Para agregar datos a la tabla "Usuarios" en la base de datos sin necesidad de crear un formulario, puedes utilizar la interfaz de línea de

comandos (CLI) de Django llamada "shell". La CLI de Django proporciona un entorno interactivo donde puedes interactuar con tu aplicación y

realizar operaciones en la base de datos.

Sigue estos pasos para agregar datos a la tabla "Usuarios" utilizando la CLI de Django:

1. Asegúrate de que estás en la raíz de tu proyecto Django en la línea de comandos.

2. Ejecuta el siguiente comando para abrir la CLI de Django:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

python manage.py shell

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IMAGEN DE EJEMPLO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Texto

Descripción generada automáticamente

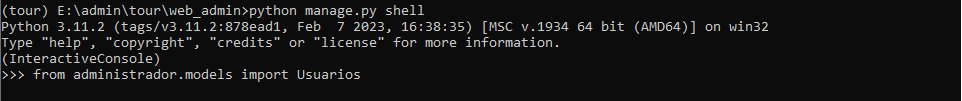
3. Una vez en la CLI de Django, importa el modelo "Usuarios" ejecutando el siguiente comando:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

python:

from administrador.models import Usuarios

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IMAGEN DE EJEMPLO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Asegúrate de reemplazar "nombre\_de\_tu\_aplicacion" con el nombre real de tu aplicación Django donde se encuentra el modelo "Cliente".

4. Ahora puedes crear una instancia de la clase "Usuarios" y establecer los valores de sus campos. Por ejemplo:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

python:

usuarios =Usuarios(id\_usuario=1, nombre\_usuario="admin",mail="pablo.gutierrez@duocuc.cl",clave="pablo123")

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IMAGEN DE EJEMPLO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Asegúrate de proporcionar los valores correctos para cada campo. Si necesitas obtener el objeto "id\_ciudad" para la clave externa

, deberás recuperarlo de la base de datos o crear uno nuevo. ejemplo (id\_ciudad=id\_ciudad)"se pondria solo el numero de la ciudad"

5. Guarda el objeto "Usuarios" en la base de datos ejecutando el siguiente comando:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

python:

usuarios.save()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IMAGEN EJEMPLO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Esto guardará el objeto en la tabla "Usuarios" de la base de datos.

Con estos pasos, puedes agregar datos a la tabla "Usuarios" directamente desde la CLI de Django sin necesidad de crear un formulario.

Recuerda que puedes repetir estos pasos para agregar más registros a la tabla si es necesario.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TABLAS DE PLANTILLAS PARA LLENAR LA BASE DE DATOS\*\*\*\*\*\*\*\*\*\*

\*\*\*TABLAS SIN FK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

usuarios = Usuarios(id\_usuario=1, nombre\_usuario = "admin" ,mail = "pablo.gutierrez@duocuc.cl",clave = "pablo123")

contacto = Contacto(id\_contacto=1, nom\_contacto = "Pablo" , ape\_contacto = "Gutierrez", des\_contacto = "AAAAAAAAAAAAAAAA AAAAAAAAAAA")

calidad\_tour = CalidadTour(id\_calidad = 1, des\_calidad= "aaaaaaaaaaaaaaa")

categoriaEva = CategoriaEva(id\_cat\_eva = 1, nom\_cate = "nose")

estado = Estado(id\_estado = 1, des\_estado = "anulado")

factura = Factura(id\_factura = 1, fecha = "2023/06/21" porc\_factura = "9%", porc\_iva ="19%", monto\_iva = "$9.990")

genero = Genero(id\_genero = 1, des\_genero = "Hombre")

idioma= Idioma(id\_idioma = 1, des\_idioma = "Español")

pais = Pais(id\_pais = 1, nom\_pais = "Chile" , nacionalidad = "Chilena")

rrss = Rrss(id\_rrss = 1, nom\_rrss = "Facebool")

tipo\_empresa = TipoEmpresa(id\_tipo\_empr = 1 , des\_tipo\_empr = "noseeee")

tipo\_guia = TipoGuiaid\_tip\_guia = 1 , nom\_tipo\_guia = "noseeee")

\*\*\*TABLAS CON PK Y FK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ciudad = Ciudad(id\_ciudad= 1 , nom\_ciu = "Santiago", id\_pais=1)

cliente = Clienteid\_cliente= 1 , dni = "19279652-0", nom\_user = "Pablo",pape\_user = "Gutierrez", sape\_user = "Alvarez", contacto="+569 33341396", direccion = "kumu 987978", email = "pab.gutierrez@duocuc.cl", id\_ciudad = 1)

Empresa = Empresa(id\_empresa = 1, nom\_empresa ="FREE", rut\_empr = "66666666-6", email\_empr = "pab.gutierrez@duocuc.cl", contacto\_empr = "+5699999999",direccion\_empr = "freire 111" , representante = "diego madrid" , nom\_dueño = "reyharasdask" , rut\_dueño ="66666666-6", contacto\_dueño = "+569 12345678", email\_dueño = "aaaaaaa@gmail.com, id\_tipo\_empr =1, id\_ciudad =1)

\*\*\*\*\* **GUIA PARA VER LOS DATOS EN LA TABLA MEDIANTE SHEL**L\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Para ver los registros de una tabla en el intérprete interactivo de Django (`python manage.py shell`), puedes seguir estos pasos:

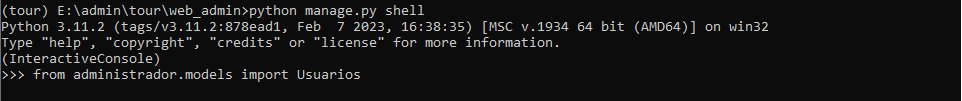
1. Importa el modelo correspondiente al nombre de la tabla que deseas consultar. Por ejemplo, si deseas ver los registros de la tabla "Usuarios", importa el modelo de esta manera:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Python:

from administrador.models import Usuarios

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* IMAGEN DE EJEMPLO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



2. Luego, puedes utilizar el ORM de Django para obtener los registros de la tabla. Puedes utilizar el método `.all()` para obtener todos los registros o el método `.filter()` para realizar una consulta más específica. Por ejemplo:

python

# Obtener todos los registros de la tabla

clientes = Usuarios.objects.all()

# Filtrar los registros que cumplan cierta condición

Usuarios\_filtrados = Usuarios.objects.filter(nombre\_usuario='Pablo')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IMAGEN DE EJEMPLO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



3. Para mostrar los registros en el shell, puedes recorrer la lista de objetos obtenida y acceder a sus atributos. Por ejemplo:

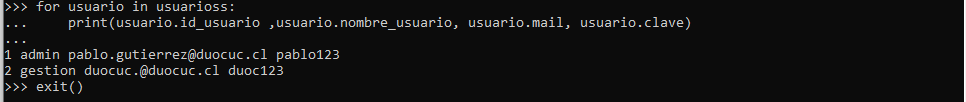
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

python

for usuario in usuarioss:

print(usuario.id\_usuario ,usuario.nombre\_usuario, usuario.mail, usuario.clave)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IMAGEN DE EJEMPLO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Esto mostrará los campos específicos de cada registro de la tabla "Usuarios" en el shell. Asegúrate de reemplazar "nombre\_de\_tu\_aplicacion" por el nombre real de tu aplicación Django y ajustar los nombres de los campos y modelos según corresponda en tu proyecto.

\*\*\*\*\* TENER A CONSIDERACION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**En este ejemplo, el `print(usuario)` está indentado con cuatro espacios, lo que indica que es parte del bloque de código del bucle `for`. Asegúrate de aplicar la indentación adecuada en tu código para resolver el error.**

**Texto

Descripción generada automáticamente**

**Para hacer el llamado y que podamos ver todas las tablas en el bucle for, se tiene que considerar las reglas del lenguaje, y mínimo luego del “for usuario in usuarioss:” tiene que haber 4 espacios para mostrar los datos que uno quieren en el print abajo.**