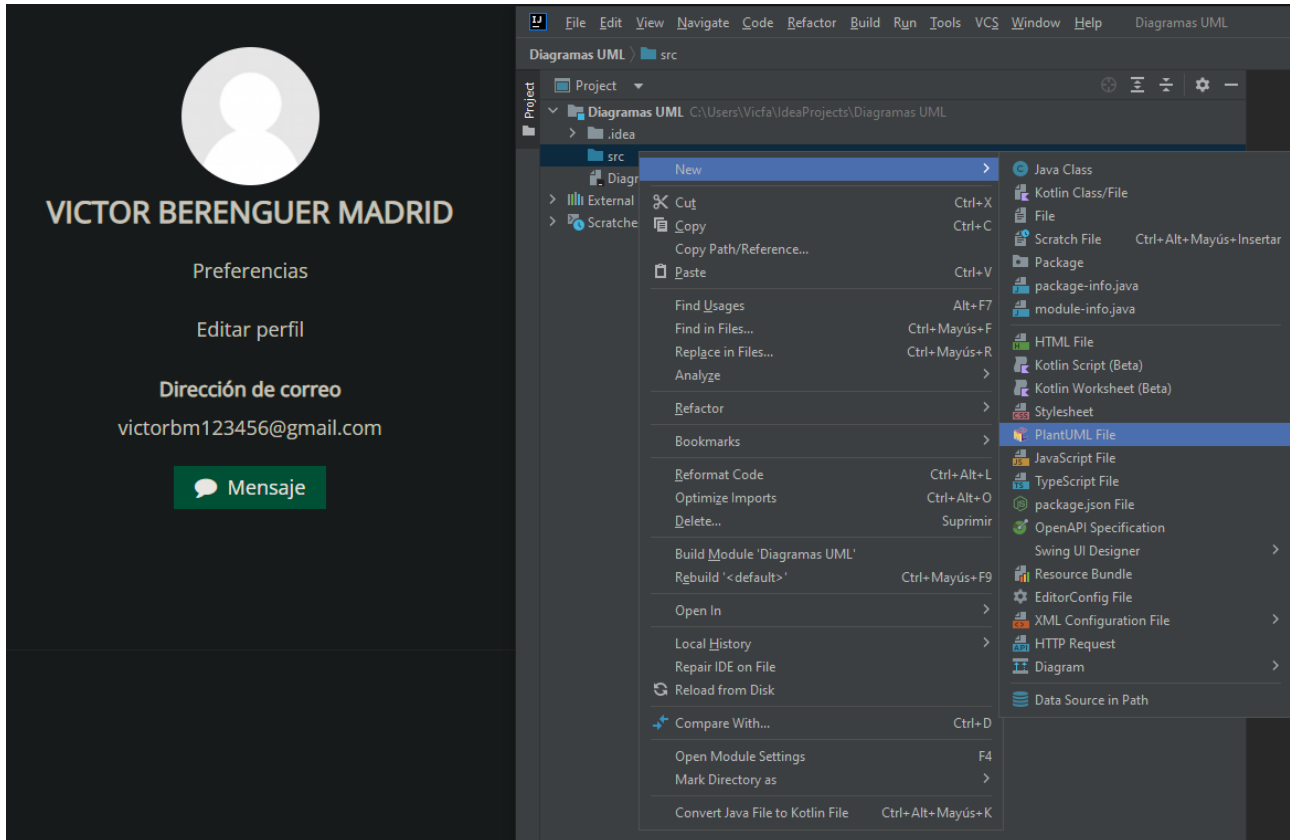


Practica UD 11. Diagramas UML, Clases.

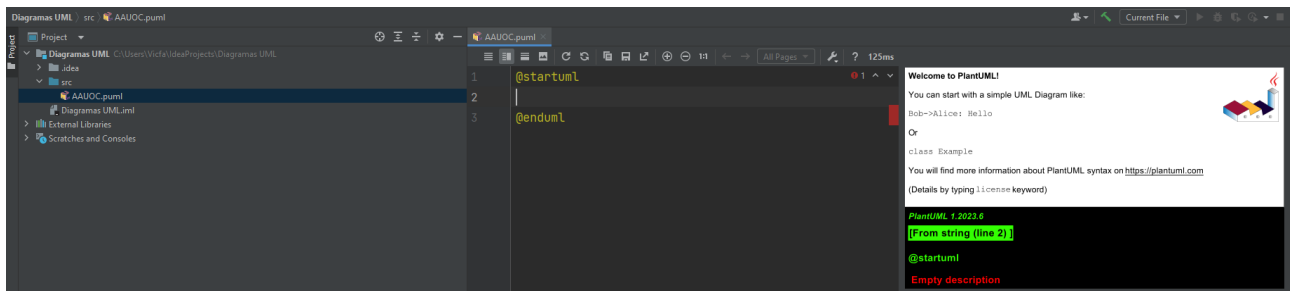
Repositorio:

<https://github.com/vicbermad/UMLClasesVictorBerenguer>

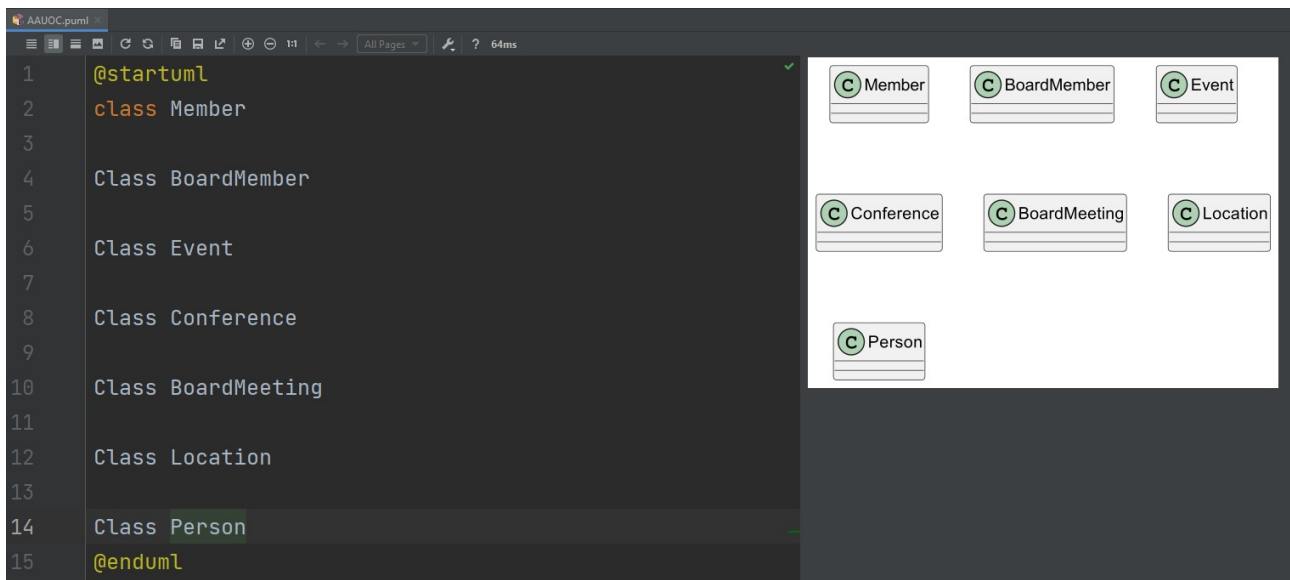
1. Identificación de las clases



En primer lugar, una vez instalado el plugin PlantUML, en un proyecto de java nuevo, creamos un fichero PlantUML en el que construiremos el diagrama.

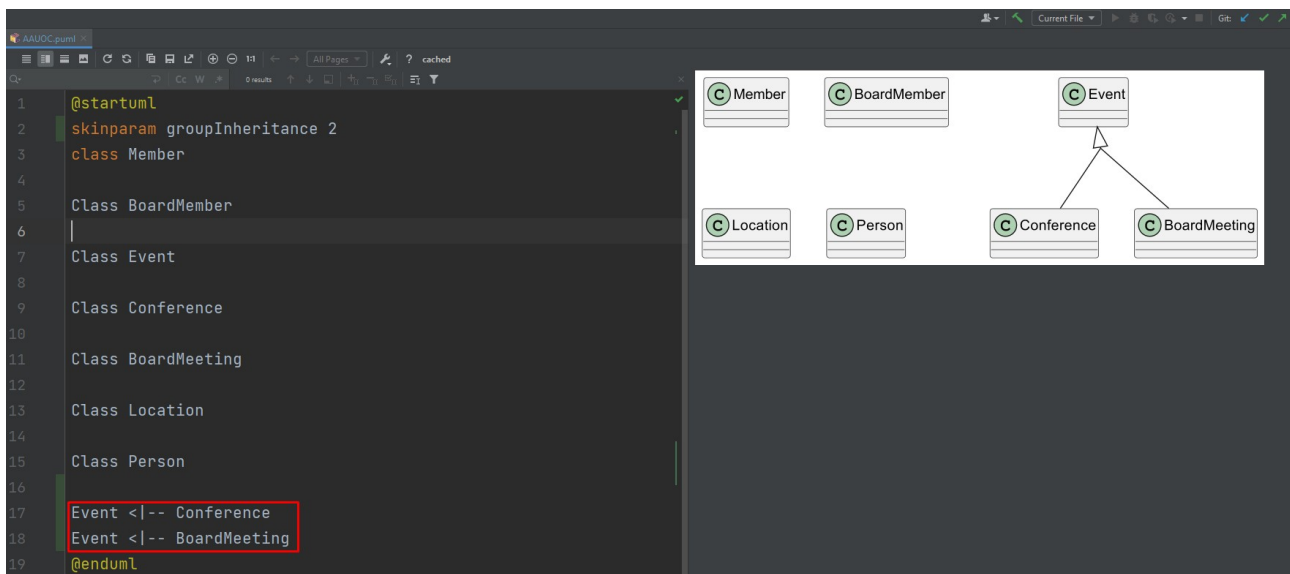


El nuevo fichero, con extensión .puml, tendrá unas delimitaciones en las que escribir el código UML. Estas serán @startuml y @enduml.



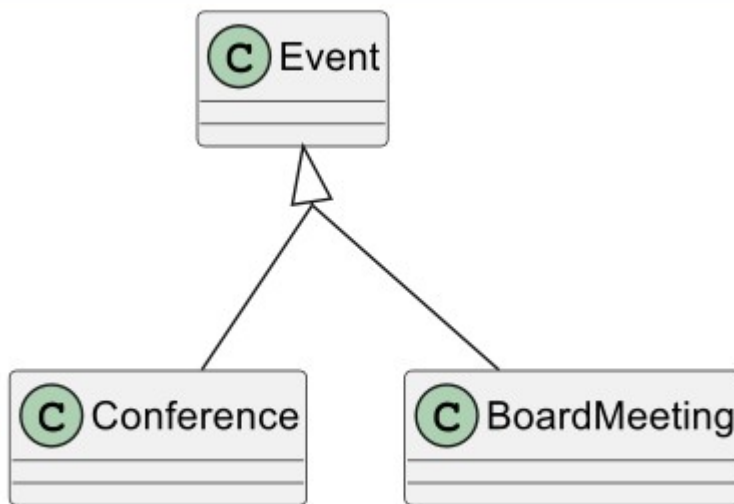
A continuación, estableceremos las clases que compondrán el proyecto. En este caso, para determinarlas, simplemente se escribe Class seguido del nombre de la clase.

2. Creación del modelo de datos



Para crear la jerarquía de herencia, escribimos el nombre de la superclase seguido de una flecha apuntando hacia ella y finalizando con el nombre de la subclase. Para cada relación de herencia se anota en una línea de código diferente. Según el tipo de relación (en este caso extensión), se “dibuja” la flecha de diferente manera, siendo la punta de la flecha un “*” para composición, “o” para agregación y sin punta de flecha en el caso de asociación.

```
AAUOC.puml x
1 @startuml
2 skinparam groupInheritance 2
3 class Member
```



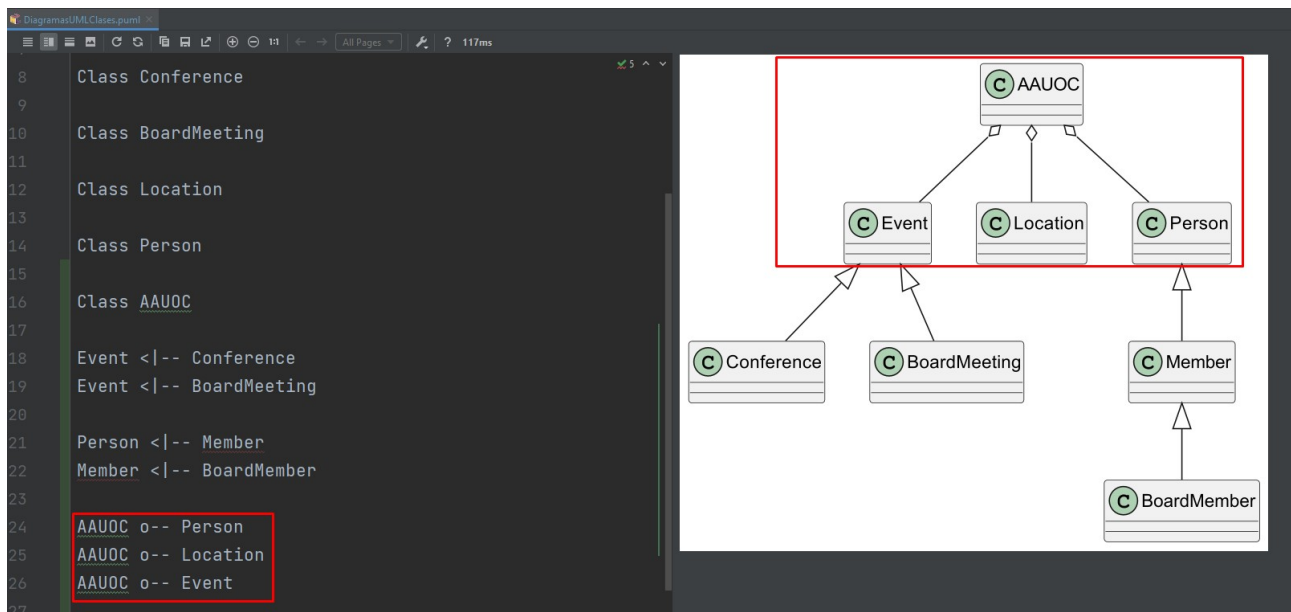
Es posible mezclar las puntas de las flechas con la línea de código *skinparam groupInheritance* seguida del número mínimo de relaciones con las que se realizará la unión.

```
AAUOC.puml
1 classDiagram
2   class BoardMember
3   class Event
4   class Conference
5   class BoardMeeting
6   class Location
7   class Person
8   Event <|-- Conference
9   Event <|-- BoardMeeting
10  Person <|-- Member
11  Member <|-- BoardMember
12 @enduml
```

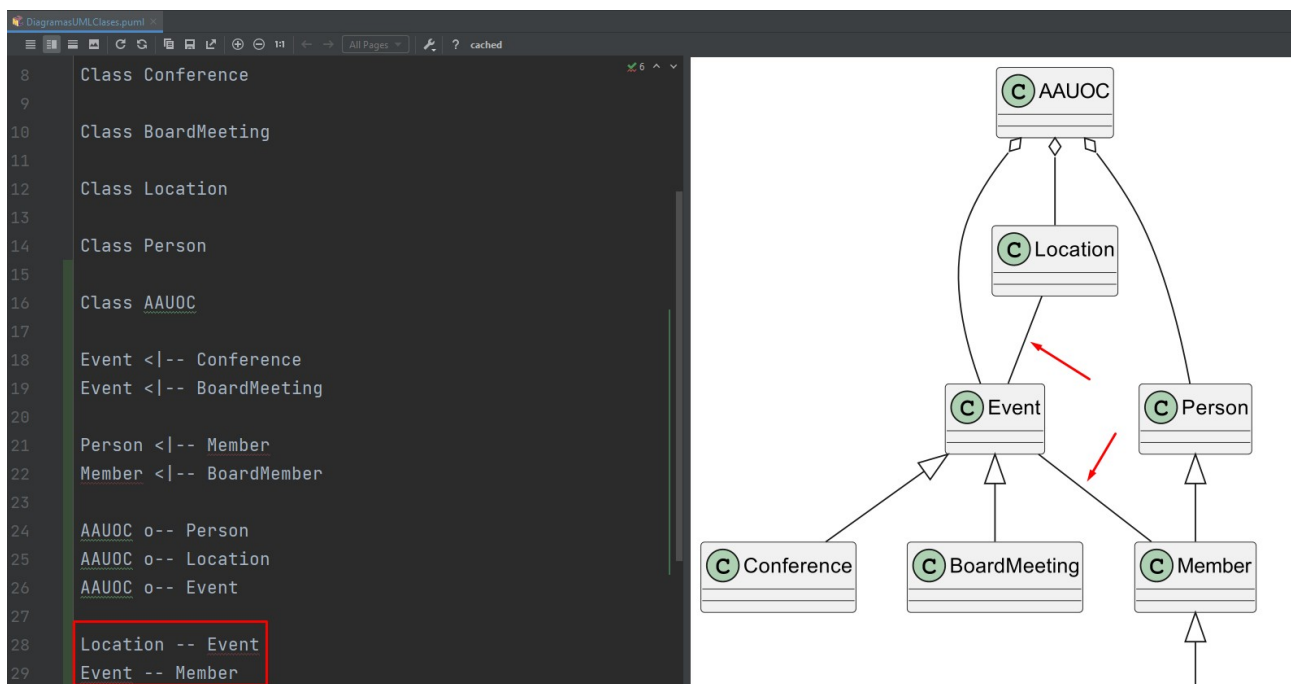
```
classDiagram
    Person <|-- Member
    Member <|-- BoardMember
    Event <|-- Conference
    Event <|-- BoardMeeting
```

A screenshot of a UML modeling tool interface. On the left, a code editor shows a Puml script. On the right, a preview window displays the resulting UML class diagram. The diagram shows a hierarchy where 'Person' is the superclass of 'Member', and 'Member' is the superclass of 'BoardMember'. Additionally, 'Event' is the superclass of 'Conference' and 'BoardMeeting'. The 'BoardMember' class is highlighted with a red box in the code editor.

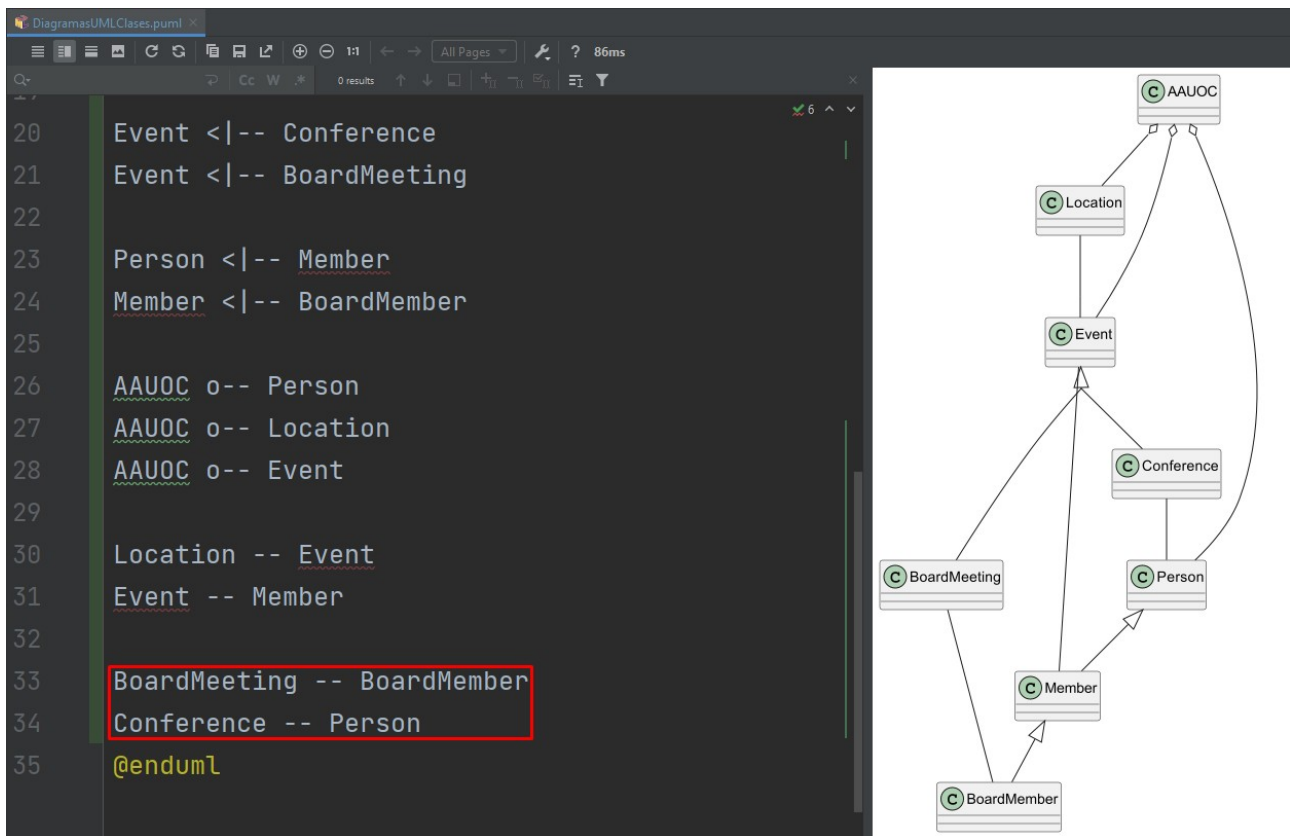
Posteriormente, establecemos el resto de jerarquías. En este caso, la clase Person será la superclase y Member la subclase. A su vez, Member será la superclase de BoardMember. Ambas relaciones de extensión.



La relación de la superclase principal, AAUOC y las clases Event, Location y Person serán de tipo agregativas.

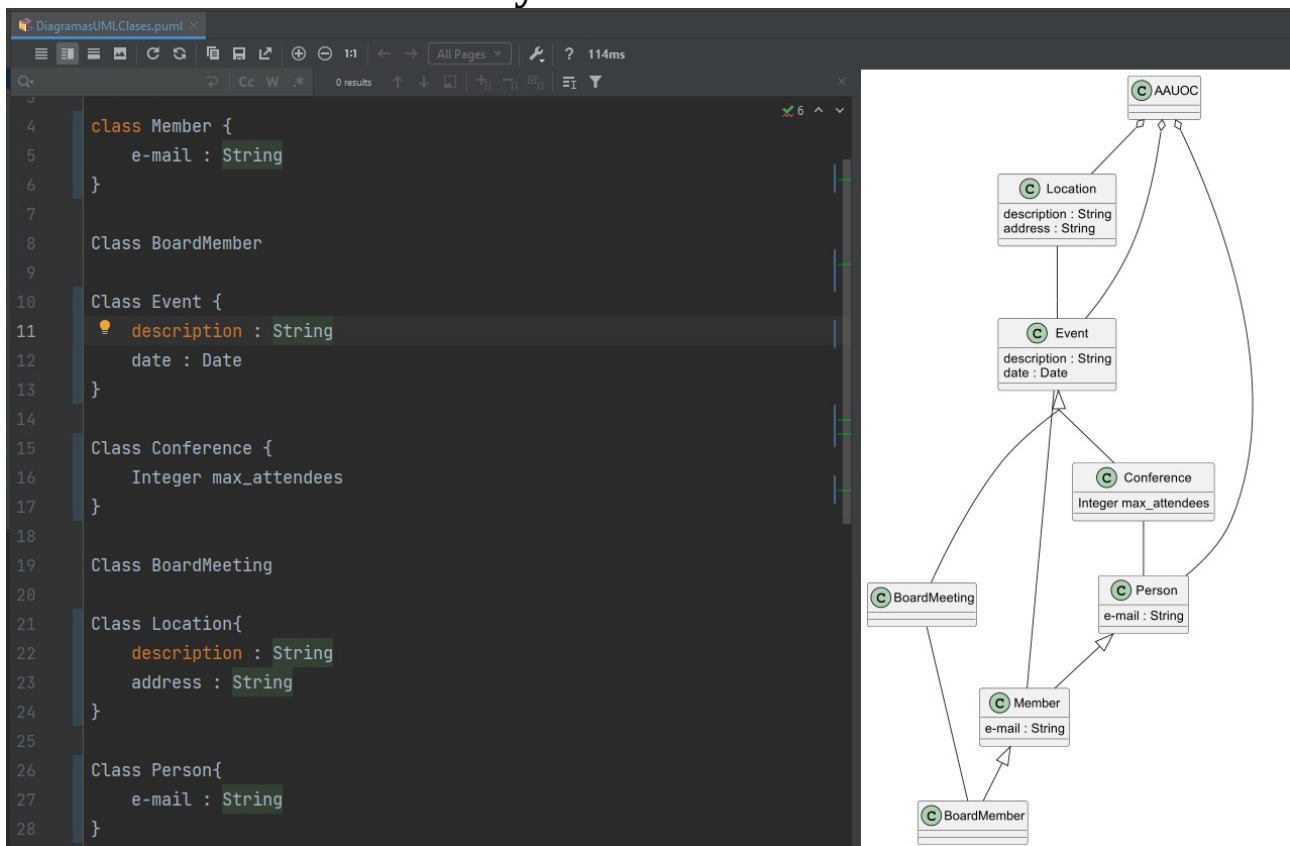


Las relaciones entre Event y Location y Member y Event serán de tipo asociativas, es decir, dibujada con dos guiones.



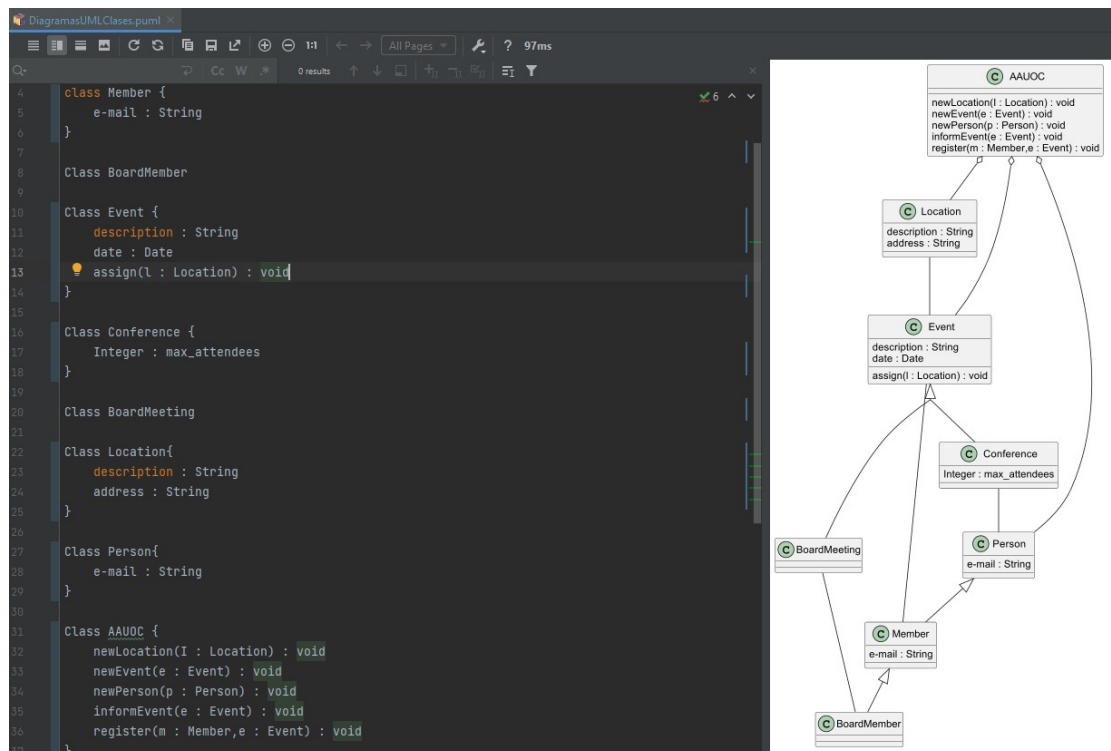
También serán asociativas las relaciones entre BoardMeeting y BoardMember y Conference y Person.

3. Inclusión de atributos y métodos

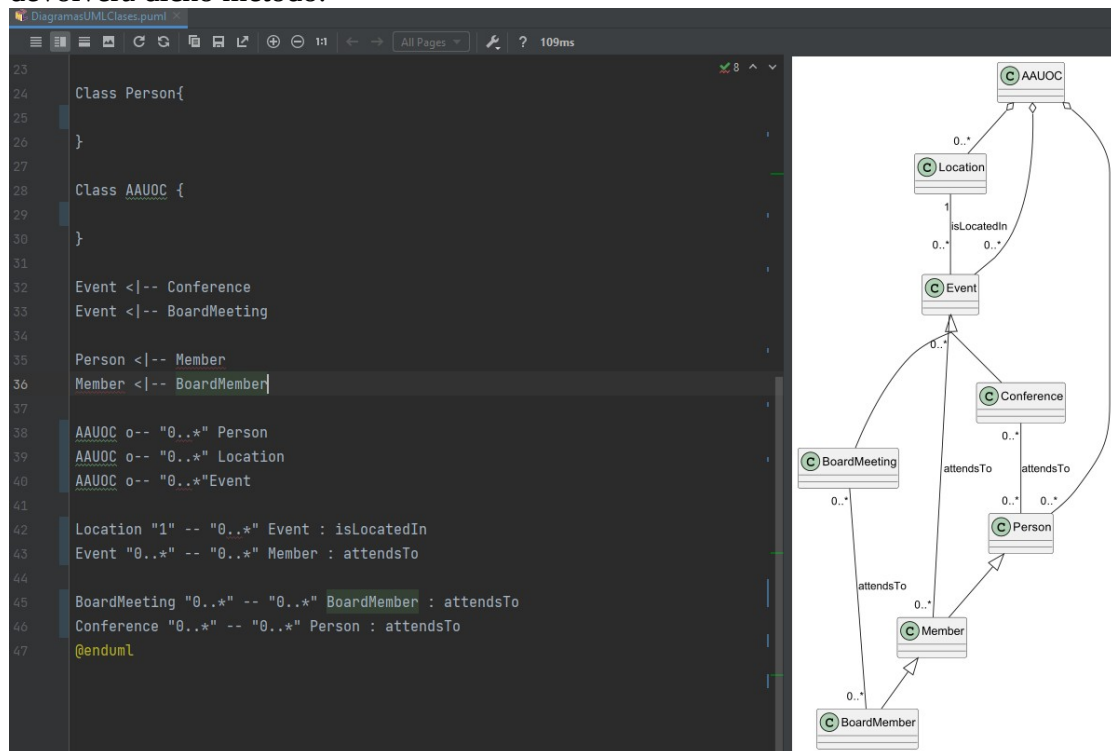


El siguiente paso, será la inclusión de los atributos a sus respectivas clases. Para introducirlos, entre llaves delimitadas en cada clase, se escribirá el nombre del atributo seguido del tipo de datos, separados entre dos puntos.

4. Inclusión de la cardinalidad y navegabilidad de las relaciones



Para añadir los métodos a las clases, de la misma forma que los atributos, dentro de las llaves introducimos el nombre del método, a continuación, entre paréntesis, escribimos el dato que queremos que sea enviado al método. Separado por un paréntesis, establecemos el tipo de dato que devolverá dicho método.




A continuación, antes de implantar la cardinalidad y la navegabilidad de las relaciones, se eliminarán los métodos y atributos de las clases. Para realizar la cardinalidad, en la relación que habíamos fijado anteriormente, se introduce entre comillas dicha cardinalidad. Se escribirá antes o después de la relación según el lado en la que queramos establecer la cardinalidad. La navegabilidad, se introduce después de la relación separada por dos puntos.

Área personal

Eventos

🔔

💬



VICTOR BERENGUER MADRID

Preferencias

Editar perfil

Dirección de correo

victorbm123456@gmail.com

Mensaje

DiagramasUMLClasses.puml

DiagramasUMLClasses.puml

Diagrama de clases UML

```
classDiagram
    class AAUOC {
    }
    class Location {
    }
    class Event {
    }
    class Conference {
    }
    class BoardMeeting {
    }
    class Person {
    }
    class BoardMember {
    }
    class Member {
    }
    AAUOC "0..*" -- "0..*" Location
    Location "0..*" -- "0..*" Event
    Event "0..*" -- "0..*" Conference
    Event "0..*" -- "0..*" BoardMeeting
    BoardMeeting "0..*" -- "0..*" Person
    BoardMeeting "0..*" -- "0..*" Member
    Person "0..*" -- "0..*" Member
    Member "0..*" -- "0..*" BoardMember
    Member "0..*" -- "0..*" BoardMember
```

Font size: 12pt Reset to 13pt