

1. Dibuje un árbol con la estructura básica de un proyecto y una aplicación de Django

```
proyecto
├── aplicacion
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py
│   ├── tests.py
│   └── views.py
├── manage.py
└── proyecto
    ├── __init__.py
    ├── __pycache__
    │   ├── __init__.cpython-36.pyc
    │   └── settings.cpython-36.pyc
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

2. Mencione tres argumentos que gestiona el comando manage.py

1. **python manage.py runserver:** Inicia el servidor web
2. **python manage.py startapp application\_name:** Crea una aplicación de Django
3. **python manage.py migrate:** Realiza las migraciones pendientes

3. Explique la diferencia de utilizar urls global y urls local

**Las url locales son solamente de una aplicación determinada, y las url globales son de todo el proyecto.**

4. Escriba un patrón de url para una vista basada en funcion

```
path('bootstrap/', include('bootstrap.urls'))
```

5. Escriba un patron de url para una vista basada en clase

```
path('list_tweet/', views.List_Tweet.as_view(),
name="List_Tweet_view")
```

6. Escriba el comando para hacer un proyecto en Django

**django-admin startproject project\_name**

7. Escriba el comando para hacer una aplicación en Django

**django-admin startapp application\_name**  
**python manage.py startapp application\_name**

8. Mencione para que sirve el archivo Settings.py

**Es el archivo de configuraciones principal del proyecto, en el se agregan las configuraciones referentes a la base de datos, aplicaciones, archivos estáticos, etc.**

9. Escriba el comando para registrar las migraciones de los modelos

**python manage.py makemigrations**

10. Escriba el comando para sincronizar el modelo con el motor de base de datos

**python manage.py migrate**