



HOLA MUNDO - BIOS

PYTHON+DIJANGO

En este tutorial exploraremos el proceso de creación de una aplicación web utilizando **Django**, abordando el levantamiento y la conexión con una base de datos **PostgreSQL**, así como la configuración y ejecución de **Docker**. Finalmente, desarrollaremos un “**Hola Mundo**”

Lenguaje y Framework:

- **Lenguaje:** Python 3.12
- **Framework:** Django 5.x
- **ORM:** ORM nativo de Django
- **Base de datos:** PostgreSQL
- **Interfaz:** Plantilla HTML

Crear entorno de trabajo

```
mkdir proyecto_animales  
cd proyecto_animales  
python -m venv venv  
source venv/bin/activate
```

Instalar dependencias

```
pip install django psycopg2-binary
```

Crear proyecto base

```
django-admin startproject core .  
python manage.py startapp animales
```

Configuración de la base de datos en PostgreSQL



```
version: '3.8'

services:
  db:
    image: postgres:15
    container_name: postgres_django
    restart: always
    environment:
      POSTGRES_DB: animales_db
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: admin123
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data

volumes:
  postgres_data:
```

```
docker-compose up -d
```

Conexión con PostgreSQL

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'animales_db',
        'USER': 'admin',
        'PASSWORD': 'admin123',
        'HOST': '127.0.0.1',
        'PORT': '5432',
    }
}
```



Registrar app

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

Definir entidades

```
from django.db import models  
from django.contrib.auth.models import User  
  
class Animal(models.Model):  
    nombre = models.CharField(max_length=100)  
    especie = models.CharField(max_length=50)  
    raza = models.CharField(max_length=50)  
    edad = models.PositiveIntegerField()  
    peso = models.FloatField()  
    propietario = models.ForeignKey(User, on_delete=models.CASCADE)  
  
    def __str__(self):  
        return f"{self.nombre} ({self.especie})"
```

Pasar a la BD

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



Formulario de registro

```
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm

class RegistroForm(UserCreationForm):
    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2']
```

Registro y login de Usuario

```
from django.shortcuts import render, redirect
from django.contrib.auth import login, authenticate
from django.contrib.auth.forms import AuthenticationForm
from .forms import RegistroForm
from .models import Animal

def registro_usuario(request):
    if request.method == 'POST':
        form = RegistroForm(request.POST)
        if form.is_valid():
            user = form.save()
            login(request, user)
            return redirect('listar_animales')
    else:
        form = RegistroForm()
    return render(request, 'registro.html', {'form': form})
```

```
def login_usuario(request):
    if request.method == 'POST':
        form = AuthenticationForm(data=request.POST)
        if form.is_valid():
            user = form.get_user()
            login(request, user)
            return redirect('listar_animales')
    else:
        form = AuthenticationForm()
    return render(request, 'login.html', {'form': form})
```



CRUD

```
from django.contrib.auth.decorators import login_required

@login_required
def listar_animales(request):
    animales = Animal.objects.filter(propietario=request.user)
    return render(request, 'listar.html', {'animales': animales})

@login_required
def crear_animal(request):
    if request.method == 'POST':
        nombre = request.POST['nombre']
        especie = request.POST['especie']
        raza = request.POST['raza']
        edad = request.POST['edad']
        peso = request.POST['peso']
        Animal.objects.create(
            nombre=nombre,
            especie=especie,
            raza=raza,
            edad=edad,
            peso=peso,
            propietario=request.user
        )
        return redirect('listar_animales')
    return render(request, 'crear.html')
```

HTML

1. Registro

```
<h2>Registro de Usuario</h2>
<form method="post">{% csrf_token %}{{ form.as_p }}<button>Registrarse</button></form>
```

2. Login

```
<h2>Iniciar Sesión</h2>
<form method="post">{% csrf_token %}{{ form.as_p }}<button>Entrar</button></form>
```



3. Crear

```
<h2>Mis Animales</h2>
<ul>
{% for a in animales %}
<li>{{ a.nombre }} - {{ a.peso }} kg</li>
{% empty %}
<li>No tienes animales registrados.</li>
{% endfor %}
</ul>
<a href="{% url 'crear_animal' %}">Agregar Animal</a>
```

```
<h2>Nuevo Animal</h2>
<form method="post">{% csrf_token %}
Nombre: <input name="nombre"><br>
Especie: <input name="especie"><br>
Raza: <input name="raza"><br>
Edad: <input name="edad"><br>
Peso: <input name="peso"><br>
<button>Guardar</button>
</form>
```

Rutas

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.login_usuario, name='login'),
    path('registro/', views.registro_usuario, name='registro'),
    path('animales/', views.listar_animales, name='listar_animales'),
    path('animales/nuevo/', views.crear_animal, name='crear_animal'),
]
```



Ejecución Final

Navegador: <http://127.0.0.1:8000/>

```
python manage.py runserver
```