**ACCESO DATOS ORACLE DJANGO**

* Vamos a visualizar cómo podemos acceder a datos dentro del entorno de django.
* Lo primero que debemos hacer es ejecutar, en nuestro entorno virtual, la instrucción: **pip install oracledb**



* En django no tenemos ningún lugar dónde incluir el concepto que tenemos de Servicios, es decir, tener un servicio desde el que podamos recuperar datos.
* Para poder utilizar **Servicios**, es decir, clases de acceso a datos, es necesario hacerlo desde una única clase llamada **models.py**
* En dicha clase, iremos incluyendo las diferentes clases que vayamos teniendo de acceso a datos.

Ejemplo de **MODELS.PY**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Necesitamos también, dentro de **models.py**, las clases que vayamos a devolver

**Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Posteriormente, debemos utilizar el Servicio creado en **models.py** dentro de **views.py**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Y ya podremos realizar el dibujo que deseemos

1. Comenzamos creando una nueva página llamada **saludo.html** dentro de **templates/aplicación** y dibujamos un formulario con una caja llamada **cajanombre**.

**SALUDO.HTML**

**<form method*="post"*>**

**{% csrf\_token %}**

**<input type*="text"* name*="cajanombre"***

**class*="form-control"*/>**

**<button class*="btn btn-info"*>**

**Enviar nombre**

**</button>**

**</form>**

**<h1>**

**Su nombre es {{nombre}}**

**</h1>**

1. Recibimos el valor del color dentro de **views.py**

Para recibir un valor debemos utilizar la siguiente sintaxis:

**request.POST[“variable”]**

***def* saludo(*request*):**

**# PREGUNTAMOS SI RECIBIMOS COLOR O NO...**

**if ('cajanombre' in request.POST):**

**nombreRecibido = request.POST['cajanombre']**

**context = {**

**'nombre': nombreRecibido**

**}**

**return render(request, "aplicacion/saludo.html", context)**

**else:**

**#No enviamos nada**

**return render(request, "aplicacion/saludo.html")**

1. Incluimos dentro de **urls.py** la nueva ruta

Texto

El contenido generado por IA puede ser incorrecto.

1. Dentro de la página **Layout.html** ponemos el nuevo Link

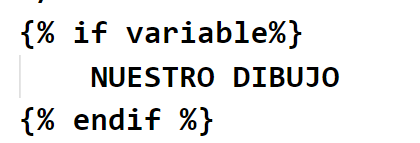
Imagen que contiene Gráfico

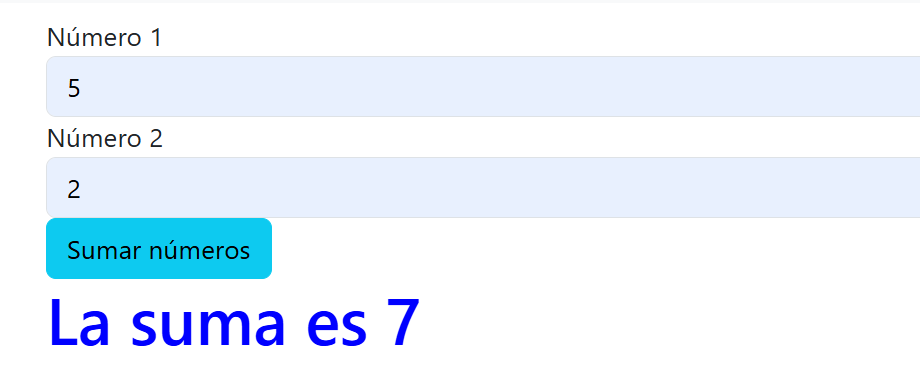
El contenido generado por IA puede ser incorrecto.

A continuación, vamos a Sumar dos números

Comprobaremos que hemos enviado una variable **suma** para realizar el dibujo.

Para comprobar si hemos recibido algún dato en el **template** se utiliza el siguiente código





Sobre **templates/aplicación** creamos una nueva página llamada **sumarnumeros.html**

**SUMARNUMEROS.HTML**

**<form method*="post"*>**

**{% csrf\_token %}**

**<label>Número 1</label>**

**<input type*="number"* name*="cajanumero1"***

**class*="form-control"*/>**

**<label>Número 2</label>**

**<input type*="number"* name*="cajanumero2"***

**class*="form-control"*/>**

**<button class*="btn btn-info"*>**

**Sumar números**

**</button>**

**</form>**

**{% if suma%}**

**<h1 style*="color:blue"*>**

**La suma es {{suma}}**

**</h1>**

**{% endif %}**

El siguiente paso es crear un nuevo método dentro de **views.py** que reciba los valores de **cajanumero1** y **cajanumero2**

**VIEWS.PY**

***def* sumarNumeros(*request*):**

**if ('cajanumero1' in request.POST):**

**dato1 = request.POST['cajanumero1']**

**dato2 = request.POST['cajanumero2']**

**suma = *int*(dato1) + *int*(dato2)**

**context = {**

**'suma': suma**

**}**

**return render(request, "aplicacion/sumarnumeros.html", context)**

**else:**

**#No enviamos nada**

**return render(request, "aplicacion/sumarnumeros.html")**

Incluimos nuestra nueva ruta/path dentro de **urls.py** de **aplicacion**

**URLS.PY**

**from django.urls import path**

**from aplicacion import views**

**urlpatterns=[**

**path('',views.*index*,*name*='index'),**

**path('peliculas/',views.*peliculas*,*name*='peliculas'),**

**path('futbol/',views.*futbol*,*name*='futbol'),**

**path('jugadores/',views.*jugadores*,*name*='jugadores'),**

**path('colores/',views.*colores*,*name*='colores'),**

**path('saludo/',views.*saludo*,*name*='saludo'),**

**path('sumar/',views.*sumarNumeros*,*name*='sumar'),**

**]**

El siguiente paso será dibujar la conjetura de Collatz.

Recibiremos un número y debemos devolver una lista con todos los números de la conjetura

Recordamos que la conjetura de Collatz dice así:

Todo número natural siempre llegará a ser 1 siguiendo estas instrucciones:

* + - * Si el número es Par, dividimos entre 2
      * Si el número es Impar, multiplicamos por 3 y sumamos 1

Ejemplo con el número 6:

* + - * 6,3,10,5,16,8,4,2,1

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Sobre **templates/aplicación** creamos una nueva página llamada **collatz.html**

**COLLATZ.HTML**

**<h2 style*="color:blue"*>**

**Conjetura de Collatz**

**</h2>**

**<form method*="post"*>**

**{% csrf\_token %}**

**<label>Introduzca número</label>**

**<input type*="text"* name*="cajanumero"***

**class*="form-control"*/>**

**<button class*="btn btn-warning"*>**

**Collatz**

**</button>**

**</form>**

**<ul>**

**{% for num in listacollatz %}**

**<li>{{ num }}</li>**

**{% endfor %}**

**</ul>**

Creamos un nuevo método dentro de **views.py**

**VIEWS.PY**

***def* collatz(*request*):**

**if ('cajanumero' in request.POST):**

**dato = request.POST['cajanumero']**

**numero = *int*(dato)**

**listacollatz = []**

**while (numero != 1):**

**if (numero % 2 == 0):**

**#PAR**

**numero = *int*(numero / 2)**

**else:**

**#IMPAR**

**numero = *int*(numero \* 3 + 1)**

**listacollatz.append(numero)**

**context = {**

**'listacollatz': listacollatz**

**}**

**return render(request, "aplicacion/collatz.html", context)**

**else:**

**#No enviamos nada**

**return render(request, "aplicacion/collatz.html")**

Por último, incluimos una nueva ruta/path dentro de **urls.py** de aplicación

**URLS.PY**

**from django.urls import path**

**from aplicacion import views**

**urlpatterns=[**

**path('',views.*index*,*name*='index'),**

**path('peliculas/',views.*peliculas*,*name*='peliculas'),**

**path('sumar/',views.*sumarNumeros*,*name*='sumar'),**

**path('futbol/',views.*futbol*,*name*='futbol'),**

**path('jugadores/',views.*jugadores*,*name*='jugadores'),**

**path('colores/',views.*colores*,*name*='colores'),**

**path('saludo/',views.*saludo*,*name*='saludo'),**

**path('collatz/',views.*collatz*,*name*='collatz'),**

**]**

Necesito una página que muestre la Tabla de multiplicar de un número.

Página: **tabla.html**

Views.py: **def** **tablaMultiplicar(request)**

Urls.py: path **tabla/**

Tendremos una caja de texto, escribiremos un número y mostraremos en una tabla el resultado de las operaciones de la tabla de multiplicar de dicho número.

**VERSION 1 VERSION 2**

Tabla

El contenido generado por IA puede ser incorrecto. Tabla

El contenido generado por IA puede ser incorrecto.