**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.

**Texto

Descripción generada automáticamente**

**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

**Texto

Descripción generada automáticamente**

**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.

**Captura de pantalla de computadora

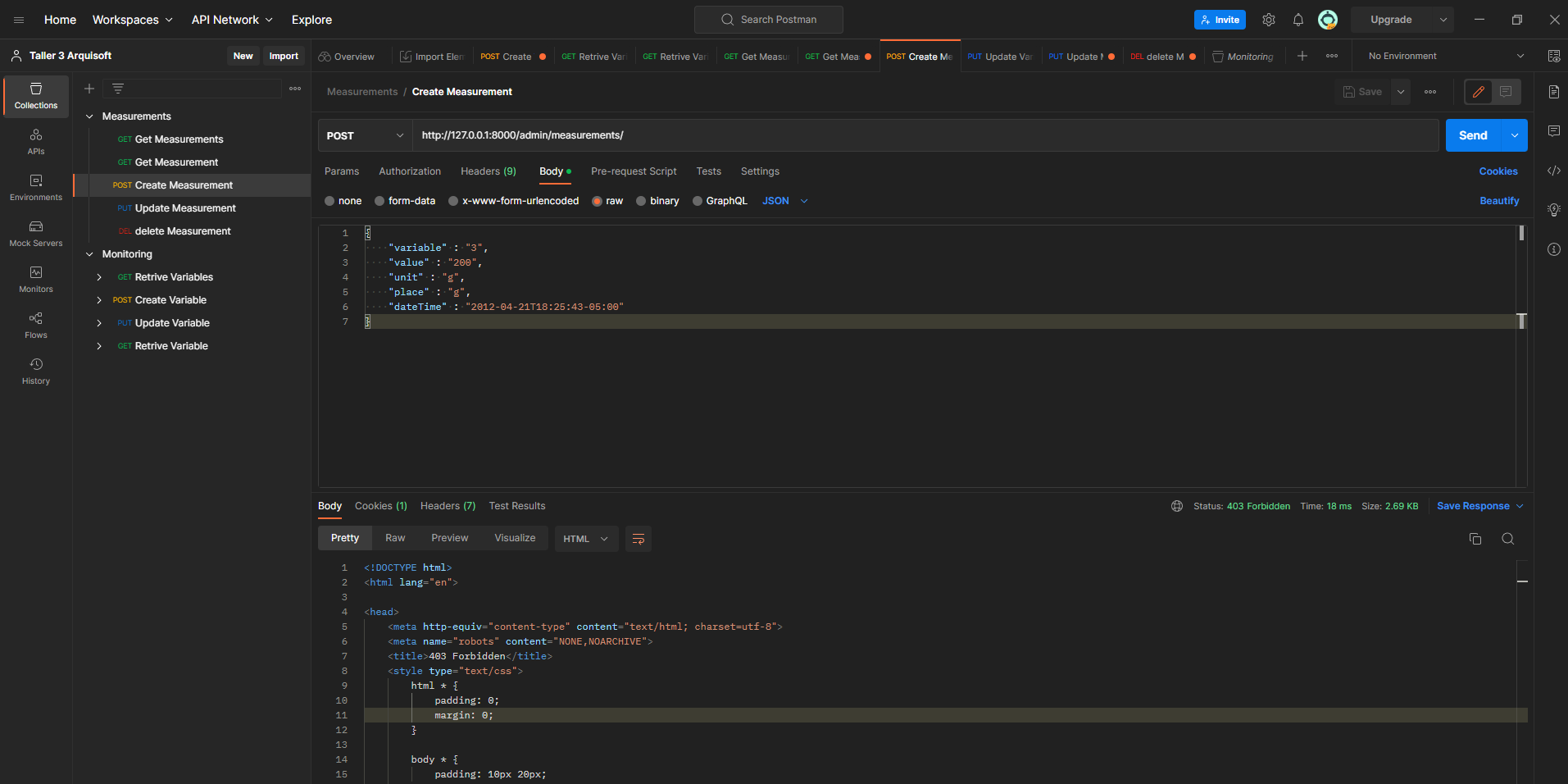
Descripción generada automáticamente**

**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 Texto

Descripción generada automáticamente

**Evidencia Pruebas Postman:**

**Evidencia 1: Create Measurement**

****

**Evidencia 2: Update Measurement**

**Captura de pantalla de un celular

Descripción generada automáticamente**

**Evidencia 3: Delete Measurement**

**Captura de pantalla de computadora

Descripción generada automáticamente**

**Evidencia 4: Get Measurements**

**Texto

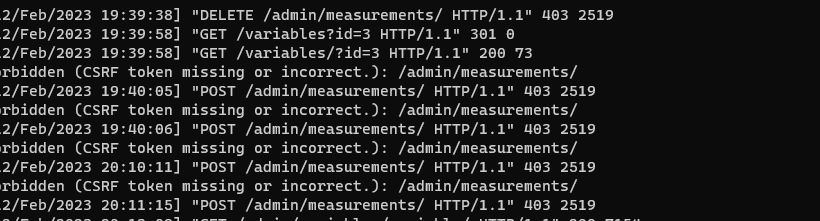
Descripción generada automáticamente**

**Evidencia 1: Get Measurement**

**Captura de pantalla de computadora

Descripción generada automáticamente**

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:



**Todas las pruebas están en el repositorio**