Taller 3 -Arquisoft

Felipe Rueda- 202010903

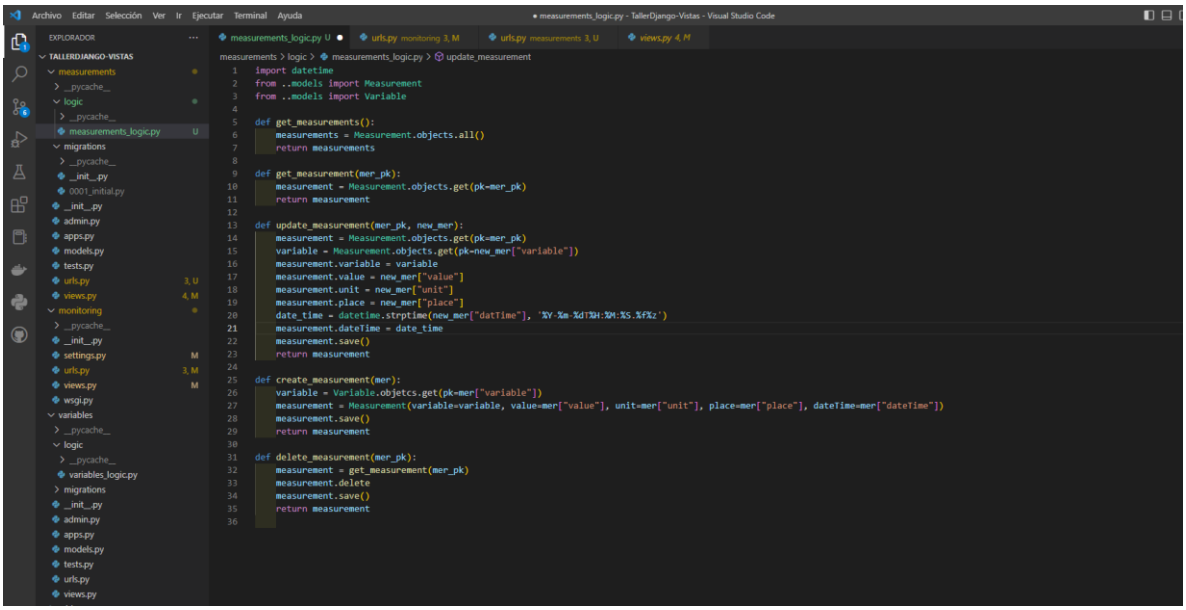
Repositorio: https://github.com/Feru34/TallerDjango-Vistas.git

Evidencias del código creado para Measurements

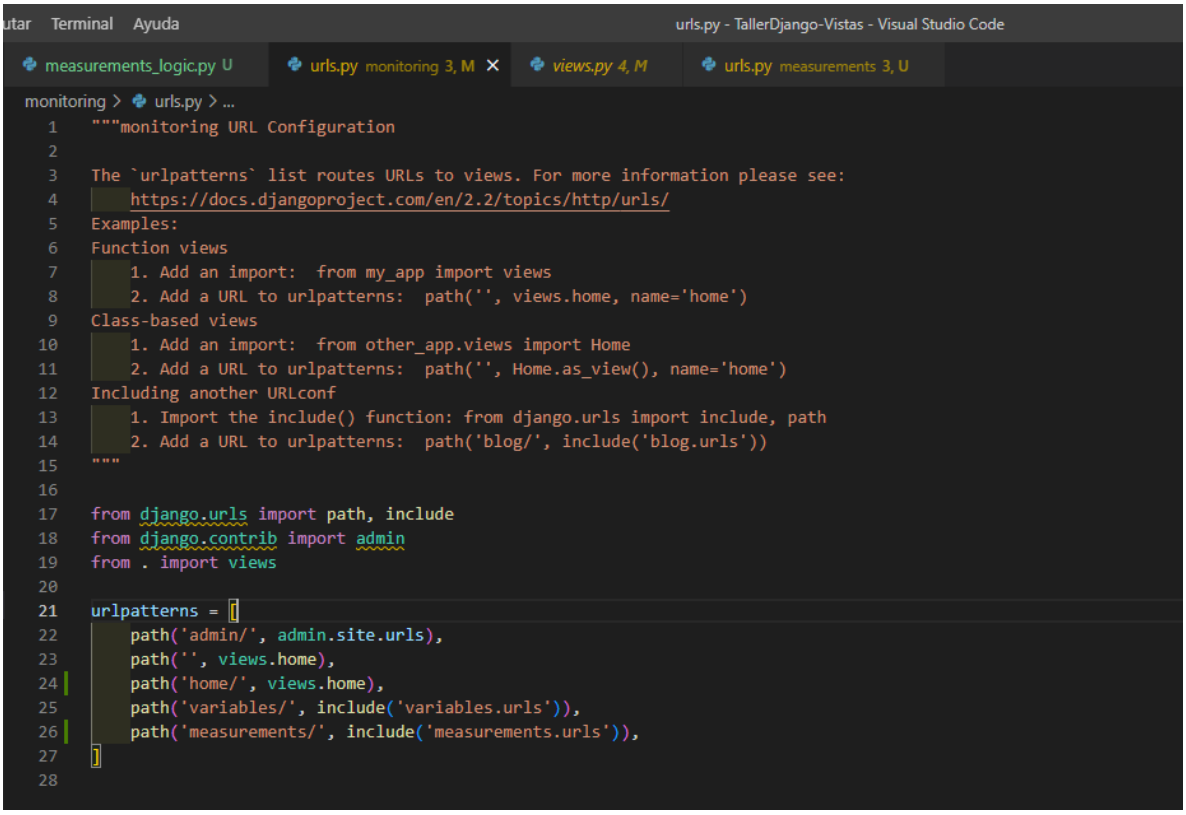
Evidencia 1: Dentro de l carpeta variables, se crea el archivo que se encarga de las actualizaciones, Measurements_logic.py.

- Se crea la función get_measurements() Consultar lista de todas las medidas, se utiliza .objects.all() para extraer todas las Measurements.
- Se crea la función get_measurement() para tomar una en específico requeriere una PK de objeto al quiere consultar.
- Se crea la función create_measurement() el cual crea una medida con una de la creación de otra variable cuya llave primaria corresponde al parámetro.
- Se crea la función update_measurement() la cual cambia una medida común identificador en específico, esta también incluye una llave foránea de Variable.
- Se crea la función delete_measurement() () la cual elimina una medida común identificador en específico.

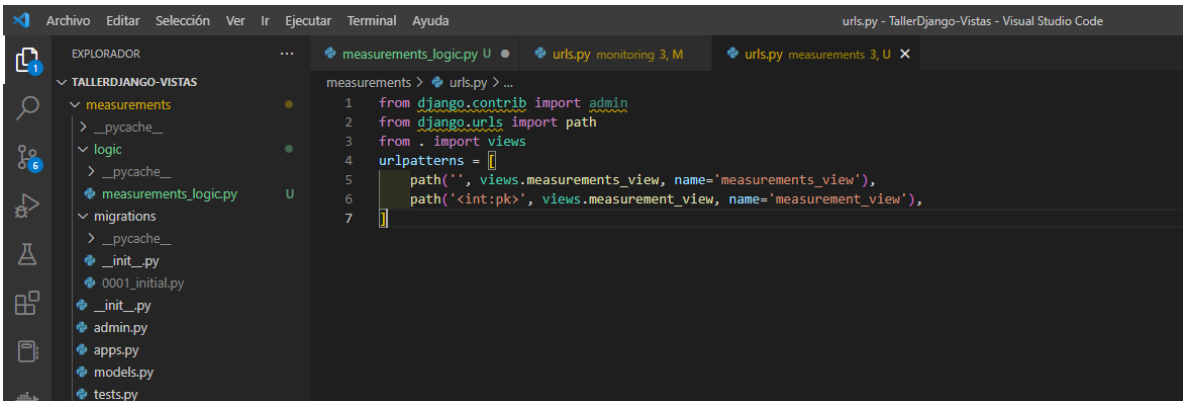
Todas llamando los respectivos atributos del objeto para su consulta o modifción. Así mismo, se respeta la llave foránea que tiene con la clase variables como vemos en la línea 15 y 26.



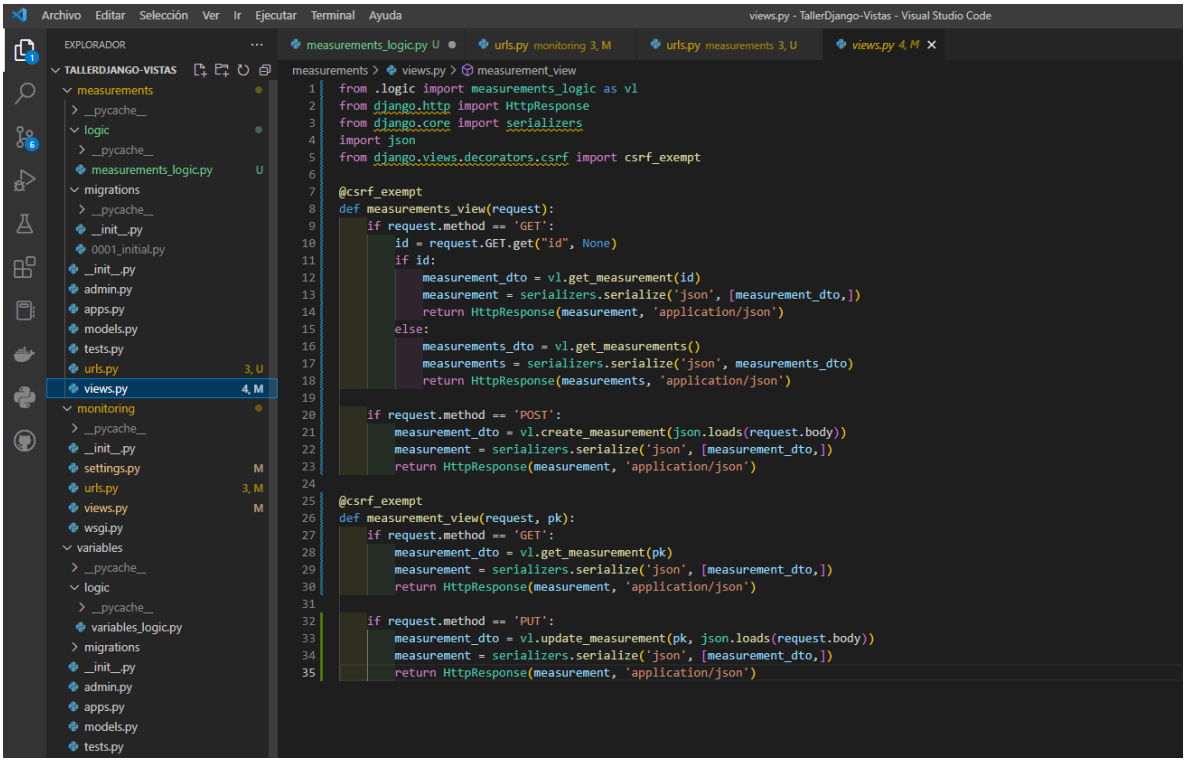
Evidencia 2: Se crea el path con Measurements junto con la directiva include, el cual marca como comenzarán la base de las direcciones URLs del objeto



Evidencia 3: Se crean las URLs que se usarán en la página para measurements, son bastante parecidas a las de variables pues tienen la misma lógica.

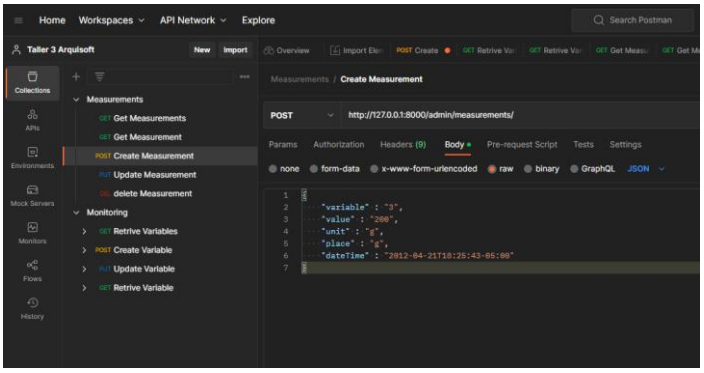


Evidencia 3: Posteriormente, dentro del archivo que se encarga de las consultas, views.py, se agrega el decorador @csrf:exempt para poder realizar las pruebas postman sin necesidad de un token de seguridad. En measurements/views.py se crean las vistas de forma casi iguales a como se hizo con variables/views.py

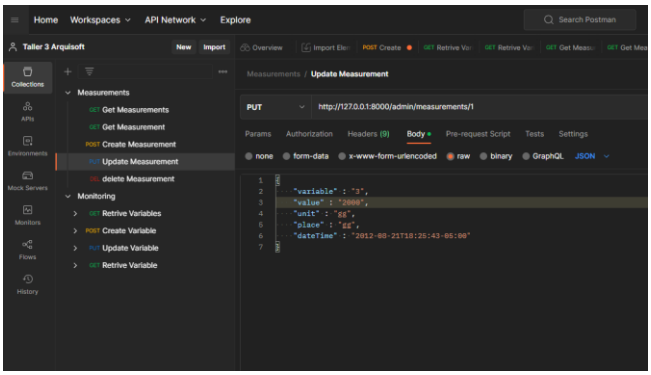


Evidencia Pruebas Postman:

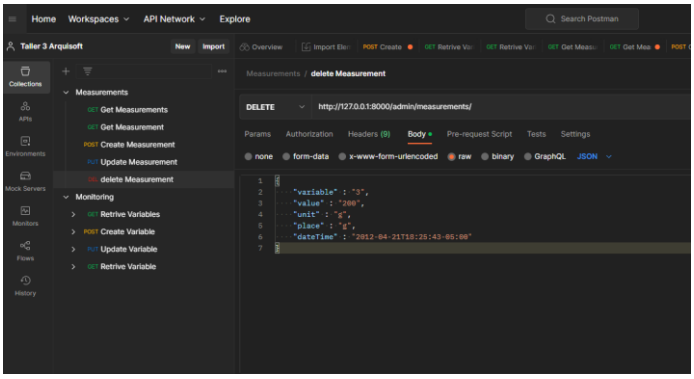
Evidencia 1: Create Measurement



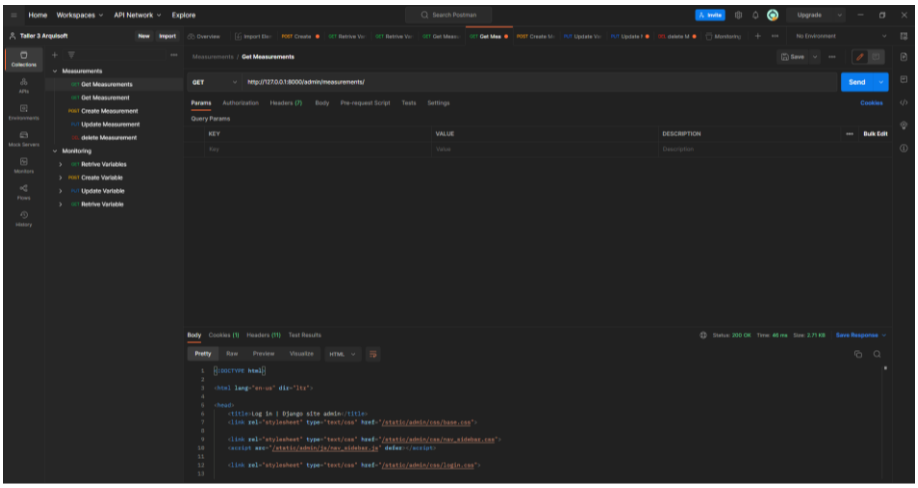
Evidencia 2: Update Measurement



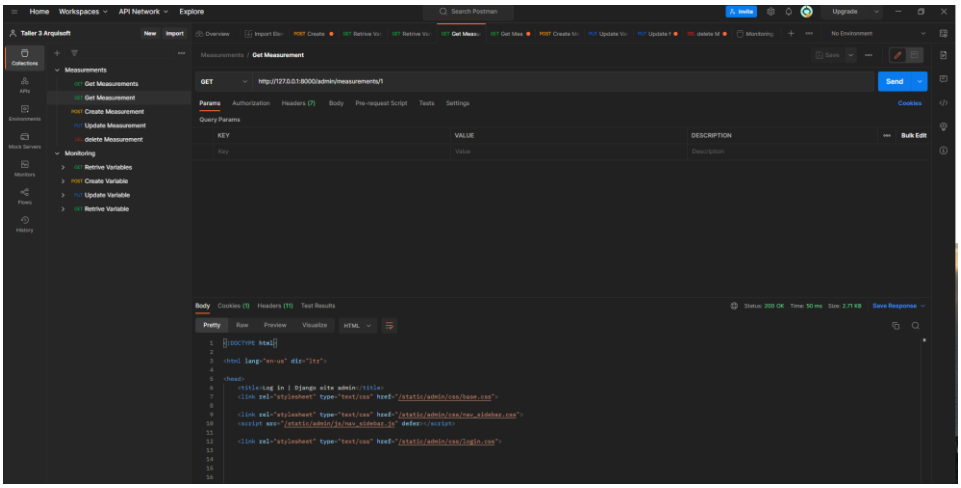
Evidencia 3: Delete Measurement



Evidencia 4: Get Measurements



Evidencia 1: Get Measurement



Desafortunadamente las pruebas Postman no funcionaron, aunque el código y la lógica estuviera correcta al criterio del redactor, aparecía en consola un error inesperado el cual no se pudo solucionar, este marcaba que el token de seguridad era incorrecto. Algo extraño pues se estaba usando el decorador @csrf:exempt justamente para evitar este problema. A continuación, el error en cuestión:

```
2/Feb/2023 19:39:38] "DELETE /admin/measurements/ HTTP/1.1" 403 2519
2/Feb/2023 19:39:58] "GET /variables?id=3 HTTP/1.1" 301 0
2/Feb/2023 19:39:58] "GET /variables/?id=3 HTTP/1.1" 200 73
rbidden (CSRF token missing or incorrect.): /admin/measurements/
2/Feb/2023 19:40:05] "POST /admin/measurements/ HTTP/1.1" 403 2519
rbidden (CSRF token missing or incorrect.): /admin/measurements/
2/Feb/2023 19:40:06] "POST /admin/measurements/ HTTP/1.1" 403 2519
rbidden (CSRF token missing or incorrect.): /admin/measurements/
2/Feb/2023 20:10:11] "POST /admin/measurements/ HTTP/1.1" 403 2519
rbidden (CSRF token missing or incorrect.): /admin/measurements/
2/Feb/2023 20:11:15] "POST /admin/measurements/ HTTP/1.1" 403 2519
```

Todas las pruebas están en el repositorio