

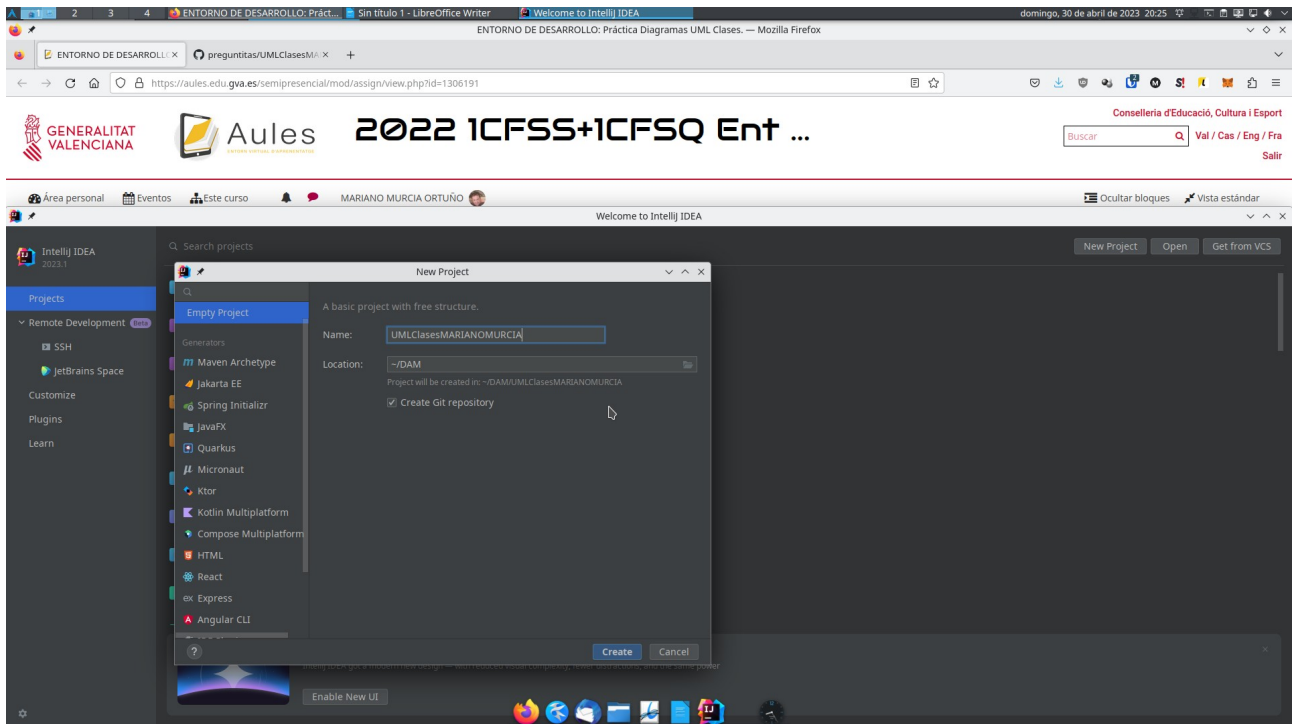
PRÁCTICA

ENTORNOS DE DESARROLLO
Memoria UMLClasesMARIANOMURCIA

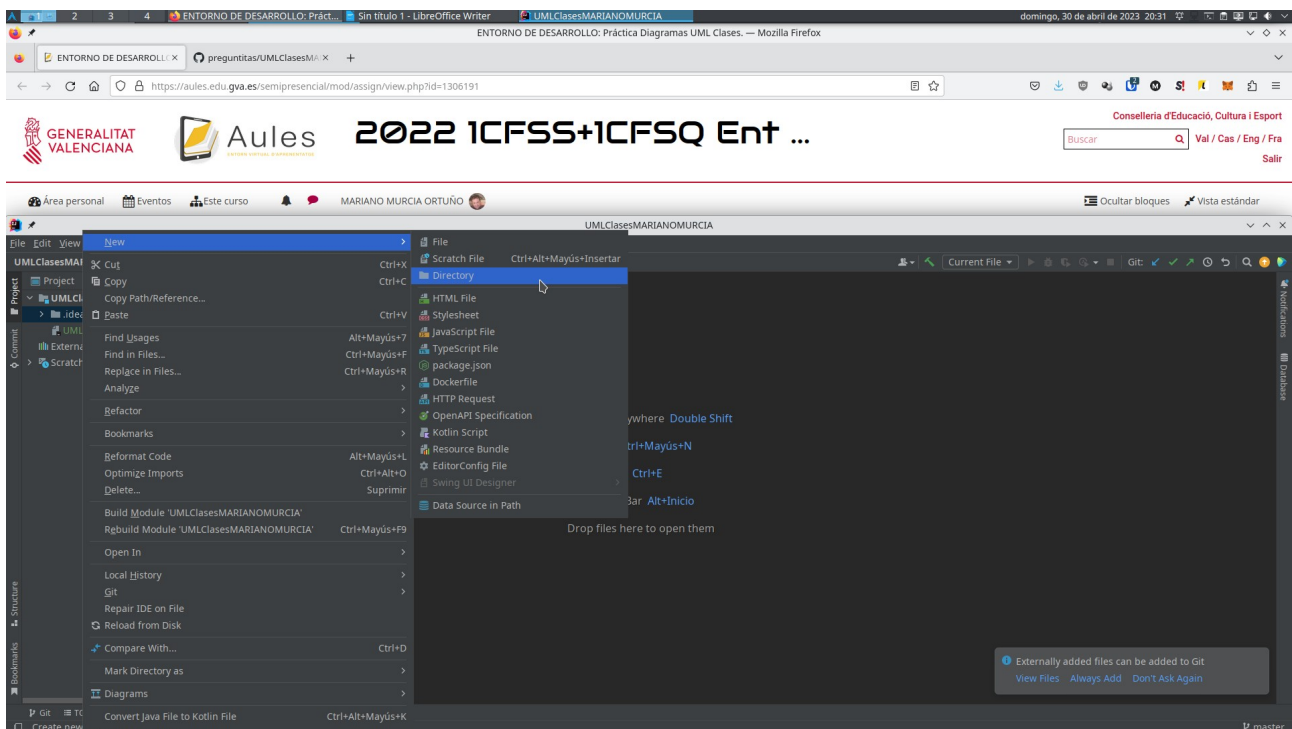
Índice

1. Creación del proyecto.....	3
2. Instalación plugin PlantUML.....	4
3. Inicialización del repositorio.....	5
4. Ejercicio.....	6
5. Repositorio remoto.....	9

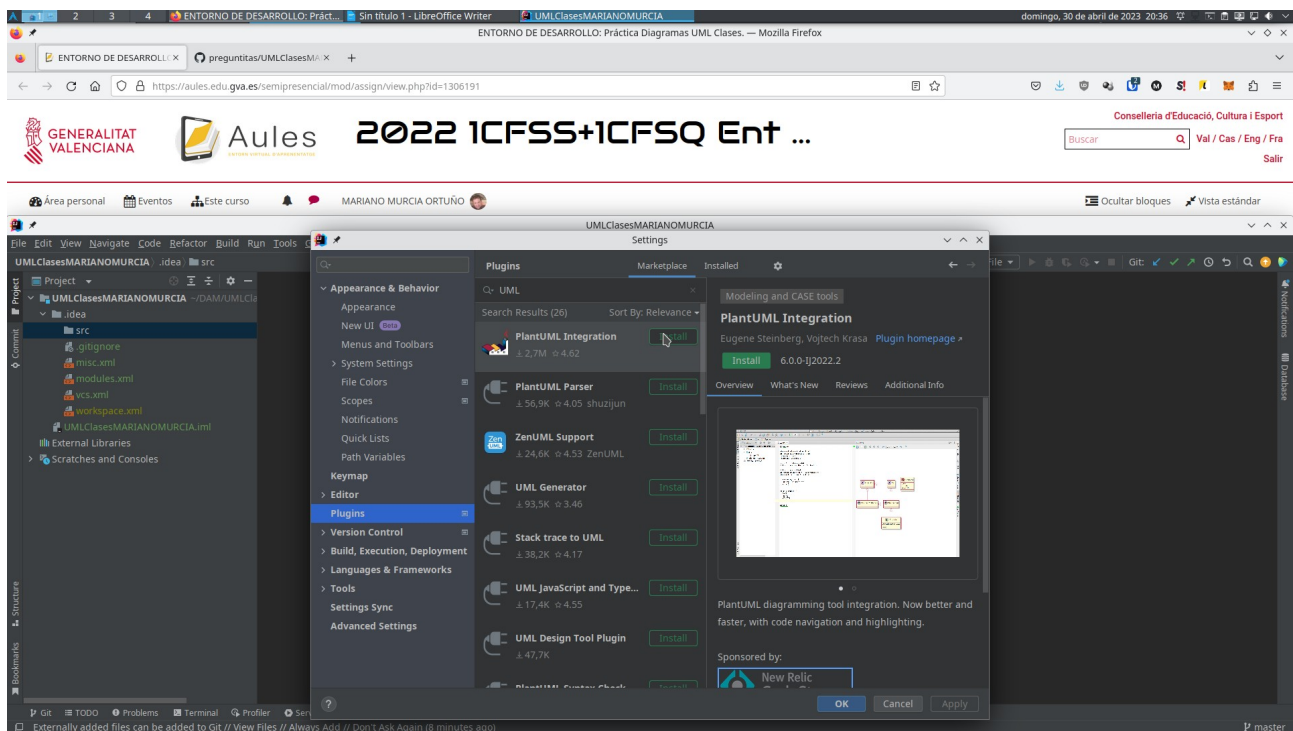
1. Creación del proyecto



Creamos nuestro proyecto en IntelliJ IDEA, usando las opciones de "Empty Project" y "Create Git repository".

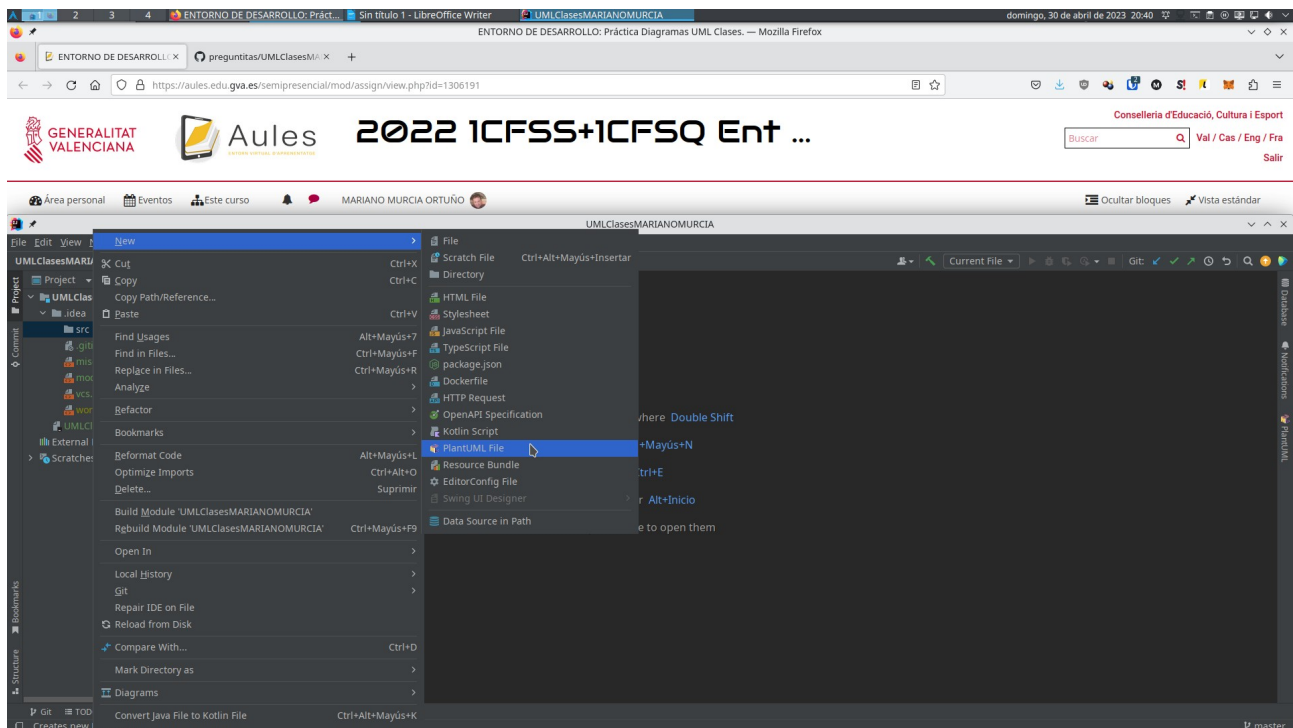


Creamos un nuevo directorio, en el que guardaremos el source del proyecto.



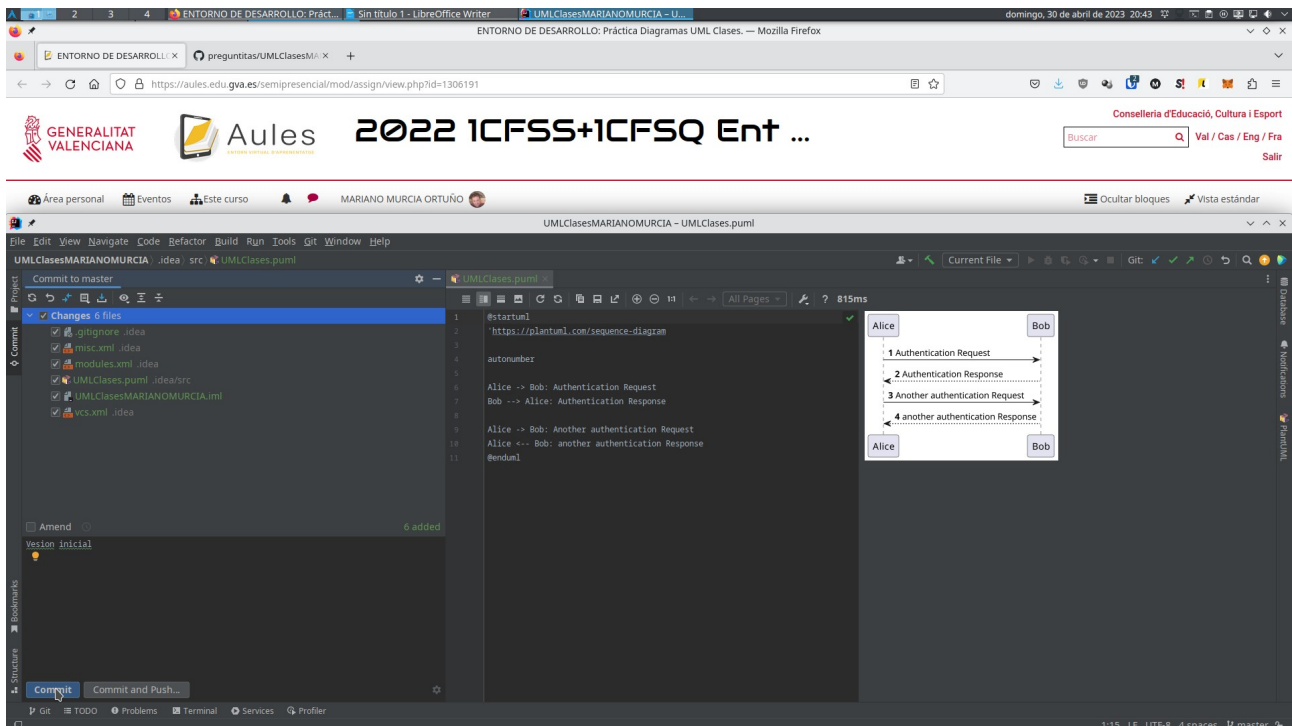
2. Instalación plugin PlantUML

Intalamos el pugin "PlantUML integration", a traves de la opcion "File->Settings->Plugins"



Creamos un nuevo archivo PlantUML. Con el nombre de UMLClases

3. Inicialización del repositorio



Creamos el primer Commit, con la version inicial.

4. Ejercicio

The screenshot shows the IntelliJ IDEA interface with the 'UMLClasses.puml' file open. The code defines an Event class and two subclasses, Conference and BoardMeeting. The UML diagram on the right shows Event as the superclass with two subclasses, Conference and BoardMeeting.

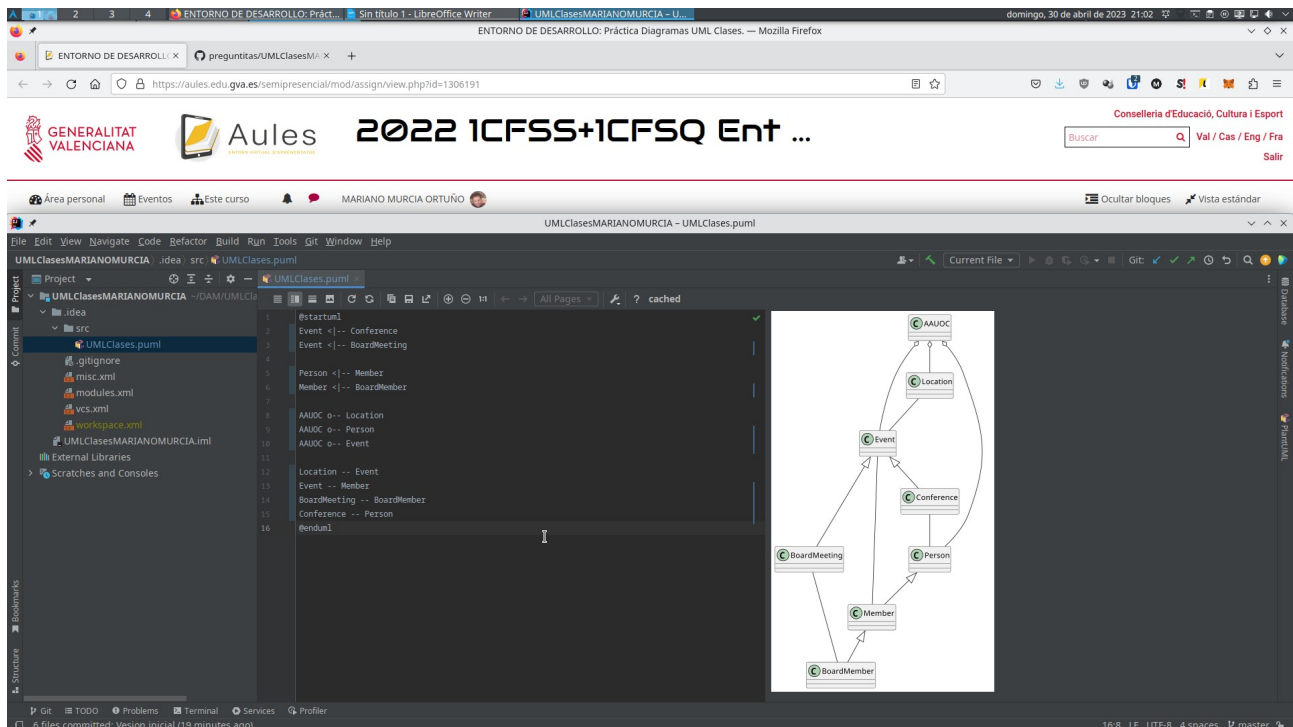
```
@startuml
class Event
class Conference
class BoardMeeting
Event <|-- Conference
Event <|-- BoardMeeting
@enduml
```

Modificamos el código con las relaciones entre la superclase Event y sus subclases Conference y BoardMeeting.

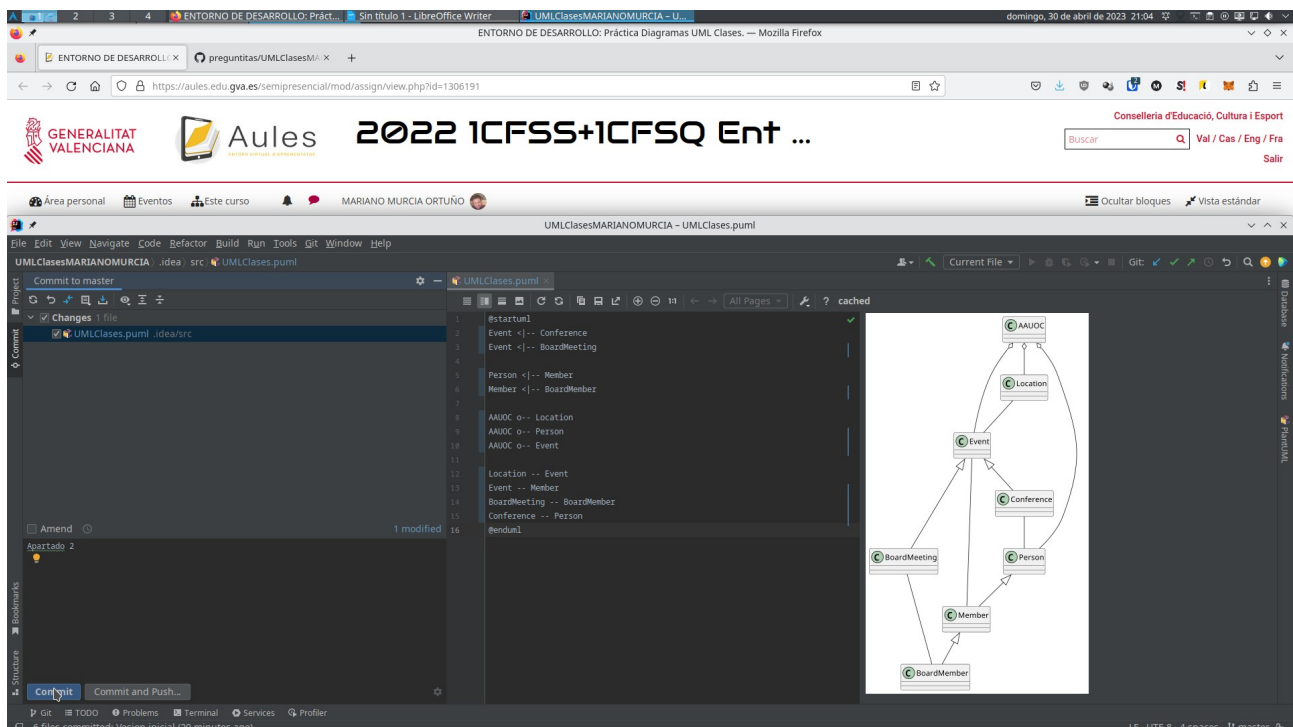
The screenshot shows the IntelliJ IDEA interface with the 'UMLClasses.puml' file open. The code defines an Event class and two subclasses, Conference and BoardMeeting. Additionally, a new class Member is added, which is a subclass of Person. The UML diagram on the right shows Event as the superclass with two subclasses, Conference and BoardMeeting. A new class Member is added, which is a subclass of Person.

```
@startuml
class Event
class Conference
class BoardMeeting
Event <|-- Conference
Event <|-- BoardMeeting
class Person
class Member
Member <|-- BoardMember
@enduml
```

Y la jerarquía entre los miembros de la asociación



Además, establecemos las clases Location y AAUOC, y las relaciones con el resto de clases



Creamos un commit habiendo terminado el apartado 2.

The screenshot shows the IDE with the UML diagram and code for the first commit. The UML diagram includes classes: `AAUOC`, `Location`, `Event`, `Conference`, `BoardMeeting`, `Person`, and `Member`. The code defines the classes and their methods.

```

1 @startuml
2 Event <|-- Conference
3 Event <|-- BoardMeeting
4
5 Person <|-- Member
6 Member <|-- BoardMember
7
8 AAUOC o-- Location
9 AAUOC o-- Person
10 AAUOC o-- Event
11
12 Location -- Event : islocated In
13 Event -- Member : attendsTo
14 BoardMeeting -- BoardMember : attendsTo
15 Conference -- Person : attendsTo
16
17 class AAUOC {
18     newLocation(l : Location):void
19     newEvent(e : Event) : void
20     newPerson(p : Person) : void
21     informEvent(e : Event) : void
22     register(m : Member,e : Event) : void
23 }
24
25 class Location {
26     description : String
27     address : String
28 }

```

The code defines the classes and their methods. The UML diagram shows the relationships between the classes.

Añadimos las etiquetas a las relaciones y declaramos los campos y los métodos en sus clases correspondientes. Acabamos haciendo commit para guardar los cambios.

The screenshot shows the IDE with the UML diagram and code for the second commit. The UML diagram includes classes: `AAUOC`, `Location`, `Event`, `Conference`, `BoardMeeting`, `Person`, and `Member`. The code defines the classes and their methods, including cardinalities.

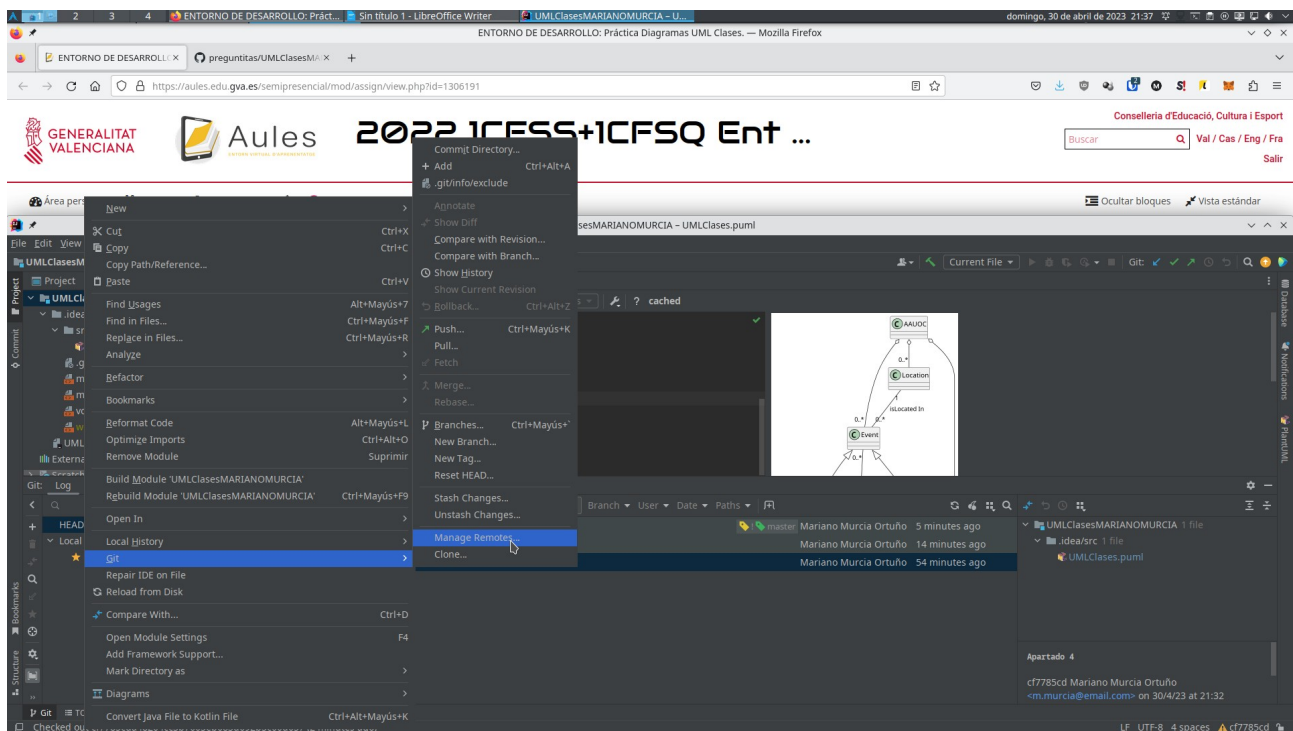
```

1 @startuml
2 Event <|-- Conference
3 Event <|-- BoardMeeting
4
5 Person <|-- Member
6 Member <|-- BoardMember
7
8 AAUOC o-- "0..*" Location
9 AAUOC o-- "0..*" Person
10 AAUOC o-- "0..*" Event
11
12 Location "1" -- "0..*" Event : islocated In
13 Event "0..*" -- "0..*" Member : attendsTo
14 BoardMeeting "0..*" -- "0..*" BoardMember : attendsTo
15 Conference "0..*" -- "0..*" Person : attendsTo
16
17 enduml

```

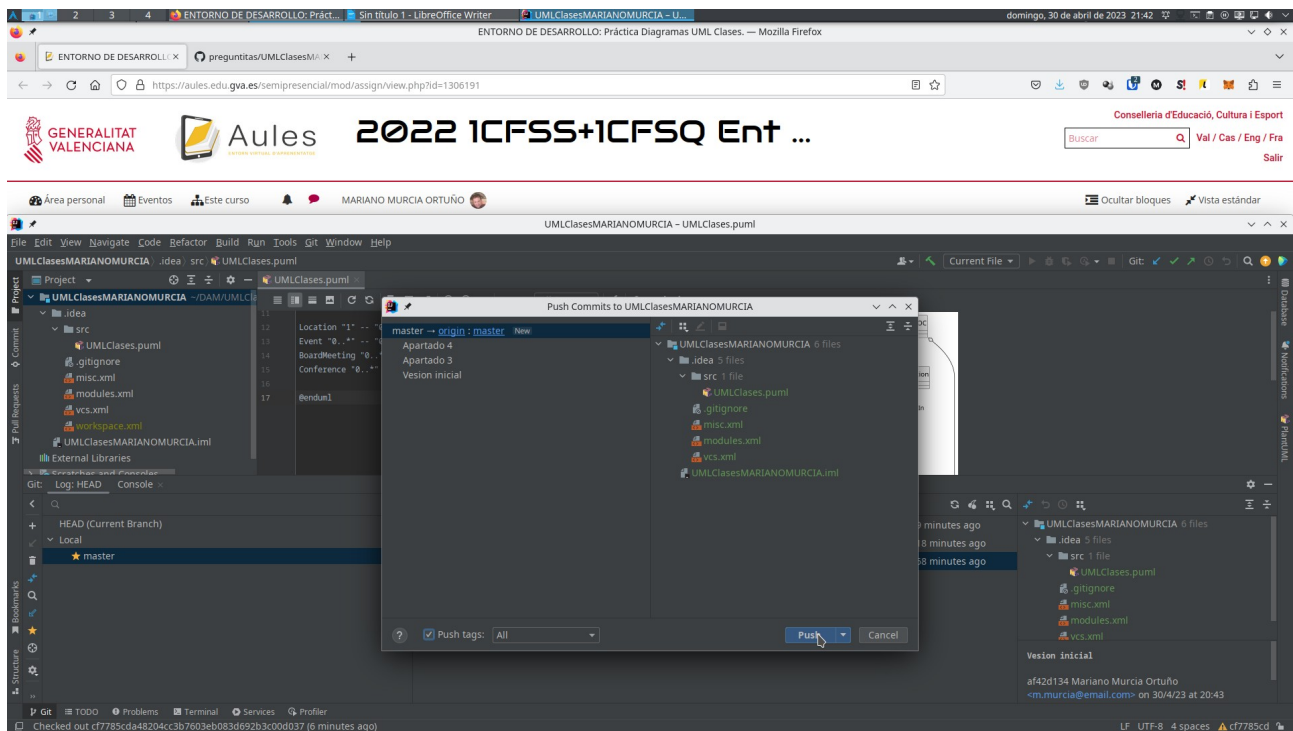
The code defines the classes and their methods, including cardinalities. The UML diagram shows the relationships between the classes with cardinalities.

Eliminamos las declaraciones de las clases y añadimos las cardinalidades a las relaciones. Terminando con un commit con el nombre de "Apartado 4".



5. Repositorio remoto

Añadimos el repositorio remoto de GitHub



Y hacemos un Push al repositorio remoto "origin"