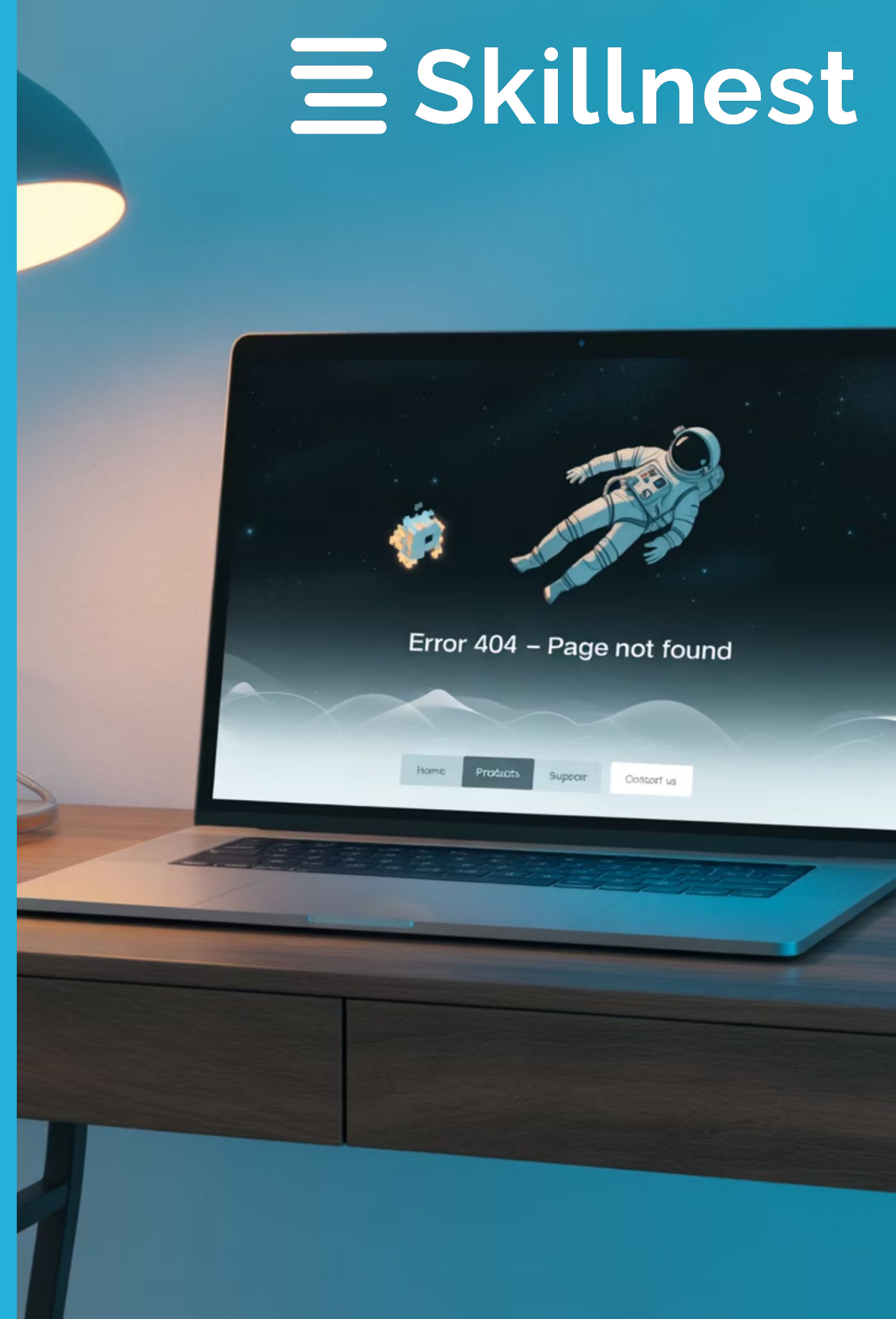
## Beneficio:

**Código modular, fácil de mantener.**

# Manejo de Errores en Django

```
from django.http import Http404

def detalle_producto(request, id):
    try:
        producto = Producto.objects.get(pk=id)
    except Producto.DoesNotExist:
        raise Http404("Producto no encontrado")
    return render(request, 'detalle.html', {'producto': producto})
```



# Actividad Final

 **En parejas:**

01

**Crear un nuevo proyecto Django**

02

**Configurar carpeta static/**

03

**Crear base.html con navbar,  
footer y jumbotron**

04

**Crear vista y template productos.html**

05

**Mostrar lista dinámica con condicionales y  
Bootstrap**

# Cierre y Preguntas

Hoy aprendimos a:

✓ Servir contenido estático

✓ Integrar Bootstrap

✓ Usar templates dinámicos

✓ Aplicar herencia e inclusión

✓ Diferenciar MVC y MTV

👉 ¿Qué diferencia clave encuentras entre un sitio estático y uno dinámico en Django?