

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

1: Dibuje un arbol con la estructura basica de un proyecto y una aplicacion de Django

Project Name

- Demo
  - `__pycache__`
  - `__init__.py`
  - `settings.py`
  - `urls.py`
  - `wsgi.py`
- App Name
  - `migrations`
  - `__pycache__`
  - `admin.py`
  - `apps.py`
  - `forms.py`
  - `__init__.py`
  - `models.py`
  - `tests.py`
  - `urls.py`
  - `views.py`
- Templates
  - App Name
    - `index.html`
- `db.sqlite3`
- `manage.py`

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

`manage.py runserver` (corre el servidor)

`manage.py makemigrations` (registra nuevas migraciones de los modelos)

`manage.py migrate` (Sincroniza el modelo con el motor de base de datos)

3: Explique la diferencia de utilizar urls global y urls local;

Cuando utilizamos una URL global, estamos utilizando una url que todo el proyecto “demo” (por dar un ejemplo) puede utilizar, sin importar que app tenga dentro, y cuando utilizamos una url local, estamos diciendo que solo cierta aplicacion o cosa puede acceder a esa URL.

4: Escriba un patron url para hacer una vista basada en funcion  
`path("", views.index, name="index"),`

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

`<a href="{% url 'NFLP_app:listaCorredor' %}">Lista de jugadores con Posicion Corredor</a>`

6: Escriba el comando para hacer un proyecto en Django

`django-admin.py startproject NombreDelProyecto`

7: Escriba el comando para hacer una aplicacion en Django

`python manage.py startapp NombreDeLaApp`

8: Mencione para que sirve el archivo `Settings.py`

Es el archivo en donde podemos agregar, modificar o quitar configuraciones para nuestro proyecto

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`

## PARTE II:

Lista de jugadores de un equipo proporcionado por el usuario:

The screenshot shows a web browser window with the URL `127.0.0.1:8000/agregarEquipo/`. The page title is "NFLP - 'National League Football Parra'". There are four tabs: "Jugadores", "Equipos", "Estadios", and "Posiciones". Below the tabs are four buttons: "Agregar Jugador", "Agregar Posicion", "Agregar Equipo", and "Agregar Estadio". The "Agregar Estadio" button is active. Below the buttons, there are two input fields: "Equipo:" and "Estadio:". The "Estadio:" field contains the text "WEBEX Stadium". At the bottom of the form is a blue button labeled "Crear".

DetalleEquiposDjangoPythonIntegratobjecpythORMComo(23)W DjangoExamipythopythoMigraExam

127.0.0.1:8000/equipos/

☆M

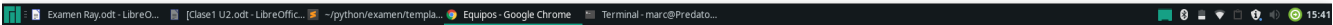
# NFLP - "National League Football Parra"

JugadoresEquiposEstadiosPosiciones

Agregar JugadorAgregar PosicionAgregar EquipoAgregar Estadio

## Lista General de Equipos!

Equipo	Actions	Detalles de los Equipos
Exentos	Jugadores en este Equipo	
Mortales	Jugadores en este Equipo	



DetalleDetalleDjangoPythonIntegratobjecpythORMComo(23)W DjangoExamipythopythoMigraExam

127.0.0.1:8000/detalleeequipo/Exentos/

☆M

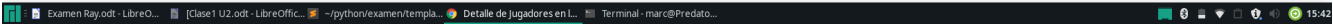
# NFLP - "National League Football Parra"

JugadoresEquiposEstadiosPosiciones

Agregar JugadorAgregar PosicionAgregar EquipoAgregar Estadio

## Detalle de Jugadores en los Equipos

Equipo	Jugadores	Actions
Exentos	El agregado	Delete
Exentos	Juan Manuel	Delete



Lista de todos los jugadores de una posicion especificada por el usuario:

Detalle

Agres

Djang

pytho

Pytho

Integr

objec

pytho

ORM

Como

(23) W

Djang

Exam

pytho

pytho

Migra

Exam

127.0.0.1:8000/agregarPosicion/

NFLP - "National League Football Parra"

JugadoresEquiposEstadiosPosiciones

Agregar JugadorAgregar PosicionAgregar EquipoAgregar Estadio

Posicion:

Corredor

Crear

Examen Ray.odt - LibreO...[Clase1 U2.odt - LibreOffic...~/python/examen/templa...Agregar Posicion - Google...Terminal - marc@Predato...15:42

Detalle

Lista

Djang

pytho

Pytho

Integr

objec

pytho

ORM

Como

(23) W

Djang

Exam

pytho

pytho

Migra

Exam

127.0.0.1:8000/posiciones/

NFLP - "National League Football Parra"

JugadoresEquiposEstadiosPosiciones

Agregar JugadorAgregar PosicionAgregar EquipoAgregar Estadio

Lista General de Posiciones

Posicion	Actions
Corredor	<a href="#">Jugadores con esta posicion</a>
Mariscal	<a href="#">Jugadores con esta posicion</a>

Examen Ray.odt - LibreO...[Clase1 U2.odt - LibreOffic...~/python/examen/templa...Lista General de Posicione...Terminal - marc@Predato...15:42

NFLP - "National League Football Parra"

Jugadores Equipos Estadios Posiciones

[Agregar Jugador](#) [Agregar Posicion](#) [Agregar Equipo](#)

### Detalle de Jugadores por posicion

Jugador	Posicion	Actions
El agregado	Corredor	<a href="#">Delete</a>

Examen Ray.odt - LibreOffice... [Clase1 U2.odt - LibreOffice... ~./python/examen/templa... Jugadores por posición - G... Terminal - marc@Predato... 15:43

Lista general de Jugadores:

← → ↻ 127.0.0.1:8000/jugadores/ 🔍 ☆ U

NFLP - "National League Football Parra"

Jugadores Equipos Estadios Posiciones

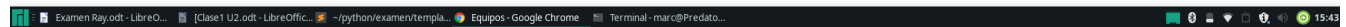
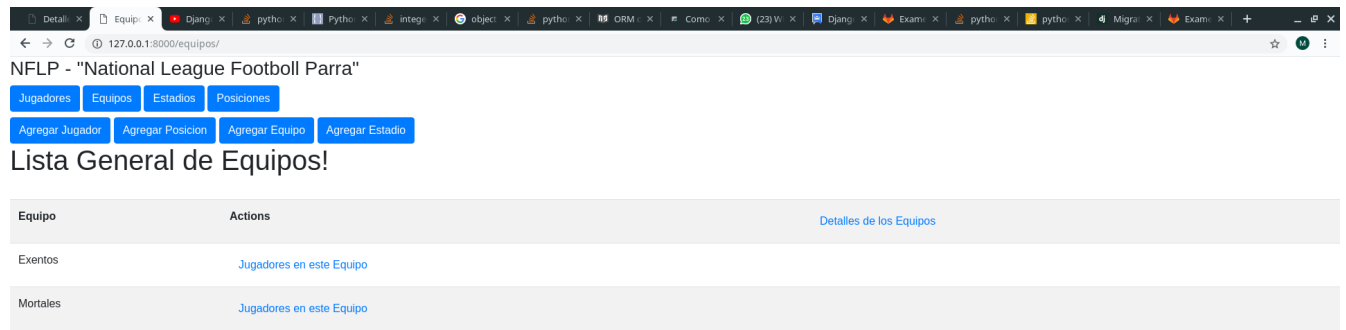
[Agregar Jugador](#)
[Agregar Posicion](#)
[Agregar Equipo](#)

## Lista General de Jugadores

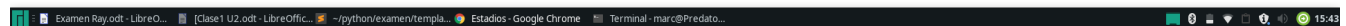
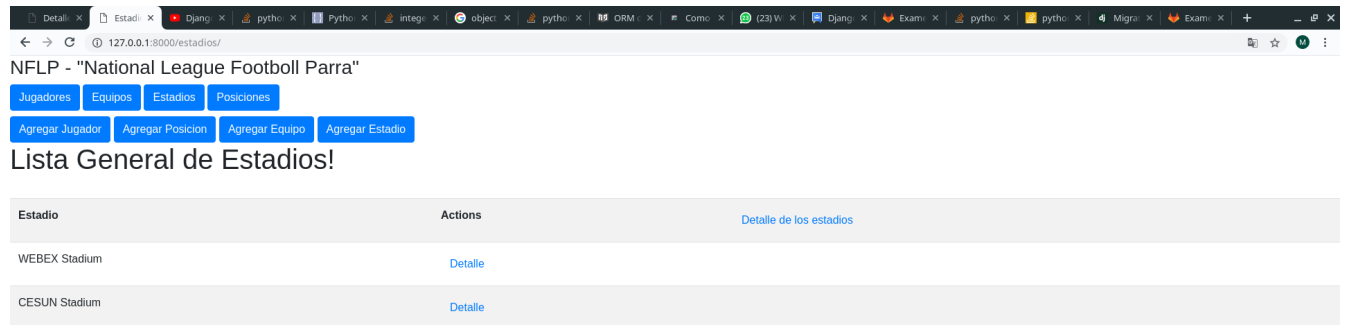
Numero	Nombre	Actions	Detalles Generales de Jugadores
1	El agregado	<a href="#">Detalle por Jugador</a>	
13	Juan Manuel	<a href="#">Detalle por Jugador</a>	
2	Mortal1	<a href="#">Detalle por Jugador</a>	

Examen Ray.odt - LibreO... [Clase1 U2.odt - LibreOffic... ~/python/examen/templa... Equipos de NFL! - Google... Terminal - marc@Predato... 15:43

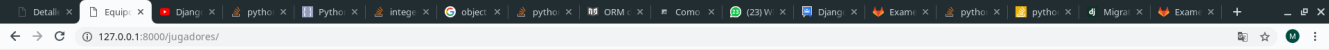
### Lista General de Equipos:



Lista General de estadios:



Detalle de Jugadores:

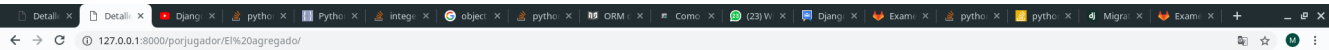
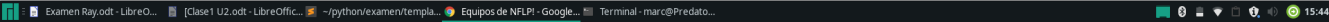


NFLP - "National League Football Parra"

- Jugadores Equipos Estadios Posiciones
- Agregar Jugador Agregar Posicion Agregar Equipo Agregar Estadio

Lista General de Jugadores

Numero	Nombre	Actions	Detalles Generales de Jugadores
1	El agregado	Detalle por Jugador	
13	Juan Manuel	Detalle por Jugador	
2	Mortal1	Detalle por Jugador	

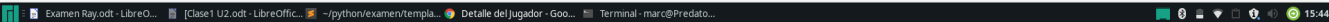


NFLP - "National League Football Parra"

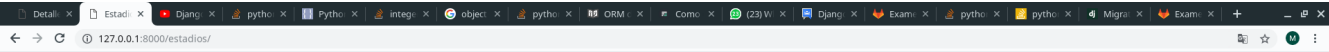
- Jugadores Equipos Estadios Posiciones
- Agregar Jugador Agregar Posicion Agregar Equipo Agregar Estadio

Detalle del Jugador

Equipo	Numero	Nombre	Fuerza	Velocidad	Resistencia	Actions
Exentos	1	El agregado	999	999	999	Delete



# Detalle de Estadios:

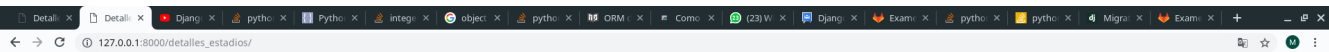
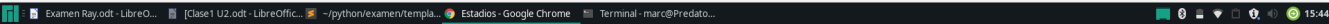


NFLP - "National League Football Parra"

Jugadores Equipos Estadios Posiciones  
Agregar Jugador Agregar Posicion Agregar Equipo Agregar Estadio

## Lista General de Estadios!

Estadio	Actions
WEBEX Stadium	<a href="#">Detalle</a>
CESUN Stadium	<a href="#">Detalle</a>

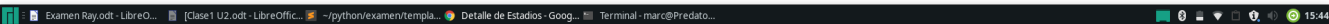


NFLP - "National League Football Parra"

Jugadores Equipos Estadios Posiciones  
Agregar Jugador Agregar Posicion Agregar Equipo Agregar Estadio

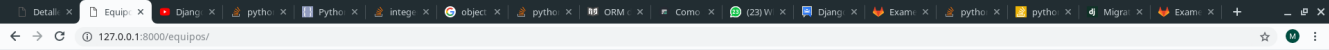
## Detalle de Estadios

Estadio	Pais	Actions
WEBEX Stadium	USA	<a href="#">Delete</a>
CESUN Stadium	USA	<a href="#">Delete</a>





# Detalle de Equipos:



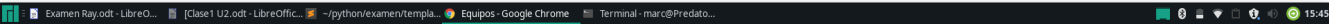
NFLP - "National League Football Parra"

Jugadores Equipos Estadios Posiciones

Agregar Jugador Agregar Posicion Agregar Equipo Agregar Estadio

## Lista General de Equipos!

Equipo	Actions	Detalles de los Equipos
Exentos	Jugadores en este Equipo	
Mortales	Jugadores en este Equipo	



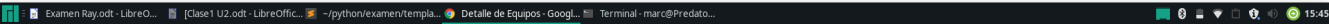
NFLP - "National League Football Parra"

Jugadores Equipos Estadios Posiciones

Agregar Jugador Agregar Posicion Agregar Equipo Agregar Estadio

## Detalle de Equipos

Equipo	Estadio	Actions
Exentos	WEBEX Stadium	Delete
Mortales	EstadioPerro	Delete



# Codigo:

```
from django.contrib import admin
from models import Jugador, Equipo, Estadio, Posicion, Numero
# Register your models here.

admin.site.register(Jugador)
admin.site.register(Equipo)
admin.site.register(Estadio)
admin.site.register(Posicion)
admin.site.register(Numero)

class JugadorAdmin(admin.ModelAdmin):
    list_display = [
        "id",
        "equipo",
        "numero",
        "nombre",
        "fuerza",
        "velocidad",
        "resistencia",
        "posicion"
    ]

class AdminEquipo(admin.ModelAdmin):
    list_display = [
        "equipo",
        "estadio"
    ]

class AdminEstadio(admin.ModelAdmin):
    list_display = [
        "estadio"
    ]
```

```
from django.db import models
from django.conf import settings

# Create your models here.

class Jugador(models.Model):
    equipo = models.ForeignKey(
        'Equipo', on_delete=models.CASCADE)
    numero = models.ForeignKey(
        'Numero', on_delete=models.CASCADE)
    nombre = models.CharField(max_length=60)
    posicion = models.ForeignKey(
        'Posicion', on_delete=models.CASCADE)
    fuerza = models.IntegerField()
    velocidad = models.IntegerField()
    resistencia = models.IntegerField()

    def __str__(self):
        return self.nombre

class Equipo(models.Model):
    equipo = models.CharField(max_length=60)
    estadio = models.ForeignKey(
        'Estadio', on_delete=models.CASCADE)

    def __str__(self):
        return self.equipo

class Estadio(models.Model):
    estadio = models.CharField(max_length=60)

    def __str__(self):
        return self.estadio

class Numero(models.Model):
    numero = models.IntegerField()

    def __int__(self):
        return int(self.numero)
```

The screenshot shows a code editor with the file explorer on the left and the code editor on the right. The file explorer shows a project structure with folders like 'examen', 'NFLP', and 'templates'. The code editor displays the 'urls.py' file, which imports 'path' from 'django.urls' and 'views' from 'NFLP'. It defines 'app\_name' as 'NFLP\_app' and lists various URL patterns for views like 'index', 'jugadores', 'equipos', 'estadios', 'detalles\_jugadores', 'detalles\_equipos', 'listaExentos', 'listaMortales', 'listaPythons', 'listaDjangueros', 'listaCorredor', 'listaMariscal', 'listaSafetie', and 'listaCornerBack'.

```
1 from django.urls import path
2
3 from NFLP import views
4
5 app_name = "NFLP_app"
6
7 urlpatterns = [
8     path('', views.index, name="index"),
9     path('jugadores/', views.jugadores, name="jugadores"),
10    path('equipos/', views.equipos, name="equipos"),
11    path('estadios/', views.estadios, name="estadios"),
12    path('detalles_jugadores/', views.detalles_jugadores, name="detalles_jugadores"),
13    path('detalles_equipos/', views.detalles_equipos, name="detalles_equipos"),
14    path('detalles_estadios/', views.detalles_estadios, name="detalles_estadios"),
15    path('listaExentos/', views.listaExentos, name="listaExentos"),
16    path('listaMortales/', views.listaMortales, name="listaMortales"),
17    path('listaPythons/', views.listaPythons, name="listaPythons"),
18    path('listaDjangueros/', views.listaDjangueros, name="listaDjangueros"),
19    path('listaCorredor/', views.listaCorredor, name="listaCorredor"),
20    path('listaMariscal/', views.listaMariscal, name="listaMariscal"),
21    path('listaSafetie/', views.listaSafetie, name="listaSafetie"),
22    path('listaCornerBack/', views.listaCornerBack, name="listaCornerBack"),
23
24 ]
```

The screenshot shows a code editor with the file explorer on the left and the code editor on the right. The file explorer shows a project structure with folders like 'examen', 'NFLP', and 'templates'. The code editor displays the 'views.py' file, which imports 'render' from 'django.shortcuts' and 'Equipo', 'Jugador', and 'Estadio' from 'models'. It defines several view functions: 'index', 'jugadores', 'detalles\_jugadores', 'equipos', and 'detalles\_equipos', each using 'render' to return a response with a context dictionary.

```
1 from django.shortcuts import render
2 from models import Equipo, Jugador, Estadio
3
4 # Create your views here.
5
6 def index(request):
7     context = {
8         "lista": [1,2,3,4,5],
9     }
10    return render(request, "NFLP/index.html", context)
11
12 def jugadores(request):
13    queryset1 = Jugador.objects.all()
14    context = {
15        "objects": queryset1
16    }
17    return render(request, "NFLP/jugadores.html", context)
18
19 def detalles_jugadores(request):
20    queryset11 = Jugador.objects.all()
21    context = {
22        "objects": queryset11
23    }
24    return render(request, "NFLP/detalle_jugador.html", context)
25
26 def equipos(request):
27    queryset2 = Equipo.objects.all()
28    context = {
29        "objects": queryset2
30    }
31    return render(request, "NFLP/equipos.html", context)
32
33 def detalles_equipos(request):
34    queryset22 = Equipo.objects.all()
35    context = {
36        "objects": queryset22
37    }
38    return render(request, "NFLP/detalle_equipo.html", context)
39
40
41
```

