Crea un directorio para tu proyecto. Por ejemplo, si tu proyecto se llama “tareas”, puedes crear el directorio con el siguiente comando:

mkdir tareas/

cd tareas

Crea el entorno virtual. Puedes hacerlo con el módulo “venv” de Python. El nombre del entorno virtual es a tu elección, pero en este caso lo llamaremos “venv”:

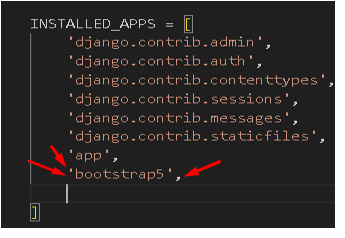
python3 -m venv venv otro ejemplo es py -m venv nombre\_del\_proyecto

Activa el entorno virtual: Para activar el entorno, ejecuta1:

nombre\_del\_proyecto\\Scripts\\activate.bat

hay instalar todas las versiones de las librerías deseadas si se esta abriendo un programa nuevo instalar todas las librerías que se necesiten esto para que pueda correr a app

las librerías usadas en una aplicación se encuentran en INSTALLED\_APPS

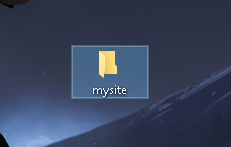


Activa el entorno virtual. Es importante activar el entorno virtual cada vez que vayas a instalar algún paquete o a correr el backend del proyecto. Para activarlo, usa el siguiente comando:

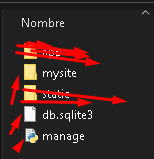
source venv/bin/activate

Creación del proyecto creamos con

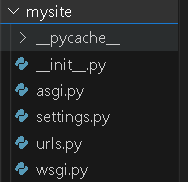
La carpeta del proyecto en el escritorio



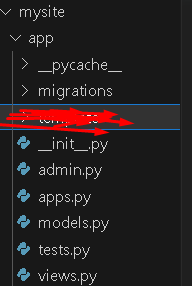
Luego se crea la carpeta proyecto en este caso se llama y nos trae los archivos que se ven la imagen



El archivo mysite es el archivo del proyecto que nos trae los siguientes documentos .py



Luego se crea el archivo app con este archivo contiene la aplicación de los cuales nos trae los siguientes documentos.py

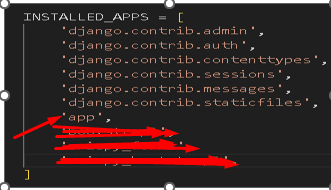


Views.py: contiene todas los renderizados de los html

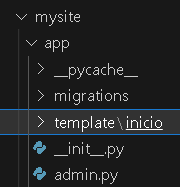
Tests.py: para testear

Models.py: para modelar los los datos que se van a introducir en el slq

Luego se debe registrar la carpeta app esto ayuda a que la app sea registrada con Django se tiene que registrar en

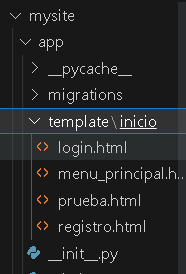


Para crear un html se debe crear un archivo en la carpeta de la app

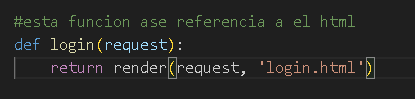


Esta es llamada témplate\adentro contiene otra carpeta en este caso inicio

Allí dentro se debe colocar los html



Para mostrar un html se debe colocar en views una función o class que renderice las vistas del html y luego usar una dirección URL para redirigir las vista

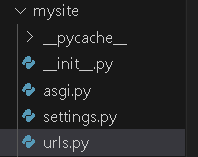


En este caso se llama la función login y retorna un renderizado que en el login.html

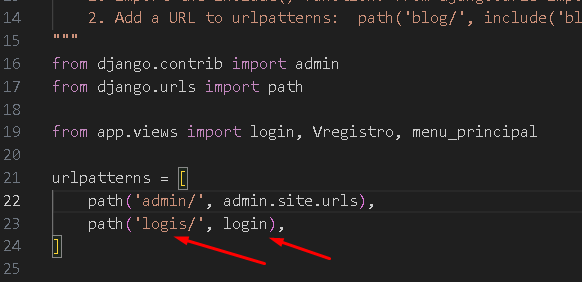
Luego se tiene que crear las url estas nos ayudan a buscar el html por medio del busardo web, por ejemplo



Nos vamos a la carpeta del proyecto y buscamos 



Cuando estemos ayi debemos colocar path(‘logis/’, login)para crear la url de la vista que se creo en viws



**USANDO BOOSTTRAP5**

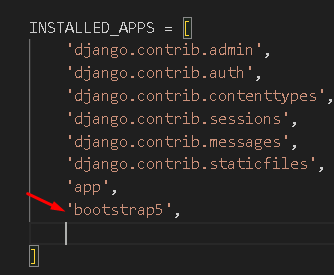
**pip install django-bootstrap-v5**

**para usar bootstrap5 en los html usar pip install django-bootstrap-v5**

**entonces se instala**

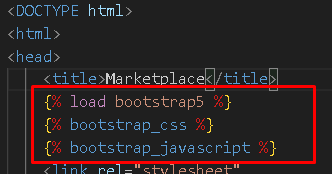


Luego de tenerlo instalado debemos registrarlo en en



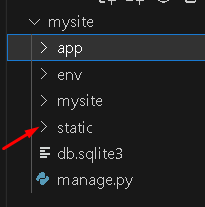
Con esto hacemos que bootstrap5 que de cargado registrado en app y puede buscar para aplicar los estilos

Luego nos vamos al html y usarmos

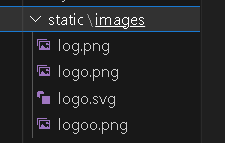


**Para crear logo**

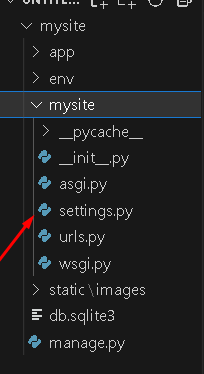
Se debe crear una carpeta que se llame static/imágenes



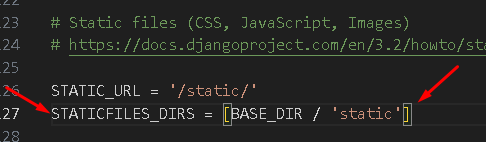
y hay colocar los logos como se ve en la imagen en static/images en este caso



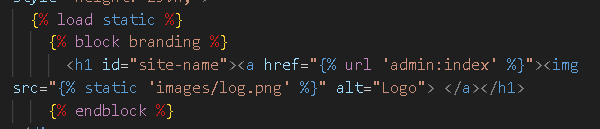
Luego se debe colocar un código en



Donde se colocará un directorio statico que contendrá las logo



Luego que se aya colocado el código nos vamos a donde queremos colocar el logo en este caso es el que se puede observar en la imagen

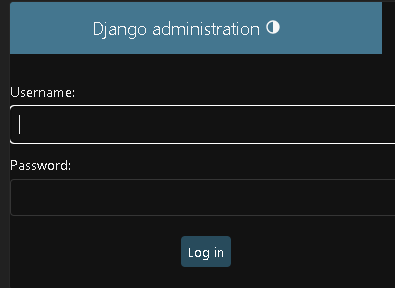


cuando este colocado el código se deberá ver “asi”



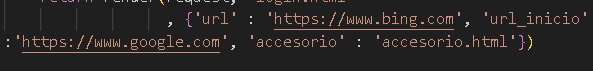
En url esta la dirección del admin de django si se le da click nos lleva



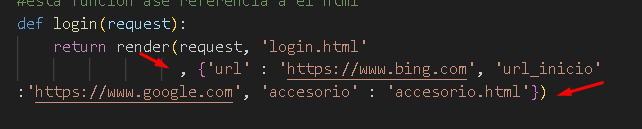


Donde ‘images/log.com’ es la dirección de carpeta donde se encuentra el logo y el nombre del logo en este caso el nombre es log.png

Para crear un url ose un botón con un url incorporado que te redireccione a una pagina se debe hacer

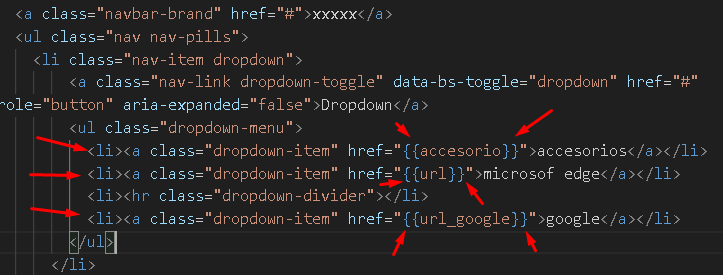


Donde url en una variable y contendrá el url igual mente url\_inicio y accesorio entonces esto debería quedar asi en una función



Luego nos vamos al html donde vamos a colocar la variable que contiene el url de la página donde nos queremos redirigir dentro de la {{ aquí }} usar la variable que contenga el link o la url

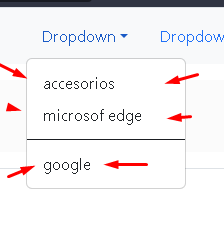
 un ejemplo de un link completo es este



En la página web se debe ver así en accesorios la url usadas es 

En Microsoft Edge es 

En Google es 

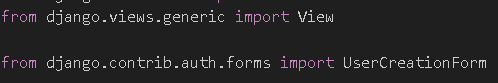


**REGISTRO DE LA PAGINA**

**C**reamos un views donde crearemos una class como se muestra en la imagen

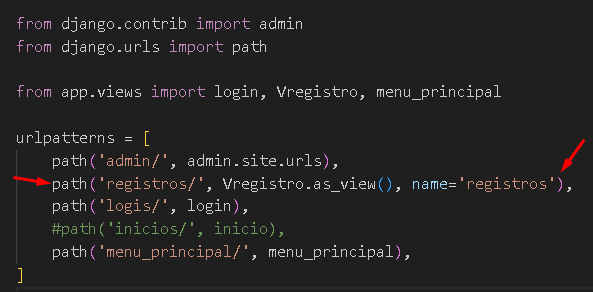


Recordar dando que se deben importar y las librerías



Para que la views funcione esto ayudara a que la class pueda crear un formulario que estará ligado automáticamente la base de datos sqlite3

Ahora toca crear la url que nos redireccionara al html de registro



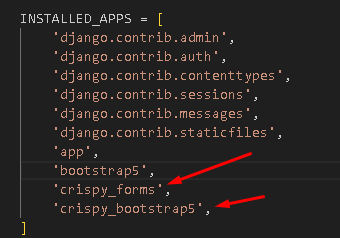
**pip install crispy-bootstrap5**

instalar desde la carpeta env ósea entorno virtual y entrando en la cónsola

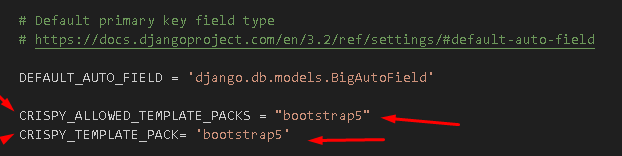
$ pip install django-crispy-forms

$ pip install crispy-bootstrap5

para usar el crispy de debe registra en al

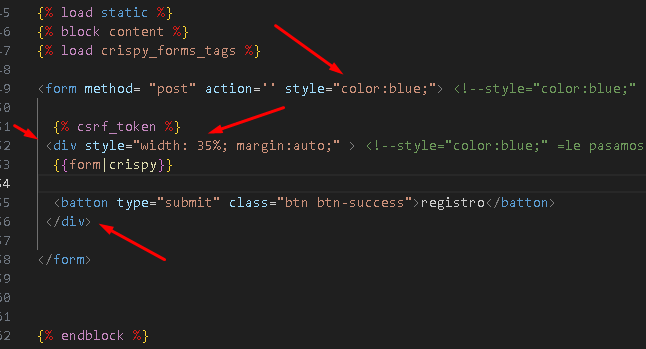


Luego se debe colocar al final del documento settings lo siguiente



Luego hay que ir al html y colocar para testear que está cogiendo los estilos **nota es button y**

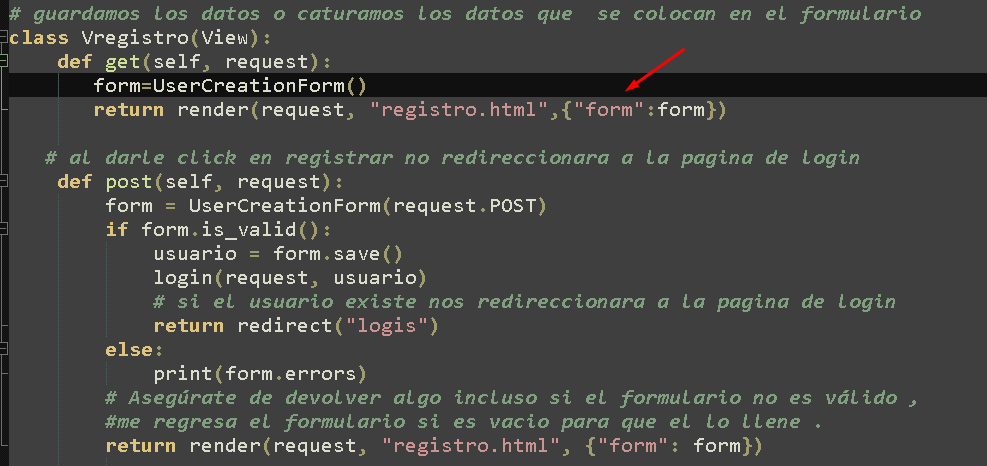
Mirar los cambios en la pagina



Los datos o capturar los datos de formulario de registro se deber usar el código el método form.save crea automáticamente una autentificación a la base de datos el if es para el capturado de datos del registro

Y el else es para control de errores y devolver el formulario el formulario si no es valido

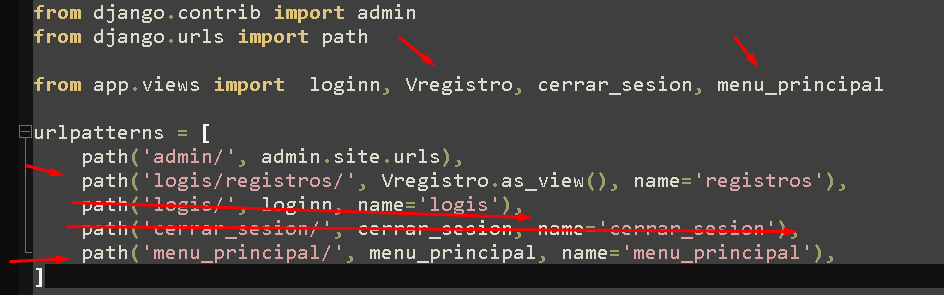
El  es del ulr del formulario que esta en html registro 



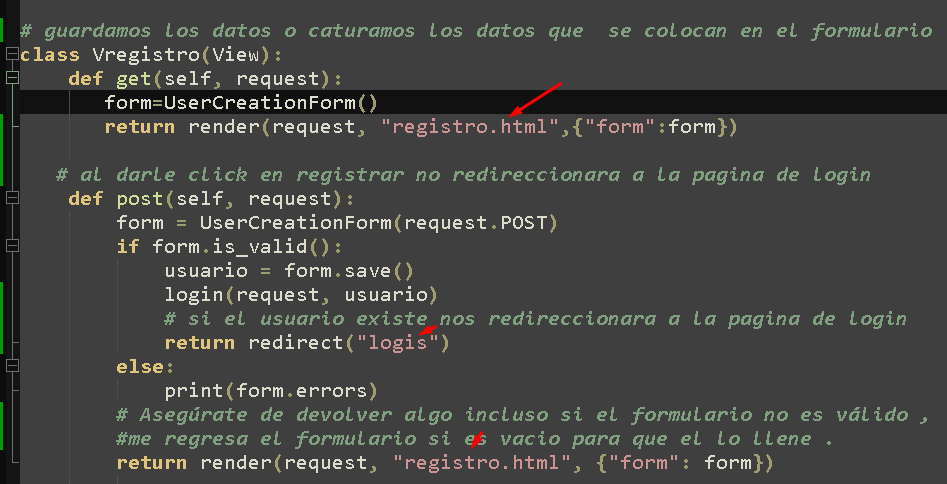
También hay que importar 

Ya que se utiliza em 

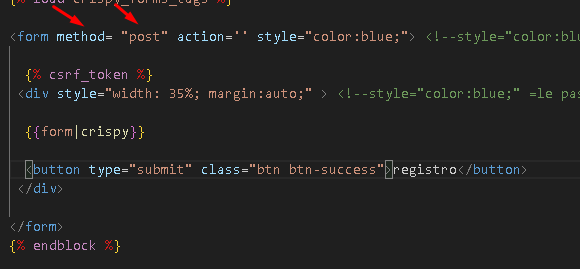
En url se debe colocar Ya que se esta utilizando los url en la vista



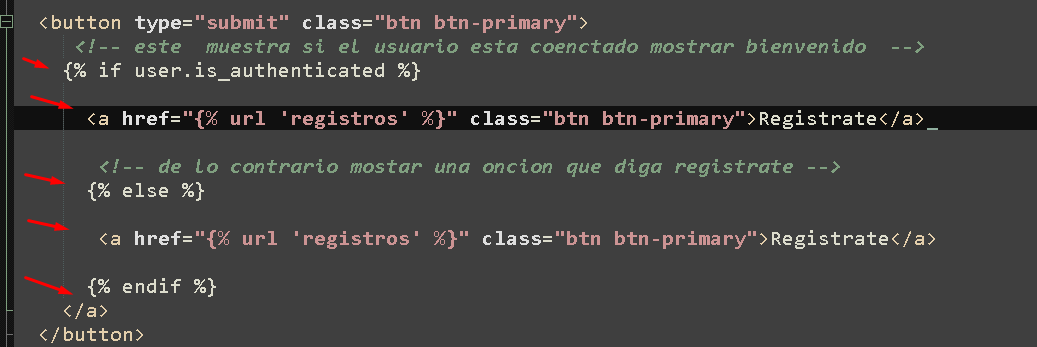
para hacer referencia a los redireccionamientos de las paginas al dar click en registrar



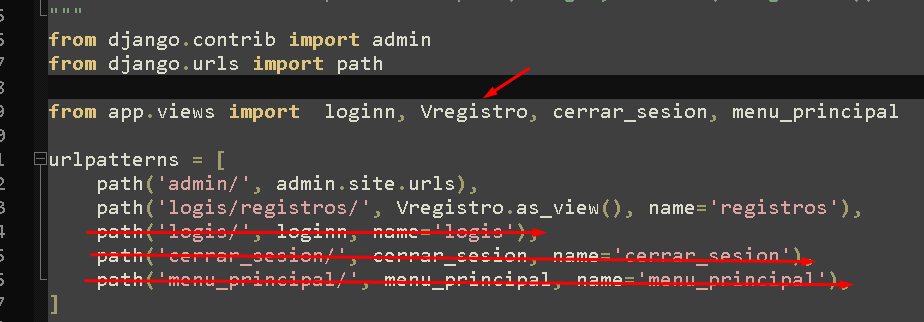
en el html se debe usar el method “post” para el envió de los datos



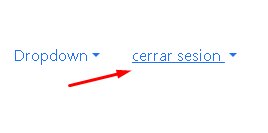
Para el apartado de login se realizó de esta manera



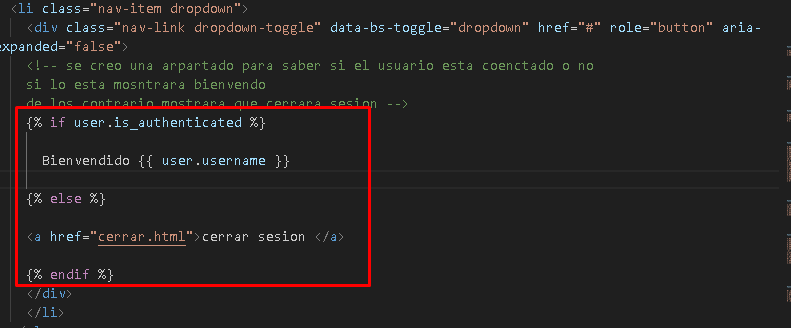
Luego ir la el documento urls.py y colocar la url del registro como se puede observar en la imagen



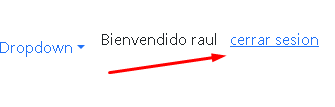
Hora para colocar una vista que diga si el usuario esta logia en el sistema, así como se muestra en la imagen se debe hacer



Se debe buscar el documento.html donde se va a colocar la vista luego se coloca

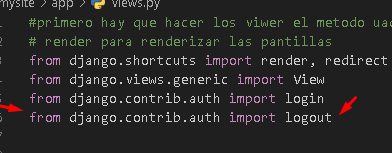


Con esto se crea una vista en el html donde se muerta una modulo o opción como el de la imagen

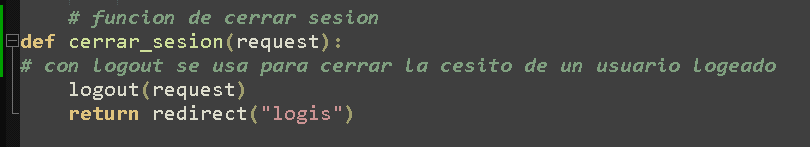


Para mostrar si esta logiado el usuario esto es para que lo muestre después que esta dentro del sistema

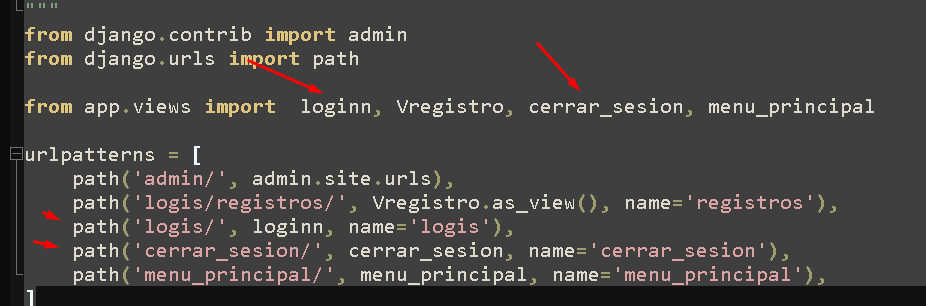
Antes de crear la función de debe importar logout ya que se esta usando en la función y perimitira cerar la sesión del usuario



Para cerrar sesión se debe realizar una views donde se define una función como se ve en la imagen de los cual no va a redireccionar a logis ose a la pagina de logrease



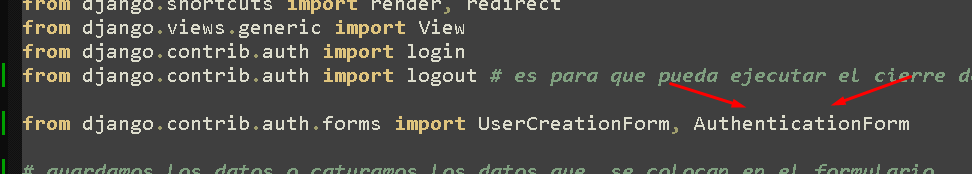
Luego se importa cerrar\_sesion y crea la urls ne el archivo url.py debe quedar asi como la imagen



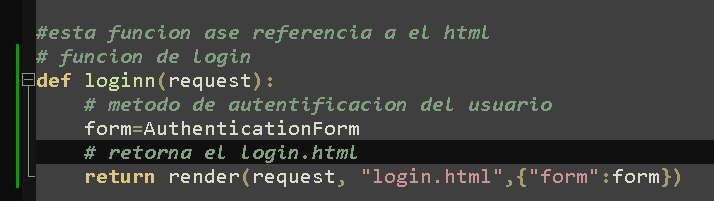
**login creando**

luego de crear el registro y que guarde los datos en la base de datos y nos redireccione a la ventana principal de login donde luego de loguearse el usuario accede a la app o página principal , usando métodos propios de python django para crear formulario de registro y autenticar también se usó bootstrap y crispy\_bootstrap5 ,se procedera a **crear el login y la autenticación de login**

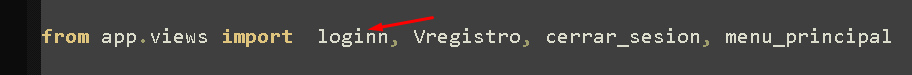
**se importa la vista en views**  ya que se usa método propios de djando como el que se muestra en pantalla



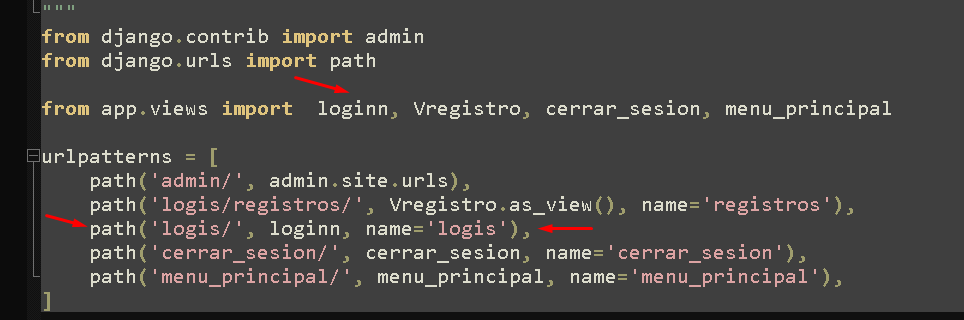
luego de que se aya importado se crea la función en la vista osea en la views.py como se muestra en la imagen



luego se realiza la importación de la vista login y luego

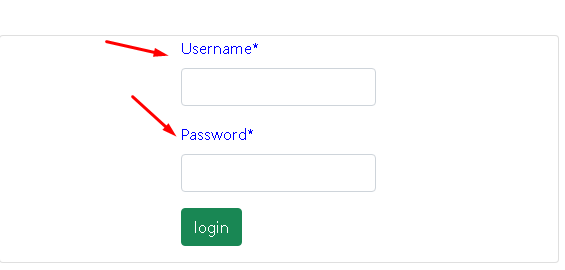


se crea la url como se observa en la imagen en **path('logis/', loginn, name='logis'),**



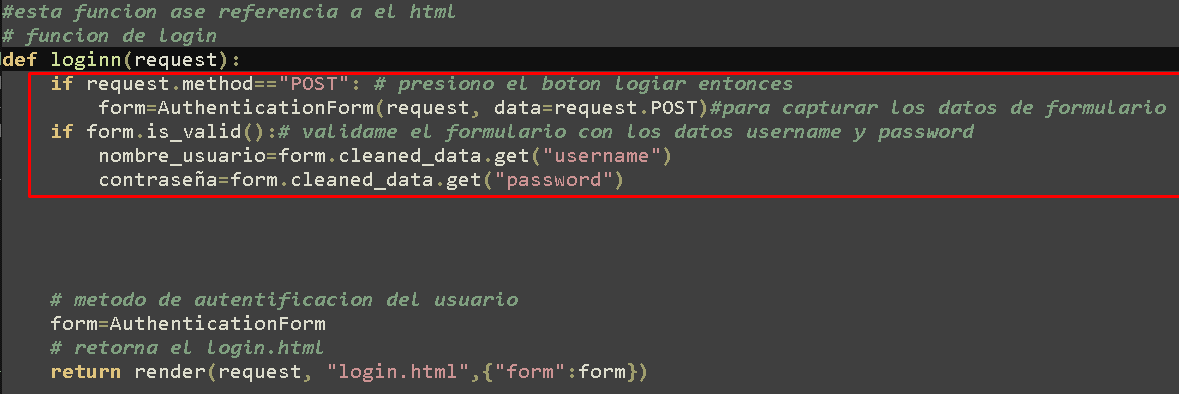
luego se observa la página de login colocando en el buscado de navegador

[**http://127.0.0.1:8000/logis/**](http://127.0.0.1:8000/logis/) deberia sali asi un ejemplo



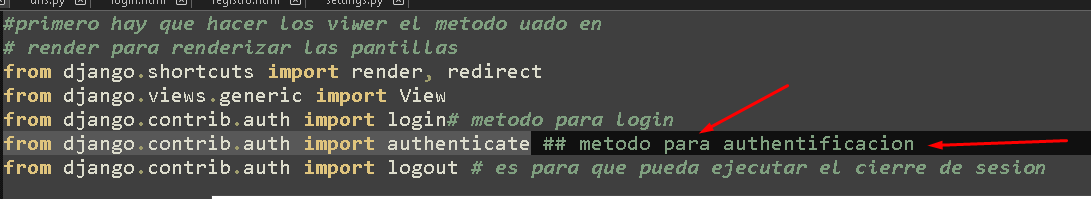
al darle al boton login no hace nada hay que **crear la validacion de username y password**

**como se muestra en la imagen**

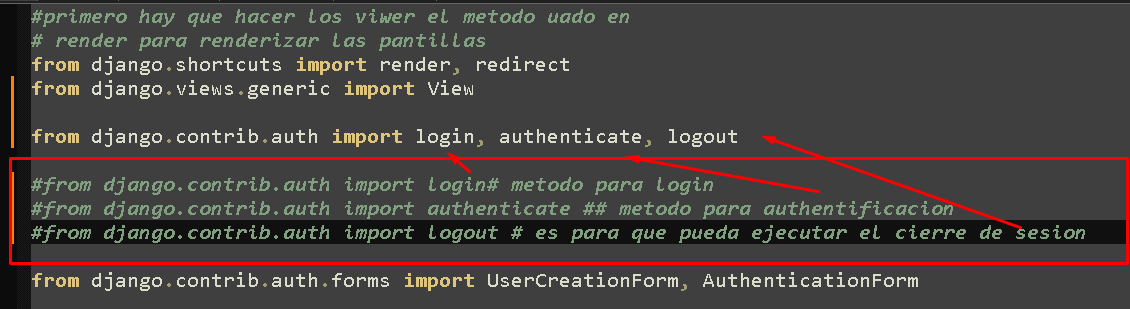
****

ahora se debe crear una importación para poder usar la autenticación esta es

from django.contrib.auth import authenticate

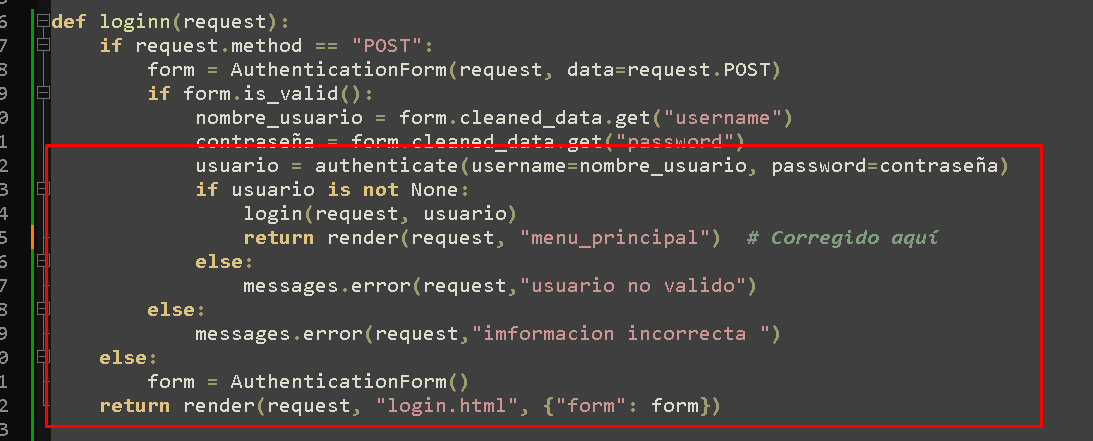


**nota otra forma de colocar las importaciones que son del mismo paquete por ejemplo la imagen anterior es usar como puede observar en la imagen**

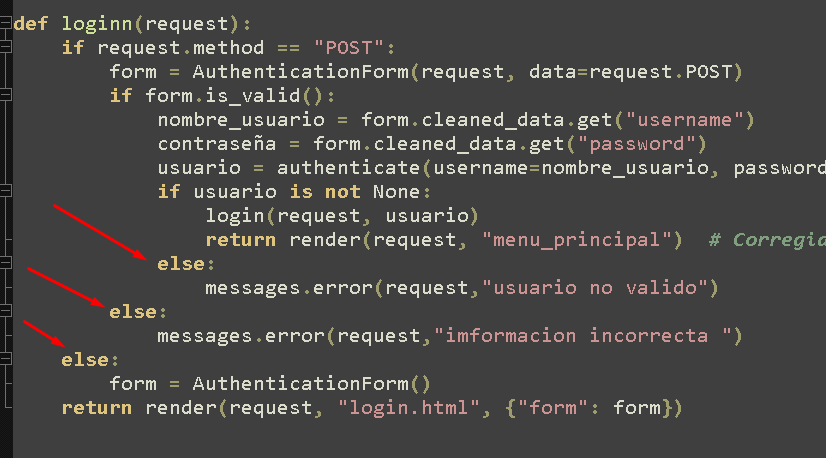
****

**ahora se usara el metodo importado**

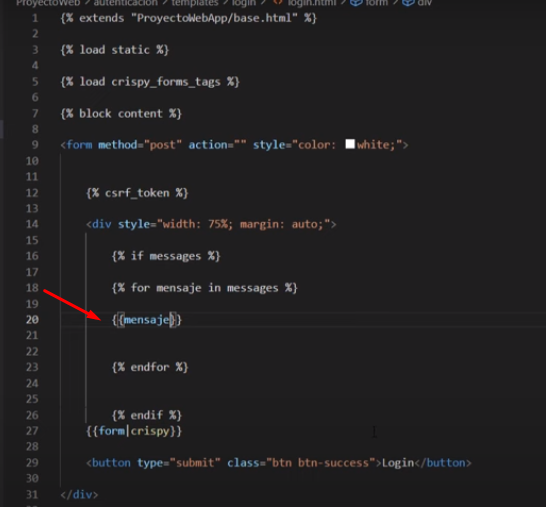
**donde se muestra la imagen views.py con eso autenticamos al usuario en el login**

****

**y contiene control de errores por ejemplo**

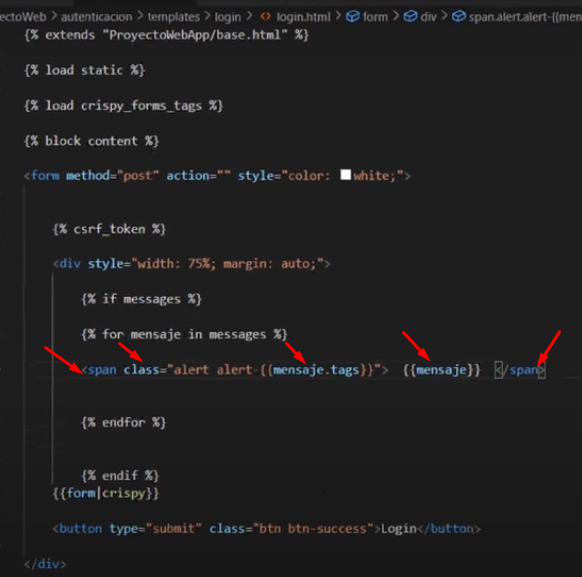


asi es el estado normal con alerta sin estilo

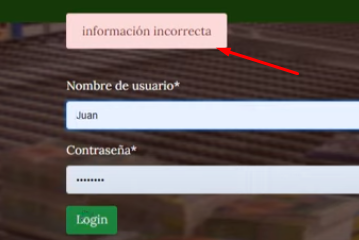


para colocar alertas el formulario usando bootstrap se se llama alert alert , se coloca





como ejemplo deberia quedar asi

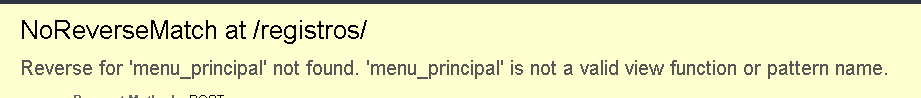


**ERRORES**

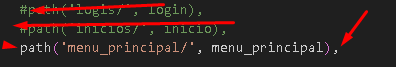
**1 es <button type="submit" class="btn btn-success">registro</button>**



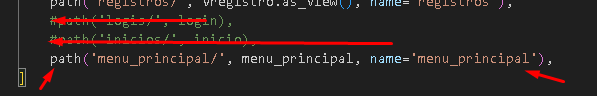
**2**



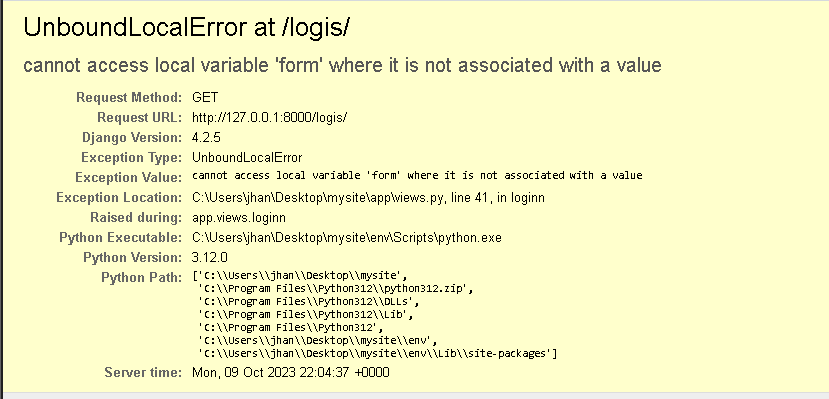
**NO DEBE COLOCAR EN EL DOCUEMNTO URL**



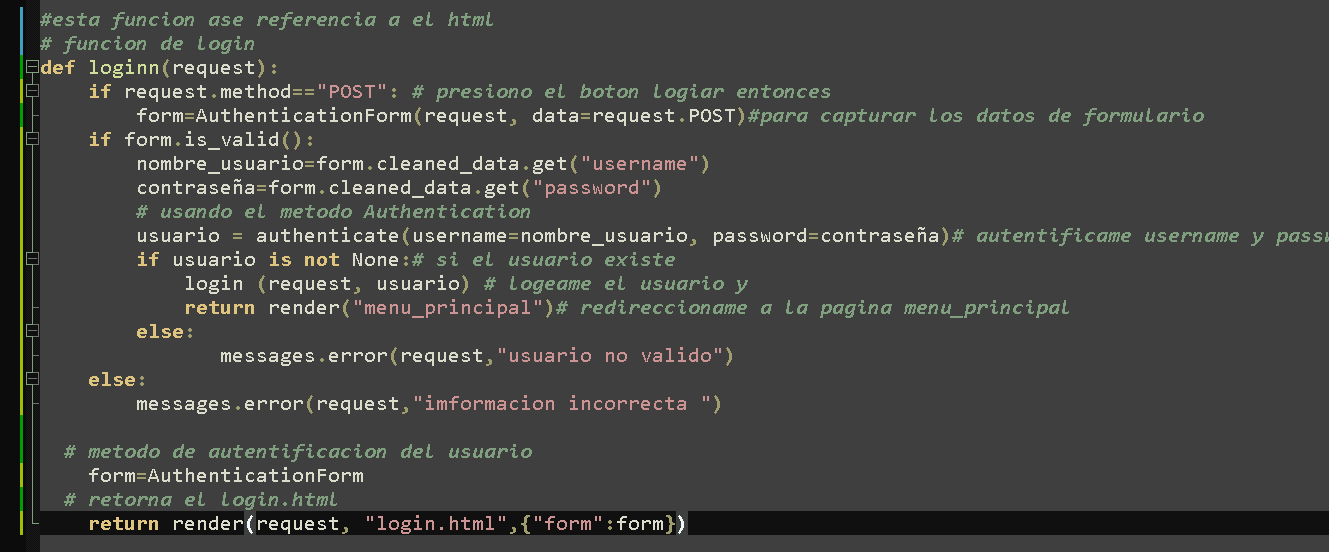
**EL CORRECTO ES ESTE**



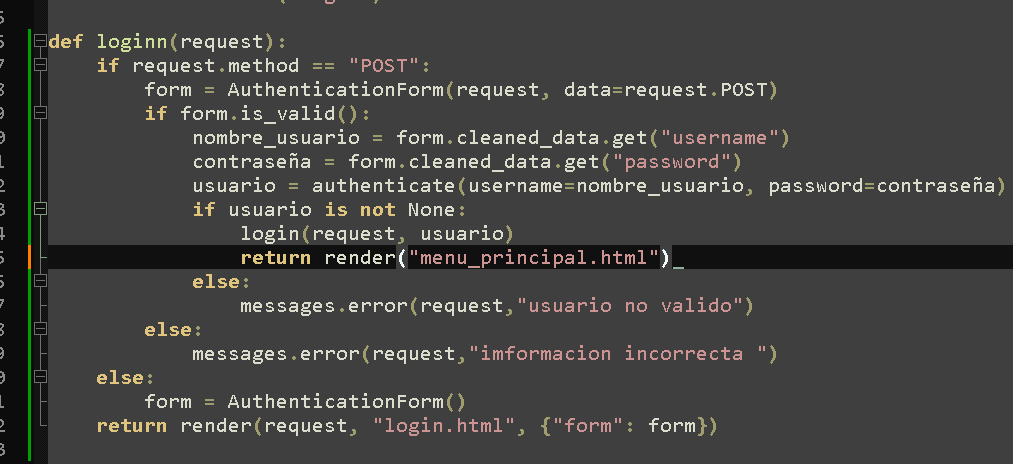
**3 error**

****

**el incorrecto**

****

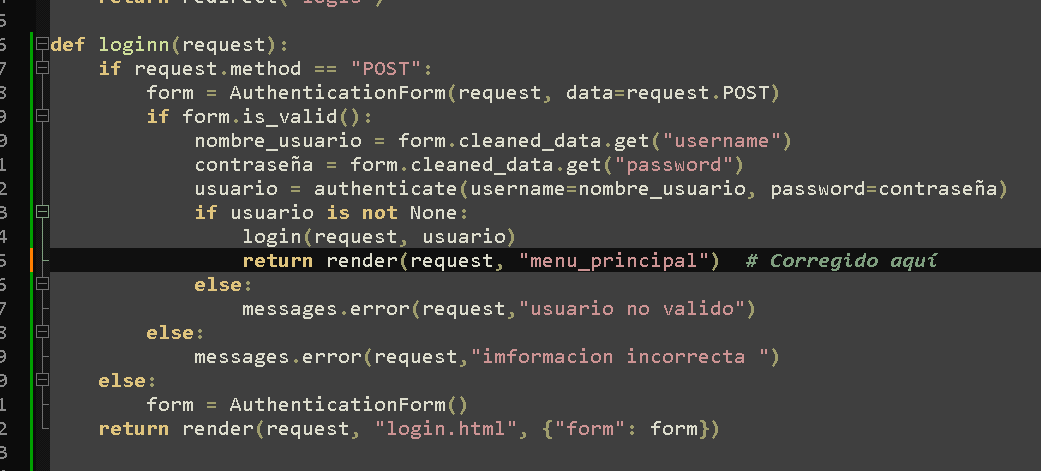
* **~~el correcto es~~**

****

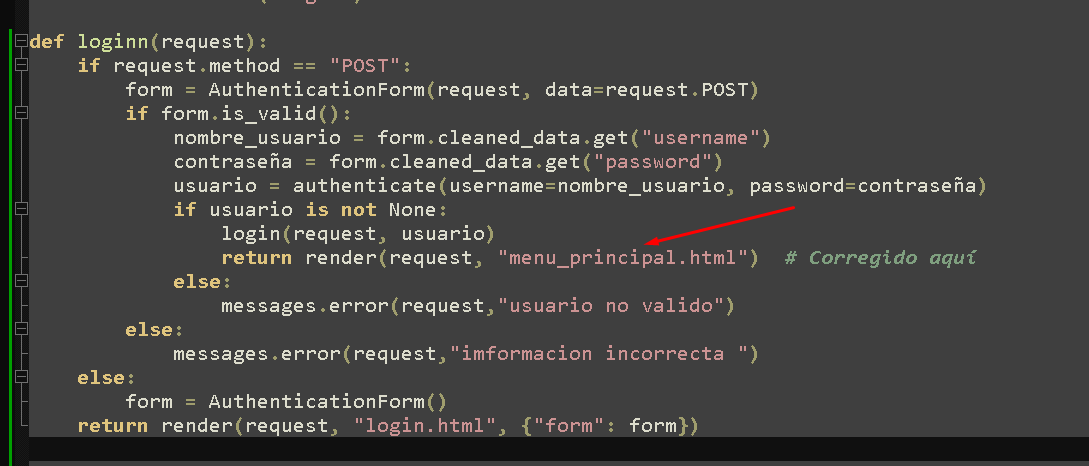
**4 error**

****

**el codigo que esta mal: return render(request, "menu\_principal") # Corregido aquí**

****

* **~~correccion~~**

****