

CASO DE ESTUDIO: PROYECTO

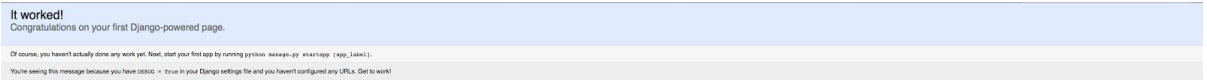
Un agente de tránsito diariamente realiza unas 40 multas por infracciones por toda la ciudad, pero esto lo hace de manera manual escribiendo en un cuaderno los datos de la persona, los vehículos implicados, la dirección dónde ocurrió y toda la información relacionada a esto (observaciones), posteriormente en la noche, él debe ingresar todas estas infracciones al sistema web, esto es una tarea de doble trabajo y bastante tediosa. Por este motivo, se requiere realizar una aplicación web en Django, que permita al agente de tránsito ingresar con un usuario y contraseña, a una vista donde pueda realizar dos tareas:

- Visualizar todas las multas realizadas.
- Ingresar una nueva multa.

También, se le debe permitir a una persona normal (usuario general), ver todas las multas que se le ha realizado.

CONSTRUCCIÓN DE LA PLATAFORMA

1. Creación de un entorno virtual que se denominará: **transito_app_env**
2. Instalación de Django
 - a. Recomendación: Utilizar Python 3
3. Creación del proyecto **plataforma_transito**
4. Ejecutar el proyecto para visualizar la pantalla inicial de Django para comprobar la correcta instalación de Django:



5. Generar las migraciones y migrar los modelos iniciales de Django a la base de datos. (Ver punto 9.b.ii)
6. Configuración de la dirección de la carpeta **templates**:
 - a. Esto se logra ingresando **os.path.join(BASE_DIR, 'templates')** dentro del apartado de **DIRS** de la variable **TEMPLATES** dentro del archivo **settings.py**.
7. Crear las aplicaciones que se usarán dentro de la aplicación (Véase: Aplicaciones a usar dentro de la plataforma).
8. Crear los modelos dentro de las aplicaciones correspondientes.
9. Generar las migraciones y migrar los modelos a la base de datos.
 - a. Nota: Para este proyecto se usará una base de datos sqlite3 que es el gestor de bases de datos que viene por defecto.
 - b. Comandos:
 - i. Crear migraciones: **python manage.py makemigrations**
 - ii. Migrar modelos: **python manage.py migrate**
10. Registrar los modelos dentro del administrador de Django.
 - a. Cada aplicación tiene varios archivos, entre estos tenemos **admin.py**, el cual es el encargado de registrar los modelos al administrador de Django.
11. Crear un superusuario para entrar a visualizar los modelos que se han creado.
 - a. Comando: **python manage.py createsuperuser**
12. Crear URL's y vistas para las URL dentro de cada aplicación.
 - a. Crear las url para cada una de las aplicaciones (Véase: Aplicaciones a usar dentro de la plataforma en el apartado de Url's y views).
 - b. Para poder general las url dentro del archivo **urls.py**, se le debe asignar una view, si aún no ha creado la vista para una url específica, puede usar una vista por defecto como la siguiente y posteriormente cambiarla:

```
from django.http import HttpResponse

def generalView(request):
    return HttpResponse('Esto es una vista general')
```

13. Crear el usuario para el agente.
 - a. Recomendación: Crear este usuario desde el administrador de Django, en el apartado *Authentication and authorization, Users*.
14. Se deben crear las plantillas HTML para la plataforma.

APLICACIONES A USAR DENTRO DE LA PLATAFORMA

1. Aplicación: **inicioSesion**
 - a. Modelos: En está aplicación no existirán modelos.
 - b. URL's y views:

URL	View	Descripción
/login	loginIndex	Esta url le permitirá al agente a ingresar al dashboard con usuario y contraseña, en caso de que estos campos sean correctos, se .

2. Aplicación: **cerrarSesion**

- a. Modelos: En esta aplicación no existirán modelos.
- b. URL's y views:

URL	View	Descripción
/logout	logoutIndex	Esta url le permitirá al agente cerrar sesión para luego ser redirigido a /login.

3. Aplicación: **agenteApp**

a. Modelos:

Modelo	Campos		
	Campo	Tipo de campo	Observaciones
Multa	placa_vehiculo	CharField	Debe ser un campo de máximo 6 letras o números.
	documento_persona	CharField	Debe contener como máximo 15 dígitos.
	lugar	CharField	Debe contener máximo 40 caracteres.
	fecha	DateField	default=datetime.now
	observaciones	CharField	Debe contener un máximo de 255 caracteres.

b. URL's y views:

URL	View	Descripción
/dashboard	dashboard	En esta url se encuentran las opciones de: <ul style="list-style-type: none">- Visualizar el historial de multas.- Ingresar una nueva multa.
/historial	historialMultas	Permite ver el historial de multas ingresadas al sistema.
/nuevaMulta	crearMulta	Permitirá al agente a ingresar una nueva multa al sistema con base en los campos que se encuentran en el modelo Multa

Nota: Todas estas URL pueden ser accesadas **si y sólo si** existe un usuario con sesión activa.

4. Aplicación: **usuarioGeneral**

- a. Modelos: En esta aplicación no existirán modelos.
- b. URL's y Vistas:

URL	View	Descripción
/	index	Página inicial de la aplicación en la cual se visualizará un campo de texto para el documento y un botón para búsqueda.
/multas/(?P<document>\d+)/\$	multasUsuario	Esta página recibirá la variable <i>document</i> para poder realizar la búsqueda y mostrar las multas de la persona con ese documento.