**Taller 5**

**Nombre**: Daniel Eduardo Rivera Acevedo

**Código**: 202123493

**Proyecto:** django-tracking2

**UML**: <https://github.com/DanielRivera10000/Taller-5.git>

**Información general**

El proyecto “django-tracking2” parece ser un sistema de seguimiento de usuarios en un sitio web. Su objetivo es registrar la información clave sobre los usuarios que entren al sitio web como al mismo tiempo sus interacciones con el sitio. Todo esto con el propósito para analizar el comportamiento de estos individuos.

La estructura del diseño la podemos dividir en 3 partes por ahora:

1. **Registro Información Visitantes**: Esta parte la podemos encontrar en la carpeta tracking, en los archivos “models.py”, “views.py” y “middleware.py”. También en la carpeta de tests. Viendo el código, como el nombre de las funciones o los nombres de las clases en estos archivos, podemos decir que el proyecto registra la información de los usuarios cuando entran y explorar el sitio web. Por ejemplo, en el archivo “models.py”, podemos ver como registra la IP, tiempo en el sitio, entre otras.
2. **Estadísticas**: Esta parte la podemos encontrar en la carpeta tracking, metiéndonos en templates y luego en snippets, hasta llegar al archivo “stats.html”. Aunque no se pudo entender completamente el código, en mi opinión esta parte es la encargada de mostrar las estadísticas de un usuario.
3. **Seguimiento:** Esta parte la podemos encontrar principalmente en la carpeta test, de tracking, en el archivo “test\_middleware.py”. Este fragmento del código esta centrado en el seguimiento. Mas precisamente se realizan pruebas de seguimiento como en visitantes anónimos, en el manejo de diferentes códigos HTTP, entre otros aspectos de seguimiento y comportamiento del usuario en su recorrido por la página.

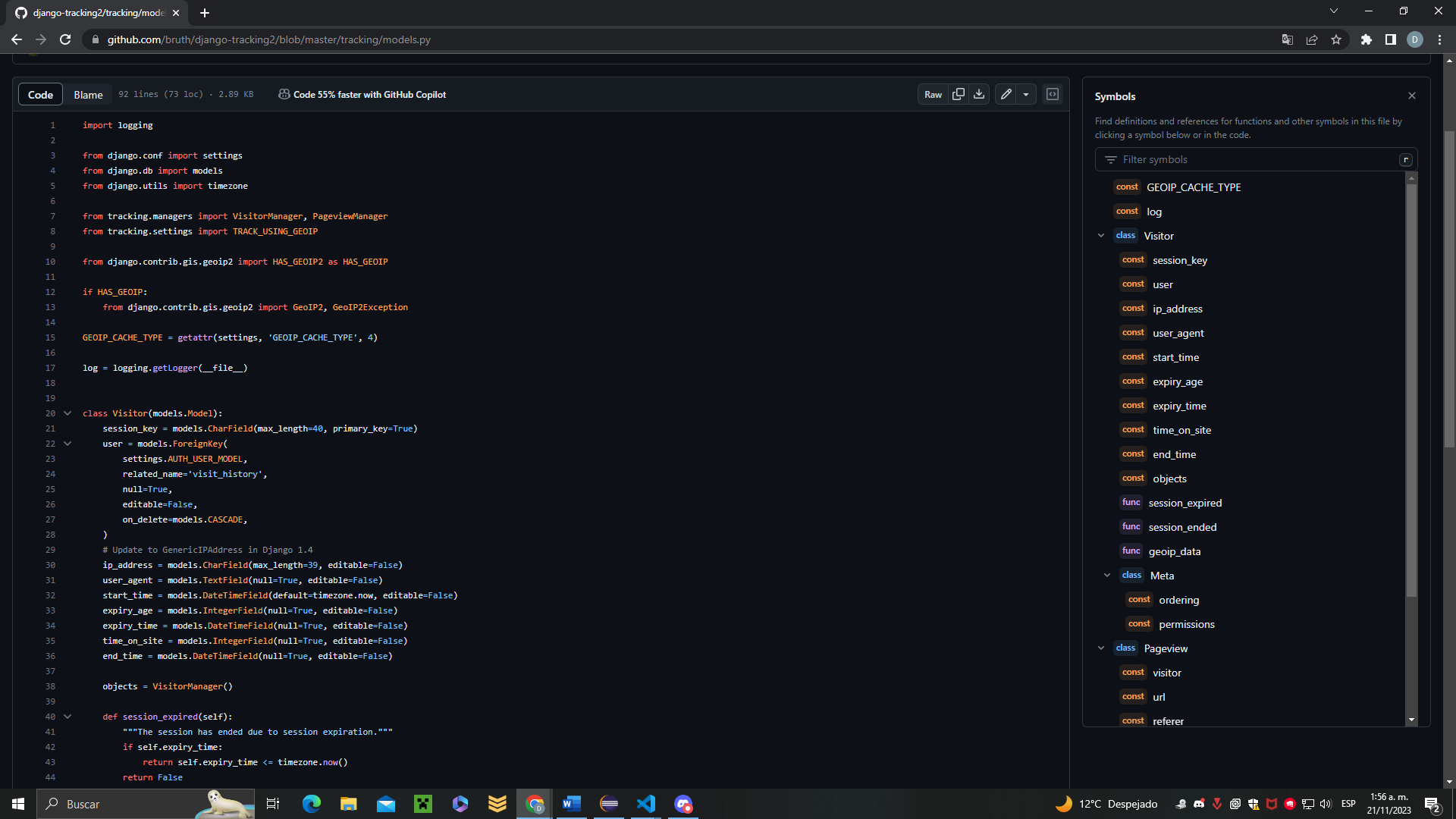
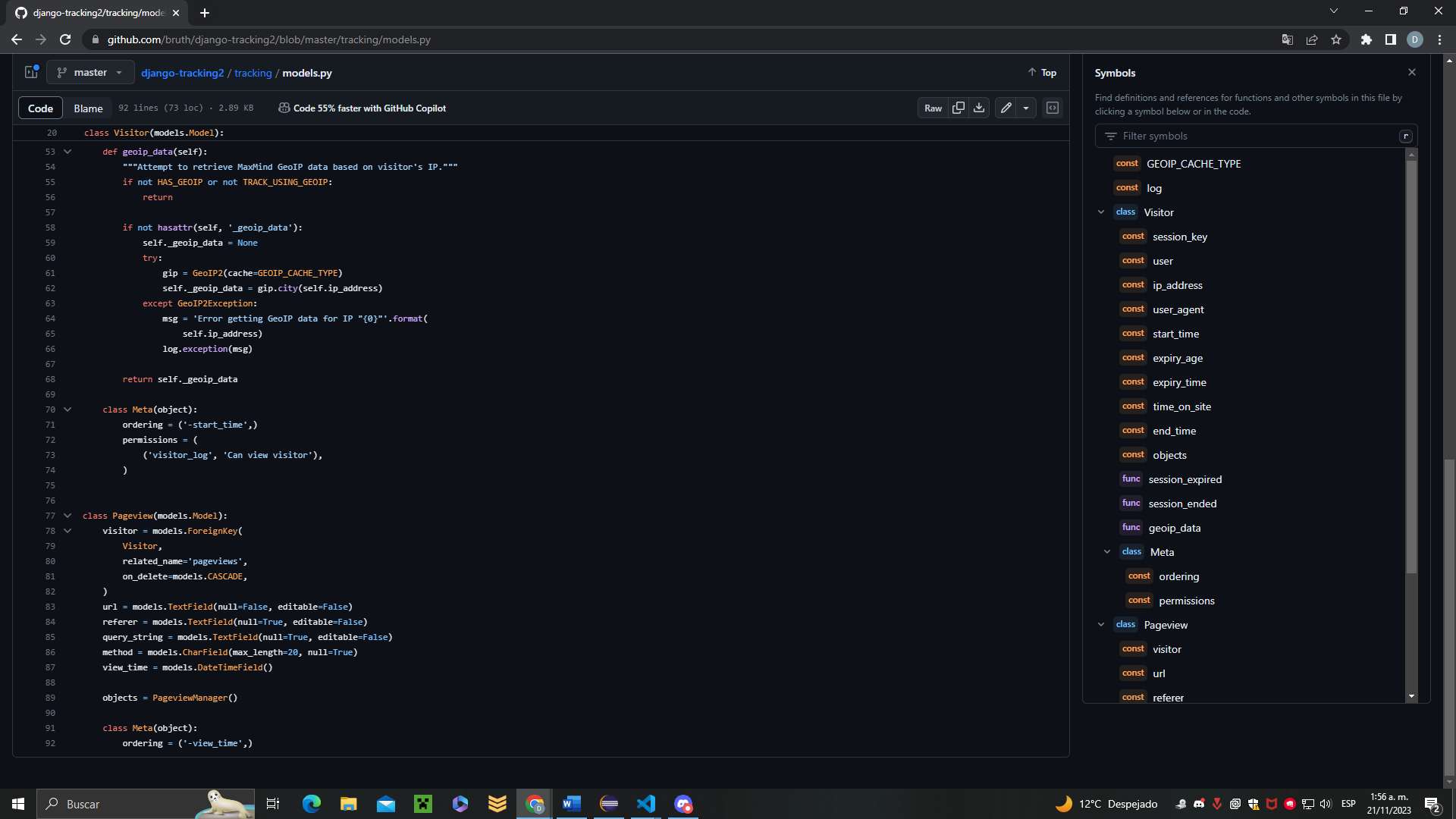
Toca resaltar que este proyecto tiene más partes en la estructura de diseño pero estas tres son las que captaron mi atención.

**Patrón de Diseño**

El patrón de diseño que se implemente en este proyecto es “Visitor”. El patrón “Visitor” es un patrón de comportamiento que representa una operación que se realiza sobre los elementos de una estructura de objeto. El visitor o su nombre en español, el visitante permite definir una nueva operación sin cambiar las clases de los elementos.

**Aplicación en el Proyecto**

El patrón de diseño “Visitor” podemos observarlo en la carpeta tracking, en su archivo “models.py”.

En este archivo, las clases Visitor y Pageview podrían considerarse una forma de patrón “Visitor”. En este proyecto, estos dos códigos representan modelos de datos que almacenan información sobre los visitantes y las vistas de la página. La clase Pageview tiene un campo (patron) “Visitor” que establece una relación con la clase visitor, ya que Pageview son visitas realizadas por Visitor. Este es una aplicación que podríamos considerar indirecta puesto que no es la funcionalidad principal del patrón.

Dentro del proyecto, el patrón de diseño se utiliza en la gestión de estadísticas y seguimiento de los usuarios en la página web. Esto se puede ver en la carpeta de tests, en el archivo “test\_managers.py” (from tracking.models import Visitor, Pageview). Cuando se ejecuta, se crean instancias de Visitor y Pageview y se realizan diferentes consultas y operaciones sobre estos modelos. En ese proceso, siento que es cuando se refleja el patrón “Visitor” donde se realizan visitas y se obtienen estadísticas.

**Ventajas y Desventajas**

Una ventaja es que el patrón “Visitor” facilita la adición de nuevas operaciones. Este patrón facilita la adición de operaciones que dependen de los componentes de objetos complejos. En nuestro caso, esta característica se aplica porque le permite al código agregar nuevas operaciones sin modificar la estructura.

Otra ventaja es que el patrón reúne operaciones relacionadas y separa las no relacionadas. En este proyecto, también aplica esta ventaja porque le está permitiendo agrupar operaciones relacionadas en visitantes específicos, manteniendo así la cohesión y separación de responsabilidades…

• ¿Por qué tiene sentido haber utilizado el patrón en ese punto del proyecto? ¿Qué ventajas tiene?

• ¿Qué desventajas tiene haber utilizado el patrón en ese punto del proyecto?

• ¿De qué otras formas se le ocurre que se podrían haber solucionado, en este caso particular, los problemas que resuelve el patrón?