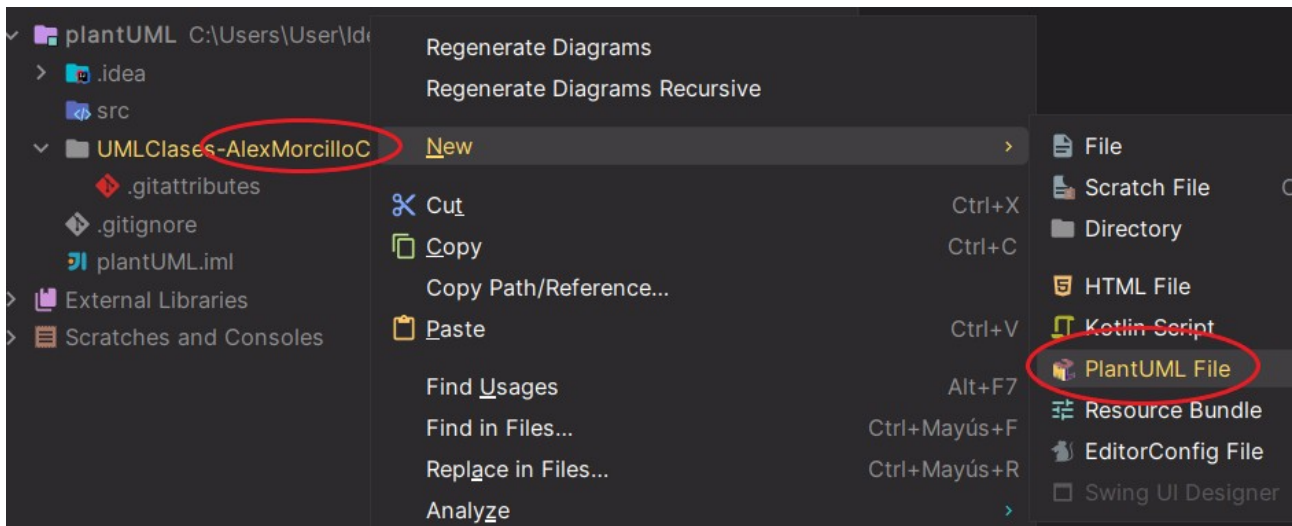


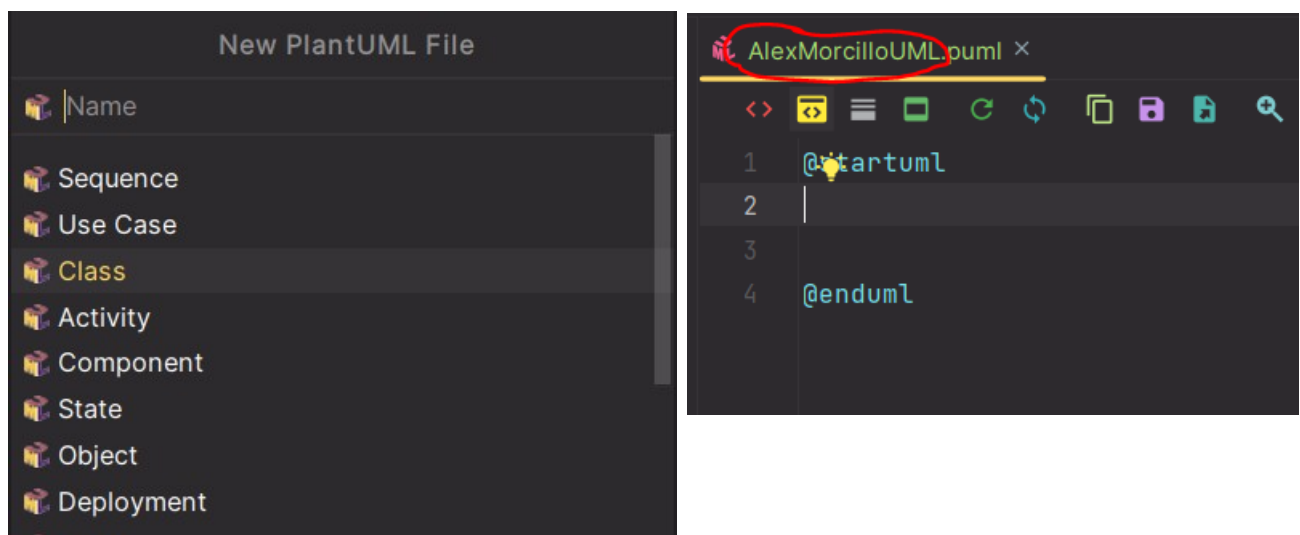
Práctica Diagramas UML Clases.

Alex Morcillo Caubera

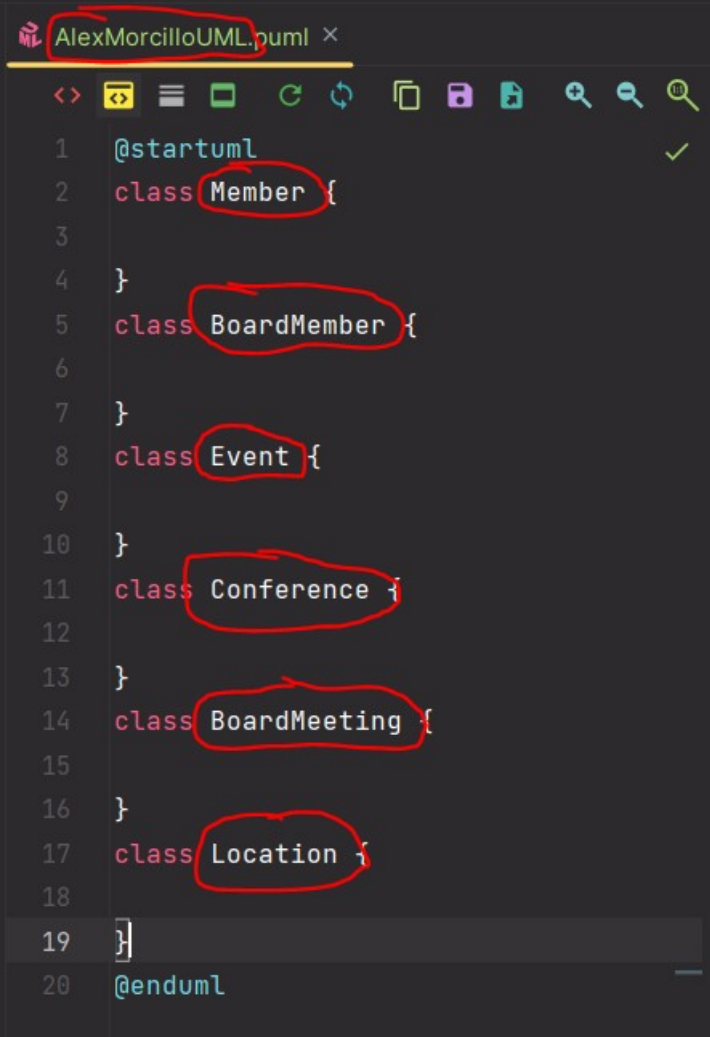
1. Lo primero que haremos será descargar el plugin de PlantUML, basta con ir a la sección de “Plugins” en el menú de opciones que puedes abrir con el atajo “Ctrl+Alt+S”, buscarlo e instalarlo. Yo ya lo tengo así que pasará al siguiente paso, que consiste en crear un archivo PlantUML. Hacemos click secundario en la carpeta deseada y seleccionamos “PlantUML file”.



Se nos abrirá este menú, seleccionaremos la opción “Class” y luego borraremos el contenido de esta. Se quedaría así:

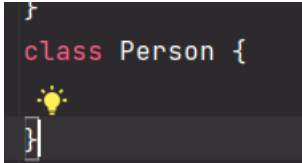


2. Ahora que tenemos el archivo creado, vamos a identificar las clases que hayan e introducirlas de esta manera:



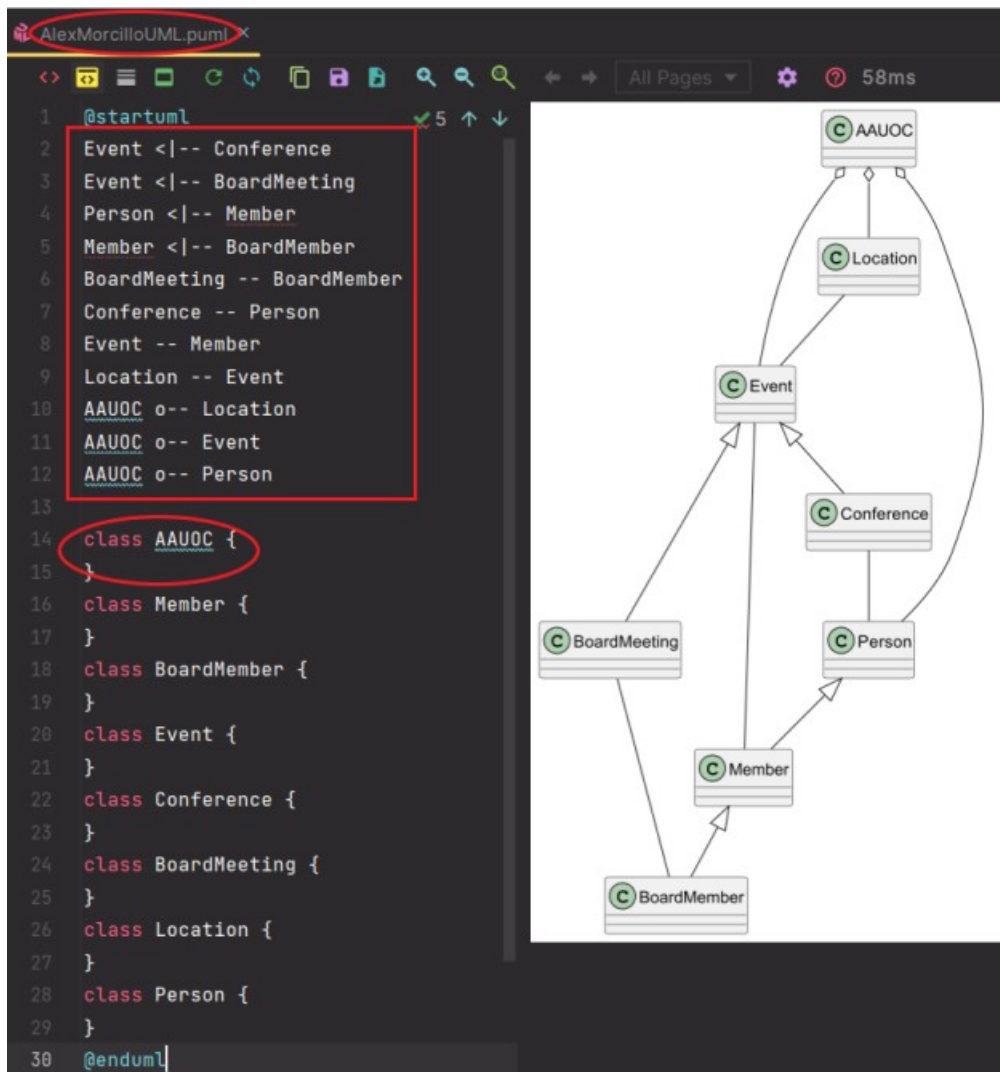
```
1 @startuml
2 class Member {
3
4 }
5 class BoardMember {
6
7 }
8 class Event {
9
10 }
11 class Conference {
12
13 }
14 class BoardMeeting {
15
16 }
17 class Location {
18
19 }
20 @enduml
```

Adicionalmente, creamos la clase “Person” para identificar a los conferenciantes.

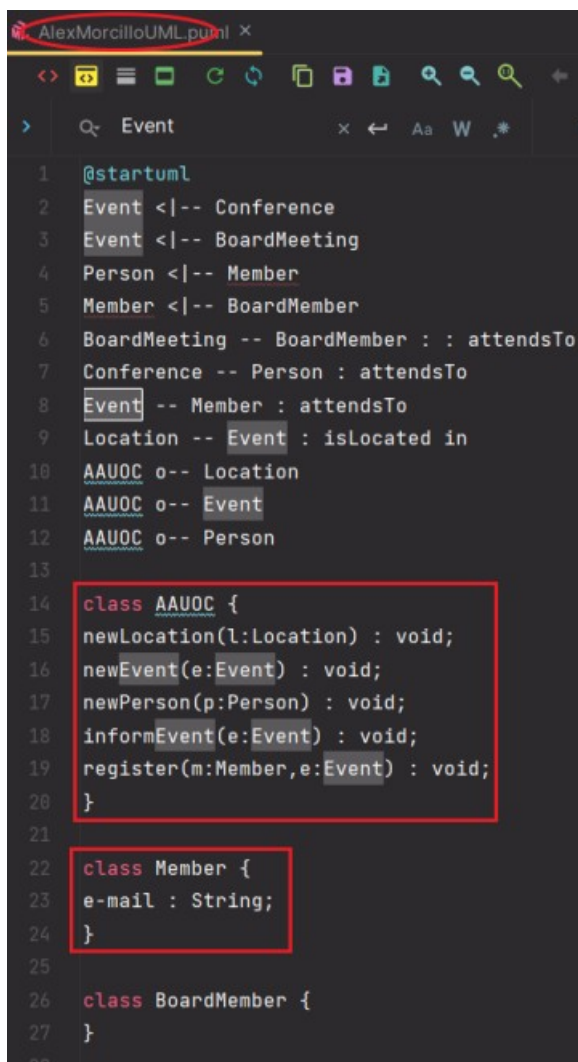


```
}
class Person {
  ⚡
}
```

3. Una vez creadas las clases necesarias, vamos a establecer las relaciones entre éstas. Para relacionar todas las clases de manera lógica, hemos de añadir una clase general llamada “AAUOC”. Las asociaciones se harían de la siguiente forma y obtendríamos un diagrama como el de la derecha:



4. Ahora es turno de declarar atributos y métodos de cada clase. Para los atributos, basta con escribir su nombre seguido del tipo de dato que es, y para los métodos hay dos formas, para la primera se escribe el tipo de dato que devuelve seguido de su nombre y unos paréntesis, dentro de los cuáles van los parámetros descritos de la siguiente forma “e:Event”, donde “e” es el nombre del parámetro y “Event” el tipo de parámetro. Y para la segunda, se escribe el nombre del método y luego unos paréntesis con sus parámetros dentro, por último escribimos “ : ” al final y añadimos el tipo de valor que devuelve. Añadiendo los métodos y atributos necesarios, quedaría de la siguiente forma:

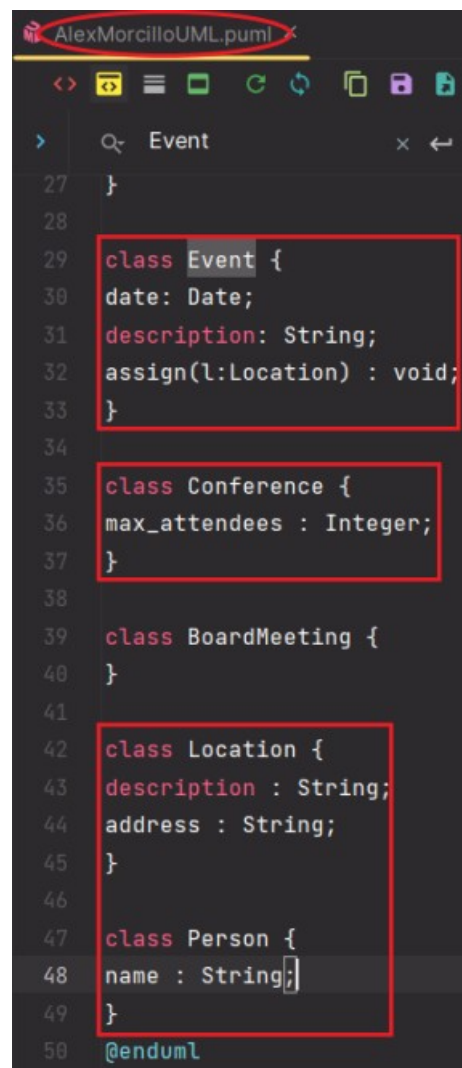


```
@startuml
Event <|-- Conference
Event <|-- BoardMeeting
Person <|-- Member
Member <|-- BoardMember
BoardMeeting -- BoardMember : attendsTo
Conference -- Person : attendsTo
Event -- Member : attendsTo
Location -- Event : isLocated in
AAUOC o-- Location
AAUOC o-- Event
AAUOC o-- Person

class AAUOC {
  newLocation(l:Location) : void;
  newEvent(e:Event) : void;
  newPerson(p:Person) : void;
  informEvent(e:Event) : void;
  register(m:Member,e:Event) : void;
}

class Member {
  e-mail : String;
}

class BoardMember {
```



```

}

class Event {
  date: Date;
  description: String;
  assign(l:Location) : void;
}

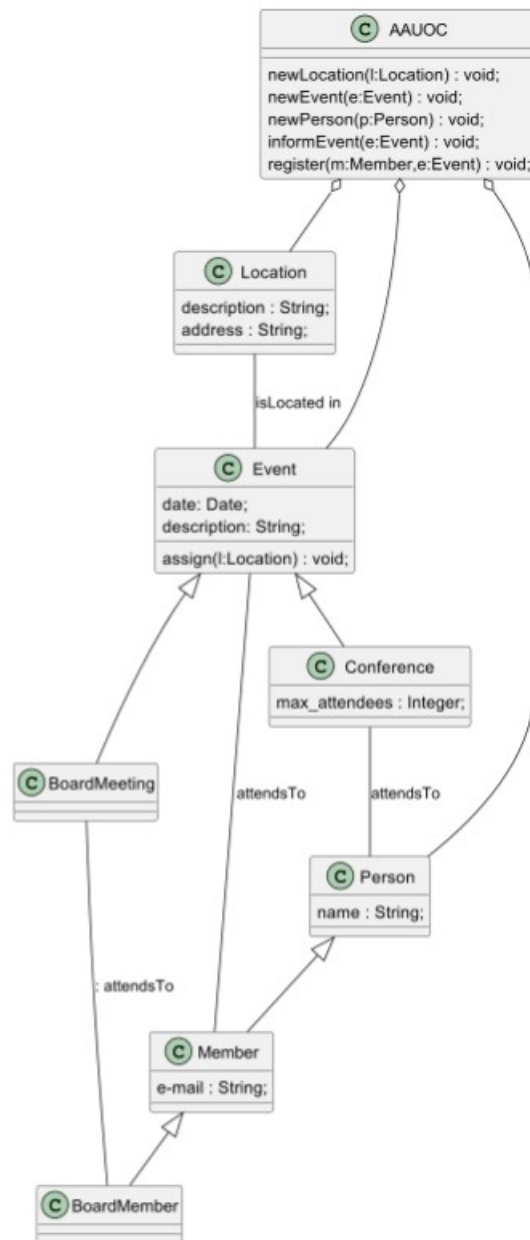
class Conference {
  max_attendees : Integer;
}

class BoardMeeting {
}

class Location {
  description : String;
  address : String;
}

class Person {
  name : String;
}

@enduml
```



5. Por último, añadimos las cardinalidades.

```
1 @startuml
2 Event <|-- Conference
3 Event <|-- BoardMeeting
4 Person <|-- Member
5 Member <|-- BoardMember
6 BoardMeeting "0..*" -- "0..*" BoardMember : : attendsTo
7 Conference "0..*" -- "0..*" Person : attendsTo
8 Event "0..*" -- "0..*" Member : attendsTo
9 Location "1" -- "0..*" Event : isLocated in
10 AAUOC o-- "0..*" Location
11 AAUOC o-- "0..*" Event
12 AAUOC o-- "0..*" Person
```

