

## Laboratorio DIA 4: CREAR PROYECTO CON MODELO Y FORMULARIO Y BD

Crear otro Proyecto **AdmProyectosModelForm** y una aplicación **moduloProyectoModelForm**

Pasos:

1. Abrir CMD
2. Ubicarse dentro del espacio de trabajo **workspace-django-project**

```
Simbolo del sistema
Microsoft Windows [Versión 10.0.19043.1826]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\TERESA>cd ..

C:\Users>cd ..

C:\>d:

D:\>cd workspace-django-project

D:\workspace-django-project>
```

3. Crear el proyecto **AdmProyectosModelForm** ejecutando la siguiente instrucción:

**django-admin startproject AdmProyectosModelForm**

```
D:\workspace-django-project>django-admin startproject AdmProyectosModelForm
D:\workspace-django-project>
```

4. Entrar al proyecto **AdmProyectosModelForm**

```
D:\workspace-django-project>cd AdmProyectosModelForm
D:\workspace-django-project\AdmProyectosModelForm>
```

5. Crear la aplicación **moduloProyectoModelForm** ejecutando la siguiente instrucción:

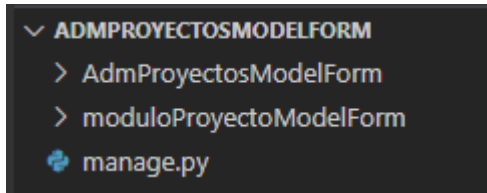
**python manage.py startapp moduloProyectoModelForm**

```
D:\workspace-django-project\AdmProyectosModelForm>python manage.py startapp moduloProyectoModelForm
D:\workspace-django-project\AdmProyectosModelForm>
```

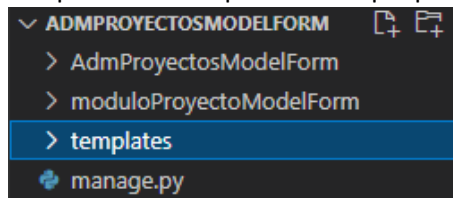
6. Abrir visual code

```
D:\Workspace-django-project\AdmProyectosModelForm>code .  
D:\Workspace-django-project\AdmProyectosModelForm>
```

En la siguiente imagen, se puede visualizar el proyecto **AdmProyectosModelForm** y la aplicación **moduloProyectoModelForm**

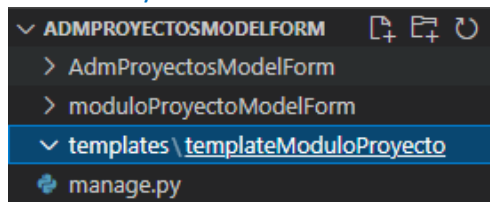


7. En el proyecto **AdmProyectosModelForm**, crear la carpeta **templates** que almacenará los templates de las aplicaciones que posee el proyecto. Revisar la creación en la imagen siguiente.

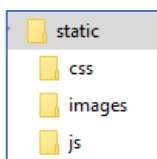


Nota: también se puede crear desde el CMD con la instrucción mkdir

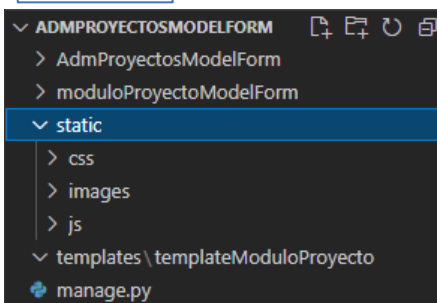
8. Crear la carpeta **templateModuloProyecto** almacenará los templates de la aplicación dentro de la carpeta **templates**. La carpeta **templateModuloProyecto** almacenará los templates de la aplicación **moduloProyectoModelForm**



9. Para almacenar contenido estático crear la siguiente estructura de carpetas dentro del proyecto **AdmEmpleadosForm**



Nota: la carpeta static está a la misma altura de la carpeta templates. Se usan los mismos pasos de creación de carpetas del punto anterior.



**Nota:** para crear las carpetas css, js e images hacer click derecho del mouse sobre la palabra static.

10. Modificar el archivo de configuración `settings.py` del proyecto `AdmProyectosModelForm`. Las modificaciones son:

- a) Incluir la aplicación `moduloProyectoModelForm` al Proyecto `AdmProyectosModelForm`

```
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     'moduloProyectoModelForm'  
41 ]
```

- b) Indicar la ruta de los templates del Proyecto `AdmProyectosModelForm`

Las instrucciones son:

- i. Incluir import os

```
12 import os  
13 from pathlib import Path  
14
```

- ii. Crear la variable `TEMPLATE_DIR` con la ruta de la carpeta templates del Proyecto y agregarla

```
15 # Build paths inside the project like this: BASE_DIR / 'subdir'.  
16 BASE_DIR = Path(__file__).resolve().parent.parent  
17 TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')
```

- iii. Especificar la variable `TEMPLATE_DIR` en el código que se indica en el recuadro amarillo

```
55 TEMPLATES = [  
56     {  
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
58         'DIRS': [],  
59         'APP_DIRS': True,  
60         'OPTIONS': {  
61             'context_processors': [  
62                 'django.template.context_processors.debug',  
63                 'django.template.context_processors.request',  
64                 'django.contrib.auth.context_processors.auth',  
65                 'django.contrib.messages.context_processors.messages',  
66             ],  
67         },  
68     },  
69 ]
```

El resultado es:

```
55 TEMPLATES = [  
56     {  
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
58         'DIRS': [TEMPLATE_DIR],  
59         'APP_DIRS': True,  
60         'OPTIONS': {  
61             'context_processors': [  
62                 'django.template.context_processors.debug',  
63                 'django.template.context_processors.request',  
64                 'django.contrib.auth.context_processors.auth',  
65                 'django.contrib.messages.context_processors.messages',  
66             ],  
67         },  
68     },  
69 ]
```

- c) Revisar y configurar contenido estático
- Revisar la existencia de la variable **STATIC\_URL** con la asignación de la carpeta **static** del Proyecto, sino agregarla

```
116 # Static files (CSS, JavaScript, Images)  
117 # https://docs.djangoproject.com/en/4.1/howto/static-files/  
118  
119 STATIC_URL = 'static/'
```

- ii. Especificar la variable **STATICFILES\_DIRS** en el código como se indica en el recuadro amarillo. En otras palabras, agregar la variable **STATICFILES\_DIRS**

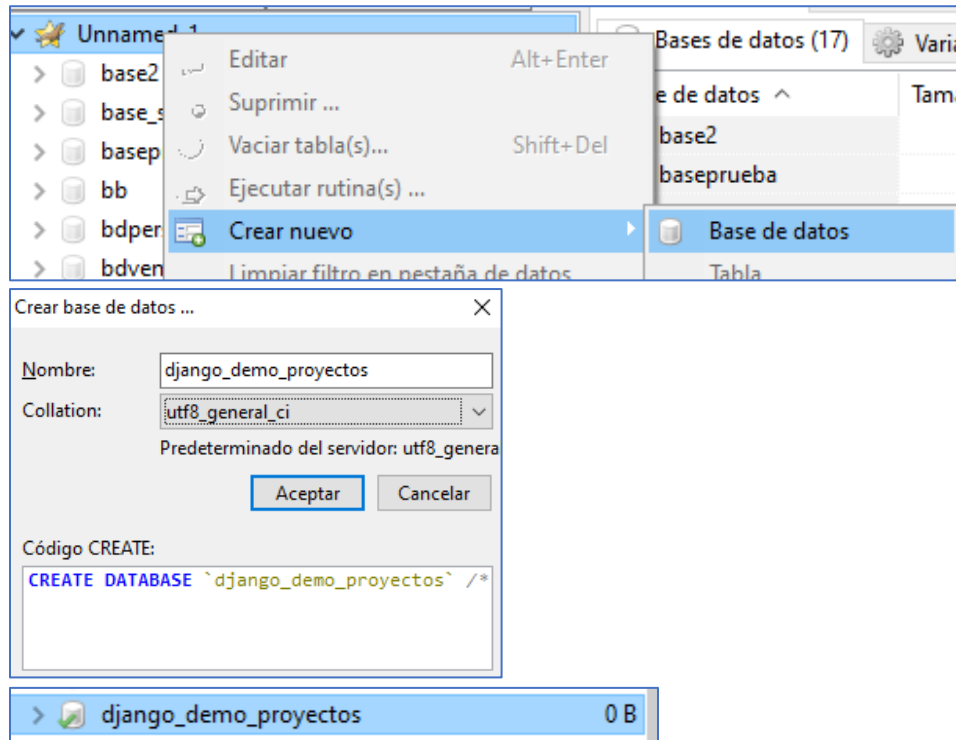
```
116 # Static files (CSS, JavaScript, Images)  
117 # https://docs.djangoproject.com/en/4.1/howto/static-files/  
118  
119 STATIC_URL = 'static/'  
120 STATICFILES_DIRS=[os.path.join(BASE_DIR, 'static')]
```

- d) Configurar que la **consola de administración** sea en español

```
LANGUAGE_CODE = 'es-Es' # 'en-us' en ingles
```

e) Crear y configurar la Base de Datos

1. Crear la Base de Datos **django\_demo\_proyectos** en Maria DB



**Nota:** no cerrar Heidi, dejar la sesión abierta.

2. Si no está instalado Pymysql  
**pip install PyMySQL**
3. Configurar la base de datos en el archivo **settings.py**. En otras palabras, reemplazar el siguiente código que se ilustra en la siguiente imagen

```
77 DATABASES = {
78     'default': {
79         'ENGINE': 'django.db.backends.sqlite3',
80         'NAME': BASE_DIR / 'db.sqlite3',
81     }
82 }
```

Se reemplaza por:

```
#pip install PyMySQL
import pymysql
pymysql.install_as_MySQLdb()

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'django_demo_proyectos',
        'USER': 'root',
        'PASSWORD': 'root1502'
    }
}
```

→ Password: admin  
(LABORATORIO INACAP)

Nota: Recuerda cambiar la password. Esto es, que sea la que tu le asignaste a tu base de datos. En el laboratorio de Inacap, la password es **admin**

4. Iniciar el servidor (opcional, no es necesario este paso. Solo se realiza para validar que no tengamos errores iniciales)

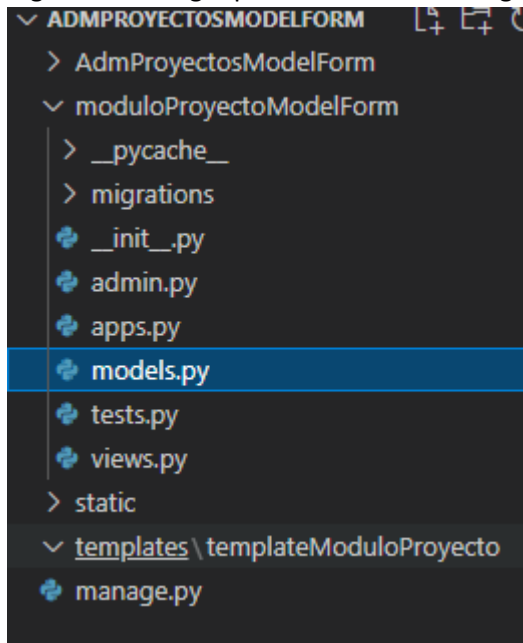
```
PS D:\Workspace-django-project\AdmProyectosModelForm> Python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the
migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
August 17, 2022 - 09:21:26
Django version 4.1, using settings 'AdmProyectosModelForm.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



11. Crear el modelo. Editar el archivo `models.py` de la aplicación `moduloProyectoModelForm` e ingresar el código que se ilustra en la imagen.



```
moduloProyectoModelForm > models.py > Proyecto
3  # Create your models here.
4
5  class Proyecto(models.Model):
6      fechaInicio =models.DateField()
7      fechaTermino = models.DateField()
8      nombre =models.CharField(max_length=50)
9      responsable = models.CharField(max_length=50)
10     prioridad = models.IntegerField()
11
```

**Nota:** Ahora con el modelo se pueden realizar las migraciones.

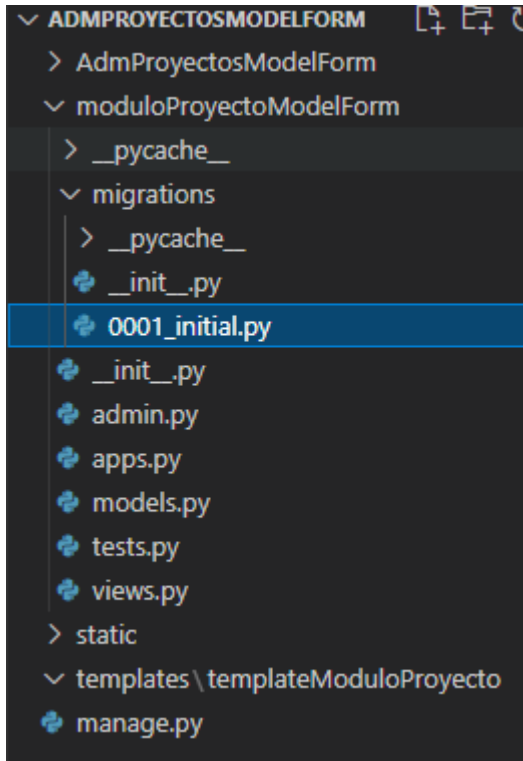
12. Realizar las migraciones ejecutando la siguiente instrucción:

`python manage.py makemigrations`

```
PS D:\Workspace-django-project\AdmProyectosModelForm> python manage.py makemigrations
Migrations for 'moduloProyectoModelForm':
  moduloProyectoModelForm\migrations\0001_initial.py
    - Create model Proyecto
PS D:\Workspace-django-project\AdmProyectosModelForm>
```

Nota: Se genero la estructura de la tabla asociada al modelo Proyecto en el archivo 0001\_initial.py

13. Revisar el archivo generado por Django **moduloProyectoModelForm\migrations\001\_initial.py**



Este archivo le dice al python lo que debe de crear. Le dice como construir la tabla en una base de datos. Agrega el campo ID que no fue indicado en el modelo. Indica que es clave primaria. Cada clase del modelo lo lleva a tabla. Una vez creada todas las clases va a generar todos los archivos de migraciones automáticamente.

Archivo 0001\_initial.py

```
operations = [
    migrations.CreateModel(
        name='Proyecto',
        fields=[
            ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False)),
            ('fechaInicio', models.DateField()),
            ('fechaTermino', models.DateField()),
            ('nombre', models.CharField(max_length=50)),
            ('responsable', models.CharField(max_length=50)),
            ('prioridad', models.IntegerField()),
        ],
    ),
]
```



14. Ejecutar las migraciones ejecutando la siguiente instrucción:

```
python manage.py migrate
```

```
PS D:\Workspace-django-project\AdmProyectosModelForm> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, moduloProyectoModelForm, 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 moduloProyectoModelForm.0001_initial... OK
  Applying sessions.0001_initial... OK
PS D:\Workspace-django-project\AdmProyectosModelForm>
```

Nota: Se crean todas las tablas en la base de datos para la aplicación. Pobra algunas tablas de configuración que crea python.

15. Revisar las tablas creadas en la base de datos:

Para que la base de datos muestre la actualización debes hacer click sobre ella y presionar refrescar o recargar.

| >  django_demo_proyectos 368,0 KiB |       |          |
|------------------------------------|-------|----------|
| Nombre ^                           | Filas | Tamaño   |
| auth_group                         | 0     | 32,0 KiB |
| auth_group_permissions             | 0     | 48,0 KiB |
| auth_permission                    | 28    | 32,0 KiB |
| auth_user                          | 0     | 32,0 KiB |
| auth_user_groups                   | 0     | 48,0 KiB |
| auth_user_user_permissions         | 0     | 48,0 KiB |
| django_admin_log                   | 0     | 48,0 KiB |
| django_content_type                | 7     | 32,0 KiB |
| django_migrations                  | 19    | 16,0 KiB |
| django_session                     | 0     | 16,0 KiB |
| moduloprojectomodelform_proyecto   | 0     | 16,0 KiB |

Se generaron varias tablas de configuración interna de django.

La tabla [moduloprojectomodelform\\_proyecto](#) corresponde a la clase creada en el modelo.

Tabla de proyectos sin datos

| 🔑 id      | fechaInicio | fechaTermino | nombre | responsable | prioridad |
|-----------|-------------|--------------|--------|-------------|-----------|
| sin datos |             |              |        |             |           |

16. Crear un usuario usando: `python manage.py createsuperuser`

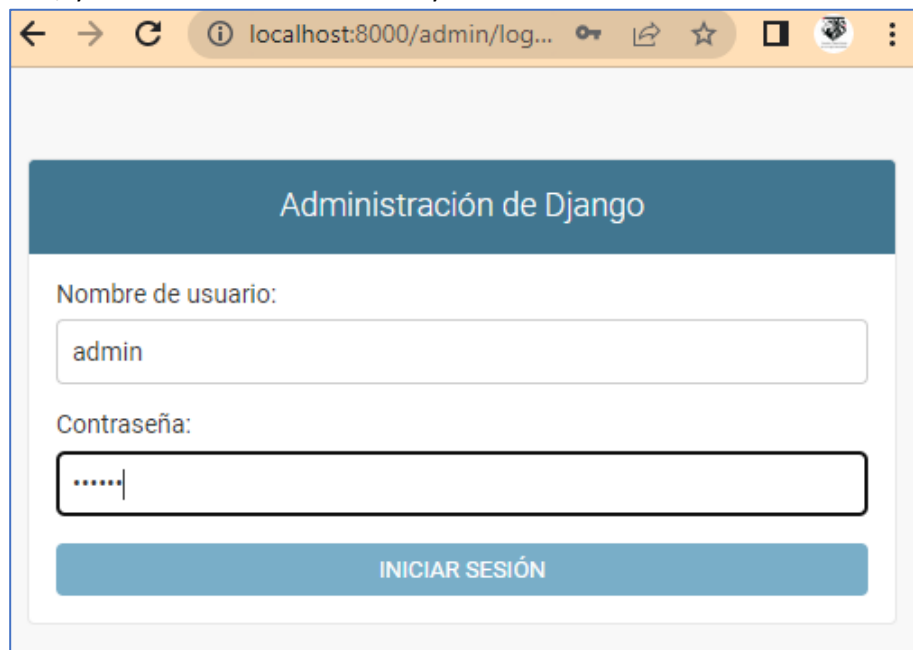
```
PS D:\Workspace-django-project\AdmProyectosModelForm> python manage.py createsuperuser
Nombre de usuario (leave blank to use 'teresa'): admin
Dirección de correo electrónico: tere_tapia@hotmail.com
Password: secret
Password (again):
Esta contraseña es demasiado corta. Debe contener al menos 8 caracteres.
Esta contraseña es demasiado común.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
PS D:\Workspace-django-project\AdmProyectosModelForm>
```

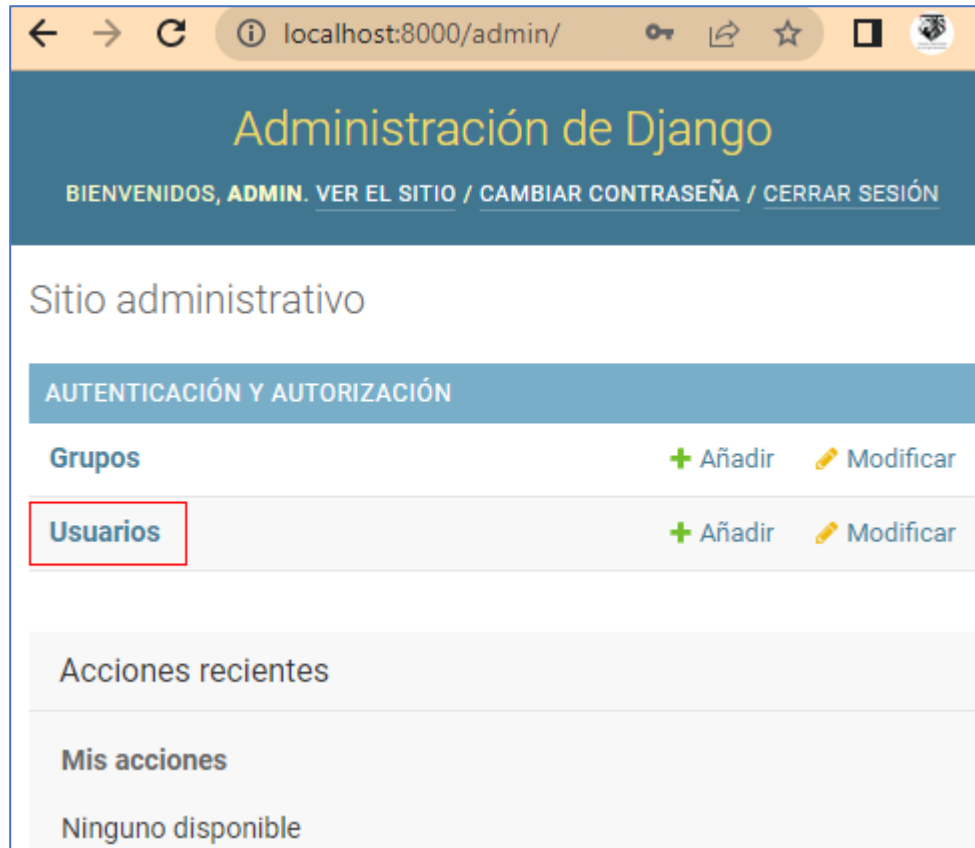
17. Iniciar el servidor

```
PS D:\Workspace-django-project\AdmProyectosModelForm> Python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 17, 2022 - 10:11:42
Django version 4.1, using settings 'AdmProyectosModelForm.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

18. Revisar admin/ y entrar con el usuario admin y la contraseña secret





| <input type="checkbox"/> | NOMBRE DE USUARIO | DIRECCIÓN DE CORREO ELECTRÓNICO | NOMBRE | APELLIDOS | ES STAFF |
|--------------------------|-------------------|---------------------------------|--------|-----------|----------|
| <input type="checkbox"/> | admin             | tere_tapia@hotmail.com          |        |           | ✓        |

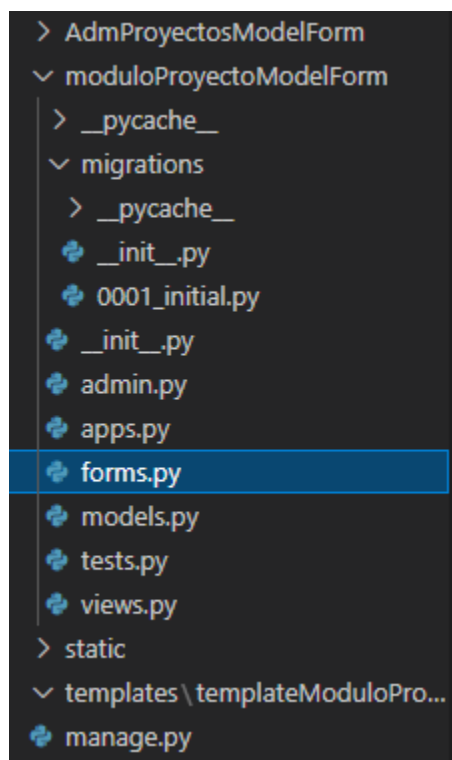
1 usuario

Completamos la información del nombre y apellidos y posteriormente guardamos.

|                                  |   |
|----------------------------------|---|
| Nombre:                          | <input type="text" value="Teresa Jesús"/>           |
| Apellidos:                       | <input type="text" value="Tapia Soto"/>             |
| Dirección de correo electrónico: | <input type="text" value="tere_tapia@hotmail.com"/> |

| Acción: <input type="text"/> |                   | <input type="button" value="Ir"/> |                 | seleccionados 0 de 1 |                                     |
|------------------------------|-------------------|-----------------------------------|-----------------|----------------------|-------------------------------------|
| <input type="checkbox"/>     | NOMBRE DE USUARIO | DIRECCIÓN DE CORREO ELECTRÓNICO   | NOMBRE          | APELLIDOS            | ES STAFF                            |
| <input type="checkbox"/>     | admin             | tere_tapia@hotmail.com            | Teresa<br>Jesús | Tapia<br>Soto        | <input checked="" type="checkbox"/> |
| 1 usuario                    |                   |                                   |                 |                      |                                     |

19. Crear y editar un archivo **forms.py** dentro de la aplicación **moduloProyectoModelForm**.



```

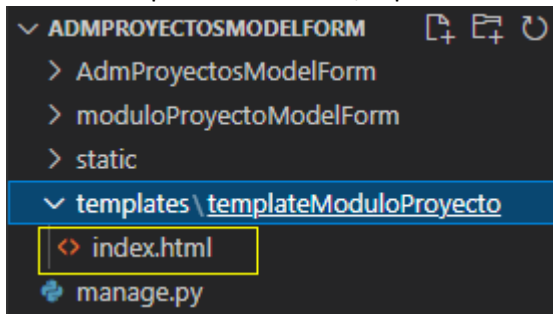
moduloProyectoModelForm > forms.py > ...
2  from django import forms
3
4  from moduloProyectoModelForm.models import Proyecto
5
6  class FormProyecto(forms.ModelForm):
7      class Meta:
8          model = Proyecto
9          fields = '__all__'
10

```

Nota: en el archivo forms.py creas un formulario llamado **FormProyecto**. La clase Meta es usada para cambiar el comportamiento de la clase **ModelForm**. Dentro de **Meta**, se debe especificar el modelo del que provienen sus campos y los campos que desea usar de ese modelo.

Tú puedes elegir sumar todos los campos o solamente unos pocos.

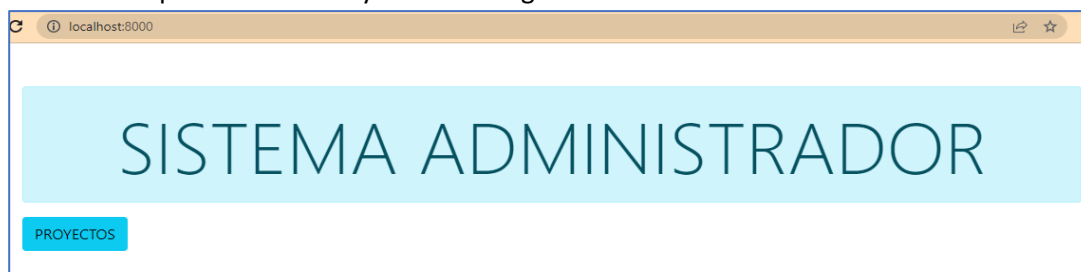
20. Crear el templates `index.html`, específicamente en la carpeta `templates/templateModuloProyecto`



21. Crear la vista para el template `index.html`. Editar el archivo `views.py` de la aplicación `moduloProyectoModelForm` y agregar el siguiente código.

```
moduloProyectoModelForm > views.py > listadoProyectos
1  from django.shortcuts import render
2
3  # Create your views here.
4  def index(request):
5      return render(request, 'templateModuloProyecto\index.html')
6
```

22. Editar el template `index.html` y diseñar la siguiente interfaz



```
<!DOCTYPE html>
{% load static %}
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <meta name="description" content="">
    <title>ADM Proyectos</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
  </head>
  <body class="container mt-5">
    <div class="alert alert-info display-1 text-center"> SISTEMA ADMINISTRADOR </div>

    <a href="#" class="btn btn-info">PROYECTOS</a>
  </body>
</html>
```

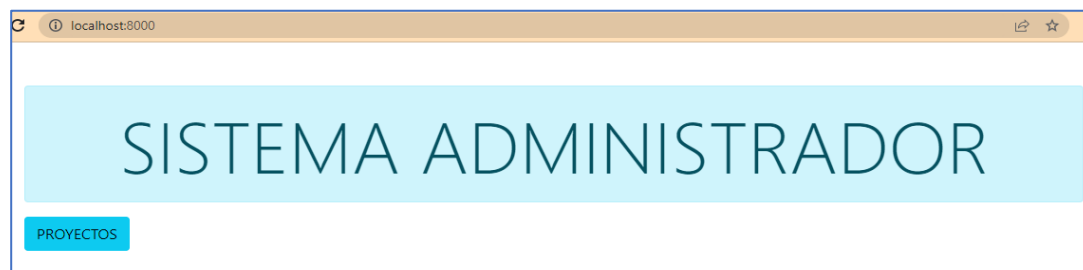
23. Modificar el archivo `urls.py` del proyecto `AdmProyectosModelForm` e incluir el path a la función `index`

```
AdmProyectosModelForm > urls.py > ..
16 from django.contrib import admin
17 from django.urls import path
18
19 from moduloProyectoModelForm import views
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('', views.index)
24 ]
```

24. Iniciar el servidor si lo tiene abajo y luego revisar. Abrir el terminal del visual code

```
PS D:\Workspace-django-project\AdmProyectosModelForm> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 17, 2022 - 16:01:31
Django version 4.1, using settings 'AdmProyectosModelForm.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

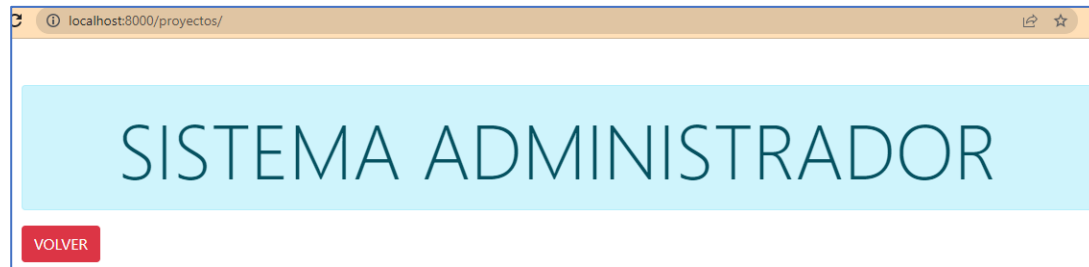


25. Crear el templates `proyectos.html`, específicamente en la carpeta `templates/templateModuloProyecto`

```
▼ ADMPROYECTOSMODELFORM
  > AdmProyectosModelForm
  > moduloProyectoModelForm
  > static
  ▼ templates\templateModuloProyecto
    <> index.html
    <> proyectos.html
  <> manage.py
```

Archivo: [proyectos.html](#)

Nota: Hacer una copia de index.html y hacerles cambios menores inicialmente.



```

1  <!DOCTYPE html>
2  {% load static %}
3  <html>
4      <head>
5          <meta charset="utf-8">
6          <meta name="viewport" content="width=device-width, initial-scale=1.0">
7          <meta http-equiv="X-UA-Compatible" content="ie=edge">
8          <meta name="description" content="">
9          <title>ADM Proyectos</title>
10         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kgtqW17K1Xvg6VCE4pQA" crossorigin="anonymous">
11         <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ktrE-aVq/8KE33RN4u66he6eA2P48QR6he/hjFy4UxwV6j+6v7jF1PtCyM" crossorigin="anonymous"></script>
12     </head>
13     <body class="container mt-5">
14         <div class="alert alert-info display-1 text-center"> SISTEMA ADMINISTRADOR </div>
15
16         <a href="#" class="btn btn-danger">VOLVER</a>
17     </body>
18 </html>

```

26. Editar el archivo `views.py` de la aplicación `moduloProyectoModelForm` y agregar la vista `listadoProyectos` que permita redireccionar al template `proyectos.html`

```
moduloProyectoModelForm > views.py > ...
1  from django.shortcuts import render, redirect
2  |
3  from moduloProyectoModelForm.models import Proyecto
4  |
5  # Create your views here.
6  def index(request):
7  |     return render(request, 'templateModuloProyecto\index.html')
8  |
9  def listadoProyectos(request):
10 |     return render(request, 'templateModuloProyecto\proyectos.html')
11 |
```

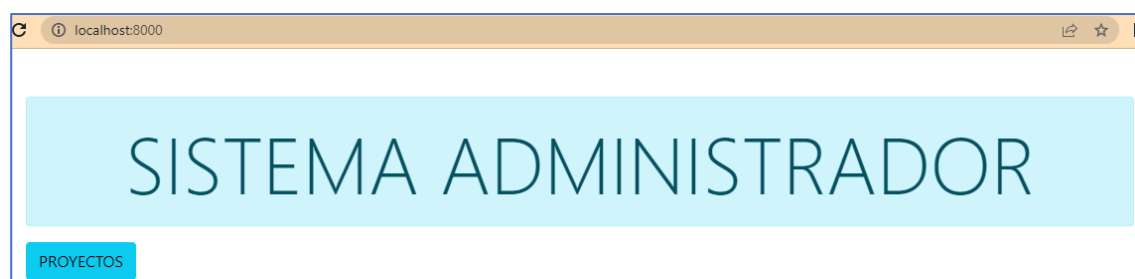
27. Modificar el archivo `urls.py` del proyecto `AdmProyectosModelForm` e incluir el `path` a la función `listadoProyectos`

```
AdmProyectosModelForm > urls.py > ...
16 from django.contrib import admin
17 from django.urls import path
18
19 from moduloProyectoModelForm import views
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('', views.index),
24     path('proyectos/', views.listadoProyectos)
25 ]
```

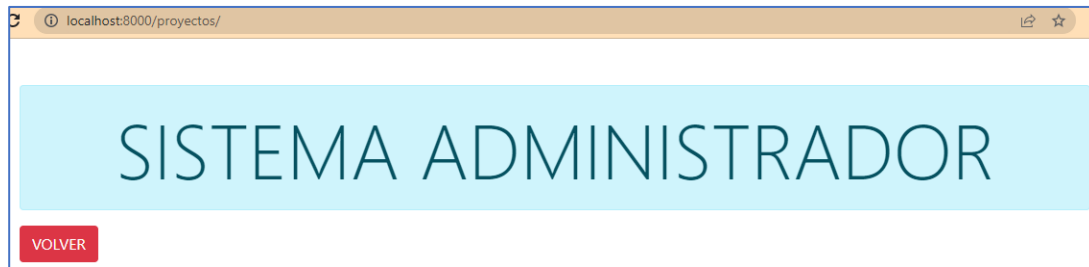
28. Modificar el templates `index.html`. Agregar el enlace en el botón proyectos.

```
<!DOCTYPE html>
{% load static %}
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <meta name="description" content="">
    <title>ADM Proyectos</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kmEgxtnW/KsZxNGUzEzwjAp60f9bBr/8/sz97650z/KZqN/tz16FIdvKxUN/Idw88" crossorigin="anonymous"></script>
  </head>
  <body class="container mt-5">
    <div class="alert alert-info display-1 text-center"> SISTEMA ADMINISTRADOR </div>
    <a href="/proyectos" class="btn btn-info">PROYECTOS</a>
  </body>
</html>
```

29. Iniciar el servidor si lo tiene abajo y luego revisar. Abrir el terminal del visual code





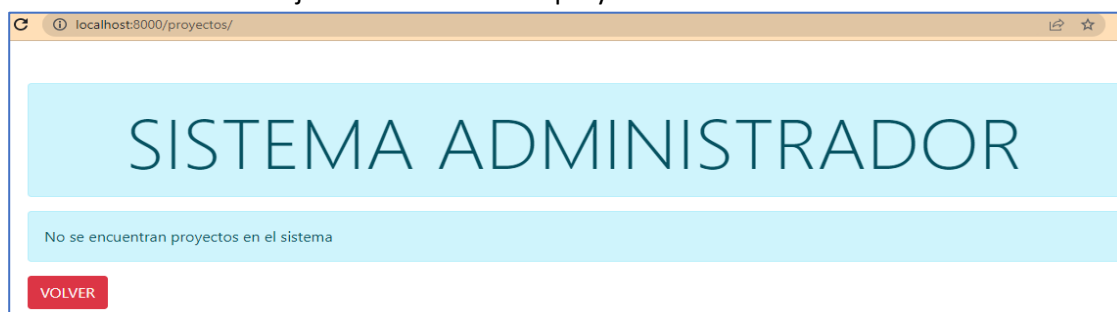


30. Modificar la vista `listadoProyectos` del archivo `views.py` de la aplicación `moduloProyectoModelForm`. Se requiere obtener los proyectos de la base de datos e imprimirlos en el template `proyectos.html`

```
moduloProyectoModelForm > views.py > ...
1  from django.shortcuts import render
2
3  from moduloProyectoModelForm.models import Proyecto
4
5  # Create your views here.
6  def index(request):
7      return render(request, 'templateModuloProyecto\index.html')
8
9  def listadoProyectos(request):
10     proyectos = Proyecto.objects.all()
11     data = {'proyectos':proyectos}
12     return render(request, 'templateModuloProyecto\proyectos.html', data)
```

**Nota:** la función ORM `Proyecto.objects.all()` obtiene los proyectos de la base de datos. Además, se define una variable `data` que almacenará el listado de proyectos para poder rescatarlo desde el template `proyectos.html`

31. Modificar el template `proyectos.html`. La modificación considera que si la lista de proyectos esta vacía muestre un mensaje “No se encuentran proyectos en el sistema”

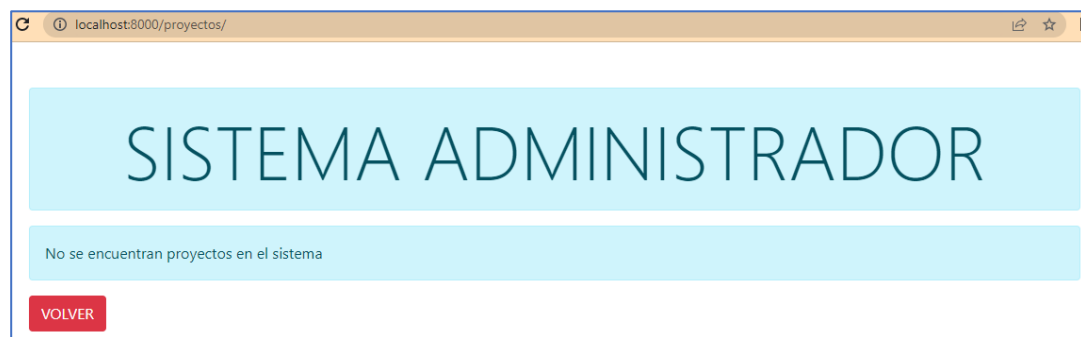


```

templates > templateModuloProyecto > <> proyectos.html
1 <!DOCTYPE html>
2 {% load static %}
3 <html>
4 <head>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <meta http-equiv="X-UA-Compatible" content="ie=edge">
8 <meta name="description" content="">
9 <title>ADM Proyectos</title>
10 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" int
11 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha
12 </head>
13 <body class="container mt-5">
14 <div class="alert alert-info display-1 text-center"> SISTEMA ADMINISTRADOR </div>
15 {% if proyectos %}
16
17
18 {% else %}
19 <div class="alert alert-info"> No se encuentran proyectos en el sistema </div>
20 {% endif %}
21 <a href="/" class="btn btn-danger">VOLVER</a>
22 </body>
23 </html>

```

32. Ejecuta y revisar en el navegador.



33. Configurar el admin para ingresar proyectos.

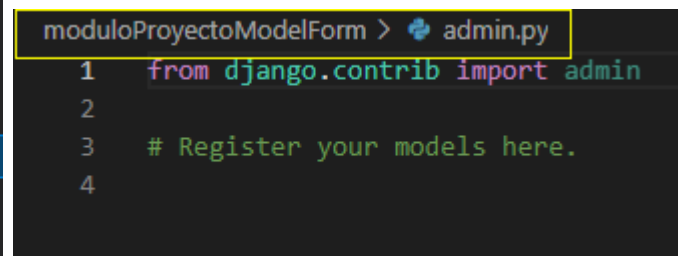
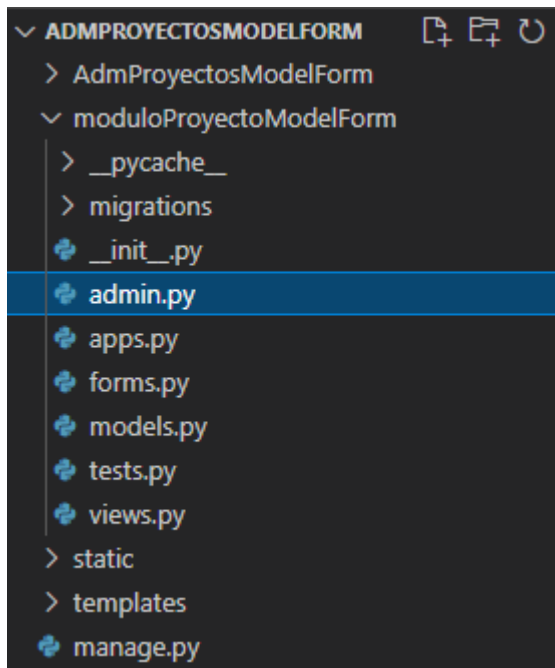
**Nota:** este paso es opcional. Tu puedes ingresar proyectos directamente desde la consola de Heidi



Ingreso con admin y secret como clave.



34. Modificar el archivo `admin.py` de la aplicación `moduloProyectoModelForm` para crear interfaces del modelo creado. Esto es para poder ver interfaces de creación de proyectos en la consola de administración Django.



Ingresa la siguiente información en `admin.py`

```

moduloProyectoModelForm > admin.py > ...
1  from django.contrib import admin
2
3  from .models import Proyecto
4
5  # Register your models here.
6  class ProyectoAdmin(admin.ModelAdmin):
7      list_display = ['fechaInicio',
8                     'fechaTermino',
9                     'nombre',
10                    'responsable',
11                    'prioridad'
12                    ]
13
14  admin.site.register(Proyecto, ProyectoAdmin)

```

ANTES

Nota: se crea una clase **ProyectoAdmin** que extiende de **admin.ModelAdmin**. Esto anterior, viene siendo una clase interface que permitirá visualizar esos campos del objeto proyecto.

A continuación, se registra el objeto proyecto y su objeto administrador para visualizarlo en el sistema de administración.

La acción de registrar estas clases en el sistema admin permite visualizar lo siguiente:

Después



35. Proceder a añadir un proyecto a través de la interfaz que proporciona el sistema de administración.

## MODULO PROYECTO MODEL FORM

### Proyectos

[+ Añadir](#)[✎ Modificar](#)

Emplee a escribir para filtrar...

#### AUTENTICACIÓN Y AUTORIZACIÓN

**Grupos** [+ Añadir](#)

**Usuarios** [+ Añadir](#)

#### MODULO PROYECTO MODEL FORM

**Proyectos** [+ Añadir](#)

### Añadir proyecto

**FechaInicio:**

Hoy | 

Nota: Usted va 4 horas por detrás de la hora del servidor.

**FechaTermino:**

Hoy | 

Nota: Usted va 4 horas por detrás de la hora del servidor.

**Nombre:**

**Responsable:**

**Prioridad:**

[Guardar y añadir otro](#)[Guardar y continuar editando](#)[GUARDAR](#)


### Añadir proyecto

**FechaInicio:**

Hoy | 

Nota: Usted va 4 horas por detrás de la hora del servidor.

**FechaTermino:**

Hoy | 

Nota: Usted va 4 horas por detrás de la hora del servidor.

**Nombre:**

**Responsable:**

**Prioridad:**

[Guardar y añadir otro](#)[Guardar y continuar editando](#)[GUARDAR](#)

✓ El proyecto "Proyecto object (1)" fue agregado correctamente.

Seleccione proyecto a modificar

AÑADIR PROYECTO +

Acción: 

-----

 Ir seleccionados 0 de 1

| <input type="checkbox"/> | FECHAINICIO         | FECHATERMINO         | NOMBRE               | RESPONSABLE | PRIORIDAD |
|--------------------------|---------------------|----------------------|----------------------|-------------|-----------|
| <input type="checkbox"/> | 9 de agosto de 2022 | 12 de agosto de 2022 | Capacitación Backend | Paul        | 1         |

1 proyecto

← → ↺

localhost:8000/admin/moduloProyectoModelForm/proyecto/

🔍

🔖

🌐

🔧

⋮

Administración de Django

BIENVENIDOS, TERESA JESÚS. [VER EL SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [CERRAR SESIÓN](#)

Inicio > ModuloProyectoModelForm > Proyectos

Empiece a escribir para filtrar...

AUTENTICACIÓN Y AUTORIZACIÓN

Grupos + Añadir

Usuarios + Añadir

MODULOPROYECTOMODELFORM

Proyectos + Añadir

✓ El proyecto "Proyecto object (1)" fue agregado correctamente.

Seleccione proyecto a modificar

AÑADIR PROYECTO +

Acción: 

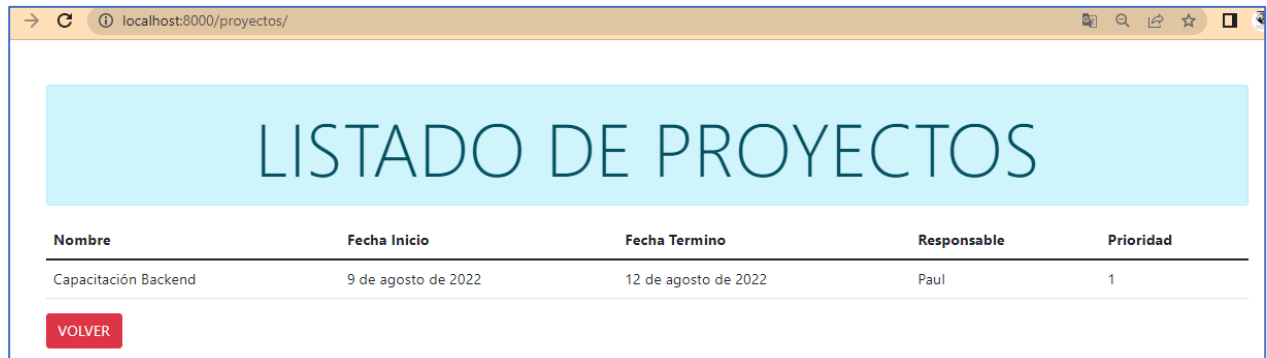
-----

 Ir seleccionados 0 de 1

| <input type="checkbox"/> | FECHAINICIO         | FECHATERMINO         | NOMBRE               | RESPONSABLE | PRIORIDAD |
|--------------------------|---------------------|----------------------|----------------------|-------------|-----------|
| <input type="checkbox"/> | 9 de agosto de 2022 | 12 de agosto de 2022 | Capacitación Backend | Paul        | 1         |

1 proyecto

36. Modificar el template `proyectos.html`. La modificación considera mostrar los proyectos de la base de datos en una tabla.

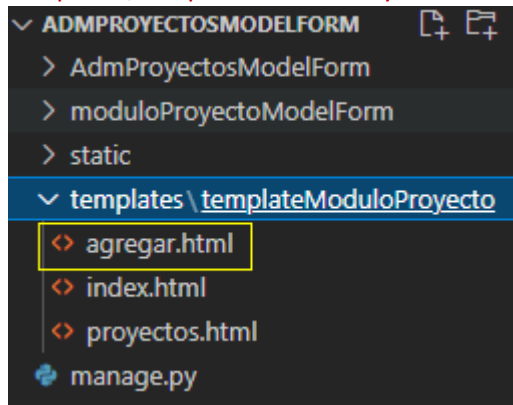


| Nombre               | Fecha Inicio        | Fecha Termino        | Responsable | Prioridad |
|----------------------|---------------------|----------------------|-------------|-----------|
| Capacitación Backend | 9 de agosto de 2022 | 12 de agosto de 2022 | Paul        | 1         |

VOLVER

```
templates > templateModuloProyecto > <> proyectos.html
13 <body class="container mt-5">
14 <div class="alert alert-info display-1 text-center"> LISTADO DE PROYECTOS </div>
15 {% if proyectos %}
16 <table class="table">
17 <thead>
18 <tr>
19 <th>Nombre</th>
20 <th>Fecha Inicio</th>
21 <th>Fecha Termino</th>
22 <th>Responsable</th>
23 <th>Prioridad</th>
24 </tr>
25 </thead>
26 <tbody>
27 {% for proyecto in proyectos %}
28 <tr>
29 <td>{{proyecto.nombre}}</td>
30 <td>{{proyecto.fechaInicio}}</td>
31 <td>{{proyecto.fechaTermino}}</td>
32 <td>{{proyecto.responsable}}</td>
33 <td>{{proyecto.prioridad}}</td>
34 </tr>
35 {% endfor %}
36 </tbody>
37 </table>
38
39 {% else %}
40 <div class="alert alert-warning text-center">
41     No se encuentran proyectos en el sistema
42 </div>
43 {% endif %}
44 <a href="/" class="btn btn-danger">VOLVER</a>
45 </body>
```

37. Crear el templates `agregar.html`, específicamente en la carpeta `templates/templateModuloProyecto`



38. Editar el template `agregar.html` y diseñar la siguiente interfaz



VOLVER

```
templates > templateModuloProyecto > <> agregar.html
1  <!DOCTYPE html>
2  {% load static %}
3  <html>
4  <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <meta name="description" content="">
9      <title>ADM Proyectos</title>
10     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" int
11     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sh
12 </head>
13 <body class="container mt-5">
14     <div class="alert alert-info display-1 text-center"> AGREGAR UN PROYECTO </div>
15
16     <a href="/" class="btn btn-danger">VOLVER</a>
17 </body>
18 </html>
```



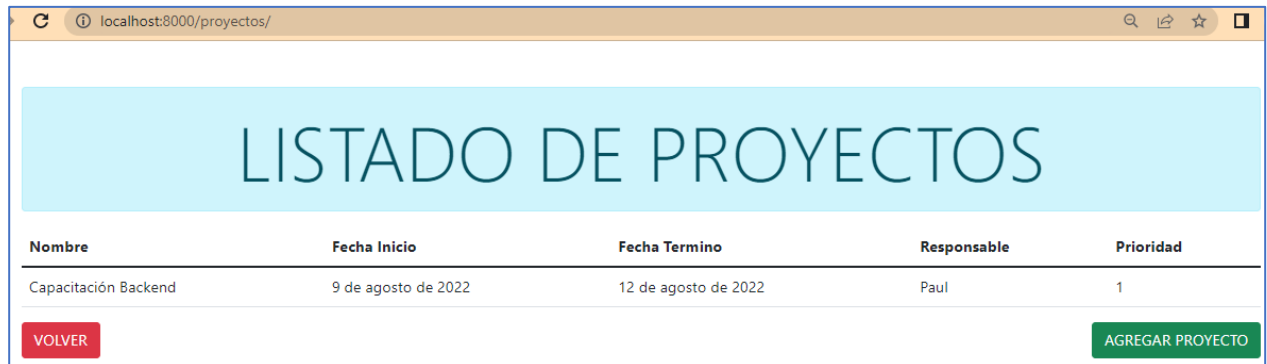
39. Editar el archivo `views.py` de la aplicación `moduloProyectoModelForm` y agregar la vista `agregarProyecto`.

```
moduloProyectoModelForm > views.py > ...
1  from django.shortcuts import render
2
3  from moduloProyectoModelForm.models import Proyecto
4
5
6  # Create your views here.
7  def index(request):
8      return render(request, 'templateModuloProyecto\index.html')
9
10 def listadoProyectos(request):
11     proyectos = Proyecto.objects.all()
12     data = {'proyectos':proyectos}
13     return render(request, 'templateModuloProyecto\proyectos.html', data)
14
15 def agregarProyecto(request):
16     return render(request, 'templateModuloProyecto'+'\\'+ 'agregar.html')
17
```

40. Modificar el archivo `urls.py` del proyecto `AdmProyectosModelForm` e incluir el `path` a la función `agregarProyecto`

```
AdmProyectosModelForm > urls.py > ...
16 from django.contrib import admin
17 from django.urls import path
18
19 from moduloProyectoModelForm import views
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('', views.index),
24     path('proyectos/', views.listadoProyectos),
25     path('proyectos/agregar', views.agregarProyecto)
26 ]
27
```

41. Modificar el template [proyectos.html](#). La modificación considera agregar un botón AGREGAR PROYECTO.



```
{% else %}
    <div class="alert alert-warning text-center">
        No se encuentran proyectos en el sistema
    </div>
{% endif %}
<a href="/" class="btn btn-danger">VOLVER</a>
<a href="agregar" class="btn btn-success float-end">AGREGAR PROYECTO</a>
</body>
</html>
```

Modificar [href](#) del botón volver....

```
<a href=".." class="btn btn-danger">VOLVER</a>
<a href="agregar" class="btn btn-success float-end">AGREGAR PROYECTO</a>
</body>
```

42. Modificar el template [agregar.html](#). Modificar [href](#) del botón volver....

```
<a href=".." class="btn btn-danger">VOLVER</a>
<input type="submit" class="btn btn-success float-end" value="GUARDAR">
```

#### 43. Iniciar el servidor y revisar la navegación

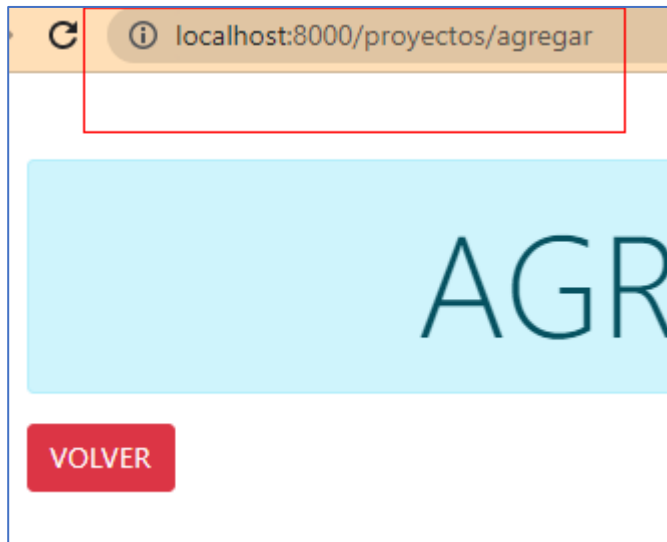
The screenshot shows a web browser at localhost:8000. The first view is the 'SISTEMA ADMINISTRADOR' header with a 'PROYECTOS' button. The second view is the 'LISTADO DE PROYECTOS' page, which contains a table with project data and two buttons: 'VOLVER' and 'AGREGAR PROYECTO'.

| Nombre               | Fecha Inicio        | Fecha Termino        | Responsable | Prioridad |
|----------------------|---------------------|----------------------|-------------|-----------|
| Capacitación Backend | 9 de agosto de 2022 | 12 de agosto de 2022 | Paul        | 1         |

This block contains two side-by-side cropped views of the project list table from the previous screenshot. The left view shows the 'Nombre' column with the value 'Capacitación Backend' and a 'VOLVER' button. The right view shows the 'Prioridad' column with the value '1' and an 'AGREGAR PROYECTO' button.

| Nombre               |
|----------------------|
| Capacitación Backend |

| Prioridad |
|-----------|
| 1         |



44. Modificar el template [agregar.html](#). La modificación considera diseñar un formulario como se ilustra en la siguiente imagen.

A screenshot of a web form titled 'AGREGAR UN PROYECTO' in a light blue header. The form contains five input fields with labels: 'FechaInicio:', 'FechaTermino:', 'Nombre:', 'Responsable:', and 'Prioridad:'. At the bottom left is a red button labeled 'VOLVER', and at the bottom right is a green button labeled 'GUARDAR'.

En Django, el diseño del formulario se crea a partir de una clase formulario (visto en el punto 19 de este documento). Lo que significa que se necesita asignarlo y validarlo en la vista `agregarProyecto` para poder verse en el html.

Las modificaciones del template [agregarProyecto.html](#) consisten en declarar el formulario a través de una variable definida en la vista. En este caso, es a través de la siguiente línea de instrucción `{{formu}}` siendo `formu` la variable que posee la estructura del formulario.

Además, otra modificación es agregar un token de seguridad `csrf_token` como se ilustra en la siguiente imagen.



|                      |                                     |                      |
|----------------------|-------------------------------------|----------------------|
| <b>FechaInicio:</b>  |                                     |                      |
| <b>FechaTermino:</b> | Atrás                               | Alt+Flecha izquierda |
|                      | Reenviar                            | Alt+Flecha derecha   |
| <b>Nombre:</b>       | Volver a cargar                     | Ctrl+R               |
| <b>Responsable:</b>  | Guardar como...                     | Ctrl+S               |
|                      | Imprimir...                         | Ctrl+P               |
| <b>Prioridad:</b>    | Transmitir...                       |                      |
|                      | Buscar imágenes con Google Lens     |                      |
| <b>VOLVER</b>        | Crear un código QR para esta página |                      |
|                      | Ver código fuente de página         | Ctrl+U               |
|                      | Inspeccionar                        |                      |

```
<input type="hidden" name="csrfmiddlewaretoken" value="Q9w8mGKzXuXs8AntoXFnXWexJQV2X4MU9GvfoHbbQhTYDCbWoHGu2b3weuKuM6b0">
<tr>
```

`value="Q9w8mGKzXuXs8AntoXFnXWexJQV2X4MU9GvfoHbbQhTYDCbWoHGu2b3weuKuM6b0">`

Al refrescar la página se va actualizando el token

`value="zKYU001qSEpuWEyUBzoDZ0G6CCL4Gp6uShX121s2br10rGmnBjpK4fv57gAwvrvA">`

45. Editar el archivo `views.py` de la aplicación `moduloProyectoModelForm` y modificar la vista `agregarProyecto`.

```
moduloProyectoModelForm > views.py > agregarProyecto
1 from django.shortcuts import render, redirect
2
3 from moduloProyectoModelForm.models import Proyecto
4 from moduloProyectoModelForm.forms import FormProyecto
5
```

```
def agregarProyecto(request):
    formu = FormProyecto()
    if (request.method == 'POST'):
        formu = FormProyecto(request.POST)
        if(formu.is_valid()):
            formu.save()
            return redirect('/proyectos')# return #index(request)

    data = {'formu': formu , 'titulo': 'AGREGAR PROYECTO'}
    return render(request, 'templateModuloProyecto/'+ 'agregar.html', data)
```

Al mandarle el **título** de manera dinámica debemos modificar **agregar.html** para que reciba el título como se ilustra a continuación.

```
<body class="container mt-5">
    <div class="alert alert-info display-1 text-center"> {{titulo}} </div>
```

Y recordar el formulario que recibirá

```
<form method="POST">
    <table class="table">
        {{ formu }}
        {% csrf_token %}
        <tr>
            <td colspan="2">
                <a href=".." class="btn btn-danger">VOLVER</a>
                <input type="submit" class="btn btn-success float-end" value="GUARDAR">
            </td>
        </tr>
    </table>
</form>
```

46. Iniciar el servidor, revisar la navegación y agregar un registro.

# SISTEMA ADMINISTRADOR

PROYECTOS

localhost:8000/proyectos/

LISTADO DE PROYECTOS

| Nombre               | Fecha Inicio        | Fecha Termino        | Responsable | Prioridad |
|----------------------|---------------------|----------------------|-------------|-----------|
| Capacitación Backend | 9 de agosto de 2022 | 12 de agosto de 2022 | Paul        | 1         |

VOLVER

AGREGAR PROYECTO

AGREGAR UN PROYECTO

FechaInicio:

08/09/2022

FechaTermino:

08/12/2022

Nombre:

Capacitación Oracle

Responsable:

Juanito

Prioridad:

3

VOLVER

GUARDAR

LISTADO DE PROYECTOS

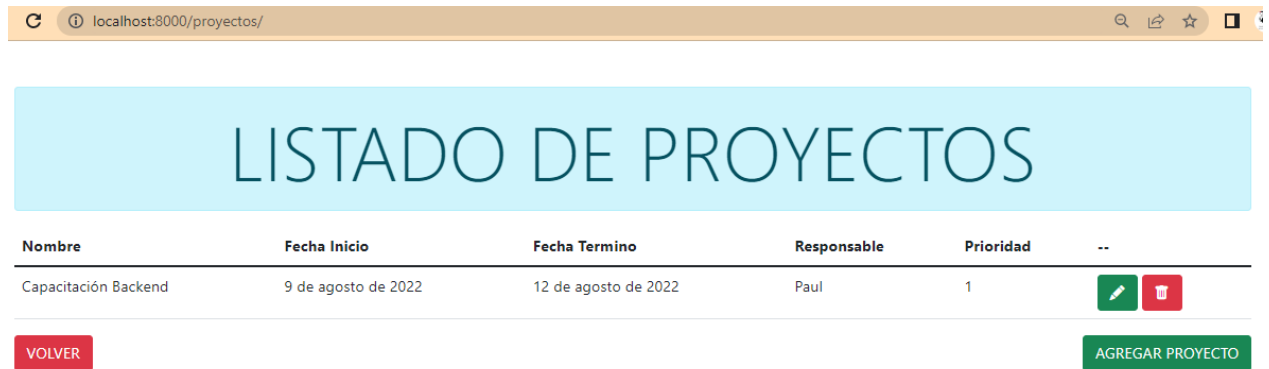
| Nombre               | Fecha Inicio            | Fecha Termino          | Responsable | Prioridad |
|----------------------|-------------------------|------------------------|-------------|-----------|
| Capacitación Backend | 9 de agosto de 2022     | 12 de agosto de 2022   | Paul        | 1         |
| Capacitación Oracle  | 8 de septiembre de 2022 | 8 de diciembre de 2022 | Juanito     | 3         |

VOLVER

AGREGAR PROYECTO



47. Modificar el template [proyectos.html](#). La modificación considera agregar dos iconos: un lápiz(editar) y un basurero (eliminar). Los iconos deberán editar y eliminar un proyecto. La interfaz debiese verse como:

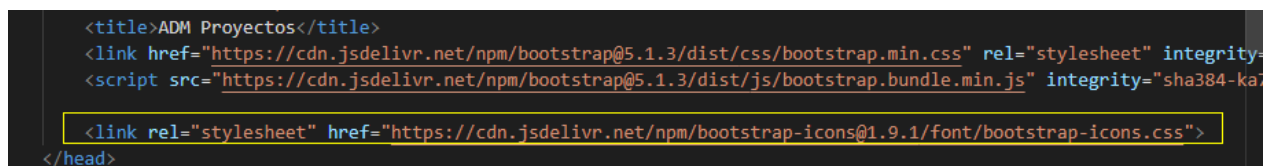


Para utilizar los iconos del lápiz y basurero obtener su especificación de la página de Bootstrap <https://icons.getbootstrap.com/#icons> CDN

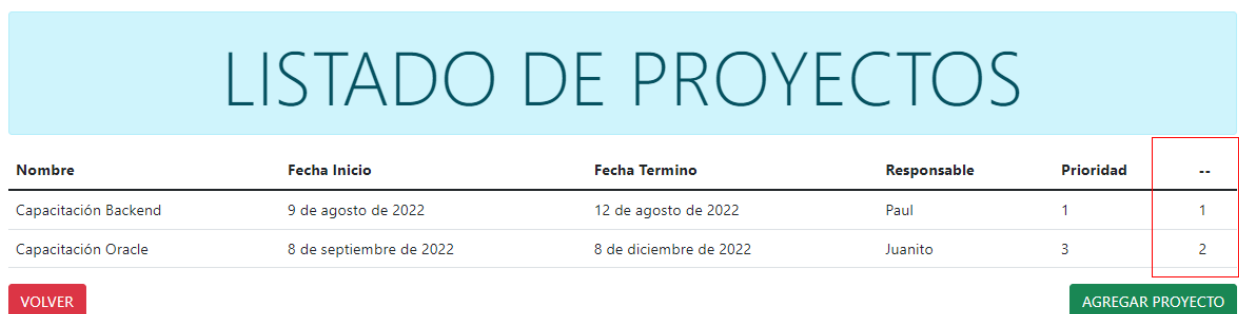
Las modificaciones son:

- En particular, agregar el siguiente link a la cabecera del archivo [proyectos.html](#)

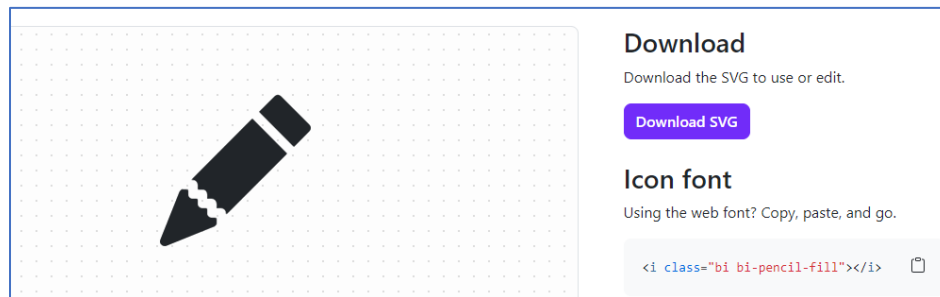
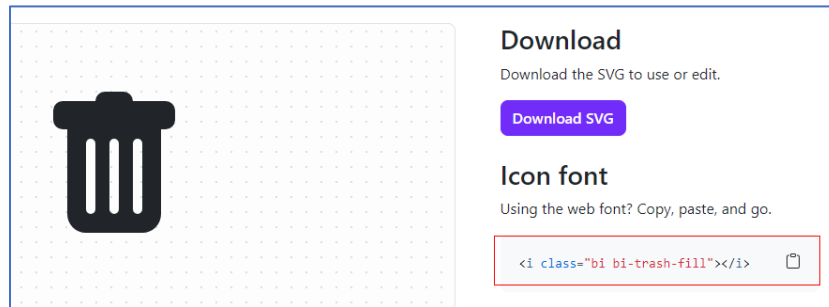
```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">
```



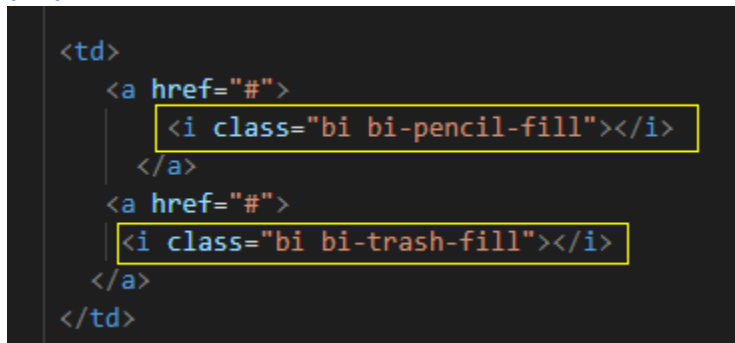
- En proyectos.html se agrega una columna más



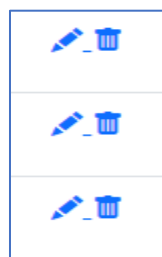
- En la página de Bootstrap, buscar los iconos y copiar su especificación enmarcada en rojo



- Agregar una columna a la tabla proyectos y la especificación de los iconos en [proyectos.html](#)



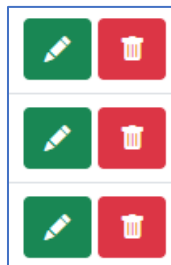
El resultado de su visualización es:



- Modificar la visualización de los iconos para verlos como botón. Esto es, se agrega la especificación de la clase.

```
<td>
    <a class="btn btn-success btn-small" href="#">
        <i class="bi bi-pencil-fill"></i>
    </a>
    <a class="btn btn-danger btn-small" href="#">
        <i class="bi bi-trash-fill"></i>
    </a>
</td>
<td>
```

El resultado de su visualización es:



48. Editar el archivo `views.py` de la aplicación `moduloProyectoModelForm` y agregar la vista `eliminarProyecto`.

```
def eliminarProyecto(request,id):
    proyecto =Proyecto.objects.get(id=id)
    proyecto.delete()
    return redirect('/proyectos')
```

49. Modificar el archivo `urls.py` del proyecto e incluir el `path` a la función `eliminarProyecto`

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

from moduloProyectoModelForm import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',views.index),
    path('proyectos/', views.listadoProyectos),
    path('proyectos/agregar/', views.agregarProyecto),
    path('proyectos/eliminar/<int:id>', views.eliminarProyecto)
]
```

50. Editar el archivo `views.py` de la aplicación `moduloProyectoModelForm` y agregar la vista `actualizarProyecto`.

```
def actualizarProyecto(request,id):
    proyecto = Proyecto.objects.get(id = id)
    formu = FormProyecto(instance=proyecto)
    if (request.method == 'POST'):
        formu = FormProyecto(request.POST, instance= proyecto)
        if(formu.is_valid()):
            formu.save()
            return redirect('/proyectos')

    data = {'formu': formu , 'titulo': 'ACTUALIZAR PROYECTO'}
    return render(request, 'templateModuloProyecto/'+'agregar.html', data)
```

51. Modificar el archivo `urls.py` del proyecto e incluir el `path` a la función `actualizarProyecto`

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

from moduloProyectoModelForm import views



urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index),
    path('proyectos/', views.listadoProyectos),
    path('proyectos/agregar/', views.agregarProyecto),
    path('proyectos/eliminar/<int:id>', views.eliminarProyecto),
    path('proyectos/actualizar/<int:id>', views.actualizarProyecto)
]
```

52. Modificar el template `proyectos.html`. La modificación considera agregar los enlaces de los iconos lápiz(editar) y un basurero (eliminar).

```
<td>
    <a class="btn btn-success btn-small" href="actualizar/{{proyecto.id}}">
        <i class="bi bi-pencil-fill"></i>
    </a>
    <a class="btn btn-danger btn-small" href="eliminar/{{proyecto.id}}">
        <i class="bi bi-trash-fill"></i>
    </a>
</td>
```

53. Iniciar el servidor y revisar la navegación  
Modificando los datos

## LISTADO DE PROYECTOS

| Nombre                 | Fecha Inicio       | Fecha Termino       | Responsable | Prioridad | --  |
|------------------------|--------------------|---------------------|-------------|-----------|---|
| Capacitación Backend 2 | 7 de julio de 2021 | 13 de julio de 2021 | Osvaldo     | 2         |   |

VOLVER

AGREGAR PROYECTO

## ACTUALIZAR PROYECTO

|               |   |
|---------------|---|
| FechaInicio:  | <input type="text" value="07/07/2021"/>             |
| FechaTermino: | <input type="text" value="13/07/2021"/>             |
| Nombre:       | <input type="text" value="Capacitación Backend 2"/> |
| Responsable:  | <input type="text" value="Osvaldo2"/>               |
| Prioridad:    | <input type="text" value="3"/>                      |

VOLVER

GUARDAR





## AGREGAR PROYECTO

|               |   |
|---------------|---|
| FechaInicio:  | <input type="text" value="09/08/2022"/>   |
| FechaTermino: | <input type="text" value="12/08/2022"/>   |
| Nombre:       | <input type="text" value="Teresa Jesus"/> |
| Responsable:  | <input type="text" value="Teresa"/>       |
| Prioridad:    | <input type="text" value="1"/>            |

VOLVER

GUARDAR

# LISTADO DE PROYECTOS

| Nombre                 | Fecha Inicio        | Fecha Termino        | Responsable | Prioridad | --  |
|------------------------|---------------------|----------------------|-------------|-----------|---|
| Capacitación Backend 2 | 7 de julio de 2021  | 13 de julio de 2021  | Osvaldo2    | 3         |   |
| Teresa Jesus           | 9 de agosto de 2022 | 12 de agosto de 2022 | Teresa      | 1         |   |

VOLVER

AGREGAR PROYECTO





# LISTADO DE PROYECTOS

| Nombre                 | Fecha Inicio        | Fecha Termino        | Responsable | Prioridad | --  |
|------------------------|---------------------|----------------------|-------------|-----------|---|
| Capacitación Backend 2 | 7 de julio de 2021  | 13 de julio de 2021  | Osvaldo2    | 3         |   |
| Teresa Jesus           | 9 de agosto de 2022 | 12 de agosto de 2022 | Teresa      | 1         |   |
| tere                   | 9 de agosto de 2022 | 12 de agosto de 2022 | Juanito     | 4         |   |

VOLVER

AGREGAR PROYECTO

# LISTADO DE PROYECTOS

| Nombre                 | Fecha Inicio        | Fecha Termino        | Responsable | Prioridad | --  |
|------------------------|---------------------|----------------------|-------------|-----------|---|
| Capacitación Backend 2 | 7 de julio de 2021  | 13 de julio de 2021  | Osvaldo2    | 3         |   |
| Teresa Jesus           | 9 de agosto de 2022 | 12 de agosto de 2022 | Teresa      | 1         |   |

VOLVER

AGREGAR PROYECTO