

# Guia mostrar contenido dinámico desde un archivo SQLite3 a una app web

## ABPro Proyecto Grupal 3 - Equipo 3

- ❖ Al ser la continuación de la guía anterior, comenzaremos desde el punto 13 asumiendo que los otros pasos ya fueron realizados.

### 13. Configuración de la base de datos:

- Abrir el archivo settings.py en la carpeta “my-project/my-project”, buscar la sección “DATABASES” y configurar la base de datos SQLite3 estableciendo el valor de ENGINE en 'django.db.backends.sqlite3' y el nombre de la base de datos en un archivo SQLite3 existente o nuevo. Por ejemplo:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

### 14. Crear una clase modelo en el archivo models.py dentro de la aplicación creada anteriormente:

```
class Usuario(models.Model):
    nombre = models.CharField(max_length=100)
    apellido = models.CharField(max_length=100)
    correo = models.EmailField()
    telefono = models.CharField(max_length=20)
    dirección = models.CharField(max_length=100)
    Ciudad = models.CharField(max_length=100)

    def __str__(self):
        return self.nombre
```

**15. Crear en consola las migraciones para el nuevo modelo ejecutando el siguiente comando:**

```
python manage.py makemigrations
```

**16. Aplicar en consola las migraciones a la base de datos con el siguiente comando:**

```
python manage.py migrate
```

**17. Registra el modelo de Usuario en el archivo admin.py de la aplicación:**

```
from django.contrib import admin
from .models import Usuario

admin.site.register(Usuario)
```

**18. Crear un “superuser” en consola y registrar usuarios en el panel de Administración de Django:**

- Crea un superusuario para acceder al panel de administración de Django usando el siguiente comando:

```
python manage.py createsuperuser
```

- Sigue las instrucciones en la terminal para ingresar un nombre de usuario, dirección de correo electrónico y contraseña para el superusuario (Ojo: por motivos de seguridad, la contraseña no será visible al momento de escribir en la consola).
- Abre un navegador web y accede a “http://localhost:8000/admin” para iniciar sesión en el panel de administración usando las credenciales del superusuario creadas anteriormente.
- Registra los usuarios que necesites en tu aplicación.

**19. Crear la vista que mostrará el contenido dinámico:**

- En el archivo “views.py” en la aplicación, agregar una vista para mostrar el contenido dinámico.

```
from django.shortcuts import render
from .models import MyModel

def lista_usuarios(request):
```

```
usuarios = Usuario.objects.all()
return render(request, 'usuarios.html', {'usuarios': usuarios})
```

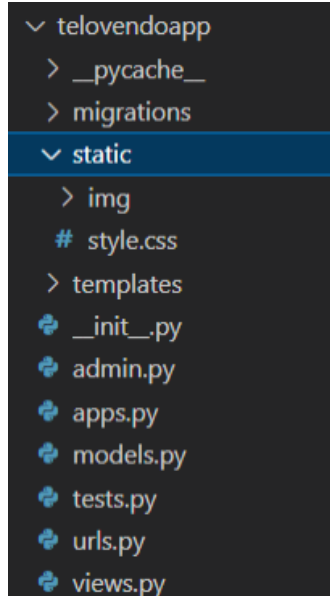
**20. Configuración de URLs en el archivo “urls.py” dentro de la aplicación, agregar una URL para la vista creada:**

```
from django.urls import path
from . import views
from .views import lista_usuarios

urlpatterns = [
    path('usuarios/', lista_usuarios,
    name='lista_usuarios') ]
```

**21. Creación de la carpeta de contenido “static” y ruta de los nuevos archivos “html” que se crearán a continuación:**

- En el interior de la carpeta de la aplicación, crear una nueva carpeta llamada “static”, la cual contendrá en su interior el archivo “style.css” con los estilos css de la aplicación y otra carpeta “img” para el contenido de imágenes del proyecto.



- Verificar que el contenido estático este registrado en el archivo “settings.py” a nivel del proyecto, en la sección “INSTALLED\_APPS”.

```
INSTALLED_APPS = [
    'django.contrib.admin',
```

```

'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'telovendoapp'
]

```

- Aún dentro de “settings.py” establecer rutas de archivos y directorios dentro del proyecto de la siguiente manera .

```

from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
PROJECT_APP_PATH = os.path.dirname(os.path.abspath(__file__))

```

## 22. Diseño del nuevo template “base.html”:

- Crea un archivo base de plantilla HTML llamado “base.html” dentro de la carpeta “templates” de la aplicación. Este archivo contendrá el diseño general de tu aplicación y se puede personalizar según se necesite (recordar cargar “static”, “style.css” y elementos de bootstrap u otras librerías a usar)

```

{% load static %}

<!DOCTYPE html>
<html lang="es">

<head>
<meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Te Lo Vendo - Venta de Productos Electrónicos</title>
    <!-- Bootstrap y CSS -->
    <link href="{% static 'style.css' %}" rel="stylesheet"
type="text/css">
    <link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstr
ap.min.css">
</head>
<body>

    {% block header%}
    ... contenido del header ...

```

```

{% endblock%}
{% block navbar%}
... contenido del navbar ...
{% endblock %}
{% block content %}
{% block img %}
{% endblock %}
{% block lista_usuarios %}
{% endblock %}
{% endblock%}
{% block carrusel%}
... contenido del header ...
{% endblock%}
{% block tarjetas%}
... contenido del header ...
{% endblock%}
{% block footer %}
... cointenido del footer ...
{% endblock %}
<!-- Bootstrap JS -->
<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

### 23. Modificación de index.html:

- Al archivo “index.html” creado anteriormente, meterlo dentro de una nueva carpeta dentro de “templates” llamada “telovendoapp” y modificarlo para definir los bloques que va a utilizar, extendiendo contenido desde la “base.html”.

```
{% extends "base.html" %}

{% block carrusel %}
{% load static%}
... contenido block carrusel(de bootstrap) ...
{% endblock %}

{% block tarjetas %}
{% load static%}
... contenido block cards(de bootstrap) ...
{% endblock %}
```

### 24. Creación página “usuarios.html”:

- Dentro de la carpeta “telovendo app” (en “templates”) crear un tercer archivo llamado “usuarios.html”, y como en el caso anterior configurar para extender de base y dar contenido a los bloques que va a utilizar.

```
{% extends "base.html" %}

{% block content %}
{% block img %}
{% load static %}
... contenido de imagen...
{% endblock %}
{% block lista_usuarios %}
<div class="card text-left m-5 rounded-6">
  <div class="card-body rounded-6">
    <h3>Lista de Usuarios Registrados:</h3>
    <ul class="list-group">
      {% for usuario in usuarios %}
        <li class="list-group-item">
          {{ usuario.nombre }} {{ usuario.apellido }} </li>
        {% endfor %}
      </ul>
    </div>
  </div>
{% endblock %}
{% endblock %}
```

- Se destaca en **negrita** el bucle “for” que itera sobre los usuarios registrados y los muestra en pantalla y en **azul** dentro de este bucle, la llamada que se hace para traer los atributos dinámicos de cada usuario

## 25. Ejecutar por consola el servidor de Django:

```
python manage.py runserver
```

## 26. Ver el contenido dinámico::

- Abrir en el navegador web la dirección “http://localhost:8000/usuarios/”
- El resultado es ver el template renderizado en el navegador, mostrando la lista de usuarios registrados en el panel de control de la aplicación

*¡Esperamos que esta guía sea de tu utilidad, para más detalles contactanos via Slack!  
¡Te esperamos en alguna siguiente guía!*

