

# Documentación del archivo settings.py

## Configuraciones generales

### 1. Importación de módulos

**Pathlib:** Facilita el manejo de rutas en el sistema de archivos.

**os:** Proporciona funciones para interactuar con el sistema operativo, como leer variables de entorno.

**environ:** Módulo para gestionar configuraciones sensibles a través de variables de entorno (útil para evitar exponer información sensible en el código).

### 2. Configuración de env

```
env = environ.Env()
```

```
environ.Env.read_env()
```

Carga las variables de entorno desde un archivo `.env`. Esto permite mantener separada la configuración del código fuente, especialmente para datos sensibles como contraseñas o claves.

### 3. Base del proyecto

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

Define la ruta base del proyecto para facilitar el uso de rutas relativas en todo el archivo de configuración.

### 4. Claves y configuraciones de seguridad

```
SECRET_KEY = os.environ.get('SECRET_KEY')
```

**DEBUG = os.environ.get('DEBUG')**

**ALLOWED\_HOSTS = env.list('ALLOWED\_HOST\_DEV')**

**SECRET\_KEY:** Clave utilizada por Django para firmar datos y garantizar la seguridad. Se lee desde variables de entorno para mayor seguridad.

**DEBUG:** Define si la aplicación está en modo de depuración. Debe estar desactivado en producción.

**ALLOWED\_HOSTS:** Lista de dominios/IPs permitidos para servir la aplicación.

## **Aplicaciones**

### **5. Instalación de aplicaciones**

**DJANGO\_APPS:** Estas son las aplicaciones principales de Django, necesarias para la funcionalidad base (autenticación, manejo de sesiones, administración, etc.).

**PROJECT\_APPS:** Aplicaciones personalizadas del proyecto.

**THIRD\_PARTY\_APPS:** Librerías externas como:

**corsheaders:** Manejo de CORS (Cross-Origin Resource Sharing).

**rest\_framework:** Django REST Framework, para construir APIs.

**ckeditor:** Editor de texto enriquecido.

**ckeditor\_uploader:** Extensión de CKEditor para subir archivos.

### **Configuración de CKEditor**

**CKEDITOR\_CONFIGS:** Configura las herramientas visibles en la barra del editor.

**CKEDITOR\_UPLOAD\_PATH:** Ruta donde se guardarán los archivos

subidos por CKEditor.

## Middleware

**Middleware:** Capas de software que procesan solicitudes/respuestas.

Por ejemplo:

**corsheaders.middleware.CorsMiddleware:** Manejo de solicitudes CORS.

**django.middleware.csrf.CsrfViewMiddleware:** Protección contra ataques CSRF.

## Bases de datos

**DATABASES:** Configuración inicial para usar SQLite como base de datos. En producción, se recomienda cambiar a PostgreSQL o MySQL.

## Archivos estáticos y multimedia

**STATIC\_ROOT:** Carpeta donde se recopilan archivos estáticos durante la implementación.

**MEDIA\_ROOT:** Carpeta donde se guardan archivos subidos por usuarios.

**STATICFILES\_DIRS:** Carpetas adicionales que contienen archivos estáticos durante el desarrollo.

## Configuración de REST Framework

**Define las políticas de permisos para APIs. Aquí, las solicitudes autenticadas tienen acceso completo, mientras que las no autenticadas solo pueden leer datos.**

## **CORS y CSRF**

**CORS\_ORIGIN\_WHITELIST:** Lista de dominios permitidos para solicitudes de origen cruzado.

**CSRF\_TRUSTED\_ORIGIN:** Lista de dominios confiables para solicitudes protegidas por CSRF.

## **Configuración de correo**

**EMAIL\_BACKEND:** Backend de correo para desarrollo. Solo imprime los correos en consola. Cambiar a un backend real en producción.

## **Producción**

**Ajusta configuraciones clave para entornos de producción:**

**ALLOWED\_HOSTS:** Dominios permitidos en producción.

**DATABASES:** Cambia a una base de datos configurada desde **DATABASE\_URL**.