

DJANGO – INSTRUCCIONES

Comprobar actualizaciones: **sudo apt-get update**

Instalar python:

sudo apt-get install python3-pip

sudo apt-get install python-is-python3

Comprobamos que tenemos django instalado con: **python -m django --version**, si no está instalado usaremos el siguiente comando: **pip install Django==4.1.5**

Comprobamos si funciona el comando **django-admin**, en el caso que no ejecutamos el siguiente comando: **sudo apt install python3-django**

Para iniciar un nuevo proyecto usamos: **django-admin startproject nombre**

Este comando ha creado:

nombre/

manage.py

mysite/

__init__.py

settings.py

urls.py

asgi.py

wsgi.py

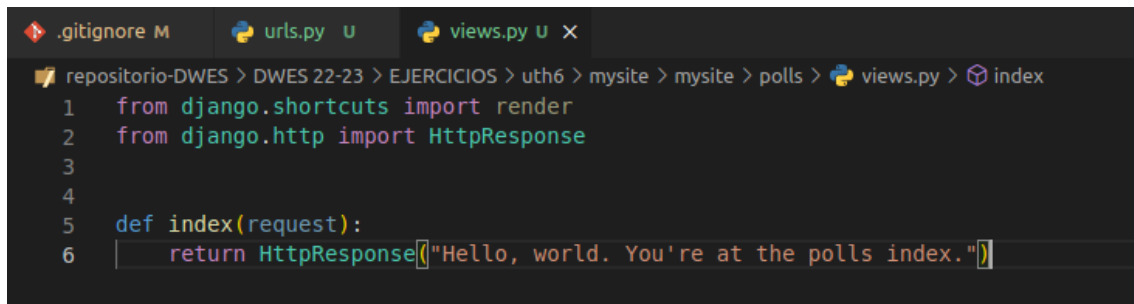
DEBEMOS ESTAR EN LA RUTA DE NUESTRO PROYECTO CONSTANTEMENTE.

Vamos a crear el directorio polls con el comando: **python manage.py startapp polls**

Si queremos revisar la estructura del proyecto instalamos **tree**: **sudo apt install tree**

Configuración básica:

Creamos la primera vista (dentro del directorio **polls**):

A screenshot of a code editor with a dark theme. The top of the editor shows three tabs: '.gitignore M', 'urls.py U', and 'views.py U X'. The main area displays the file path 'repositorio-DWES > DWES 22-23 > EJERCICIOS > uth6 > mysite > mysite > polls > views.py' followed by an 'index' icon. The code in the editor is as follows:

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4
5 def index(request):
6     return HttpResponse("Hello, world. You're at the polls index.")
```

Creamos el directorio `urls.py` y dentro le ponemos esto (*polls*):

```
.gitignore M  urls.py U X  views.py U
repositorio-DWES > DWES 22-23 > EJERCICIOS > uth6 > mysite > mysite > polls > urls.py > ...
1  from django.urls import path
2
3  from . import views
4
5  urlpatterns = [
6      path('', views.index, name='index'),
7  ]
```

En `urls.py` de *mysite* escribimos lo siguiente:

```
.gitignore M  urls.py .../polls U  urls.py .../mysite U X  views.py U
repositorio-DWES > DWES 22-23 > EJERCICIOS > uth6 > mysite > mysite > mysite > urls.py > ...
1  from django.contrib import admin
2  from django.urls import include, path
3
4  urlpatterns = [
5      path('polls/', include('polls.urls')),
6      path('admin/', admin.site.urls),
7  ]
```

En `settings.py` de *mysite* en el apartado `DATABASES`, modificamos los parametros a nuestro gusto:

```
settings.py 75
urls.py U    76
wsgi.py U    77
polls        78
migrat...    79
__init__...  80
admin.py U   81
apps.py U    82
models... U  83
tests.py U   84
              85
              86

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'mysite',
        'USER': 'user_mysite',
        'PASSWORD': '1234',
        'HOST': '127.0.0.1',
        'PORT': '3306'
    }
}
```

Creamos una nueva base de datos para conectar los ajustes usando **mariadb**.

```
Crear nueva DB:
sudo su
mariadb
CREATE DATABASE mysite;
CREATE USER 'user_mysite'@'localhost' IDENTIFIED BY '1234';
GRANT ALL PRIVILEGES ON mysite.* TO 'user_mysite'@'localhost' WITH GRANT OPTION;
```

De nuevo en **settings.py** (línea 112), cambiamos la zona horaria:

```
112 TIME_ZONE = 'Europe/Madrid'
```

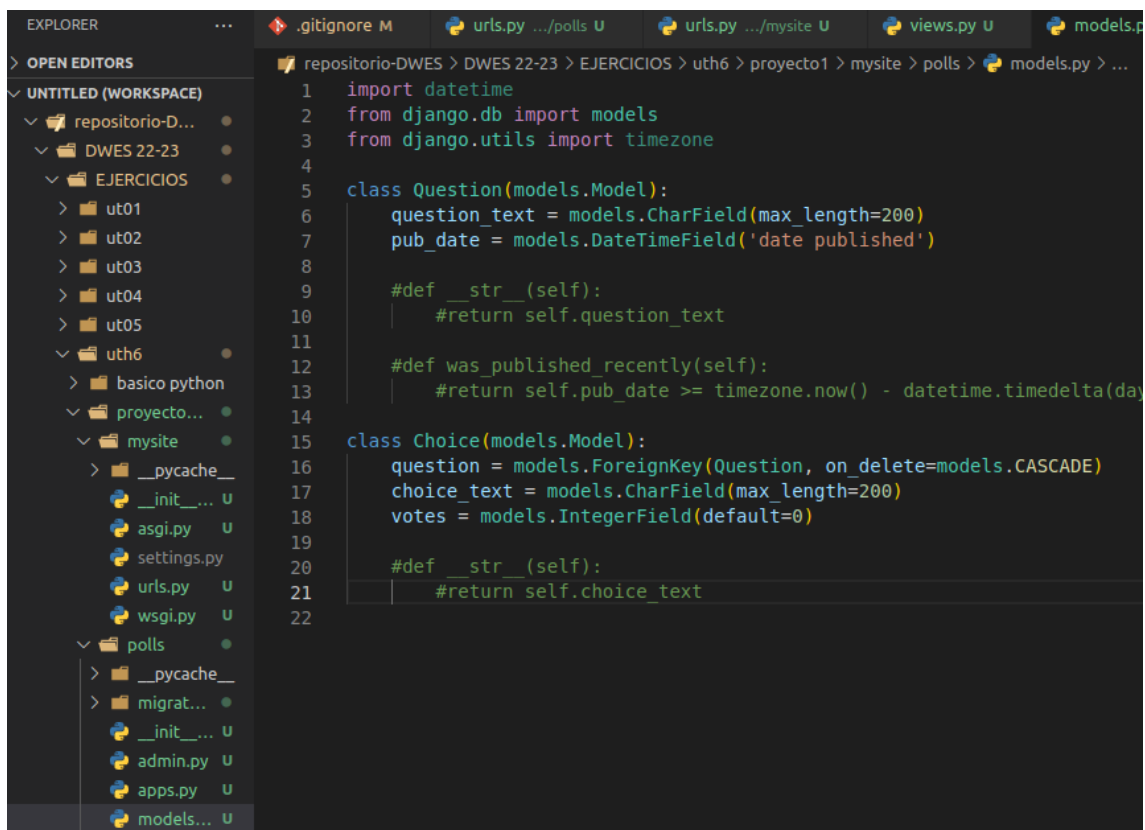
Ahora deberíamos instalar mysqlclient en caso de que no lo tengamos instalado (**terminal**):

```
apt update
sudo apt-get install python3-dev default-libmysqlclient-dev build-essential
pip install mysqlclient
```

En la ruta correcta **/mysite/** debemos usar el siguiente comando para migrar los cambios a **manage.py**:

```
mysite:
python manage.py migrate
```

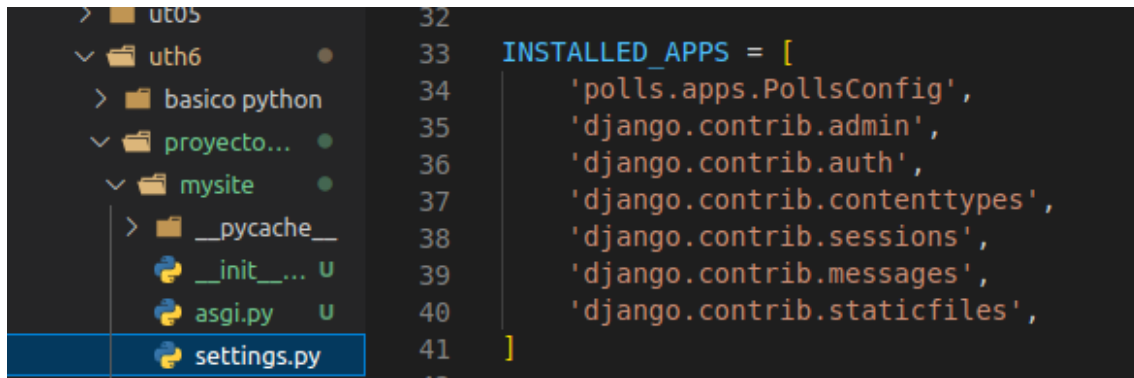
El siguiente paso sería añadir distintos modelos (objetos) para poder usarlos, en este caso he creado *Question* y *Choice* con las características que queramos. No olvidemos importar lo necesario.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'ut01' through 'ut05', 'uth6', 'basico python', 'proyecto...', 'mysite', and 'polls'. The code editor shows the content of 'models.py' in the 'polls' folder. The code defines two Django models: 'Question' and 'Choice'. 'Question' has a 'question_text' field (CharField, max_length=200) and a 'pub_date' field (DateTimeField, 'date published'). 'Choice' has a 'question' field (ForeignKey to 'Question', on_delete=models.CASCADE), a 'choice_text' field (CharField, max_length=200), and a 'votes' field (IntegerField, default=0). Both models have a '__str__' method that returns the respective text field.

```
1 import datetime
2 from django.db import models
3 from django.utils import timezone
4
5 class Question(models.Model):
6     question_text = models.CharField(max_length=200)
7     pub_date = models.DateTimeField('date published')
8
9     #def __str__(self):
10     |     #return self.question_text
11
12     #def was_published_recently(self):
13     |     #return self.pub_date >= timezone.now() - datetime.timedelta(days=1)
14
15 class Choice(models.Model):
16     question = models.ForeignKey(Question, on_delete=models.CASCADE)
17     choice_text = models.CharField(max_length=200)
18     votes = models.IntegerField(default=0)
19
20     #def __str__(self):
21     |     #return self.choice_text
22
```

En la línea 33 modificamos **INSTALLED_APPS** y añadimos lo siguiente:



```
32
33 INSTALLED_APPS = [
34     'polls.apps.PollsConfig',
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 ]
```

Para que los cambios en **models.py** se guarden hay que ejecutar el siguiente comando en la **terminal** (con la ruta correcta):

```
consola mysite:
python3 manage.py makemigrations polls
```

Con python3 **manage.py Shell** iniciamos la terminal interactiva donde debemos importar los modelos que hemos añadido (*Question* y *Choice*). También importaremos *timezone*. En la ultima línea podemos ver como se ha creado un objeto de tipo *Question* (con sus correspondientes atributos) y se ha guardado.

```
python3 manage.py shell
from polls.models import Choice, Question (or *)
Question.objects.all()
from django.utils import timezone
objeto_pregunta(Question(question_text="Whats new?", pub_date=timezone.now()))
objeto_pregunta.question_text="primera pregunta"
objeto_pregunta.save()
Question(question_text="al vuelo?", pub_date=timezone.now()).save()
```

En **models.py** añadimos las siguientes funciones a *Question* y *Choice*:

```
.gitignore M  urls.py .../polls U  urls.py .../mysite U X  views.py U  models.py U X
repositorio-DWES > DWES 22-23 > EJERCICIOS > uth6 > proyecto1 > mysite > polls > models.py > Question
1  import datetime
2  from django.db import models
3  from django.utils import timezone
4
5  class Question(models.Model):
6      question_text = models.CharField(max_length=200)
7      pub_date = models.DateTimeField('date published')
8
9      def __str__(self):
10         return self.question_text
11
12     def was_published_recently(self):
13         return self.pub_date >= timezone.now() - datetime.timedelta(days=1)
14
15     class Choice(models.Model):
16         question = models.ForeignKey(Question, on_delete=models.CASCADE)
17         choice_text = models.CharField(max_length=200)
18         votes = models.IntegerField(default=0)
19
20     def __str__(self):
21         return self.choice_text
22
```

Esta es la forma de crear un superusuario para poder acceder al cliente del servidor.

```
consola (mysite):
python manage.py createsuperuser
admin
perritomundial12
```

Modificamos el archivo **admin.py** de la siguiente forma:

```
.gitignore M  urls.py .../polls U  urls.py .../mysite U  views.py U  models.py U X
repositorio-DWES > DWES 22-23 > EJERCICIOS > uth6 > proyecto1 > mysite > polls > admin.py
1  from django.contrib import admin
2  from .models import Question
3
4  # Register your models here.
5  admin.site.register(Question)
```

Para correr el servidor usamos el siguiente comando: **python3 manage.py runserver**

Algunas otras opciones que tenemos en **models.py** son estas:

```
.gitignore M  views.py U  urls.py ../videojuegos U  models.py U X  settings.py  _str_
repositorio-DWES > DWES 22-23 > EJERCICIOS > uth6 > videojuegos > esports > models.py > Juego > __str__
1  from django.db import models
2
3  class Genero(models.Model):
4      nombre = models.CharField(max_length=50)
5      descripcion = models.TextField()
6
7      #def __str__(self):
8      |   #return f"{self.nombre} ({self.anio})"
9
10
11 class Equipo(models.Model):
12     nombre = models.CharField(max_length=50)
13     descripcion = models.TextField()
14     anio = models.DateField()
15
16     def __str__(self):
17     |   return f"{self.nombre} ({self.anio})"
18
19
20 class Juego(models.Model):
21     nombre = models.CharField(max_length=50)
22     descripcion = models.TextField()
23     anio = models.DateField()
24     genero = models.ForeignKey(Genero, on_delete=models.SET_NULL, null=True)
25     equipos = models.ManyToManyField(Equipo)
26
27     def __str__(self):
28     |   return f"{self.nombre} [{self.genero}]({self.anio})"
```

En el caso de arriba para que los modelos funcionen hay que añadir lo siguiente a **admin.py**

```
.gitignore M  views.py U  urls.py ../esports U  urls.py ../videojuegos U  models.py U  settings.py  admin.py U X
repositorio-DWES > DWES 22-23 > EJERCICIOS > uth6 > videojuegos > esports > admin.py
1  from django.contrib import admin
2  from .models import Equipo, Juego, Genero
3
4  # Register your models here.
5
6  admin.site.register(Equipo)
7  admin.site.register(Juego)
8  admin.site.register(Genero)
```

Instalamos las **django_extensions**:

```
pip install django-extensions
```

La añadimos a las **INSTALLED_APPS**:

```
INSTALLED_APPS = [  
    'django_extensions',  
    'esports.apps.EsportsConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

Hemos realizado los pasos anteriores para poder ejecutar la terminal interactiva **shell_plus**:

```
arturo@arturo-VirtualBox:~/Desktop/repositorio-DWES/DWES 22-23/EJERCICIOS/uth6/videojuegos$ python manage.py shell_plus
# Shell Plus Model Imports
from django.contrib.admin.models import LogEntry
from django.contrib.auth.models import Group, Permission, User
from django.contrib.contenttypes.models import ContentType
from django.contrib.sessions.models import Session
from esports.models import Equipo, Genero, Juego
# Shell Plus Django Imports
from django.core.cache import cache
from django.conf import settings
from django.contrib.auth import get_user_model
from django.db import transaction
from django.db.models import Avg, Case, Count, F, Max, Min, Prefetch, Q, Sum, When
from django.utils import timezone
from django.urls import reverse
from django.db.models import Exists, OuterRef, Subquery
Python 3.10.6 (main, Nov 14 2022, 16:10:14) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```