# **@ Curso Práctico: Desarrollo Web Profesional con Django** y Python

"De cero a aplicación funcional: construye tu primer sistema web con Django en Windows"

#### @ Perfil del Estudiante

- Nivel: Principiante
- Escolaridad: Preparatoria (sin experiencia previa)
- Sistema Operativo: Windows
- Objetivo: Crear aplicaciones web profesionales con Django.

#### **K** Tecnologías y Herramientas

- Lenguaje: Python (última versión)
- Framework: Django
- Frontend: HTML5, CSS3, JavaScript, Bootstrap 5
- Editor: Visual Studio Code
- Base de datos: SQLite (integrada en Django)

#### Software Requerido

- **Python** (3.11 o superior)
- Visual Studio Code con extensiones:
- o Python
- o Pylance
- o Django
- o HTML CSS Support
- o SQLite Viewer
- o Bracket Pair Colorizer
- o Live Server (opcional)

#### Estructura del Proyecto

Organiza tus proyectos en una carpeta principal:

C:\Curso\_Django\

### N Procedimiento Paso a Paso

- 1. Crear carpeta de trabajo: Curso\_Django
- 2. Crear subcarpeta: practica\_1
- 3. Abrir VS Code desde practica\_1
- 4. Verificar Python y pip: python --version, pip --version
- 5. Crear entorno virtual: python -m venv .venv
- 6. **Activar entorno**: .venv\Scripts\activate
- 7. **Seleccionar intérprete** en VS Code (Ctrl+Shift+P → "Python: Select Interpreter")
- 8. **Instalar Django**: pip install django y verificar con django-admin --version
- 9. Crear proyecto: django-admin startproject backend\_empleado .
- 10. **Ejecutar servidor**: python manage.py runserver
- 11. Ver página web: http://127.0.0.1:8000

Curso Django con Python - Guía Práctica 12. Detener servidor: Ctrl + C 13. **Realizar migración**: python manage.py migrate 14. Hacer migraciones: python manage.py makemigrations (para preparar cambios) 15. Crear superusuario: python manage.py createsuperuser (usuario: admin, contraseña: 12345678) 16. Acceder al admin: /admin 17. **Detener servidor** nuevamente 18. Crear app: python manage.py startapp app\_empleado 19. **Crear urls.py** en app\_empleado 20. Crear vista ver\_pagina con HttpResponse 21. Registrar app en INSTALLED\_APPS de settings.py 22. Incluir rutas en backend\_empleado/urls.py con include() 23. **Ejecutar servidor** y ver ver\_pagina 24. Inspeccionar página con herramientas del navegador (F12) 25. Crear modelo Empleado con nombre y edad, \_\_str\_\_ y Meta 26. Registrar modelo en admin.py 27. Hacer migraciones: makemigrations 28. Aplicar migraciones: migrate 29. Usar visor de SQLite (extensión SQLite Viewer en VS Code) 30. Acceder al admin y agregar empleados 31. Detener servidor 32. Crear vista listar empleados con render 33. Crear carpeta templates en app\_empleado 34. Crear listar\_empleados.html 35. Diseñar tabla con botones: 🥜 Editar y 🔟 Borrar (usando Bootstrap) 36. Actualizar urls.py de la app 37. **Ejecutar servidor** y ver listado 38. Detener servidor  ${\tt 39.} \ \textbf{Crear funciones CRUD}: \ \texttt{crear\_empleado} \ , \ \texttt{actualizar\_empleado} \ , \ \texttt{borrar\_empleado}$ 40. Crear archivos HTML: crear\_empleado.html, editar\_empleado.html, borrar\_empleado.html

## Restricciones y Buenas Prácticas

41. Actualizar urls.py con todas las rutas42. Ejecutar servidor y probar todo el sistema

- X No usar forms.py : formularios manuales en vistas.
- X No usar base.html : cada plantilla es independiente.
- Comentarios en código: explicaciones breves en Python y HTML.
- **Estructura completa**: carpetas y archivos bien organizados.
- **Proyecto atractivo**: diseño con Bootstrap 5, colores, iconos y estilo moderno.

#### Resultado Esperado

Al finalizar, tendrás una **aplicación web funcional** para gestionar empleados con operaciones CRUD completas, listo para incluir en tu portafolio.

© 2025 Curso Django con Python | Guía educativa para estudiantes de preparatoria

P Consejo: Usa este documento como hoja de ruta. ¡Sigue cada paso y celebra tus avances!