

**CESUN Universidad**  
**Dirección Académica**  
**Ingeniería en Desarrollo de Software**  
**Optativa II**



**Profesor:** Ray Brunett Parra Galaviz

**Alumno:** Carlos Armando Alcántar González

**Matricula:** MRID16007

**I. Instrucciones:** Leer detalladamente los reactivos y responder de manera correcta lo solicitado en los reactivos

1. Dibuje un árbol con la estructura básica de un proyecto y una aplicación de Django
2. Mencione tres argumentos que gestiona el comando manage.py
3. Explique la diferencia de utilizar urls global y urls local
4. Escriba un patrón de url para una vista basada en función
5. Escriba un patrón de url para una vista basada en clase
6. Escriba el comando para hacer un proyecto en Django
7. Escriba el comando para hacer una aplicación en Django
8. Mencione para qué sirve el archivo Settings.py
9. Escriba el comando para registrar las migraciones de los modelos
10. Escriba el comando para sincronizar el modelo con el motor de base de datos.

**II. Instrucciones:** Leer detalladamente la problemática y desarrollar un proyecto en Python y Django para solucionar la problemática establecida.

1.- En la Ciudad Parra's existe la National League Football Parra (NFLP), esta liga está solicitando a las casas desarrolladoras un proyecto a medida para registrar y controlar las estadísticas de la liga. La NFLP está interesada en tener los siguientes Reportes:

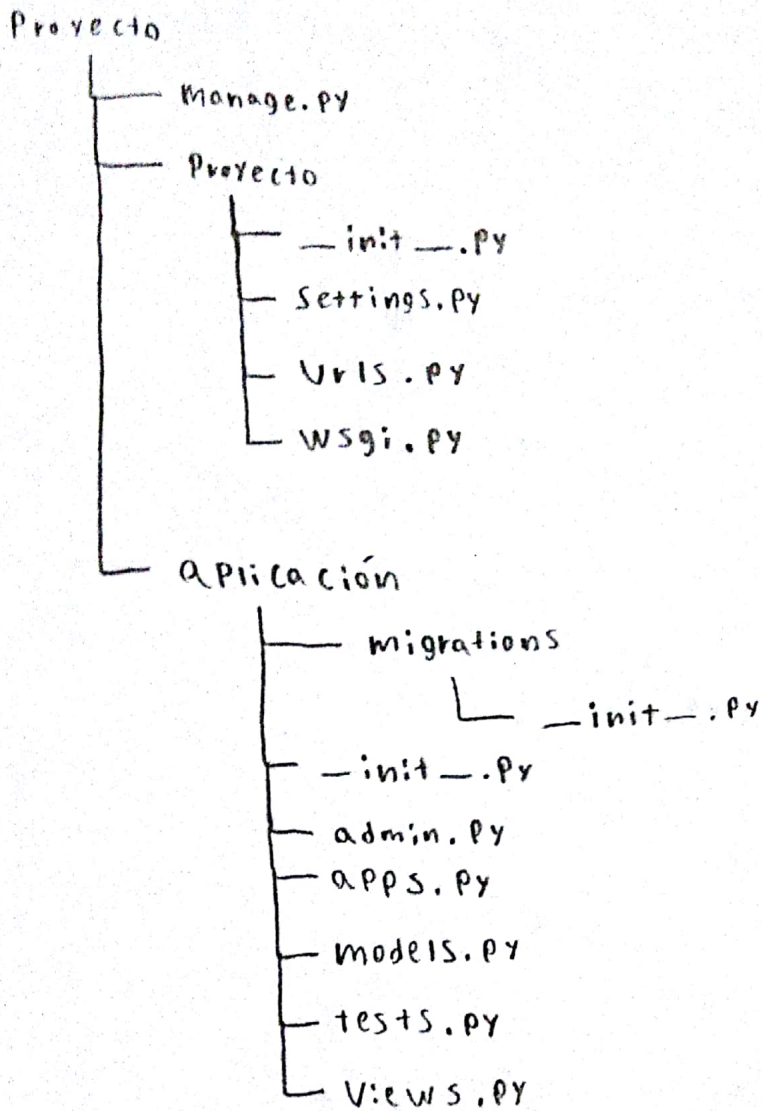
- Lista General de Jugadores
- Lista General de Equipos
- Lista General de Estadios
- Detalle de Jugadores
- Detalle de Estadios
- Detalle de Equipos
- Lista de Jugadores de un equipo proporcionado por el usuario
- Lista de todos los jugadores de una posición especificada por el usuario

**Importante:** Realizar las Vistas, Modelos, Urls, Formas y demás cosas necesarias para cumplir con la necesidad de los reportes. Los reportes deben obtener su información mediante un queryset al modelo.



# 1) Proyecto

Aplicación



2) makemigrations, migrate y runserver

3) La diferencia es que las URIs locales están en las aplicaciones que se van creando y que a su vez ayuda a modularizar las URIs para que no estén todos juntos en la URI del Proyecto y así solamente se pueda acceder a esas URIs a través de un include().

- 4) `path( '', views.index, name="index" )`
- 5) `path( 'crear/', views.crear.as_view(), name="crear" )`
- 6) `django-admin startproject Proyecto`
- 7) `django-admin startapp Aplicacion`
- 8) Sirve principalmente para dar de alta las aplicaciones que se van creando en el proyecto, sirve para conectar el proyecto de django con el motor de una base de datos, etc.
- 9) `python manage.py makemigrations`
- 10) `python manage.py migrate`