

 tutorial\_django.md

# Tutorial Django

---

El objetivo de este tutorial es desarrollar una aplicación web de un establecimiento de adopción de mascotas.

## 1. Instalar Django

Para instalar django en la consola (cmd) poner el siguiente código

```
pip install django
```

## 2. Crear un nuevo proyecto de Django

Para empezar con el desarrollo de un proyecto bajo el framework de Django, primero debemos de inicializar un proyecto.

Para esto, en **cmd**, dentro de una carpeta, en la cual se creará el proyecto poner el siguiente código (que creará el proyecto wisdompets, así estamos llamando al proyecto):

```
python-django startproject wisdompets
```

Esto creará una carpeta con el nombre del Proyecto (wisdompets) en la carpeta donde se corrió el comando indicado.

## 3. Crear una aplicación dentro del proyecto

La primera etapa es crear una aplicación (se le llamará adoptions). Para esto primero se creará el app y luego se tiene que registrar el app, en el file **settings.py**

Para crear el app, dentro de la carpeta wisdompets poner el siguiente código:

```
python manage.py startapp adoptions
```

Luego, dentro del file **settings.py**, dentro de la lista *INSTALLED\_APPS* incluir el nombre de la aplicación

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'adoptions',  
]
```

Recuerden guardar siempre el código que están modificando (Ctrl+S)

## 4. Crear las bases de datos

Primero se creará la estructura de la base de datos. Para esto, dentro del file **models.py** dentro de la carpeta **adoptions**, se crearán dos clases, cada una representará las dos bases de datos que se generarán. Para esto se incluirá el siguiente código dentro del file **models.py**

```
class Pet(models.Model):  
    SEX_CHOICES = [('M', 'Male'), ('F', 'Female')]  
    name = models.CharField(max_length=100)  
    submitter = models.CharField(max_length=100)  
    species = models.CharField(max_length=30)  
    breed = models.CharField(max_length=30, blank = True)  
    description = models.TextField()  
    sex = models.CharField(choices=SEX_CHOICES, max_length=1, blank = True)  
    submission_date = models.DateTimeField()  
    age = models.IntegerField(null=True)  
    vaccinations = models.ManyToManyField('Vaccine', blank=True)
```

```
class Vaccine(models.Model):  
    name = models.CharField(max_length=50)  
    def __str__(self):  
        return self.name
```

Luego de crear la estructura de las bases de datos se deberá de "decir" a django que efectúe esos cambios. A esta acción se le llama "migración". Para hacer las migraciones en la consola (cmd) dentro del proyecto (carpeta wisdompets) se deberá de poner lo siguiente

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

El comando showmigrations nos dirá el estado de las migraciones.

Luego de ejecutar estos códigos, se crearán las bases de datos. Una forma de poder ver estas bases de datos es usando una aplicación como **sqlite.org** Para introducir información dentro de esta base de datos se usará una función ya creada y data que se encuentra en un csv (archivo delimitado por comas).

Para esto, dentro de la carpeta de github de **carlosgl87** ([https://github.com/carlosgl87/desarrollo\\_soluciones\\_web/tree/master/clase4](https://github.com/carlosgl87/desarrollo_soluciones_web/tree/master/clase4)) extraer la carpeta **management** y el file **pet\_data.csv**. La carpeta **management** copiarla dentro del proyecto, dentro de la carpeta **adoptions** y el file **pet\_data.csv** copiarlo dentro de la carpeta inicial del proyecto.

Una vez copiado la carpeta y el archivo, en línea de comando, poner:

```
python manage.py load_pet_data
```

Con esto se insertará la información del csv en las bases de datos que has creado. Puedes verificarlo en el programa sqlite browser que te descargaste.

## 5. Administrar el Proyecto

Para poder crear un usuario que permita administrar el proyecto que estás creando, dentro del file ***admin.py*** que se encuentra en la carpeta ***adoptions*** incluir:

```
from .models import Pet
@admin .register(Pet)
class PetAdmin(admin.ModelAdmin):
    list_display = ['name', 'species', 'breed', 'age', 'sex']
```

Después tienes que crear un superusuario, para esto en la consola poner:

```
python manage.py createsuperuser
```

Al crear el usuario te pedirá un usuario y una contraseña.

Después, para ver si dió resultados puedes ir entrando a la página web, para esto poner en consola:

```
python manage.py runserver
```

Esto abrirá un puerto local (por defecto el 8000) y puedes entrar a este puerto poniendo en un browser como Chrome:

<http://localhost:8000/admin>

## 6. Crear la estructura de la página desarrollo\_soluciones\_web

Ahora empezaremos a crear la estructura lógica de la página web. Primero definiremos los patrones de la página web. Para esto en el file ***urls.py*** primero definiremos los patrones que podrán tener los urls que busquemos en el browser. Para esto pondremos el siguiente código:

```
from django.urls import path, re_path
from adoption import views

urlpatterns = [
    path('admin/', admin.site.urls),
```

```
re_path(r'^$', views.home, name='index'),  
re_path(r'adoptions/(\d+)/', views.pet_details, name='pet_detail'),  
]
```

Luego en el file **views.py** tenemos que crear los views a los que se dirigirá la página cuando busquemos en el browser el url.

```
from django.shortcuts import render  
from django.http import HttpResponse  
from django.http import Http404  
from .models import Pet  
  
# Create your views here.  
def home(request):  
    pets = Pet.objects.all()  
    return render(request, 'home.html', {'pets': pets})  
  
def pet_detail(request, id):  
    try:  
        pet = Pet.objects.get(id=id)  
    except Pet.DoesNotExist:  
        raise Http404('Pet not found')  
    return render(request, 'pet_detail.html', {'pet': pet})
```

Finalmente, estos views tendrán que llamar a unos htmls y algunas imágenes. Para esto copia algunos elementos en tu carpeta de trabajo del github ([https://github.com/carlosgl87/desarrollo\\_soluciones\\_web/tree/master/clase4](https://github.com/carlosgl87/desarrollo_soluciones_web/tree/master/clase4)).

La carpeta **template** copiala dentro de la carpeta adoptions y la carpeta **static** copiala en la carpeta inicial.

Una vez con todos estos cambios ya estará listo el proyecto.