

Informe de Django

Tema: Django

Nota

Estudiante	Escuela	Asignatura
Victor Gonzalo Maldonado Wilca vmaldonadov@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1702122

Tarea	Tema	Duración
05	Django	2 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 29 de mayo de 2024	Al 1 de junio de 2024

1. Introducción

Django es un framework de desarrollo web en Python que facilita la creación rápida de aplicaciones web seguras y escalables. Ofrece un conjunto de herramientas integradas como ORM, administrador de Django y enrutamiento de URLs, lo que simplifica el desarrollo y mejora la seguridad de las aplicaciones web.

2. Objetivos

- Entender y configurar el entorno de trabajo para Django
- Crear un primer proyecto en Django
- Crear apps e integrarlas al framework

3. Tarea

En esta tarea deberá seguir las diapositivas de DJango01, dentro de un proyecto git local, deberá hacer un commit por cada paso, si desea puede usar branch. La entrega de la tarea consistirá de una captura de pantalla (png o jpg) del siguiente comando:

```
git log --graph --pretty=oneline --abbrev-commit --all
```

Cada commit debe ser realizado con un mensaje descriptivo del paso de la diapositiva que estuvo siguiendo.

La calificación tendrá en cuenta los siguientes aspectos:

- Presentó la tarea 10 puntos
- Los mensajes de los commit son claros y descriptivos 10 puntos
- Realizó Branch +2 puntos extra

4. Entregables

- Informe hecho en Latex
- Captura de pantalla (png o jpg) del comando:

```
git log --graph --pretty=oneline --abbrev-commit --all
```

- EXTRA: URL del repositorio en GitHub

5. Equipos, materiales y temas utilizados

- Python
- Pip
- Django
- Entornos virtuales en python
- Proyectos de Django
- Aplicaciones en Django
- Modelos, Vistas, Templates y Formularios en Django

6. Captura de pantalla (png o jpg)

- Comando usado:

```
git log --graph --pretty=oneline --abbrev-commit --all
```

- Captura:

```
Usuario@DESKTOP-G0S7RR6 MINGW64 /D/UNSA/pw2/teoria/Django/src (master)
$ git log --graph --pretty=oneline --abbrev-commit --all
* 3e636c5 (HEAD -> master, origin/master) Finalización del proyecto, cambiando Read
me.md, colocando en negrita el comando git log ...
* 5b72c3c Guardando cambios en la base de datos
* 29a586f Juntando rama app con master
| \
| * 2547157 (origin/app, app) Registrar el modelo Persona en el panel de administra
ción (admin.py)
* | de066dc (origin/usuario, usuario) Creando super usuario usando comando: python
manage.py createsuperuser
| /
* 7250655 Actualizar los cambios hechos e informar al framework
* 6de2398 Agregando app en la configuración del proyecto
* f0a10d8 Cambios hechos en models.py, implementación de la clase Persona
* 88a650e Creando aplicación personas
* ca9a3b2 (origin/migraciones, migraciones) Se uso comando python manage.py migrate
* 9da616b (origin/listaContactos, listaContactos) Configuración del lenguaje y time
-zone
* 880fe30 Creación del proyecto
* 408596e Agregando .gitignore
* 86a2c68 Modificando README.md
* d4a51f0 primer commit
```

Figura 1: Captura del comando git

7. URL de Repositorio Github

- URL: Repositorio GitHub
- <https://github.com/Victor-Gonzalo-Maldonado-Vilca/Django.git>

8. Metodología

8.1. Creación del entorno de Trabajo

8.1.1. Carpeta de trabajo

Listing 1: Creación del Directorio

```
mkdir Django && cd Django
```

8.1.2. Creación del Entorno Virtual

Listing 2: Creación Entorno virtual

```
virtualenv -p python3 .
```

8.1.3. Activar entorno Virtual

Listing 3: Activar Entorno Virtual

```
Scripts/activate
```

8.1.4. Instalar Django en el entorno Virtual

Listing 4: Instalar Django

```
pip install Django
```

8.1.5. Creación carpeta del proyecto

Listing 5: Carpeta src

```
mkdir src && cd src
```

8.1.6. Inicializar git

Listing 6: Inicializar git

```
git init
```

8.1.7. Crear .gitignore

Seguimos el siguiente Repositorio <https://github.com/django/django/blob/main/.gitignore>

8.2. Creacion del proyecto y apps

8.2.1. Crear Proyecto

Listing 7: Crear proyecto

```
django-admin startproject listaContactos
```

8.2.2. Crear App

Listing 8: Crear App

```
python manage.py startapp personas
```

9. Desarrollo del trabajo

9.1. Configuramos el proyecto en settings.py

9.1.1. Cambiamos el idioma y zona del tiempo

```
104 # Internationalization
105 # https://docs.djangoproject.com/en/5.0/topics/i18n/
106
107 LANGUAGE_CODE = 'es'
108
109 TIME_ZONE = 'America/Lima'
110
111 USE_I18N = True
112
113 USE_TZ = True
114
115
```

Figura 2: Configuración a nuestra región

9.1.2. Migraciones

Uso del comando: python manage.py migrate

```
(Django) D:\UNSA\PW2\Teoria\Django\src>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

Figura 3: migraciones

9.1.3. Agregamos la aplicación en el archivo settings.py

```
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'personas',
41 ]
42
```

Figura 4: Agregar App

9.2. Definir Modelos en models.py en la App personas

9.2.1. Modelo Persona

■ Modelo Persona:

```
1 from django.db import models
2
3 # Create your models here.
4 class Persona(models.Model):
5     nombres = models.TextField()
6     apellidos = models.TextField()
7     edad = models.TextField()
8
```

Figura 5: Modelo Persona

9.3. Actualizar los cambios hechos e informar al framework

9.3.1. Informar al Framework sobre los cambios

Listing 9: Comando

```
python manage.py makemigrations
```

```
(Django) D:\UNSA\PW2\Teoria\Django\src>python manage.py makemigrations
Migrations for 'personas':
  personas\migrations\0001_initial.py
  - Create model Persona
```

Figura 6: Informar al Framework

9.3.2. Actualizar los cambios hechos

Listing 10: Comando

```
python manage.py migrate
```

```
(Django) D:\UNSA\PW2\Teoria\Django\src>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, personas, sessions
Running migrations:
  Applying personas.0001_initial... OK
```

Figura 7: Actualizar

9.4. Creando nuevo usuario

Crear un nuevo usuario con su contraseña y con permisos de administrador

- **Uso del comando:**

```
python manage.py createsuperuser
```

```
(Django) D:\UNSA\PW2\Teoria\Django\src>python manage.py createsuperuser
Nombre de usuario (leave blank to use 'usuario'): Victor
Dirección de correo electrónico: vmaldonadov@unsa.edu.pe
Password:
Password (again):
Superuser created successfully.
```

Figura 8: Crear Usuario

- **Capturas:**

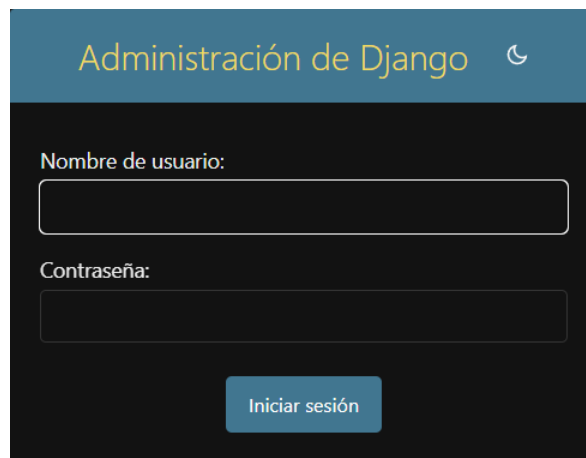


Figura 9: Captura 1

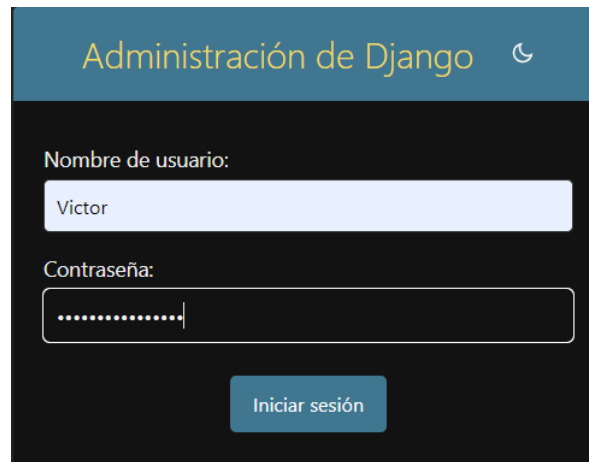


Figura 10: Captura 2

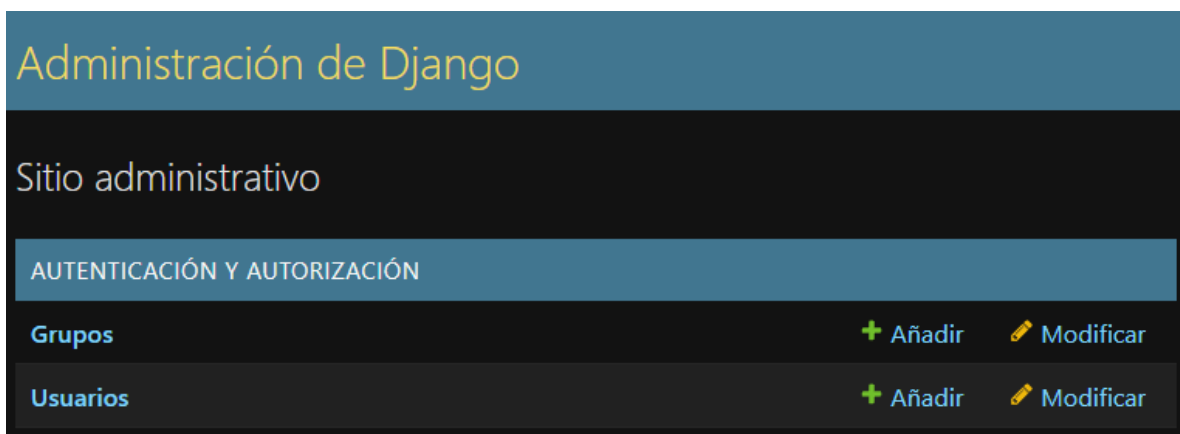


Figura 11: Captura 3

9.5. Registro de Modelos en el Administrador (admin.py)

```
1  from django.contrib import admin
2
3  # Register your models here.
4  from .models import Persona
5
6  admin.site.register(Persona)
7
```

Figura 12: admin.py

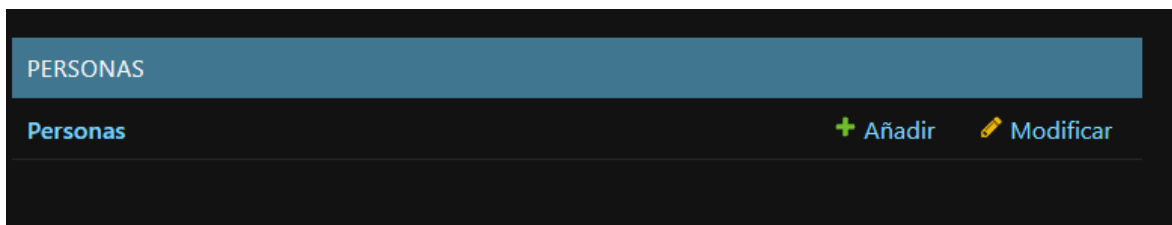


Figura 13: En el Servidor

9.6. Crear nuevo objeto de Persona

- Uso del comando:

```
python manage.py shell
```

```
(Django) D:\UNSA\PW2\Teoria\Django\src>python manage.py shell
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
(InteractiveConsole)
>>> from personas.models import Persona
>>> Persona.objects.all()
<QuerySet []>
>>> Persona.objects.create(nombres="Victor Gonzalo", apellidos="Maldonado Vilca", edad="20")
<Persona: Persona object (1)>
>>> Persona.objects.all()
<QuerySet [<Persona: Persona object (1)>]>
>>> exit()
```

Figura 14: Shell

Modificar persona

Persona object (1)

Nombres:

Victor Gonzalo

Apellidos:

Maldonado Vilca

Edad:

20

Figura 15: En el Servidor

9.7. Iniciar el servidor de desarrollo

Recordemos que para iniciar el servidor se usa el comando:

```
python manage.py runserver
```

django

Ve [la notas de la versión](#) de Django 5.0



¡La instalación funcionó con éxito! ¡Felicitaciones!

Estás viendo esta página porque `DEBUG=True` está en su archivo de configuración y no ha configurado ninguna URL.



Documentación de Django

Temas, referencias, & como hacer



Tutorial: Una aplicación de encuesta

Comienza con Django



Comunidad Django

Conéctate, obtén ayuda o contribuye

Figura 16: Servidor

Ahora colocando `http://127.0.0.1:8000/admin/` ya registrandonos con las credenciales del usuario

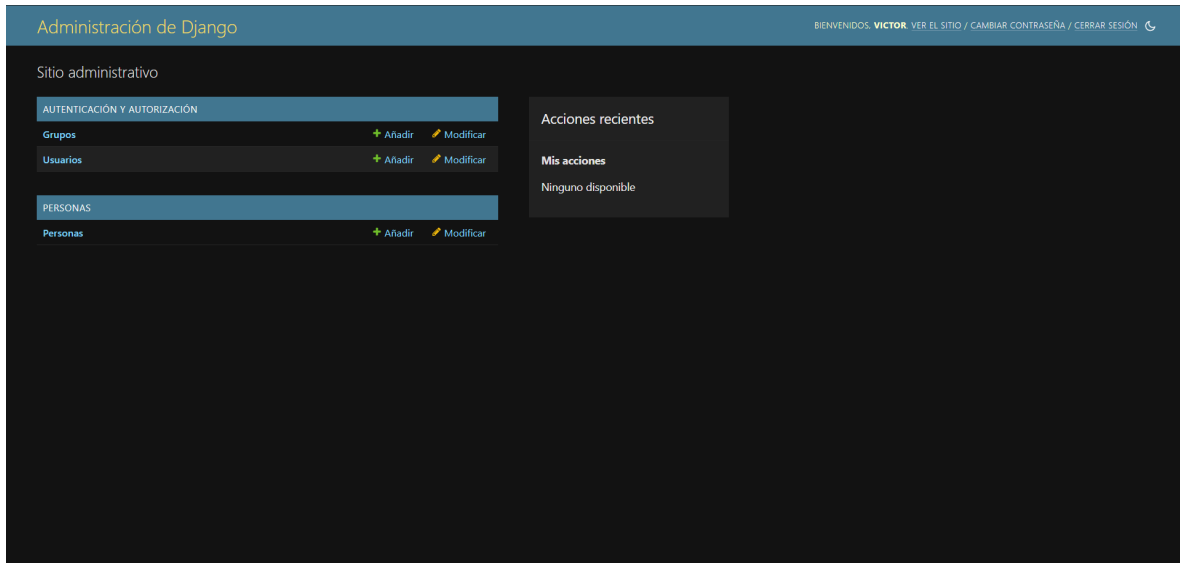


Figura 17: Servidor

9.8. Uso de GitHub

9.8.1. Usuario de GitHub

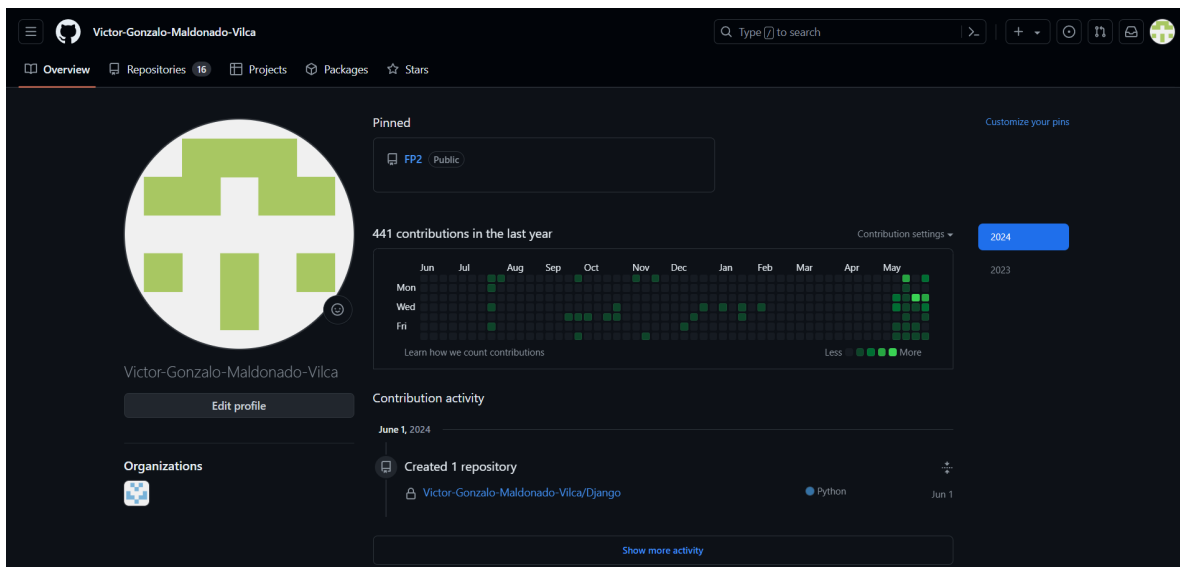


Figura 18: Usuario

9.8.2. Creación de un Nuevo Repositorio

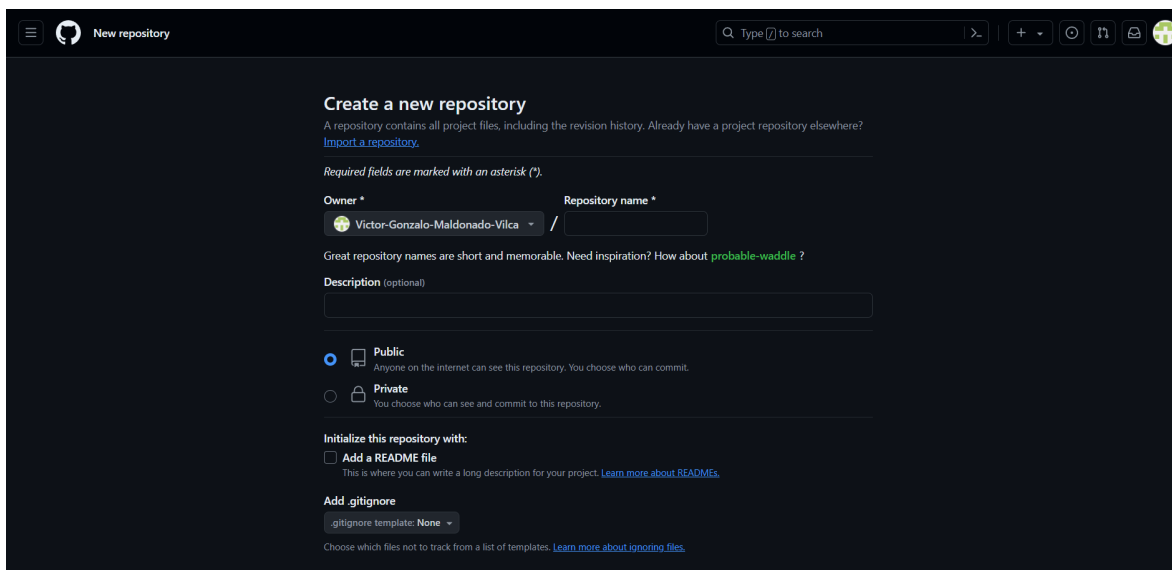


Figura 19: Crear Repositorio

9.8.3. Comandos de Configuración

Listing 11: Comando Iniciales

```
echo "# Auxiliar" >> README.md
git init
git add README.md
git commit -m "primer commit"
git branch -M master
git remote add origin https://github.com/Victor-Gonzalo-Maldonado-Vilca/Django.git
git push -u origin master
```

9.8.4. Implementación de Readme.md

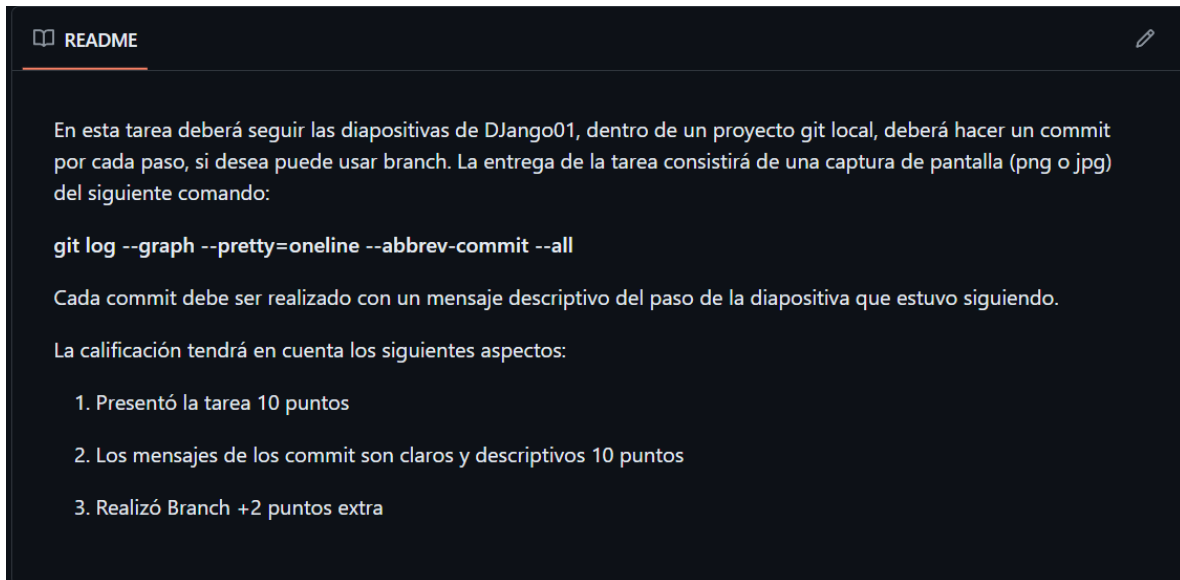


Figura 20: README.md

9.8.5. Registro de cambios en mi código

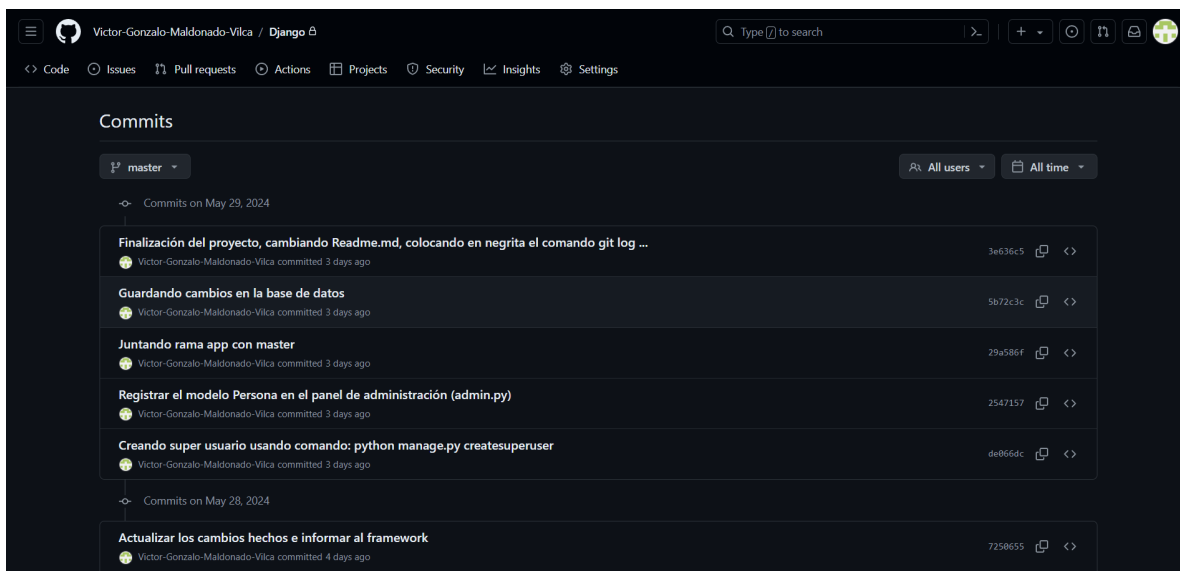


Figura 21: Commits

9.8.6. Repositorio

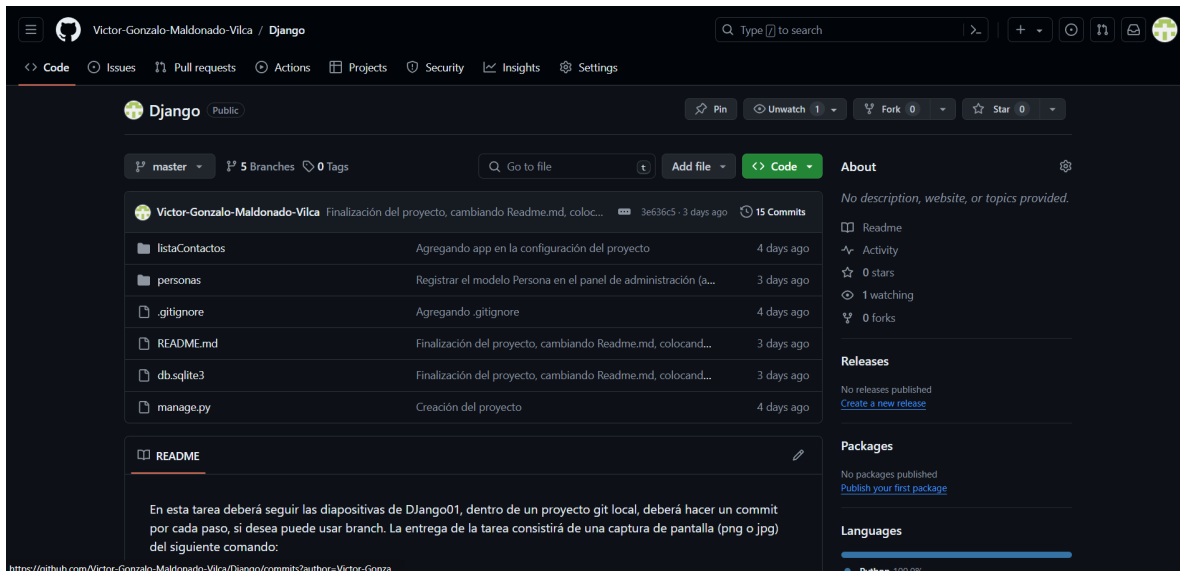


Figura 22: Repositorio

9.8.7. Uso de Ramas

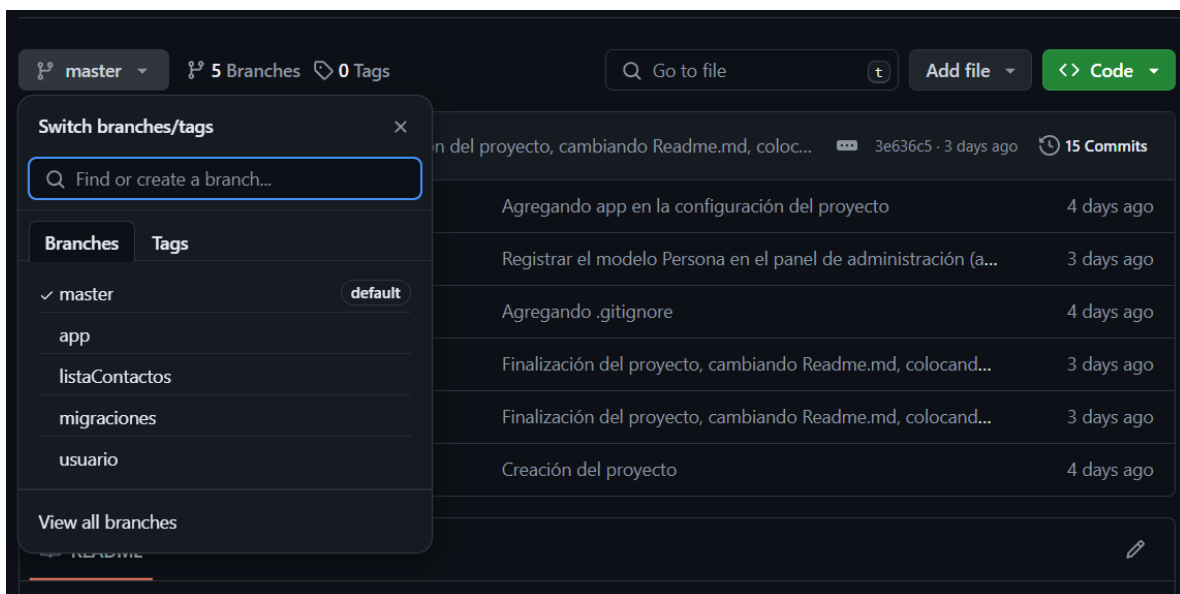


Figura 23: Uso de Ramas

9.8.8. Proyecto compartido con el profesor de github

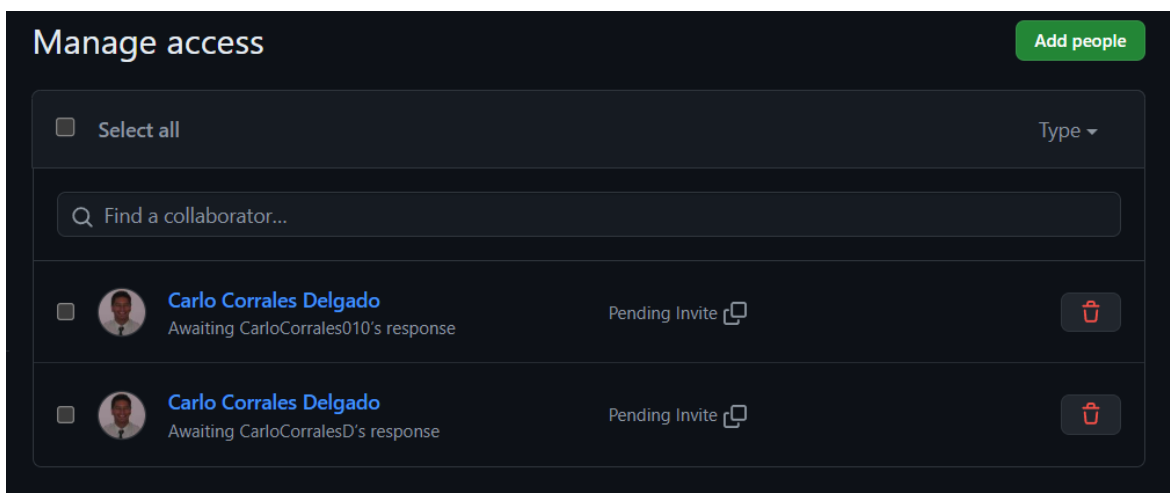


Figura 24: Compartir con el Docente

10. Recomendaciones

- Seguir las mejores prácticas de desarrollo de Django, como la separación de la lógica de negocios en vistas y la creación de plantillas reutilizables.
- Realizar pruebas unitarias y de integración para garantizar la calidad del código, y asegúrate de implementar medidas de seguridad en tu aplicación.

11. Conclusiones

- Django es un framework web potente y versátil que permite desarrollar aplicaciones web de manera rápida y eficiente.
- La arquitectura MVC (Modelo-Vista-Controlador) de Django ayuda a organizar el código de manera estructurada y modular, lo que facilita la mantenibilidad y escalabilidad de las aplicaciones.
- Con un buen entendimiento de Django y siguiendo las mejores prácticas de desarrollo, se pueden crear aplicaciones web robustas y de alto rendimiento.

11.1. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 2: Rúbrica para contenido del Informe y demostración

	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
Total		20		20	

12. Referencias

- <https://docs.djangoproject.com/en/5.0/>
- <https://docs.github.com/es>
- <https://git-scm.com/doc>