



Práctica 1.

Preparación del entorno de desarrollo con Django



Objetivo

Preparar Python, instalar Django y crear el proyecto base de Ñôtho Digital.



PASOS

1

Crear la carpeta del proyecto

Abre VS Code y utiliza la terminal de PowerShell para crear y abrir la carpeta de trabajo.

TERMINAL · PowerShell

```
1 mkdir C:\Proyectos\natho-digital
2 cd C:\Proyectos\natho-digital
```

2

Crear el entorno virtual

El entorno web aísla las bibliotecas de este proyecto de otros programas de Python.

TERMINAL · PowerShell

```
1 py -m venv web
```

3

Activar el entorno

Actívalo antes de instalar paquetes. La terminal mostrará (web) al principio.

TERMINAL · PowerShell

```
1 .\web\Scripts\Activate.ps1
```

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.



Práctica 1.

Preparación del entorno de desarrollo con Django

4

Instalar Django

Con el entorno web activo, instala Django y verifica que el paquete responda.

TERMINAL · PowerShell

```
1 python -m pip install django
2 python -m django --version
```

5

Crear el proyecto config

El punto final indica que el proyecto se creará dentro de la carpeta actual.

TERMINAL · PowerShell

```
1 django-admin startproject config .
```

6

Revisar la estructura

Comprueba que aparezcan manage.py y la carpeta config.

► EXPLORADOR · ÑÖTHO-DIGITAL

- config/
- core/ casas/ turismo/
- lengua/ videojuego/
- templates/ static/

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.



Práctica 1.

Preparación del entorno de desarrollo con Django

7

Preparar la base de datos local

Aplica las migraciones iniciales para crear las tablas que Django necesita.

TERMINAL · PowerShell

```
1 python manage.py migrate
```

8

Iniciar el servidor

Abre el navegador en la dirección local que indica la terminal.

TERMINAL · PowerShell

```
1 python manage.py runserver
```

127.0.0.1:8000

Ñötho Digital

[Inicio](#) [Casas](#) [Turismo](#) [Lengua](#) [Videojuego](#)

Raíces y Código · Página funcionando

9

Comprobar el resultado

La página inicial de Django confirma que el proyecto se ejecuta. Guarda una captura como evidencia.

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.



Práctica 2.

Arquitectura y cinco aplicaciones Django



Objetivo

Organizar el proyecto en cinco aplicaciones independientes y registrar cada una.



PASOS

1

Comprobar el proyecto

Abre la carpeta del proyecto en VS Code, activa web y verifica que Django no reporte errores.

TERMINAL · PowerShell

```
1 python manage.py check
```

2

Crear las cinco aplicaciones

Ejecuta cada comando desde la carpeta que contiene manage.py.

TERMINAL · PowerShell

```
1 python manage.py startapp core
2 python manage.py startapp casas
3 python manage.py startapp turismo
4 python manage.py startapp lengua
5 python manage.py startapp videojuego
```

3

Explorar las carpetas

Cada app contiene archivos como views.py, models.py y apps.py.

► EXPLORADOR · ÑÖTHO-DIGITAL

- config/
- core/ casas/ turismo/
- lengua/ videojuego/
- templates/ static/



Práctica 2.

Arquitectura y cinco aplicaciones Django

4

Registrar las aplicaciones

En config/settings.py, dentro de INSTALLED_APPS, agrega estos nombres sin borrar las aplicaciones propias de Django.

TERMINAL · PowerShell

```
1 'core',  
2 'casas',  
3 'turismo',  
4 'lengua',  
5 'videojuego',
```

5

Entender la separación

core: portada; casas: casas de resguardo; turismo: rutas y tradiciones; lengua: vocabulario; videojuego: ficha y descarga.

6

Verificar la configuración

Revisa las comas y las comillas antes de guardar.

TERMINAL · PowerShell

```
1 python manage.py check
```

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.



Práctica 2.

Arquitectura y cinco aplicaciones Django

7

Crear la primera vista

En `core/views.py` define una respuesta temporal para comprobar el funcionamiento de la aplicación.

TERMINAL · PowerShell

```
1 from django.http import HttpResponse
2 def inicio(request):
3     return HttpResponse("Ñötho Digital")
```

8

Conectar la URL

En `core/urls.py` y `config/urls.py` conecta la ruta principal.

TERMINAL · PowerShell

```
1 # core/urls.py
2 from django.urls import path
3 from . import views
4 urlpatterns = [path("", views.inicio)]
5
6 # config/urls.py: agregar include
7 path("", include("core.urls")),
```

9

Probar en el navegador

Visita `127.0.0.1:8000`. Debe aparecer el mensaje Ñötho Digital.

TERMINAL · PowerShell

```
1 B
2 R
3 O
4 W
5 S
6 E
7 R
```



Práctica 3.

Plantilla HTML compartida y menú principal



Objetivo

Crear una plantilla reutilizable, estilos CSS y un menú para todas las aplicaciones.



PASOS

1

Crear carpetas compartidas

Dentro de la raíz del proyecto crea templates y static; dentro de static agrega css e img.

TERMINAL · PowerShell

```
1 mkdir templates
2 mkdir static
3 mkdir static\css
4 mkdir static\img
```

► EXPLORADOR · ÑÖTHO-DIGITAL

- config/
- core/ casas/ turismo/
- lengua/ videojuego/
- templates/ static/

2

Configurar las plantillas

En config/settings.py, dentro de TEMPLATES, establece DIRS y agrega la carpeta static.

TERMINAL · PowerShell

```
1 'DIRS': [BASE_DIR / 'templates'],
2 STATIC_URL = "static/"
3 STATICFILES_DIRS = [BASE_DIR / "static"]
```

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.



Práctica 3.

Plantilla HTML compartida y menú principal

3

Crear base.html

En templates/base.html carga static, enlaza estilos y agrega un bloque para el contenido de cada sección.

TERMINAL · PowerShell

```
1 {% load static %}
2 <!DOCTYPE html>
3 <html lang="es"><head>
4 <meta charset="UTF-8">
5 <title>Ñötho Digital</title>
6 <link rel="stylesheet" href="{% static 'css/estilos.css' %}">
7 </head><body>
8 <nav>Inicio | Casas | Turismo | Lengua</nav>
9 <main>{% block contenido %}{% endblock %}</main>
10 </body></html>
```

4

Crear estilos CSS

En static/css/estilos.css agrega un diseño básico para el menú y el contenido.

TERMINAL · PowerShell

```
1 body { font-family: Arial, sans-serif; margin: 0; }
2 nav { background: #0a3470; color: white; padding: 18px; }
3 main { padding: 24px; }
```

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.



Práctica 3.

Plantilla HTML compartida y menú principal

5

Crear una página de inicio

En templates/inicio.html extiende la plantilla compartida y agrega un mensaje.

TERMINAL · PowerShell

```
1 {% extends "base.html" %}
2 {% block contenido %}
3 <h1>Bienvenidos a Ñötho Digital</h1>
4 <p>Raíces y Código</p>
5 {% endblock %}
```

6

Renderizar desde Django

En core/views.py sustituye la respuesta temporal por render.

TERMINAL · PowerShell

```
1 from django.shortcuts import render
2 def inicio(request):
3     return render(request, "inicio.html")
```

7

Comprobar en el navegador

Ejecuta el servidor y revisa el menú, los estilos y el contenido de la portada.

TERMINAL · PowerShell

```
1 python manage.py runserver
```

127.0.0.1:8000

Ñötho Digital

Inicio Casas Turismo Lengua Videojuego

Raíces y Código · Página funcionando



Práctica 4.

GitHub y trabajo colaborativo



Objetivo

Compartir el código de los cinco equipos mediante Git, ramas y solicitudes de integración.



PASOS

1

Comprobar Git

Desde la terminal de VS Code confirma que Git esté instalado.

TERMINAL · PowerShell

```
1 git --version
```

2

Configurar la identidad

Cada estudiante utiliza su propio nombre y correo de colaboración.

TERMINAL · PowerShell

```
1 git config --global user.name "Tu nombre"
2 git config --global user.email "correo@ejemplo.com"
```

3

Crear el archivo .gitignore

Evita publicar el entorno virtual, archivos temporales y secretos.

TERMINAL · PowerShell

```
1 web/
2 __pycache__/
3 *.pyc
4 db.sqlite3
5 .env
```

4

Inicializar Git

Solo la coordinación realiza este paso si todavía no existe un repositorio.

TERMINAL · PowerShell

```
1 git init
2 git add .
3 git commit -m "Proyecto Django inicial"
```



Práctica 4.

GitHub y trabajo colaborativo

5

Crear el repositorio en GitHub

La coordinación crea un repositorio vacío y conecta su dirección al proyecto local.

TERMINAL · PowerShell

```
1 git branch -M main
2 git remote add origin URL_DEL_REPOSITORIO
3 git push -u origin main
```

6

Clonar en otro equipo

Cada integrante obtiene una copia del proyecto y prepara su propio entorno virtual.

TERMINAL · PowerShell

```
1 git clone URL_DEL_REPOSITORIO
2 cd notho-digital
3 py -m venv web
4 python -m pip install django
```

7

Crear una rama

Actualiza main antes de crear una rama nueva para una tarea concreta.

TERMINAL · PowerShell

```
1 git switch main
2 git pull origin main
3 git switch -c equipo-lengua/vocabulario
```

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.



Práctica 4.

GitHub y trabajo colaborativo

8

Registrar los cambios

Haz un cambio funcional, revisa los archivos y publica tu rama.

TERMINAL · PowerShell

```
1 git status
2 git add .
3 git commit -m "Agregar vista de vocabulario"
4 git push -u origin equipo-lengua/vocabulario
```

9

Abrir un Pull Request

Desde GitHub solicita que tu rama se integre a main. Explica qué cambió y cómo lo probaste.

10

Revisar e integrar

Otra persona revisa los cambios. Ejecuten la comprobación de Django antes de integrar.

TERMINAL · PowerShell

```
1 python manage.py check
```

11

Actualizar y entregar

Después de fusionar, descarga la versión nueva. Entrega el enlace del Pull Request y capturas.

TERMINAL · PowerShell

```
1 git switch main
2 git pull origin main
```

✓ COMPROBACIÓN / EVIDENCIA

Guarda una captura y registra lo aprendido en tu portafolio.

**Práctica 5.**

Vistas dinámicas y contenido por equipo

**OBJETIVO**

Crear vistas y páginas HTML para las cinco aplicaciones de Django, usando la plantilla compartida.

PASOS**1****Identifica la estructura compartida**

Trabajaremos con las apps core, casas, turismo, lengua y videojuego. Cada equipo creará su propia vista y plantilla, pero todos heredarán base.html.

EXPLORADOR DE VS CODE

```
config/  core/  casas/  turismo/  
lengua/  videojuego/  templates/  
static/  manage.py
```

2**Crea una vista en lengua/views.py**

La función recibe la solicitud y devuelve una página HTML. El nombre de la plantilla será lengua/inicio.html.

```
lengua > views.py
```

```
from django.shortcuts import render  
  
def inicio(request):  
    return render(request, "lengua/inicio.html")
```

3**Declara la ruta de la aplicación**

Crea lengua/urls.py y conecta la vista. El nombre de ruta permitirá enlazarla desde el menú.

COMPRUEBA

Cada aplicación debe tener su propio archivo urls.py y una vista inicial.

```
urlpatterns = [  
    path("", views.inicio, name="inicio_lengua"),  
]
```



Práctica 5.

Conectar vistas y plantillas

4

Conecta las rutas en config/urls.py

En el archivo principal incluye las rutas de las cinco apps. Conserva las rutas que ya funcionan; evita duplicar prefijos.

config > urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path("admin/", admin.site.urls),
    path("", include("core.urls")),
    path("casas/", include("casas.urls")),
    path("turismo/", include("turismo.urls")),
    path("lengua/", include("lengua.urls")),
    path("videojuego/", include("videojuego.urls")),
]
```

5

Crea la plantilla del equipo

Dentro de templates crea la carpeta lengua y el archivo inicio.html. La etiqueta extends reutiliza el menú y los estilos de base.html.

templates > lengua > inicio.html

```
{% extends "base.html" %}
{% block content %}
<section class="contenido">
  <h1>Lengua ñötho</h1>
  <p>Aprendamos palabras de nuestra comunidad.</p>
</section>
```

ATENCIÓN

Si aparece TemplateDoesNotExist, verifica la ruta de la carpeta templates y la configuración TEMPLATES en settings.py.

**Práctica 5.**

Verificación y trabajo en equipo

6**Repite el patrón para las otras apps**

Cada equipo cambia el nombre de la carpeta y la vista: casas/inicio.html, turismo/inicio.html y videojuego/inicio.html. Core conserva la portada.

7**Ejecuta Django y visita las rutas**

Desde la terminal, con el entorno web activo, inicia el servidor. Comprueba que cada URL abre su página sin errores.

TERMINAL DE VS CODE

```
python manage.py check
python manage.py runserver

http://127.0.0.1:8000/lengua/
```

127.0.0.1:8000/lengua/

Ñötho Digital[Inicio](#) [Casas](#) [Turismo](#) [Lengua](#)**Palabras en ñötho****Bienvenido / Ejemplo de vocabulario**

Contenido de muestra; validar con la instructora.

8**AUTOEVALUACIÓN**

☐ Mi app tiene una vista y una URL funcional. ☐ La página hereda base.html. ☐ Verifiqué las rutas. ☐ Entregué evidencia y cambios en GitHub.