

Contenido

Configuración	2
Librería para el manejo de imágenes	2
Configurar el icono.....	2
ORM	2
Práctica	2
Personalizar la administración de Django	3
Práctica	3
Manejo de imágenes	5
Modelo Vista Template.....	6
Práctica	6

Configuración

Librería para el manejo de imágenes

Para el manejo de imágenes se requiere la librería Pillow
pip install Pillow

Configurar el icono

```
<link rel="icon" type="image/jpg" href="{% static 'core/img/courses.jpg' %}">
```

ORM

Django ofrece un ORM (Modelo Objeto Relacional)

- Objetos mapeados a la base de datos.

Referencia <https://docs.djangoproject.com/en/5.0/topics/db/queries/>

Práctica

1. Crear la aplicación proyecto
python manage.py startapp project
2. Agregar la aplicación en el archivo settings.py

```
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     'core',
41     'project',
42 ]
```

3. Creación del MODELO
 - a. Dentro de la aplicación project, abrir el archivo “models.py”, crear la clase Project

```
from django.db import models

class Project (models.Model):
    title = models.CharField(max_length=100)
    description = models.TextField()
    image = models.ImageField()
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)
```

- b. * Cada atributo de la clase, representa una columna en la tabla que será creada.
4. Realizar la migración a la base de datos.

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

5. Registrar el modelo en el archivo admin.py de la aplicación proyecto

```
1 from django.contrib import admin
2 from .models import Project
3
4 admin.site.register(Project)
5
```

6. Para acceder al administrador de Django, solo hay que colocar la url:
<http://127.0.0.1:8000/admin/>
7. Agregar un registro, seleccione una imagen cualquiera, observe:
 - a. Los nombres que aparecen, (object, en ingles)
 - b. La imagen no se muestra (marca un 404)
 - c. Observe donde se guardo el archivo de imagen
 - d. Verifique los campos de la base de datos y observe que la imagen no se guarda en la base de datos, lo que se guarda es un URL

Personalizar la administración de Django

Práctica

1. En el archivo apps.py realice los siguientes cambios

```
apps.py U x
webpersonal > project > apps.py > ProjectConfig
1 from django.apps import AppConfig
2
3
4 class ProjectConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'project'
7
8     verbose_name = 'Portafolio de proyectos'
```

2. Agregar la subclase Meta (Metadatos) en la clase del modelo

```

13     #auto_now se ejecuta cada vez que se modificar
14     updated = models.DateTimeField(auto_now=True)
15
16     class Meta:
17         verbose_name = "Proyecto"
18         verbose_name_plural = "Proyectos"
19
20         ordering = ['-created']
21
22     def __str__(self):
23         return self.title

```

3. Para que se muestre en español y con la leyenda adecuada, es necesario inicializar la propiedad `verbose_name`

```

5 class Project (models.Model):
6     title = models.CharField(max_length=100, verbose_name='Título')
7     description = models.TextField(verbose_name='Descripción')
8     image = models.ImageField(verbose_name='Imagen')
9
10     #auto_now_add solo se ejecuta cuando se crea
11     created = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de creación')
12
13     #auto_now se ejecuta cada vez que se modificar
14     updated = models.DateTimeField(auto_now=True, verbose_name='Fecha de modificación')

```

4. Para extender la clase de Admin y poder mostrar los campos que son solo de lectura se realiza la siguiente configuración

```

admin.py x
webpersonal > projects > admin.py > ...
1 from django.contrib import admin
2 from .models import Project
3
4
5 class ProjectAdmin(admin.ModelAdmin):
6     readonly_fields = ('created', 'updated')
7
8
9 admin.site.register(Project, ProjectAdmin)

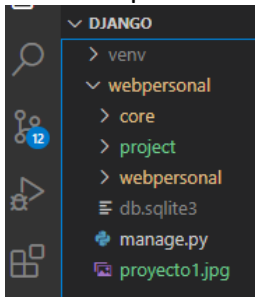
```

5. El resultado es el siguiente:

The screenshot shows the Django Admin interface for a project named 'Aprende a programar en 1 día'. The interface is in Spanish. On the left, there's a sidebar with navigation links like 'Inicio', 'Portafolio de pproyectos', 'Proyectos', and 'Aprende a programar en 1 día'. The main content area shows the 'Modificar Proyecto' form. The form includes a 'Título' field with the value 'Aprende a programar en 1 día', a 'Descripción' field with a long text area containing placeholder text, an 'Imágenes' field with a file upload button and a 'Seleccionar archivo' button, and two read-only fields for 'Fecha de creación' and 'Fecha de modificación', both showing the date and time '8 de julio de 2024 a las 16:50'.

Manejo de imágenes

1. Observe que las imágenes se están guardando en la raíz del proyecto.



2. Para que la imagen se pueda administrar correctamente, es necesario, configurar el directorio "media". En el archivo settings.py :

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

```
128 #Media files
129 MEDIA_URL = '/media/'
130 MEDIA_ROOT = os.path.join(BASE_DIR, "media")
131
```

3. La siguiente configuración es para indicarles a django que las imágenes de la aplicación proyecto se debe guardar en el directorio project.

```
7 description = models.TextField(verbose_name='Descripción')
8 image = models.ImageField(verbose_name='Imagen', upload_to='project')
```

4. Esto solo se hace en modo debug (en el archivo urls.py)

```
from django.conf.urls.static import static
from django.conf import settings
```

```
31 if settings.DEBUG:
32     urlpatterns += static(settings.MEDIA_URL,
33                           document_root=settings.MEDIA_ROOT)
```

5. Verifique su funcionamiento.
6. Observe que si elimina un registro, la imagen no se elimina del directorio, para corregirlo solo hay que instalar la siguiente librería e instalarla como aplicación.

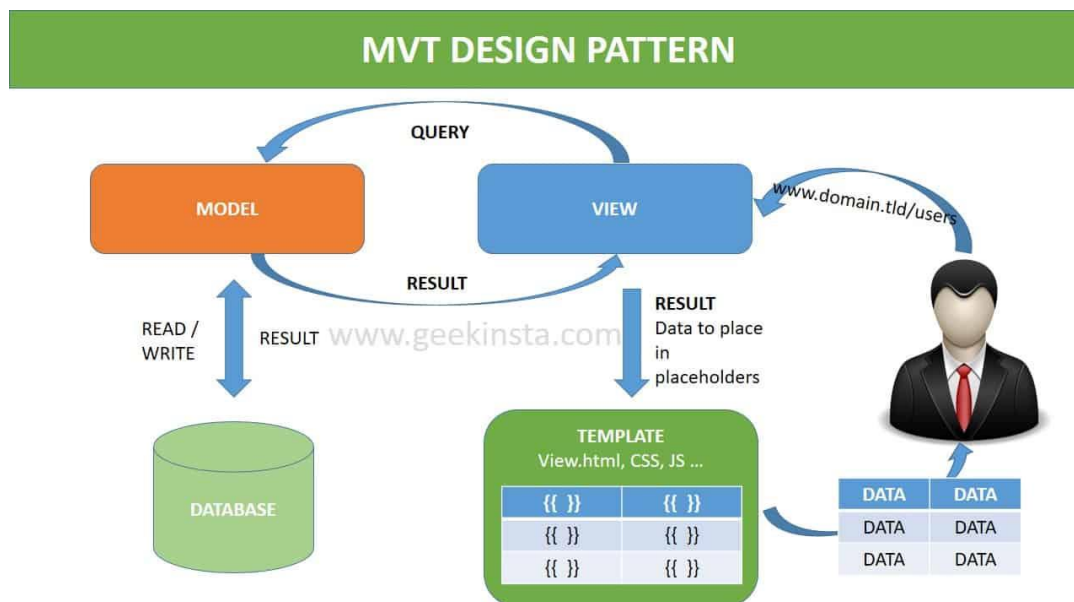
pip install django-cleanup

```

34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     'django_cleanup',
42     'core',
43     'project',
44 ]

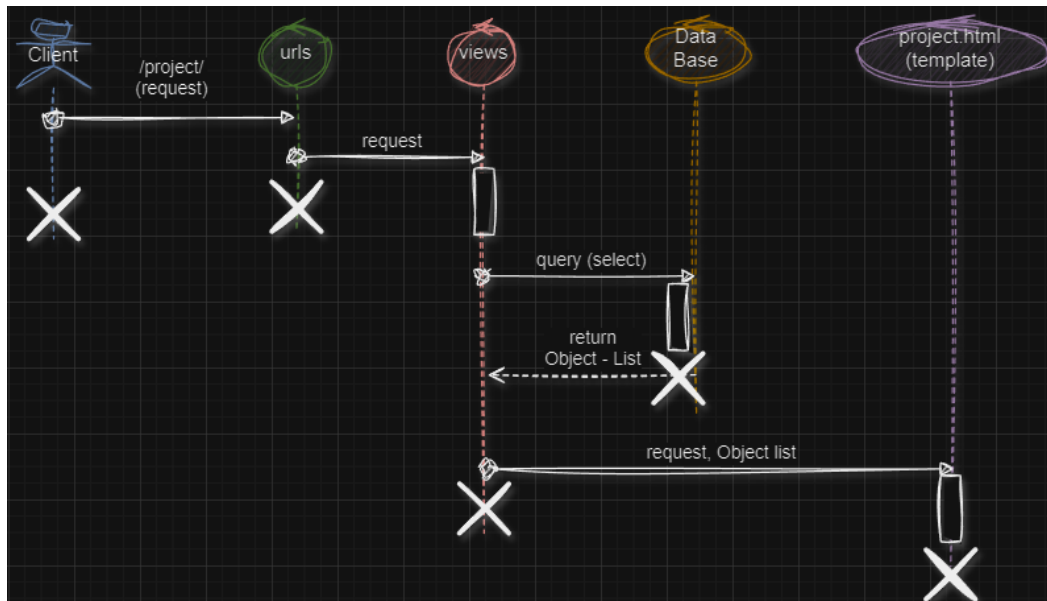
```

Modelo Vista Template



Práctica

1. El diagrama de secuencia de lo que representa la aplicación es:



2. Crear la estructura de templates/project en la aplicación project.
 - a. Crear el archivo project.html
3. Del archivo portfolio.html, copiar el código correspondiente y pegarlo dentro de projects.html, (no olvidar lo que se realizó en el archivo home.html)
4. Coloque la variable `{{projects}}`, dentro del HTML

```

9      Nuestros Proyectos
10     <i class="far fa-hands-usd"></i>
11   </h1>
12   {{projects}}
13   <div class="row">
14     <div class="col d-sm-block">
15       <!-- Proyecto -->
16       <div class="row project">
17         <div class="col-lg-3 col-md-4

```

5. Dentro del archivo views.py, consultar todos los proyectos y regresar el “querySet” en el template.

```

views.py U x
webpersonal > project > views.py > ...
1  from django.shortcuts import render
2  from .models import Project
3
4  def project(request):
5      projects = Project.objects.all()
6      return render(request, "project/project.html", {'projects':projects})

```

6. En el archivo urls.py, agregar la ruta y vista correspondiente, observe que ya tiene dos archivos que se llaman views, por lo que es necesario renombrarlo.

```

20 from . import settings
21 from project import views as views_project
22
23 urlpatterns = [
24     path('admin/', admin.site.urls),
25     path('', views.home, name='home'),
26     path('contact/', views.contact, name='contact'),
27     path('project/', views_project.project, name='project'),
28 ]

```

7. Modificar el “navbar” (menu) para que pueda acceder a los proyectos.

```

<a class="nav-link" href="{% url 'project' %}">
    <i class="far fa-file-chart-line"></i> Proyectos</a>
</li>

```

8. Verificar su funcionamiento.

💡 Nuestros Proyectos 🤝

<QuerySet [<Project: La numismática es una ciencia>, <Project: Aprende a programar en un día>]>

¿Cómo aprender a programar e

As we got further and further away, it [the Earth] diminished

9. Un queryset que es una lista de objetos, es la representación del resultado de una consulta a la base de datos, donde cada registro está representado por un objeto.
10. Meter en un ciclo los proyectos que recibirá el template. Para iterar dicha lista, lo hacemos con el template tag for

```

9     Nuestros Proyectos
10     <i class="far fa-hands-usd"></i>
11     </h1>
12     {% for project in projects %}
13     <!-- Proyecto -->
14     <div class="row project">
15         <div class="col-lg-3 col-md-4 offset-lg-1">
16             
17         </div>
18         <div class="col-lg-7 col-md-8">
19             <br>
20             <h2 class="section-heading title">
21                 {{project.title}}
22             </h2>
23             <p>
24                 {{project.description}}
25             </p>
26             <p><a href="http://google.com">Más información</a></p>
27         </div>
28     </div>
29     {% endfor %}

```

11. Verifique su funcionamiento.