

Curso: **Django con Python** para implementar aplicaciones web profesionales.

Sistema operativo: **windows**.

Nivel: **principiante**.

Escolaridad: **preparatoria**:

Tecnologías: **Python, Django, html, css, js, bootstrap**.

Software instalado: **python última versión, vs code**,

Extensiones de vs code: **python, Django, html,cs**, más lo que recomiendes.

Visores de base de datos: **lo que recomiendes**.

Procedimiento paso a paso:

- 1 Crear carpeta de trabajo "**Curso_Django**" donde se guardarán los proyectos.
- 2 Crear subcarpeta "**practica_1**" para el primer proyecto
- 3 Abrir vs code desde **practica_1**
- 4 Abrir nueva **terminal de vs code** y verificar versión de python y pip.
- 5 En la terminal **crear el entorno virtual** .venv
- 6 **Activar** entorno virtual.
- 7 Seleccionar **intérprete de python**.
- 8 **Instalar Django** y verificar versión de Django..
- 9 Crear el proyecto "**backend_empleado**".
- 10 Ejecutar **servidor integrado de Django**.
- 11 Ver página web.
- 12 Detener servidor Django..
- 13 Realizar **migración**.
- 14 Hacer **migraciones**.
- 15 crear **superusuario** indicando las opciones que conlleva.
- 16 Ejecutar servidor y acceder al **panel de administración de Django**.
- 17 Detener el servidor.
- 18 Crear aplicación "**app_empleado**".
- 19 En app_empleado **crear el archivo** [urls.py](#)
- 20 En app_empleado en [views.py](#) crear función **ver_pagina** utilizando httpresponse.
- 21 En backend_empleado de [setting.py](#) agregar la **app_empleado**.
- 22 En backend_empleado **incluir la ruta** para app_empleado.
- 23 Ejecutar servidor para observar ver_pagina.
- 24 **Inspeccionar** página web.
- 25 En app_empleado [models.py](#) crear la clase "**Empleado**" con los campos o atributos "Nombre" y "edad", con el constructor y meta.
- 26 **Registrar** el modelo "Empleado" en [admin.py](#) de app_empleado
- 27 Hacer migraciones.
- 28 Realizar migrate.
- 29 Utilizar visor de base de datos para sqlite
- 30 Ejecutar servidor y acceder al panel de administración de Django.
- 31 Detener servidor.
- 32 En app_empleado en [views.py](#) crear función **listar empleados** utilizando render.
- 33 Crear carpeta "**templates**" dentro de app_empleado.
- 34 Dentro de templates crear el archivo **listar_empleados.html**
- 35 crear el código simple para listar empleados en una tabla en listar_empleados.html con los botones de acción (editar y borrar)..
- 36 actualizar [urls.py](#) de app_empleado.

- 37 Ejecutar servidor para ver listar empleados.
- 38 detener servidor.
- 39 En app_empleado en [views.py](#) crear las demás funciones para realizar las operaciones de un crud básico (crear_empleado, actualizar_empleado y borrar_empleado).
- 40 crear los archivos correspondientes de html dentro de templates de acuerdo a las [views.py](#) del punto 39.
- 41 Actualizar la [urls.py](#) de En app_empleado
- 42 Ejecutar servidor para ver el proyecto totalmente funcionando.
- 43 No utilizar [forms.py](#)
- 44 No utilizar archivo base.html
- 45 Incluir comentarios y explicaciones breves en el código de python y html.
- 46 crear la estructura de carpetas y archivos del proyecto que esté completa.
- 47 proyecto totalmente atractivo para los estudiante.