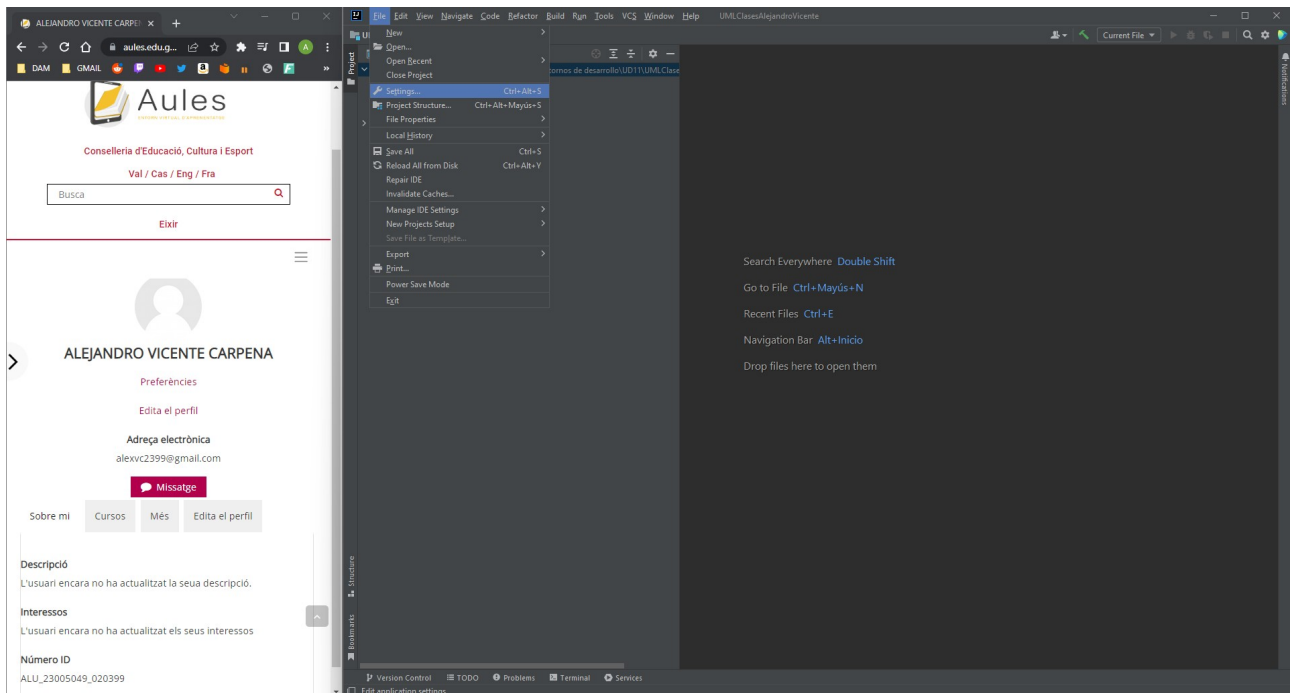


Memoria UML Clases

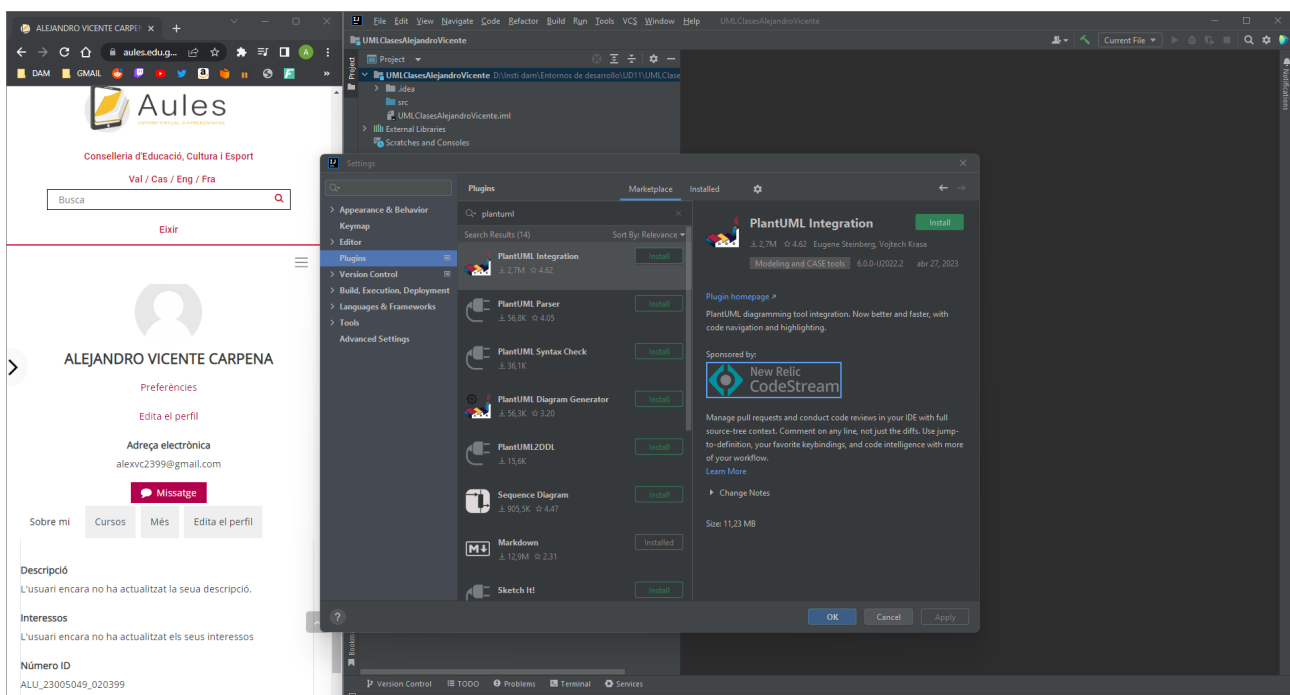
Alejandro Vicente Carpena

ENLACE AL REPOSITORIO: <https://github.com/alexvc991/UMLClasesAlejandroVicente>

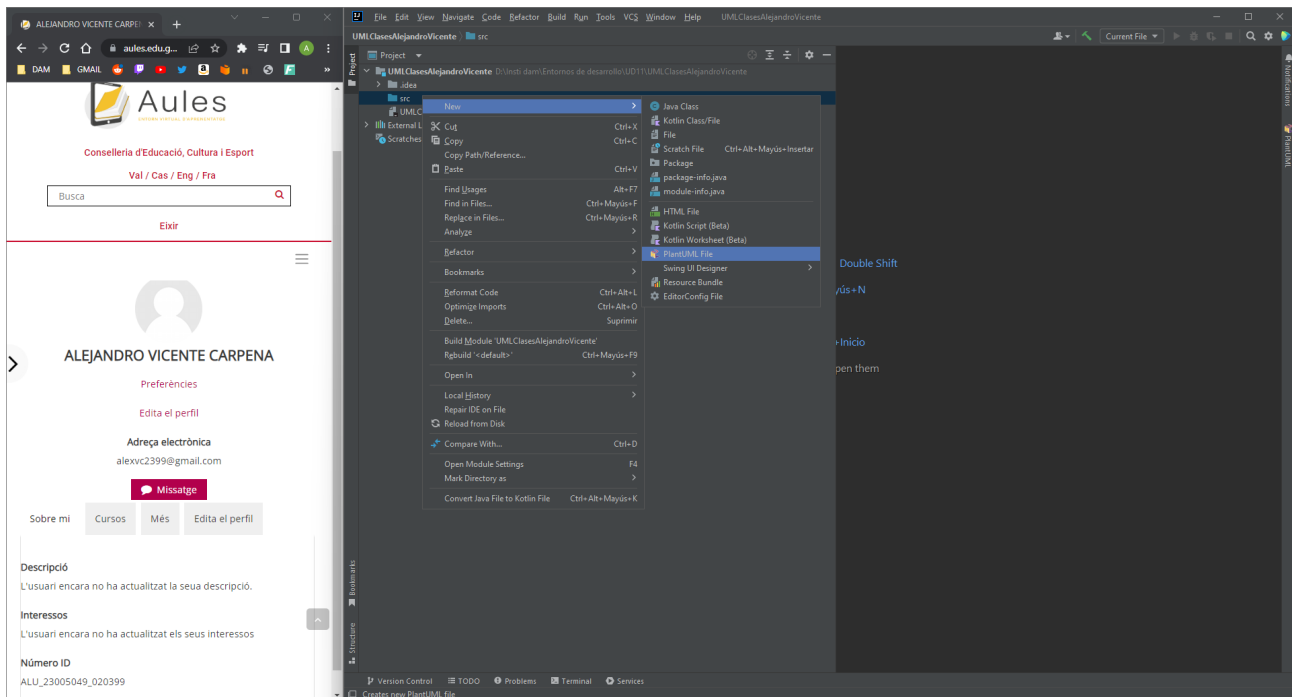
Primero creamos un proyecto JAVA en IntelliJ.
Una vez creado nos vamos a File→ Settings, para instalar el plugin de PlantUML



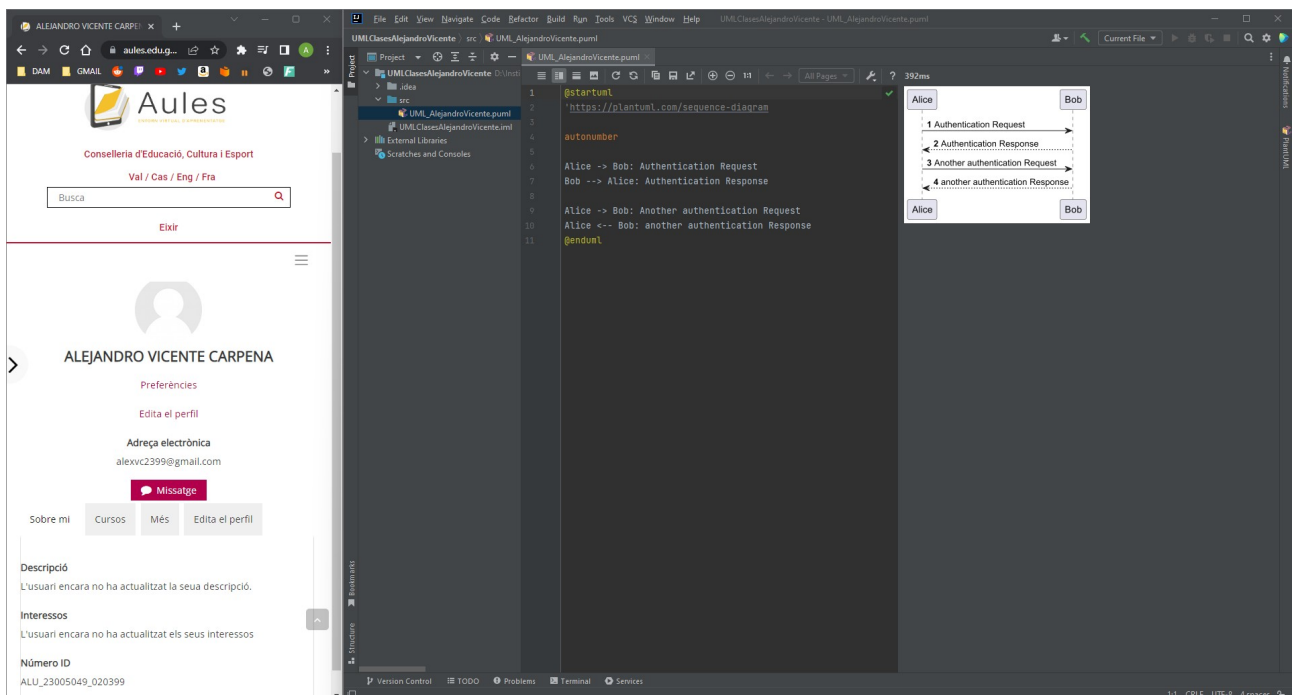
Dentro de settings, nos vamos al apartado plugins, le damos a marketplace, y buscamos plantuml y lo instalamos. Una vez se instale nos pedira que reiniciemos el entorno de desarrollo



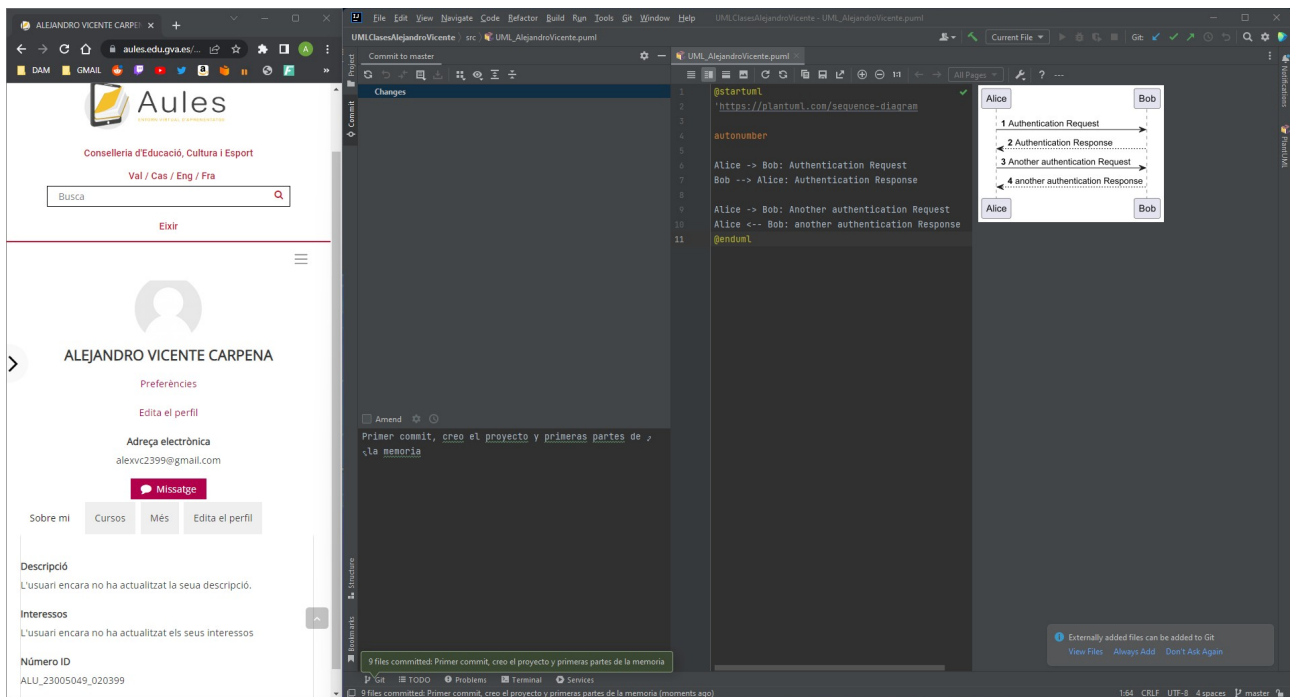
Para comenzar a usar el plugin, debemos crear un archivo PlantUML. Esto podemos hacerlo dándole clic derecho a la carpeta en la que queramos crearlo, que en este caso sera src, le damos a new→ PlantUML File



Se nos creara este archivo, en el cual podremos empezar a crear el archico UML, que tiene una previsualizacion a la derecha



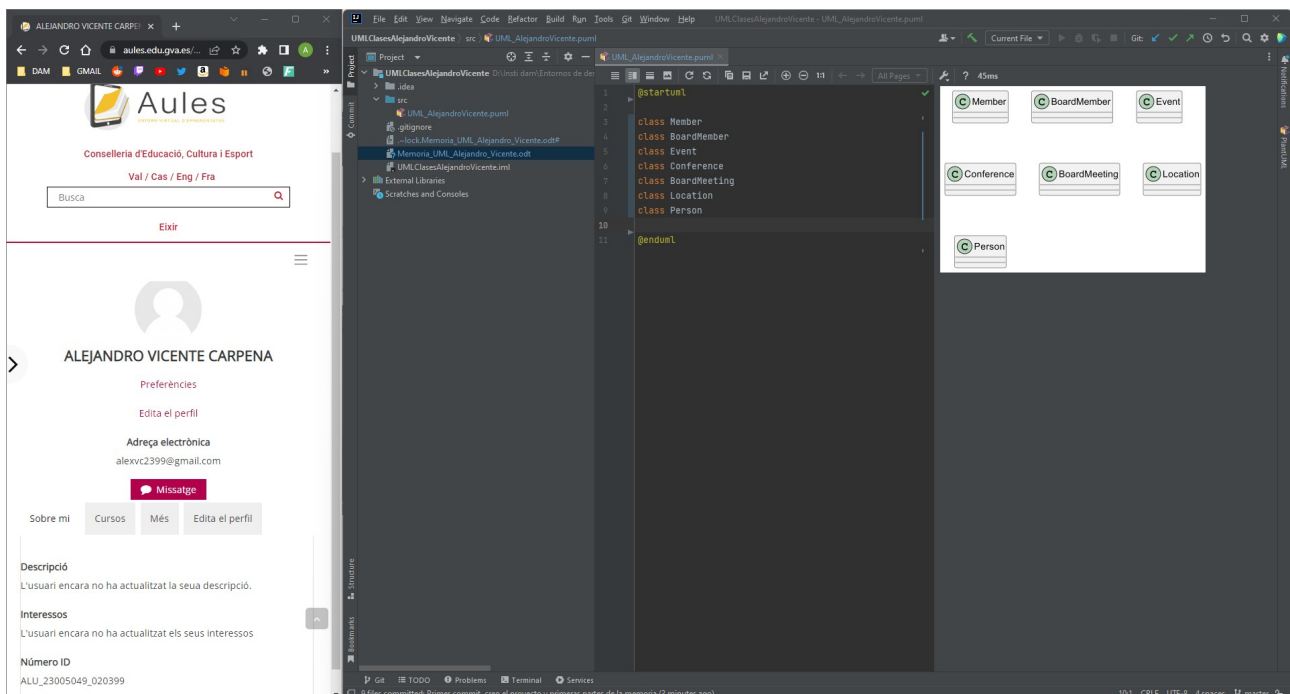
Realizo el primer commit, con el proyecto, el archivo de plantuml y la memoria creados.



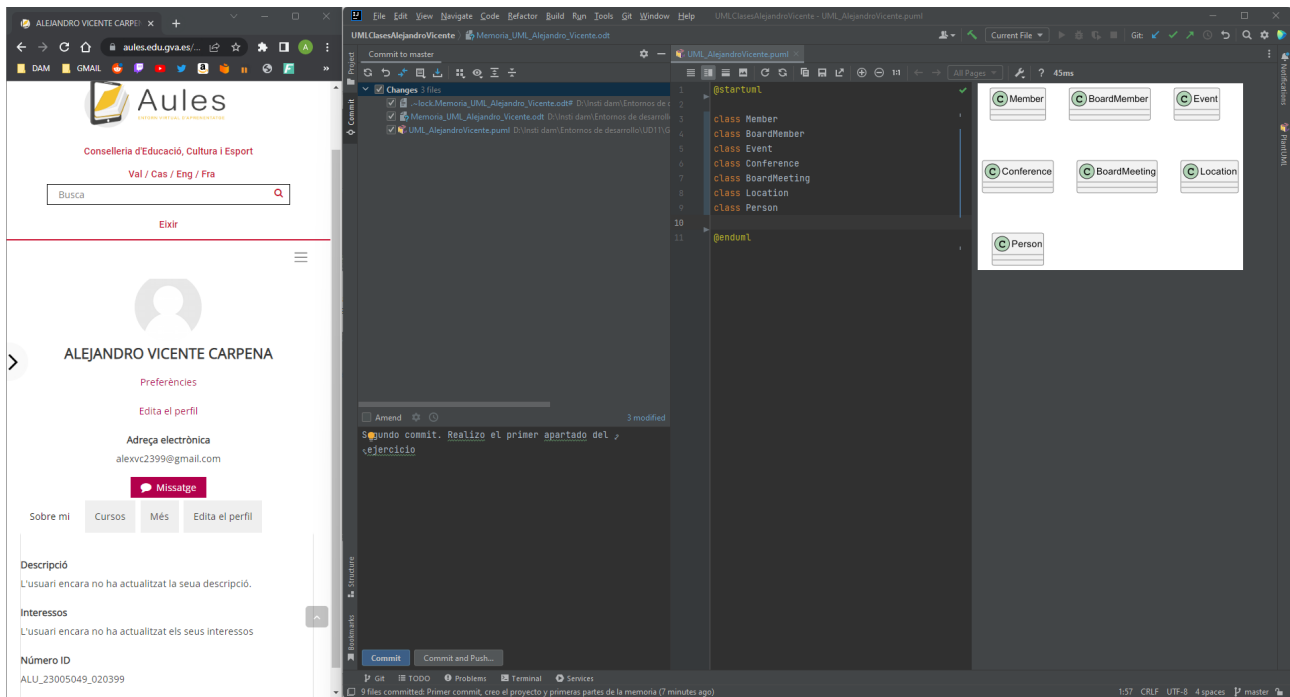
Primer punto del ejercicio realizado.

Este punto consistia en identificar las clases del problema planteado.

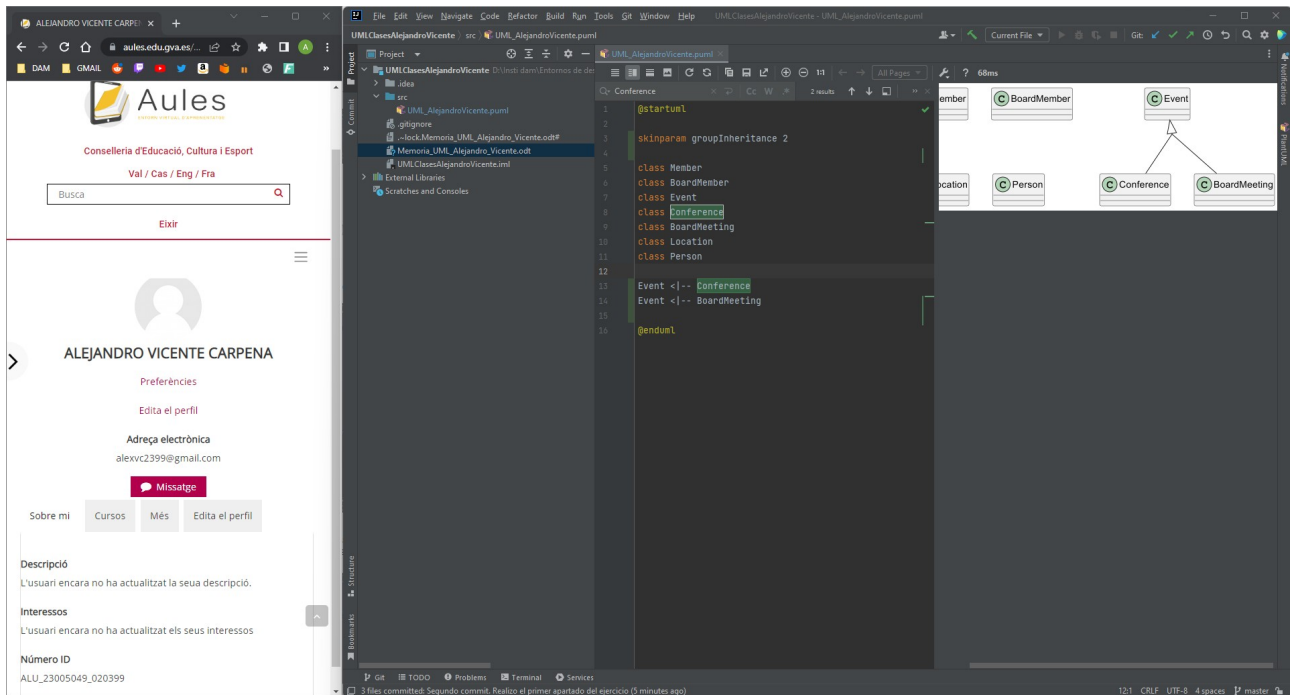
Las clases se crean poniendo "class 'nombreclase' "



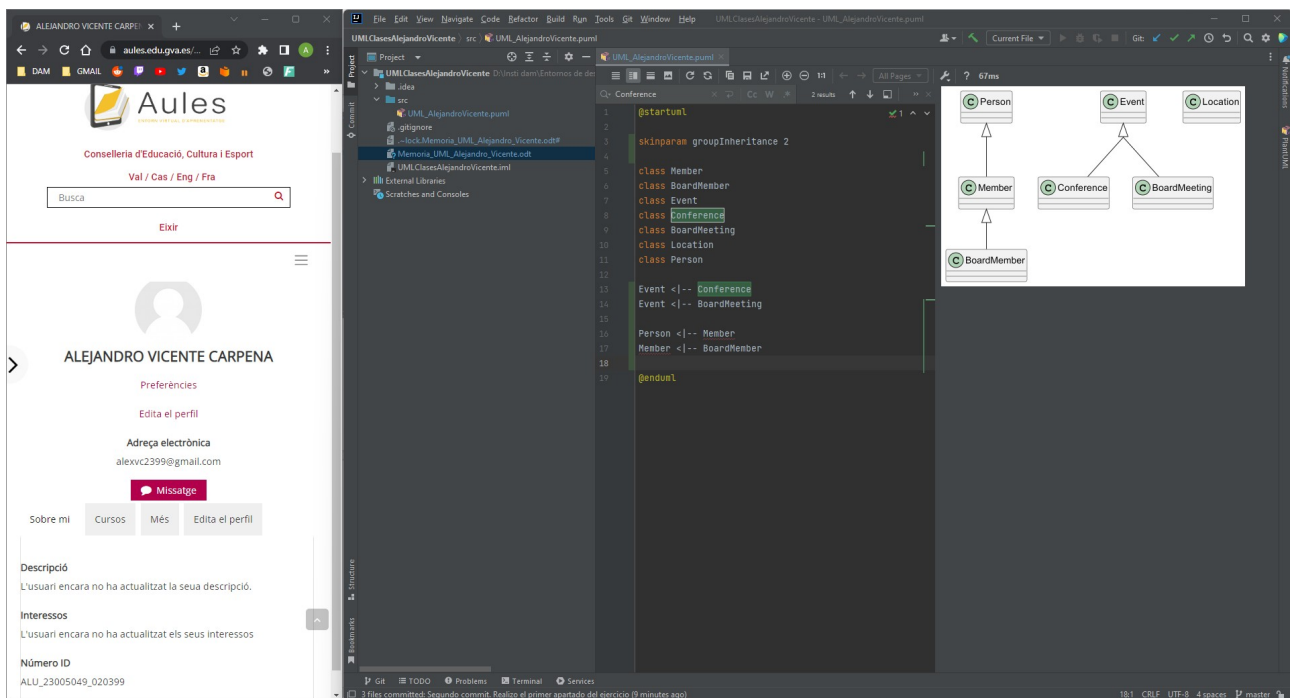
Realizo el segundo commit.



Para el primer punto del segundo apartado, utilizo “class1 <|-- class2” para mostrar las relaciones de herencia. También uso “skinparam groupInheritance 2” para agrupar las flechas, en este caso de Conference y BoardMeeting

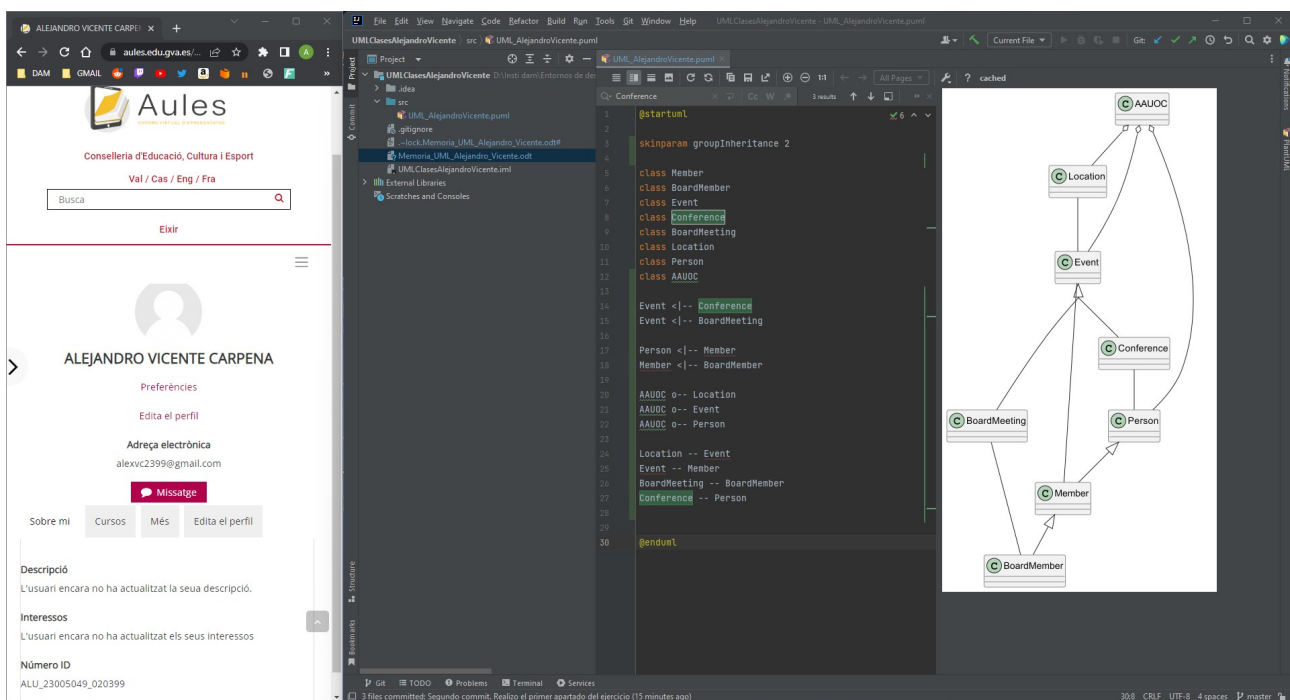


El siguiente punto para representar la jerarquia entre person member y boardmember

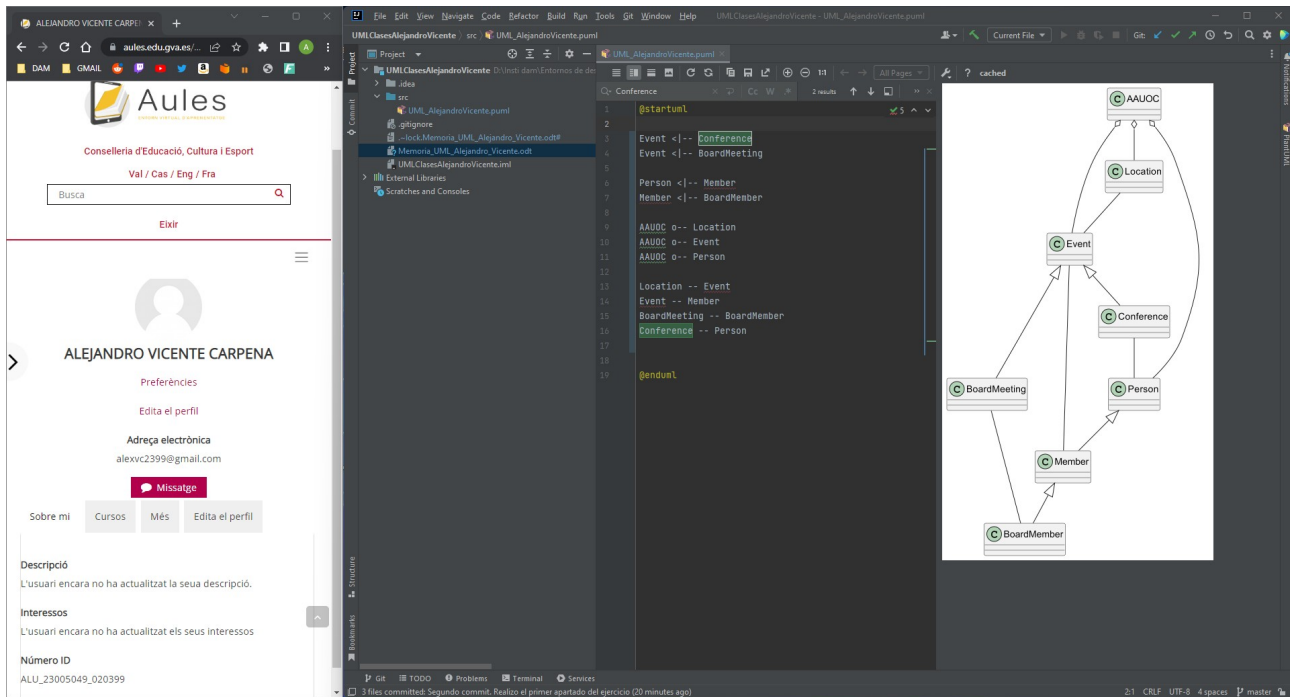


El apartado final con todas las relaciones.

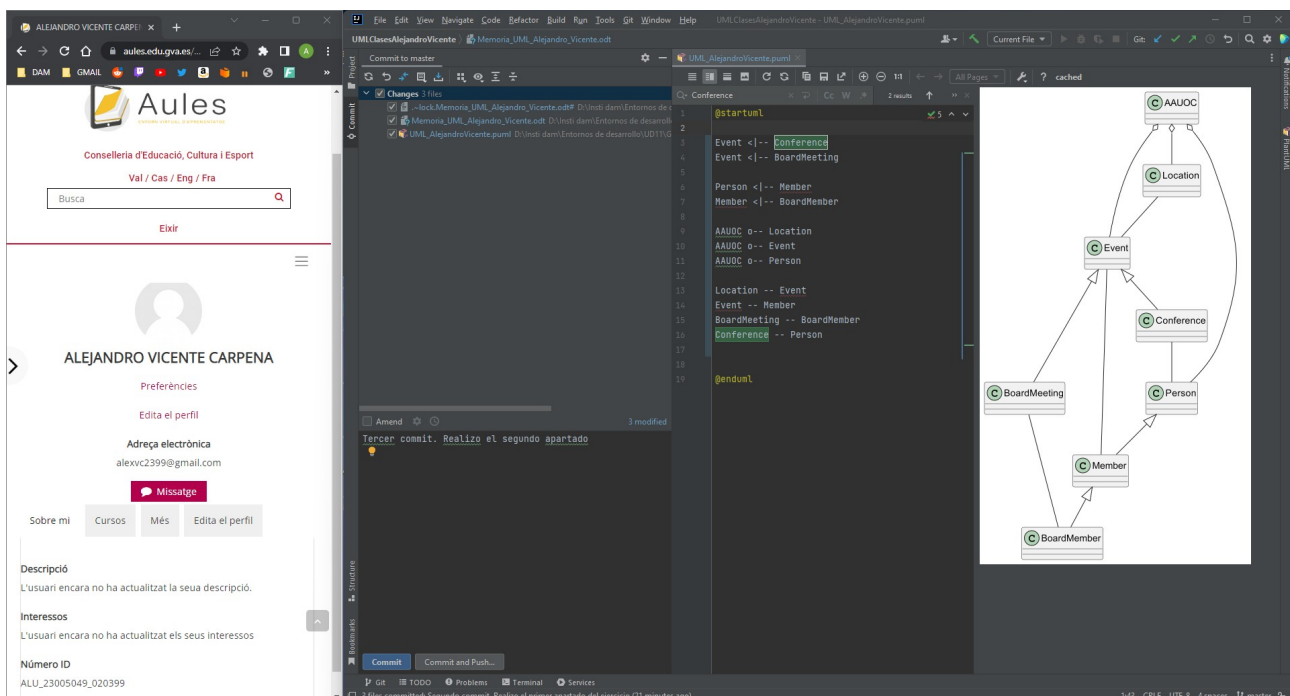
Como las líneas que convergen en Event se cruzan, quito la línea previa de skinparam groupInheritance 2. Además, ya no es necesario tener las declaraciones de las clases, ya que al hacer las relaciones es como si creásemos las clases



Asi quedaria sin la linea de skinparam y sin las declaraciones de las clases

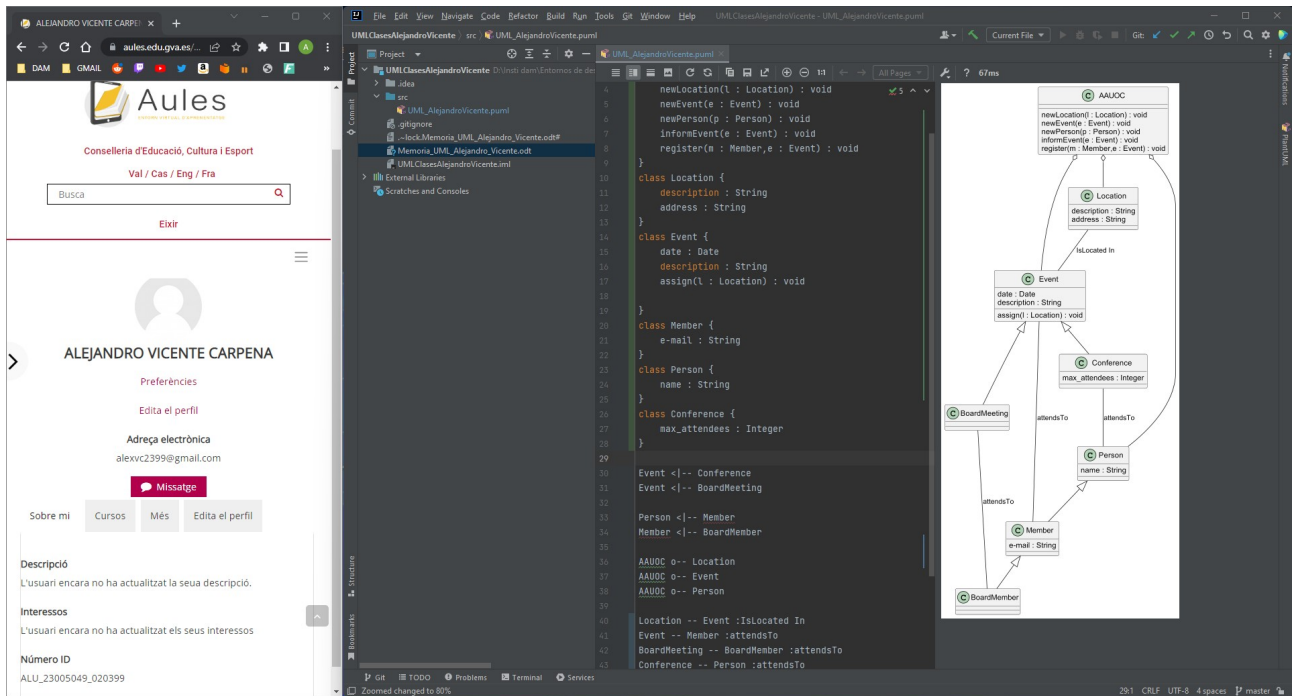


Realizo el tercer commit.

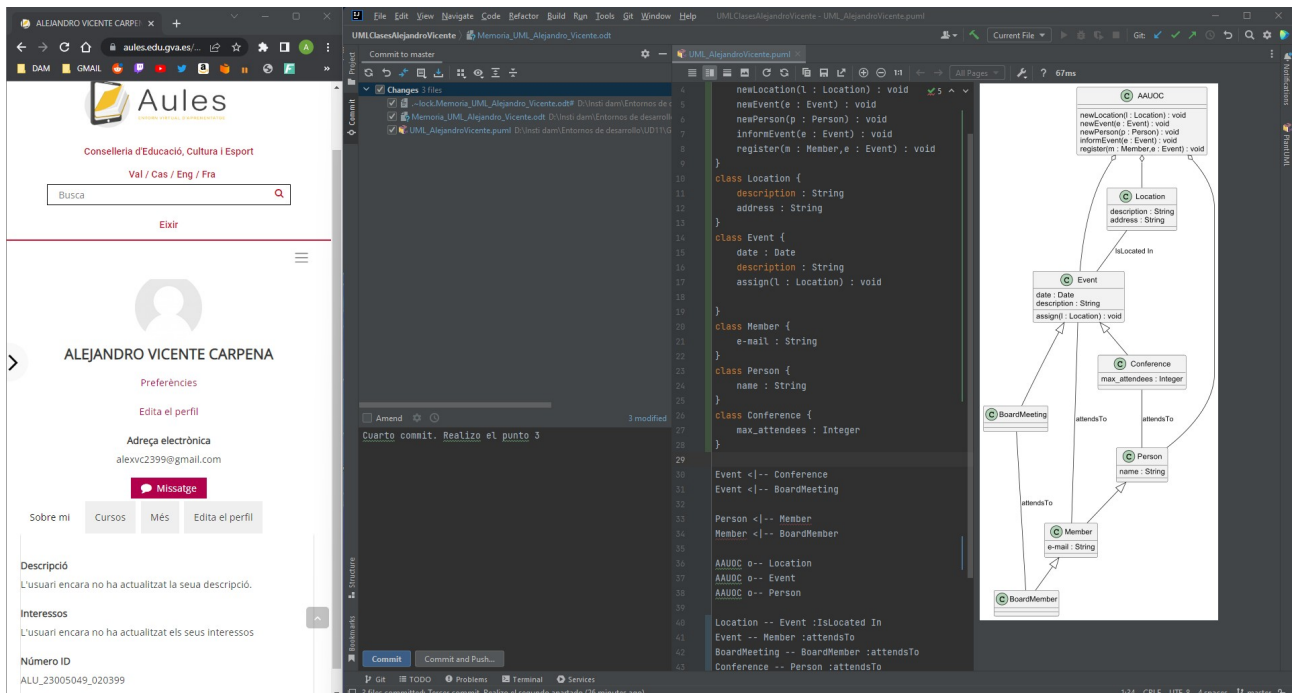


Para el punto 3 del ejercicio, vuelvo a crear las declaraciones de las clases para poder poner los metodos y los atributos. Esto se hace abriendo las llaves { } y dentro escribiendo los diferentes metodos y/o atributos. El plugin detecta si es un metodo o un atributo automaticamente, detectando si lleva parentesis o no. Para las notas de los enlaces se realiza poniendo : 'notadelenlace' al lado del enlace que queramos anotar.

Asi quedaria el punto 3



Realizo el cuarto commit



Para el cuarto punto, eliminamos los metodos y los atributos de las clases, para añadir la cardinalidad entre ellas

The screenshot shows a web browser on the left displaying the 'Aules' website profile for Alejandro Vicente Carpena. On the right, an IDE displays UML class diagrams and code. The code defines classes: `AAUOC`, `Location`, `Event`, `Conference`, `Person`, `Member`, `BoardMeeting`, and `BoardMember`. The UML diagram shows relationships: `AAUOC` has a `Location` (0..1 to 0..1, labeled 'isLocated In'), `Event` has a `Location` (0..1 to 0..1, labeled 'isLocated In'), `Event` has a `Conference` (0..1 to 0..1, labeled 'attendsTo'), `Event` has a `Person` (0..1 to 0..1, labeled 'attendsTo'), `Event` has a `Member` (0..1 to 0..1, labeled 'attendsTo'), `Event` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo'), `Conference` has a `Person` (0..1 to 0..1, labeled 'attendsTo'), `Conference` has a `Member` (0..1 to 0..1, labeled 'attendsTo'), `Conference` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo'), `Person` has a `Member` (0..1 to 0..1, labeled 'attendsTo'), `Person` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo'), `Member` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo').

Asi quedaria, sin eliminar los metodos y los atributos y añadiendo la cardinalidad

The screenshot shows the same web browser and IDE as before. The code defines classes: `AAUOC`, `Location`, `Event`, `Conference`, `Person`, `Member`, `BoardMeeting`, and `BoardMember`. The UML diagram shows relationships: `AAUOC` has a `Location` (0..1 to 0..1, labeled 'isLocated In'), `Event` has a `Location` (0..1 to 0..1, labeled 'isLocated In'), `Event` has a `Conference` (0..1 to 0..1, labeled 'attendsTo'), `Event` has a `Person` (0..1 to 0..1, labeled 'attendsTo'), `Event` has a `Member` (0..1 to 0..1, labeled 'attendsTo'), `Event` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo'), `Conference` has a `Person` (0..1 to 0..1, labeled 'attendsTo'), `Conference` has a `Member` (0..1 to 0..1, labeled 'attendsTo'), `Conference` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo'), `Person` has a `Member` (0..1 to 0..1, labeled 'attendsTo'), `Person` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo'), `Member` has a `BoardMember` (0..1 to 0..1, labeled 'attendsTo').