**pasted-image.tiffAsignatura:** Arquitectura y Diseño de Software 2022-II Sección 01

**Profesor:** Leonardo Ibarra Becerra, Andrés Felipe Rodriguez Murillo

**Estudiante**: Javier Cerino, Daniel Bernal

**Código:** 202020873, 202020706

**Programa de Estudios**: Ingeniería de Sistemas y computación

**Taller 02**

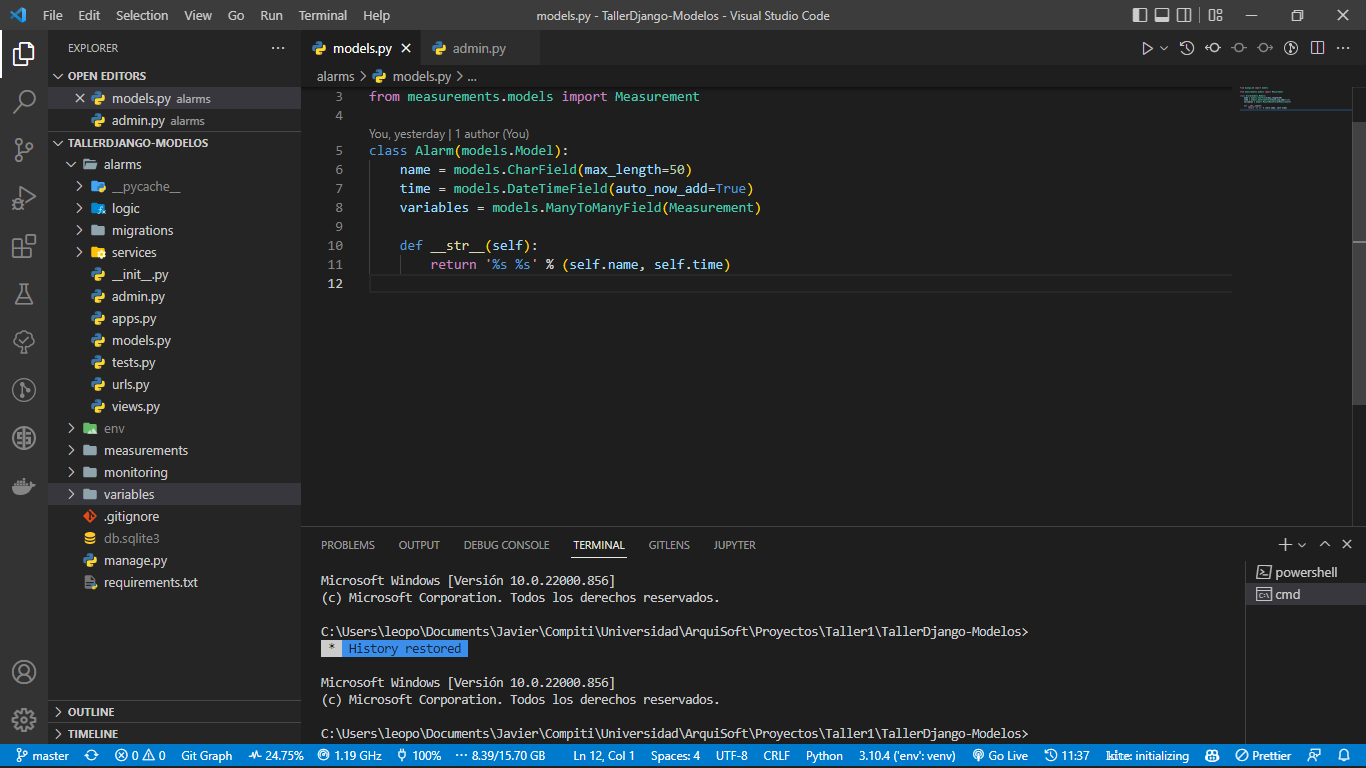


Imagen 1: models.py de Alarm

Texto

Descripción generada automáticamente

Imagen 2: Admin.py de Alarm

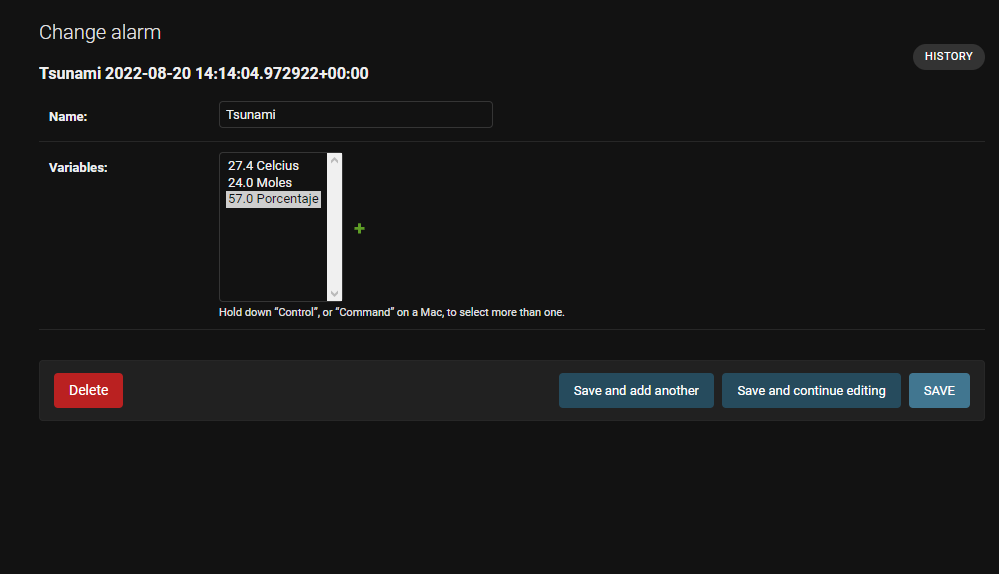


Imagen 3: Ejemplo de una instanciación de alarma.

Captura de pantalla de un celular

Descripción generada automáticamente

Imagen 4: Crear Measurments

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Imagen 5: Listado de Measurements

Captura de pantalla de computadora

Descripción generada automáticamente

Imagen 6: Crear Alarm con más de un Measurments

Captura de pantalla de computadora

Descripción generada automáticamente

Imagen 7: Crear Alarm

Definición de ORM:

Debido a que la programación orientada a objetos trabaja con objetos y no con valores escalares. Es necesario transformar dichos datos en un lenguaje entendible para la base de datos relacional. Una de estas técnicas que permite transformar los objetos a datos de una Base de datos relacional es el ORM, Object-Relational mapping.

Ventajas y desventajas de ODM y ORM

En ORM lo principal es poder trabajar con objetos y trasportarlo a la estructura de modelo relacional. Teniendo esto en consideración ORM permite unificar las diferentes sintaxis de SQL presentes en las múltiples bases de datos relacionales por lo que le ayuda a los desarrolladores de código a trabajar de una forma más fácil y sencilla entre distintas bases de datos. En consecuencia a esto, el tiempo de producción de software se ve reducido al igual que los costos de desarrollo.

No obstante, al trabar con ORM se pueden observan ciertas problemáticas como que los desarrolladores dejan de entender que es lo que está sucediendo por debajo de dicho proceso lo cual no sucedería si se trabajase directamente con SQL. En adición, ORM tiende a tener una velocidad de procesamiento menor con respecto a las sentencias SQL. Al tener un nivel de abstracción tan alto, al momento de tratar de desarrollar consultas complejas pueden presentarse fallos.

El ODM se encarga de mapear entre objetos y base de datos documental. En ODM se utilizan archivos para la comunicación con la base de datos a diferencia de las SQL que abstrae ORM. Cuando se utiliza ODM se pueden agregar nuevos campos y propiedades de una forma mucho más sencilla.

DTO: Data Transfer Object, es un objeto que se encarga de transportar información entre dos sistemas garantizando la integridad y seguridad de los datos. Son generalmente usados para comunicarse con un servidor web por medio de una API, por ejemplo. Su único propósito es contener y transportar la información.

Una DAO, Data Access Object, está encargado de suministrar una forma alternativa de visualización entre una aplicación y una o más estructuras de almacenamiento de información. Un dato interesante para tener en consideración es que al momento de enviar los datos para ser visualizados no necesita conocer el destino de estos a diferencia de los DTO que cambian en consideración a esto.

En este caso los ORM que utiliza Django son artefactos del tipo DAO ya que proveen una interfaz muy amigable entre el programa que se está creando y la base de datos relacional que permite visualizar de forma fácil y sencilla, a la vez que permite modificar los datos, sin necesidad de sentencias SQL utilizando dicha interfaz.

**Alternativas al ORM integrado de Django**

Una alternativa para el ORM de Django sería SQLAlchemy, el cual se destaca por usar la implementación de mapeo de datos, la cual genera una separación entre la estructura de datos y la estructura de objetos a diferencia de la ORM de Django que es de registros activos. De este modo, el código no depende y desconoce la estructura relacional en la base de datos para su correcto funcionamiento.

Otra alternativa para el ORM de Django sería Prisma, esta utiliza la implementación de mapeo de datos, pero lo que caracteriza a Prisma es su manera diferente de trabajar con el mapeo de datos, ya que, se define un “prisma schema” el cual sirve como fuente para las bases de datos y los modelos del lenguaje de programación. Así mismo, el cliente de Prisma sirve para leer y escribir en la base de datos de una manera eficiente y segura.