

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2025/05/03</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Programacion Web 2				
TÍTULO DE LA PRÁCTICA:	Git de Django 4 y 5				
NÚMERO DE PRÁCTICA:	8	AÑO LECTIVO:	2025	NRO. SEMESTRE:	1
FECHA DE PRESENTACIÓN	28/06/2025	Repositorio	https://github.com/JesusFSP/Curso-PWeb2.git		
INTEGRANTE (s): Silva Pino Jesus Francisco				NOTA:	
DOCENTE(s): CARLO JOSE LUIS CORRALES DELGADO					

SOLUCIÓN Y RESULTADOS
<p>I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</p> <h2>1. Introducción</h2> <p>En este informe se describen los pasos realizados para implementar una aplicación web básica en Django, utilizando vistas basadas en clases (CBV) para gestionar un CRUD (Crear, Leer, Actualizar, Borrar) de personas. Este proyecto forma parte de la práctica del curso Programación Web 2, aplicando buenas prácticas de estructura de proyecto, plantillas y manejo de archivos estáticos.</p> <h2>2. Objetivos</h2> <ul style="list-style-type: none"> - Construir un proyecto Django organizado. - Implementar modelos, vistas y rutas para operar un CRUD.

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p align="center">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2025/05/03</p>	<p>Código: GUIA-PRLE-001</p>	<p align="right">Página: 2</p>

- Usar vistas genéricas basadas en clases.
- Crear plantillas que extienden una base común.
- Configurar archivos estáticos para aplicar estilos CSS.
- Validar la funcionalidad completa de la aplicación.

3. Herramientas

- Lenguaje: Python 3.11
- Framework: Django 5.2.3
- Entorno: Entorno virtual (venv)
- Editor: VS Code (o similar)
- Sistema operativo: Linux (Ubuntu)

4. Desarrollo

4.1 Creación del entorno y del proyecto

Instalación de Django:

```
python -m venv venv
source venv/bin/activate
pip install django
```

Creación del proyecto y la aplicación:

```
django-admin startproject listaContactos .
python manage.py startapp personas
```

4.2 Definición del modelo

Se definió el modelo Persona en personas/models.py:

```
class Persona(models.Model):
    nombres = models.CharField(max_length=100)
    apellidos = models.CharField(max_length=100)
    edad = models.IntegerField()
    donador = models.BooleanField(default=False)
```

4.3 Configuración de la base de datos

Se aplicaron migraciones:

```
python manage.py makemigrations
python manage.py migrate
```

4.4 Creación de las vistas basadas en clases

En personas/views.py se implementaron:

- ListView: para listar personas con filtro por edad.
- DetailView: para ver detalles (opcional).
- CreateView: para crear nuevas personas.
- UpdateView: para modificar datos.
- DeleteView: para confirmar y borrar.

4.5 Configuración de URLs

personas/urls.py:

```
path("", PersonaListView.as_view(), name='persona-list')
path('create/', PersonaCreateView.as_view(), name='persona-create')
path('<int:pk>/update/', PersonaUpdateView.as_view(), name='persona-update')
path('<int:pk>/delete/', PersonaDeleteView.as_view(), name='persona-delete')
```

listaContactos/urls.py:

```
path('personas/', include('personas.urls'))
```

4.6 Creación de plantillas

- base.html: plantilla principal con título y hoja de estilos.
- persona_list.html: muestra título, lista de personas y botones de crear/modificar/borrar.
- persona_form.html: formulario para crear/modificar.
- persona_confirm_delete.html: formulario para confirmar borrado.

Mi sitio

Lista de personas

[Crear Nueva Persona](#)

- [jesus Silva](#) [Modificar](#) [Borrar](#)
- [Kasandra Remache](#) [Modificar](#) [Borrar](#)
- [juan Peña](#) [Modificar](#) [Borrar](#)

4.7 Configuración de archivos estáticos

Se creó la carpeta personas/static/personas/ con style.css.

Se enlazó en base.html y se configuró en settings.py con STATIC_URL.

4.8 Pruebas de funcionamiento

Se verificó /personas/ para mostrar solo:

Mi sitio

Lista de personas

Listado de registros

Botones de Crear, Modificar y Borrar.



Mi sitio

Crear nueva persona

Nombres:

Apellidos:

Edad:

Donador: ☐

Modificar

Mi sitio

Modificar persona

Nombres:

Apellidos:

Edad:

Donador: ☒

Eliminar

Mi sitio

Lista de personas

Crear Nueva Persona

- jesus Silva
- Kasandra Remache
- juan Peña
- Pablo Picasso

Mi sitio

¿Estás seguro de que quieres borrar "Pablo Picasso"?

Sí, borrar

Cancelar

Mi sitio

Lista de personas

Crear Nueva Persona

•

jesus Silva

Modificar

Borrar

•

Kasandra Remache

Modificar

Borrar

•

juan Peña

Modificar

Borrar

Commits totales

PROBLEMAS 74 SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

git - Django4 + - + - + -

Contando objetos: 100% (63/63), listo.
* 7ccd498 (HEAD -> main, origin/main, origin/HEAD) Se corrigio un error al borrar las personas y se agrego los botones a las personas para modificarlas o borrarlas
* ef998f1 Se implemento un archivo estatico que es el style.css para mejorar la vista de todo el proyecto
* 510801e Se corrigio los errores al crear una nueva persona, creando una carpeta template global con el archivo base.html
* 8717de9 Modificando personas/view.py para implementar updateview y deleteview y corrigiendo errores
* 2f042d5 Modificando personas/view.py para implementar createview
* 4044193 Modificando personas/view.py para modificar la funvion personasShowObject que ya no es necesaria por la class PersonaDetailView
* 76c27e7 Modificando personas/view.py para agregar un queryset para filtrar a las personas menores de 40 años
* 3b1b172 Modificando la plantilla personas_list.html a persona_list.html
* 9985da0 Modificando personas/urls.py se actualiza la ruta raiz para que apunte a la nueva vista basada en clases
* 7cfe7ff Modificando personas/view.py la funcion personasListView basada en funciones para que este basada en clases
* 371a347 Agregando y modificando las urls para darle mas seguridad al proyecto
* 00a9f2c Creando la plantilla personas personas_delete.html para borrar objetos persona
* 1f09941 Creando la plantilla personas personas_detail.html y el manejo de objetos no existentes
* 62da98f Creando la plantilla personas_list.html para listar todas las personas
* b188861 Corrigiendo el archivo views.py de personas para que guarde los datos ingresados en el formulario
* 32eddca Se modifica la vista para manejar datos POST, validarlos y guardarlos
* 3e27b16 Se anadio la ruta de la nueva vista y se prueba que se visualice el formulario
* c70eb5a Se creo la vista que maneja el formulario
...skipping...
* 7ccd498 (HEAD -> main, origin/main, origin/HEAD) Se corrigio un error al borrar las personas y se agrego los botones a las personas para modificarlas o borrarlas
* ef998f1 Se implemento un archivo estatico que es el style.css para mejorar la vista de todo el proyecto
* 510801e Se corrigio los errores al crear una nueva persona, creando una carpeta template global con el archivo base.html
* 8717de9 Modificando personas/view.py para implementar updateview y deleteview y corrigiendo errores
* 2f042d5 Modificando personas/view.py para implementar createview
* 4044193 Modificando personas/view.py para modificar la funvion personasShowObject que ya no es necesaria por la class PersonaDetailView
* 76c27e7 Modificando personas/view.py para agregar un queryset para filtrar a las personas menores de 40 años
* 3b1b172 Modificando la plantilla personas_list.html a persona_list.html
* 9985da0 Modificando personas/urls.py se actualiza la ruta raiz para que apunte a la nueva vista basada en clases
* 7cfe7ff Modificando personas/view.py la funcion personasListView basada en funciones para que este basada en clases
* 371a347 Agregando y modificando las urls para darle mas seguridad al proyecto
* 00a9f2c Creando la plantilla personas personas_delete.html para borrar objetos persona
* 1f09941 Creando la plantilla personas personas_detail.html y el manejo de objetos no existentes
* 62da98f Creando la plantilla personas_list.html para listar todas las personas
* b188861 Corrigiendo el archivo views.py de personas para que guarde los datos ingresados en el formulario
* 32eddca Se modifica la vista para manejar datos POST, validarlos y guardarlos
* 3e27b16 Se anadio la ruta de la nueva vista y se prueba que se visualice el formulario
* c70eb5a Se creo la vista que maneja el formulario
* b9977ac creando el archivo forms.py y agregando el formulario con los campos necesarios
* 4daa926 Corrigiendo el lab06 y creando el proyecto para git de django 4 y 5

<p>II. CONCLUSIONES</p> <p>El proyecto implementa de forma correcta la estructura de un CRUD en Django usando vistas genéricas, plantillas base reutilizables y archivos estáticos para dar estilo. Se asegura una arquitectura limpia y escalable para mantener y extender el sistema.</p>

RETROALIMENTACIÓN GENERAL

REFERENCIAS Y BIBLIOGRAFÍA