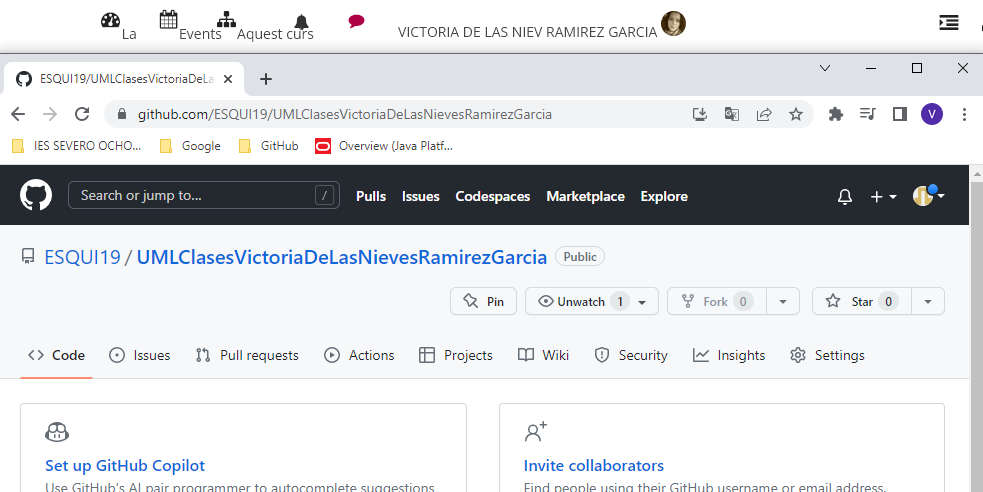
**ENLACE GITHUB**

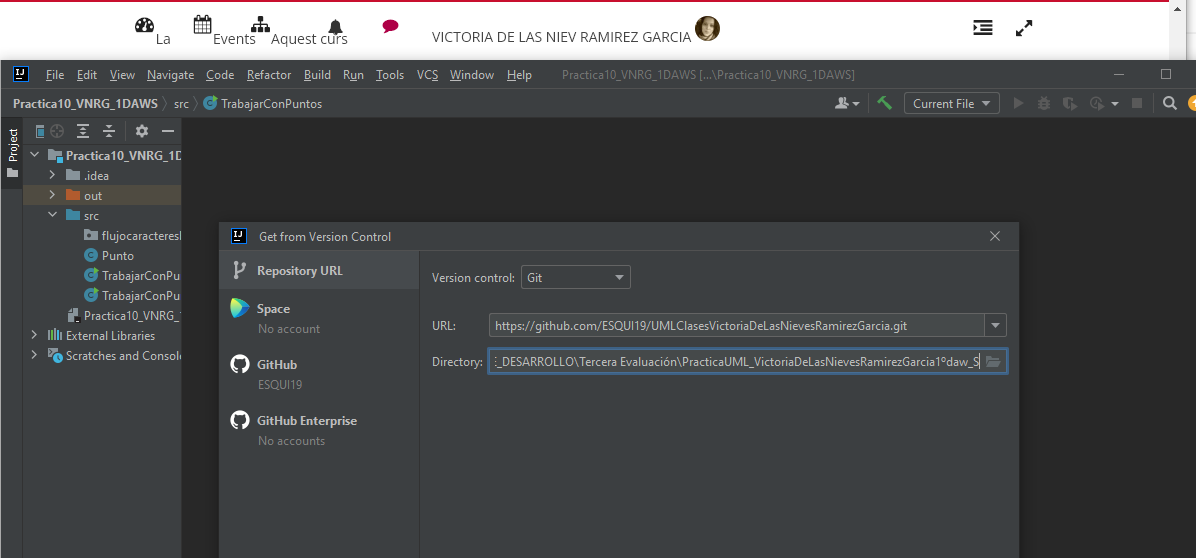
<https://github.com/ESQUI19/UMLClasesVictoriaRamirezGarcia.git>

**PASOS PREVIOS A LA PRÁCTICA**

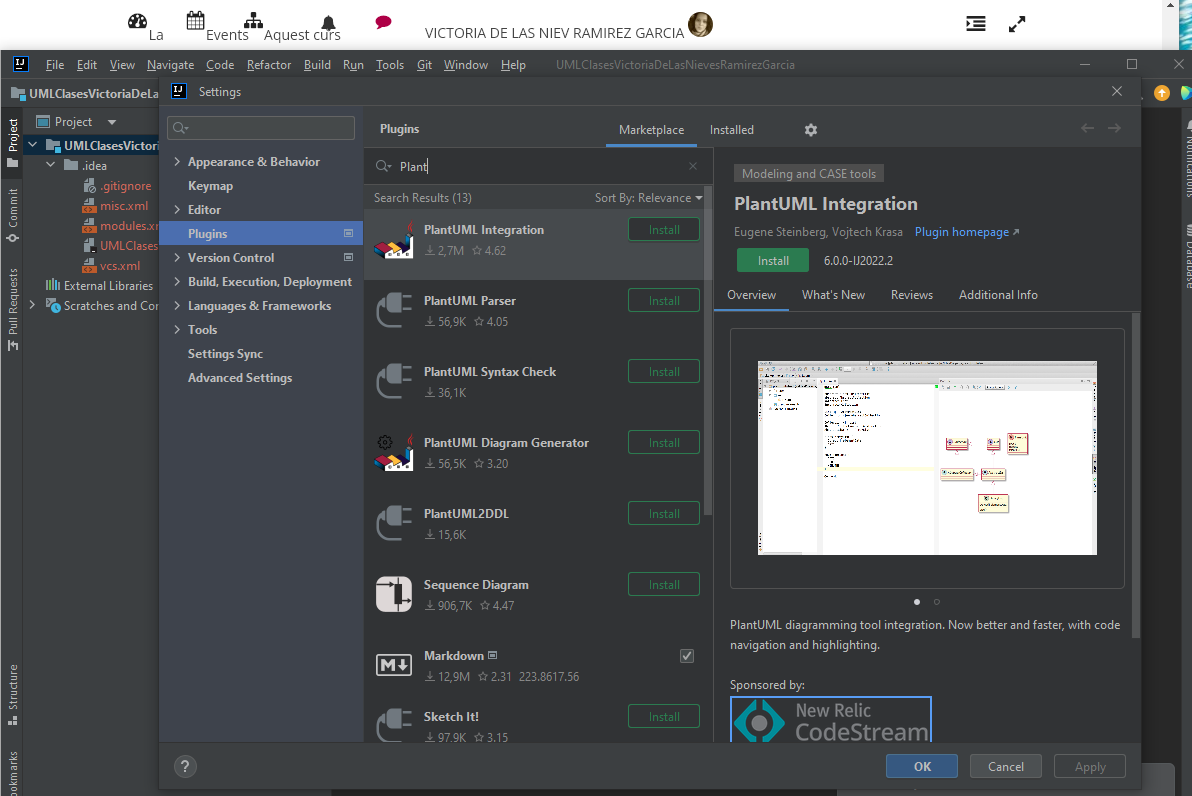
Para empezar, he creado un repositorio en mi cuenta de GitHub, para importarlo a IntelliJ e ir haciendo commits sobre la marcha.



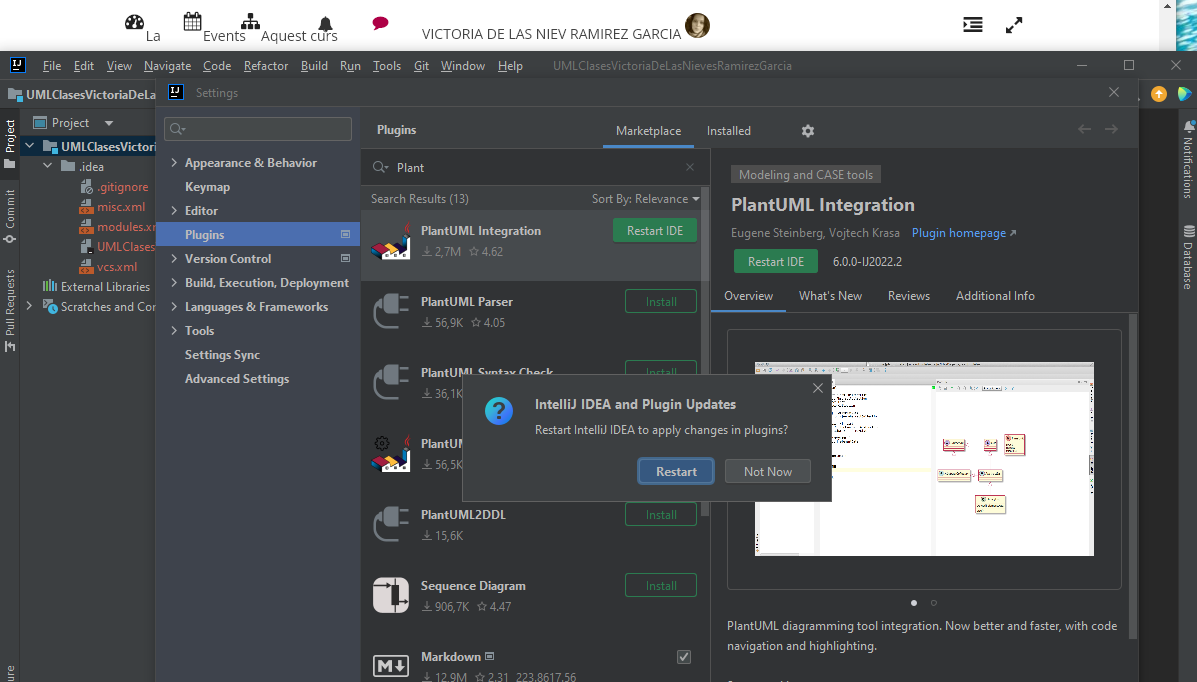
Importamos el proyecto al IDE.



Instalo el plugin PlantUml. En IntelliJ 🡪 File 🡪Settings 🡪 Plugins 🡪 Marketplace 🡪 PlantUml 🡪 Install



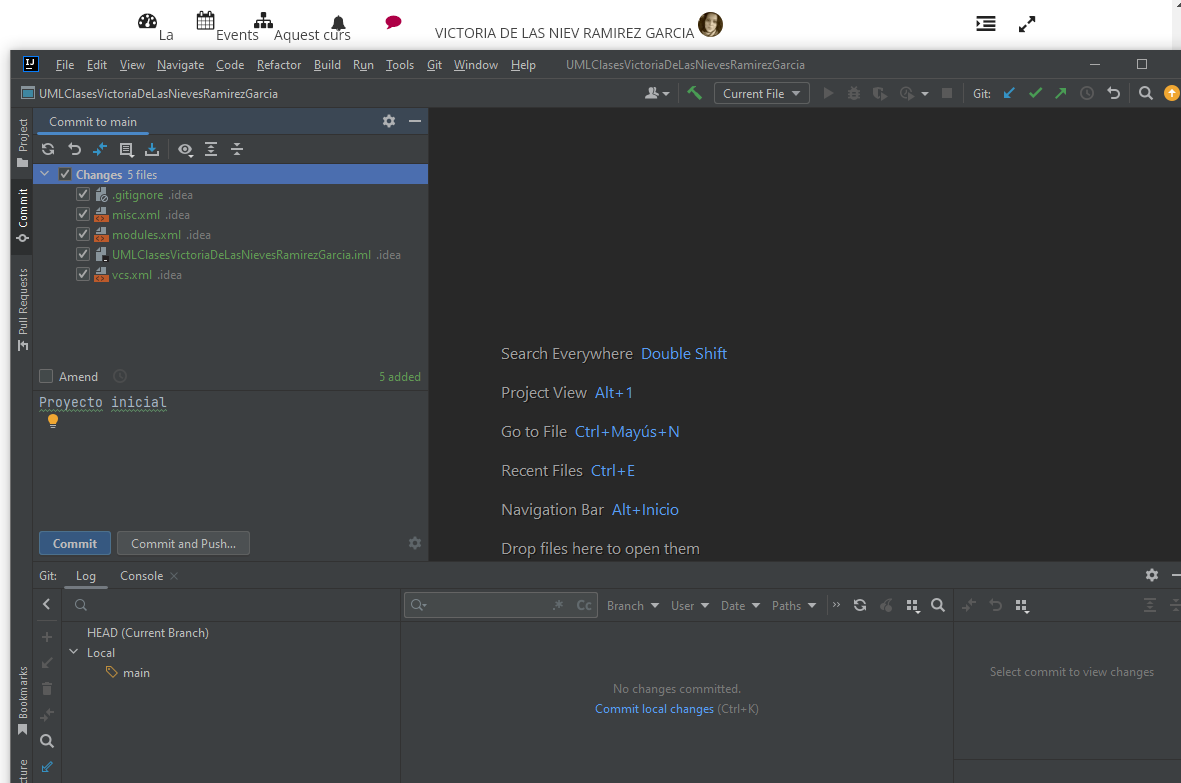
Una vez instalado el Plugin, me pide reiniciar el IDE. Lo reinicio.



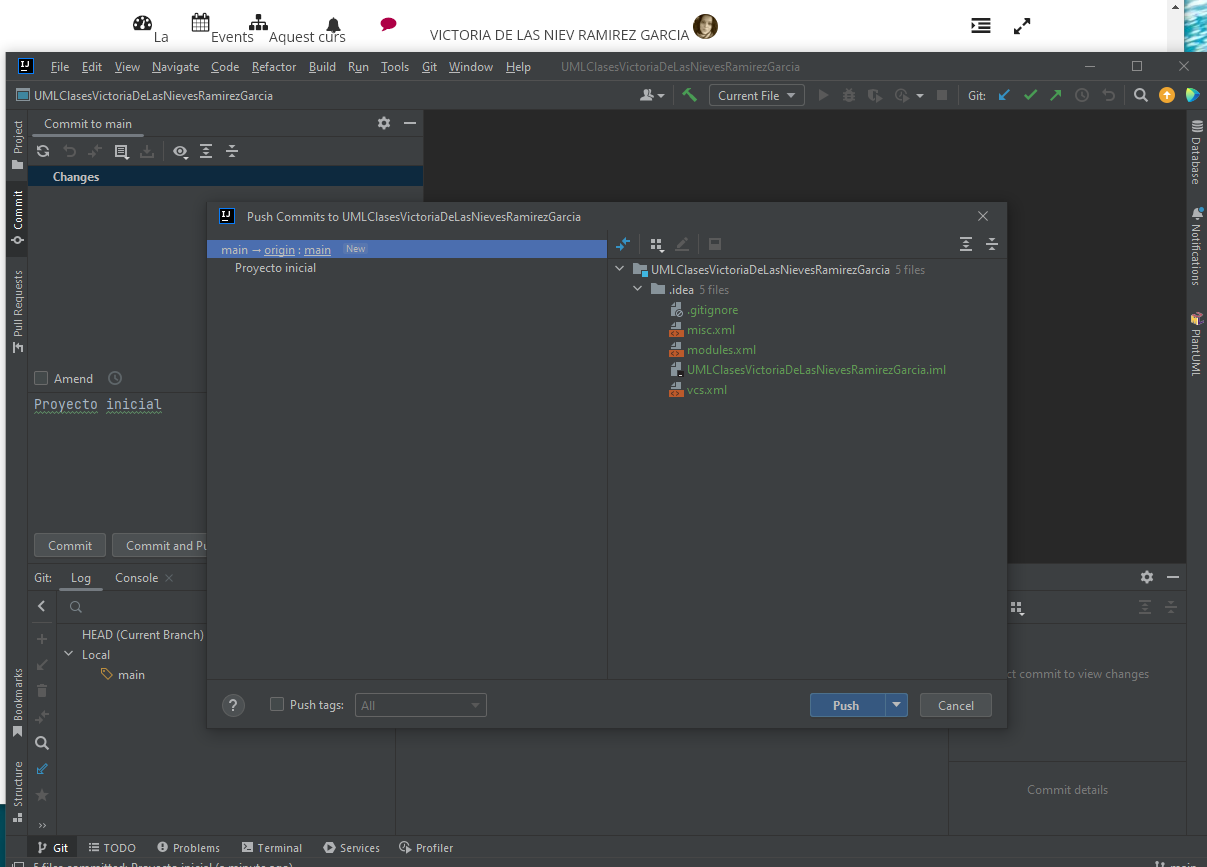
Los archivos del proyecto aparecen en rojo porque no se han añadido al staging area. Los añado y hago commit con el proyecto inicial.

Selecciono todos los archivos y Git 🡪 Selected Files 🡪 Add (Los archivos quedan en color verde).

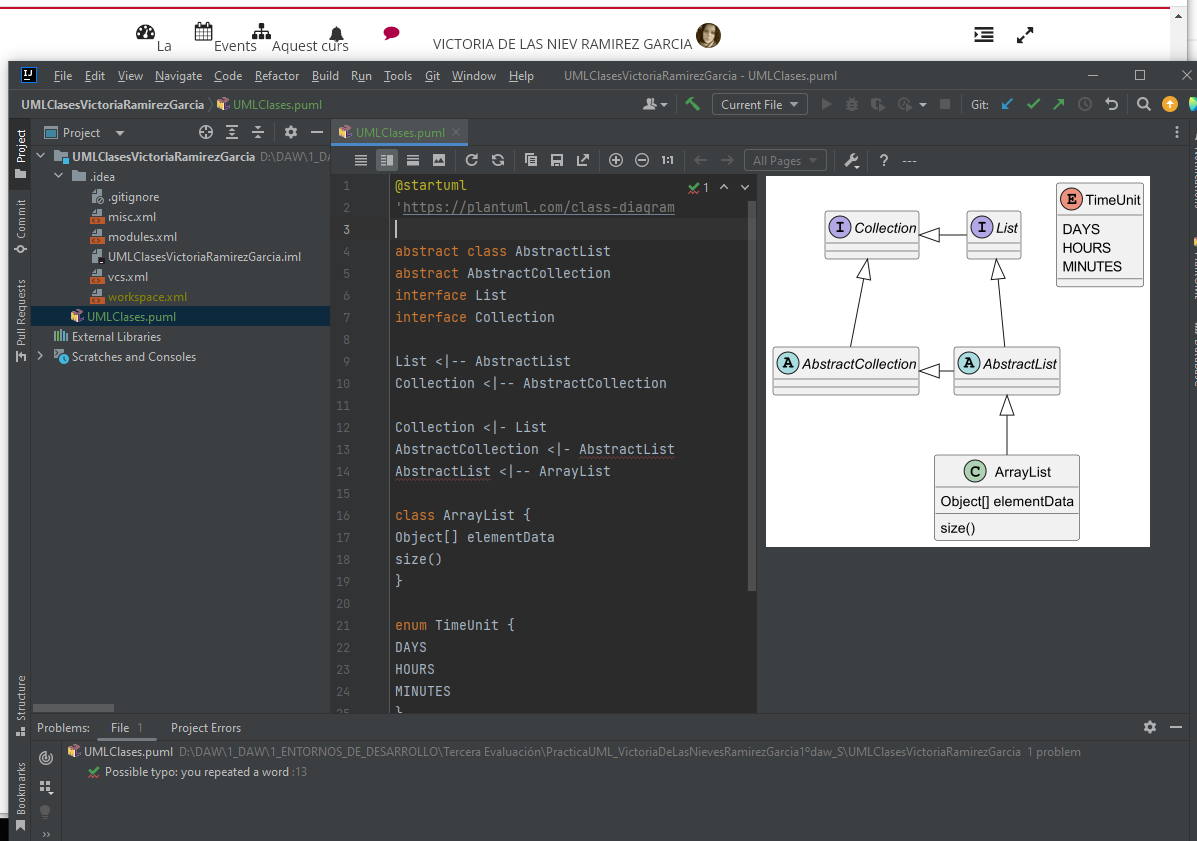
Selecciono todos los archivos de nuevo, ponemos el mensaje y hacemos commit.



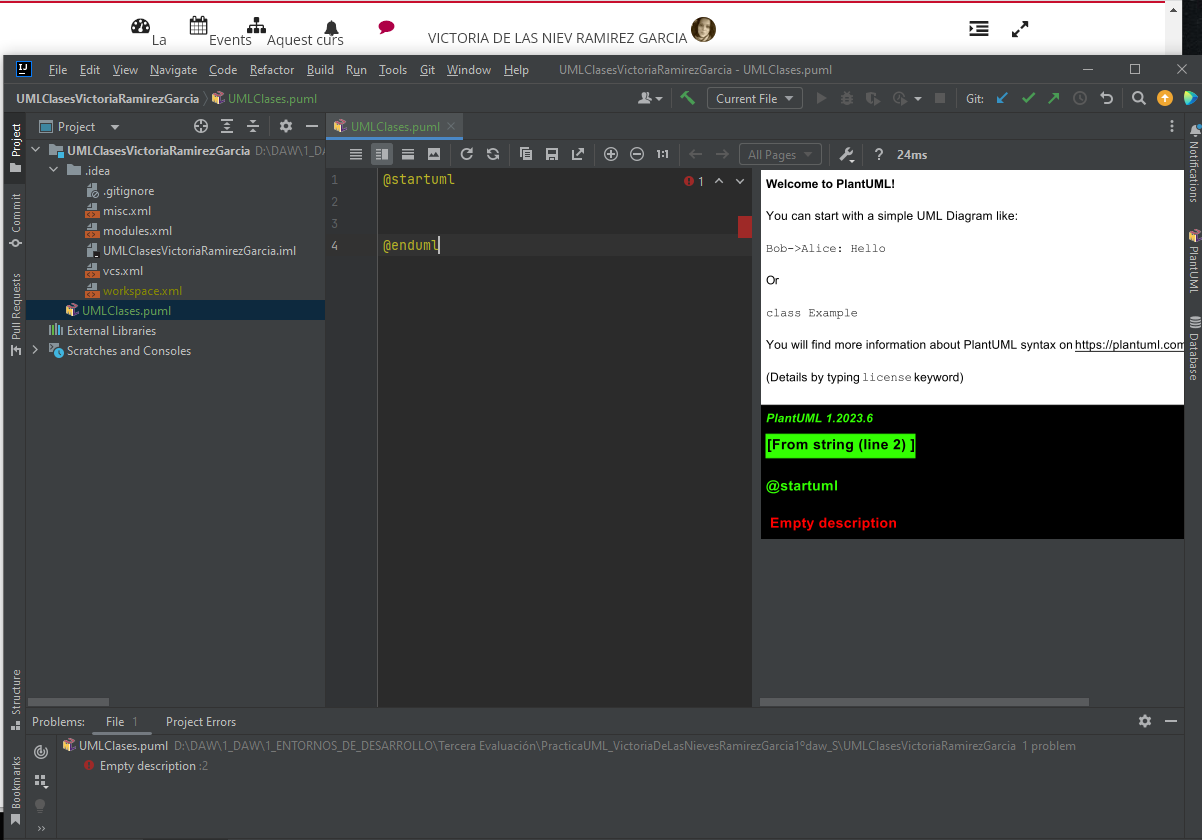
Hacemos push del repositorio para mandar todos los cambios al repositorio en GitHub.



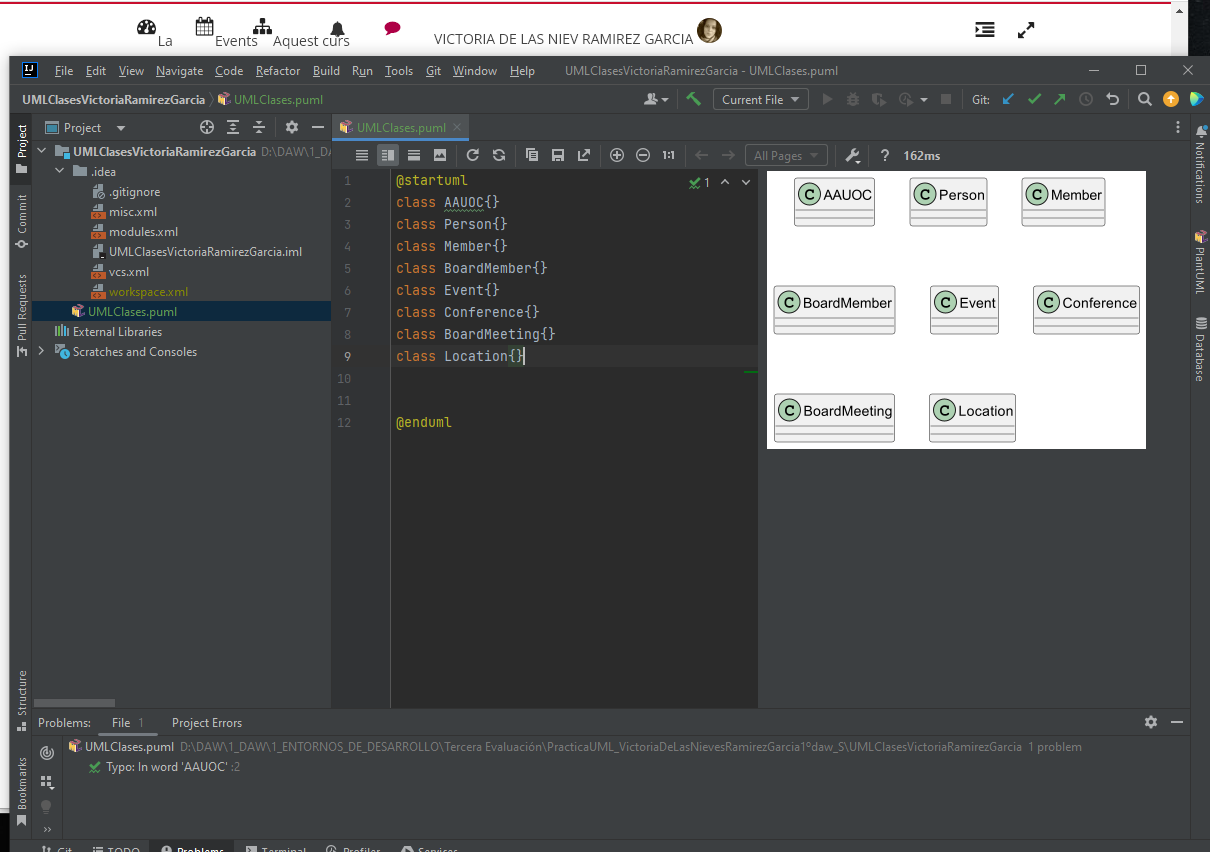
Para crear un archivo PlantUML, File 🡪 New File 🡪 PlantUML File. Seleccionamos “Class” y damos nombre a la clase. En este caso se ha creado una clase UMLClases para definir el modelo. El IDE nos pregunta si queremos añadir este archivo a Git. Aceptamos.



Borramos todo el código exceptuando las anotaciones @staruml y @enduml, que marcan el inicio y el fin del archivo.



Creamos las clases de la siguiente manera:



Además de las clases Member (Miembro o miebro numerario), BoardMember (Miembro de la junta directiva), Event (Evento), Conference (conferencia), BoardMeeting (Reunión de la junta directiva) y Location (localización), se han creados dos clases más.

Clase Persona: es la superclase de Member y BoardMember, es decir de las personas de la asociación.

Clase AAUOC: superclase de todas las del modelo, ya que, de ella heredan directamente Persona, Event y Location (y a su vez, las hijas de estas).

La jerarquía de herencia es la siguiente:

De la clase AAUOC heredan las clases Person, Event y Location.

De la clase Persona heredan Member y BoardMember.

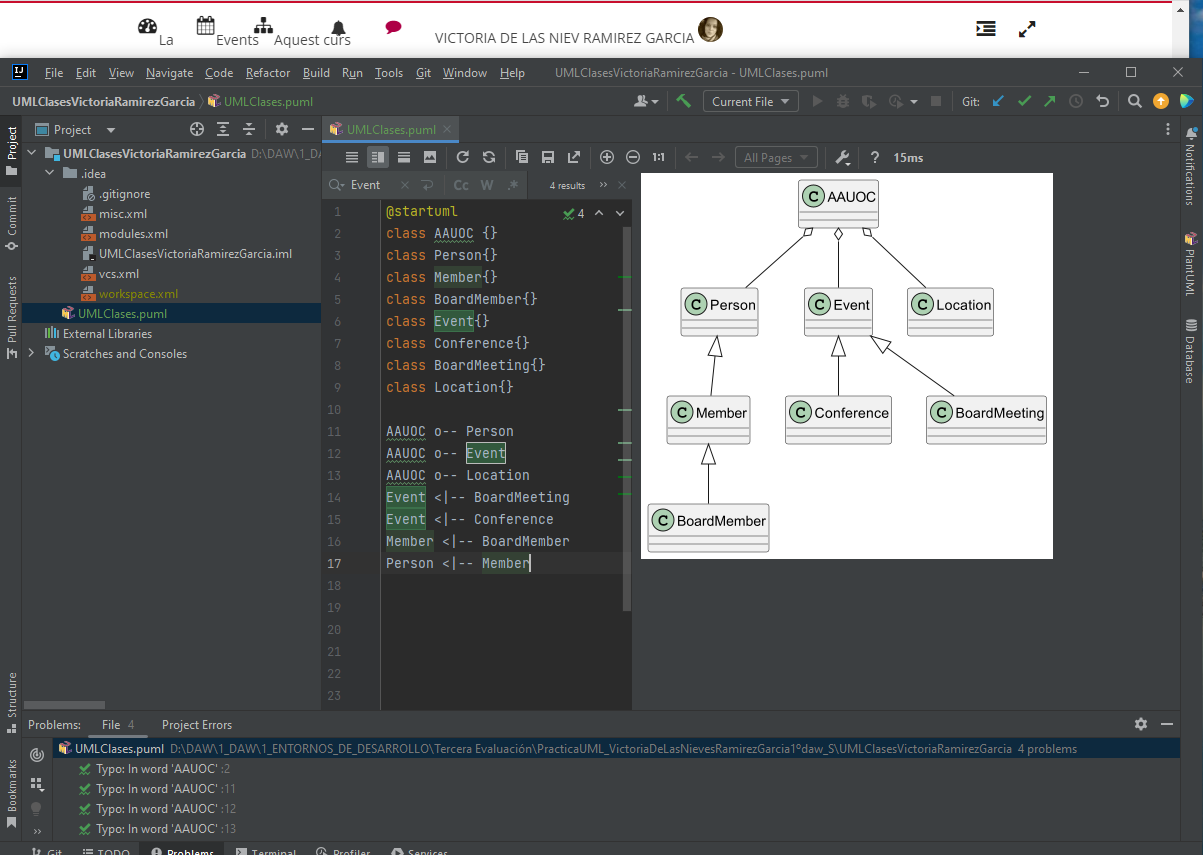
De la clase Event hereda BoardMeeting y Conference.

Por otra parte, Event se relaciona con Member y Location, y BoardMeeting se relaciona con BoardMember.

También Conference se relaciona con Person.

En el caso de la superclase AAUOC y las hijas directas Location, Event y Person, tenemos una relación de extensión, que se representa con “<|- -“.

En el caso del resto de herencia, representamos relaciones de agregación, con “o - -“.

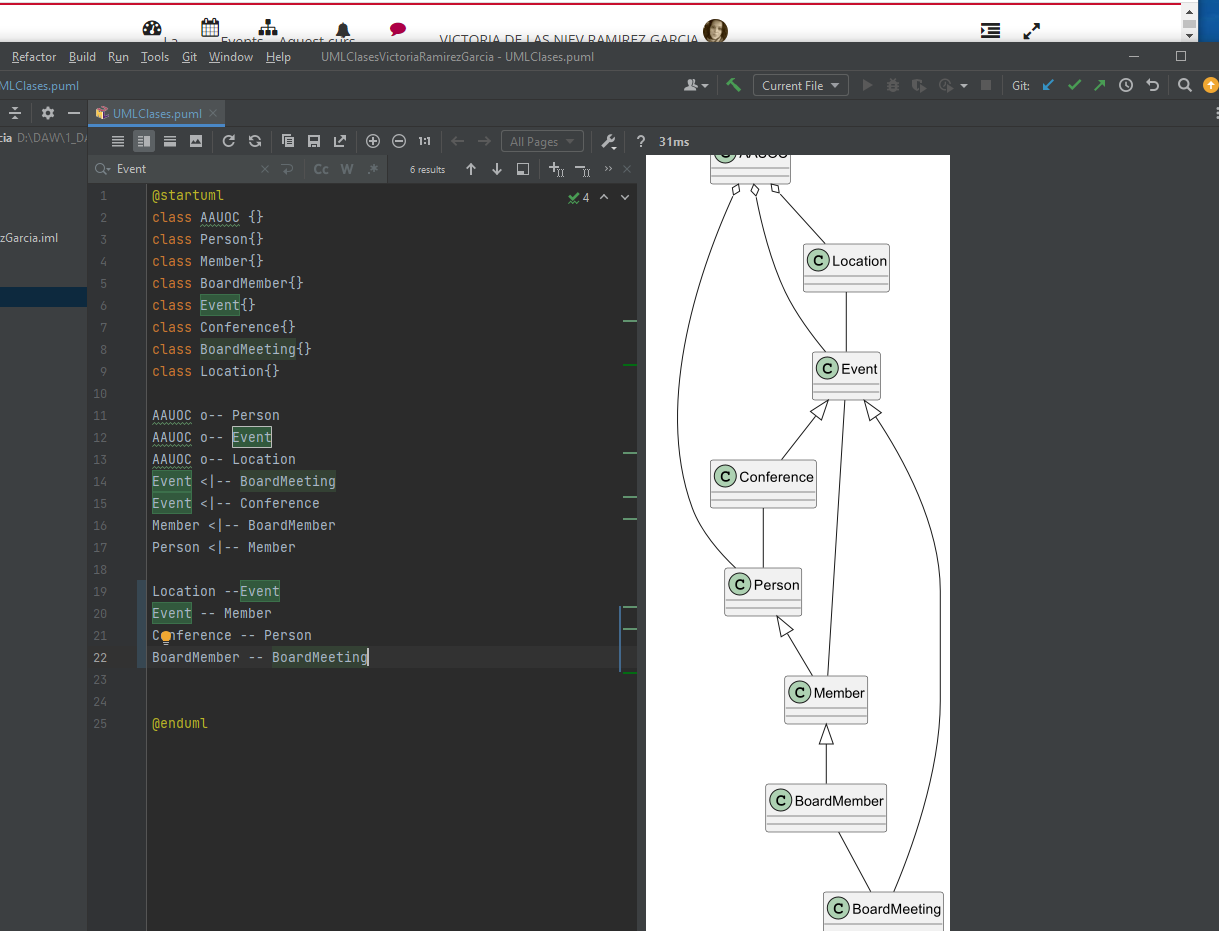


Hacemos commit and” push”, y actualizamos el repositorio remoto.

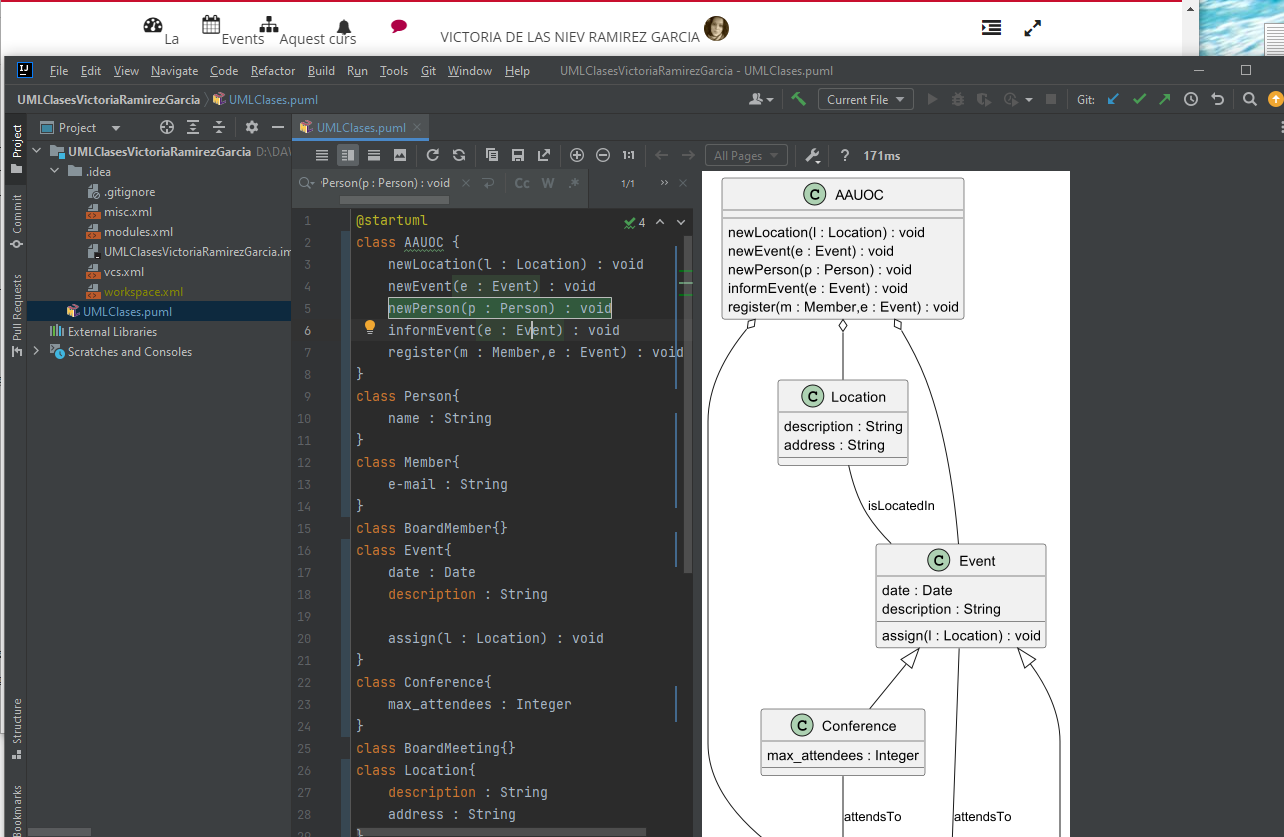
Añadimos una nueva rama “master” y hacemos checkout (¡¡Deberíamos haberla creado al principio porque por defecto en GitHub crea la rama main, pero rectificar es de sabios!!).

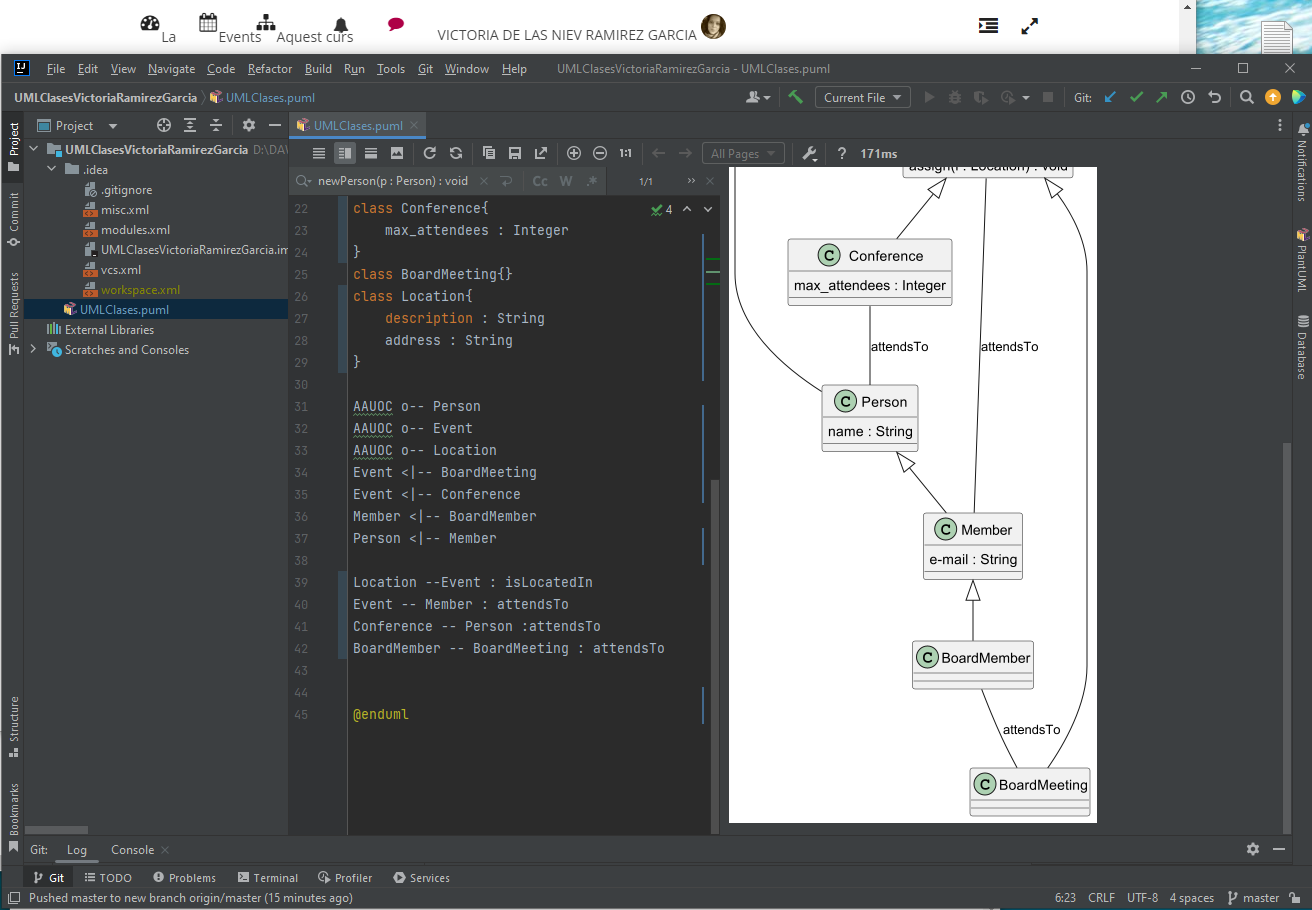
Seguimos con las relaciones binarias entre Event y Member, BoardMeeting y BoardMember, Conference y Person, Location y Event, que se representan con “- - “.

Hacemos commit and push para guardar cambios en remoto, con el mensaje “terminando relaciones”.

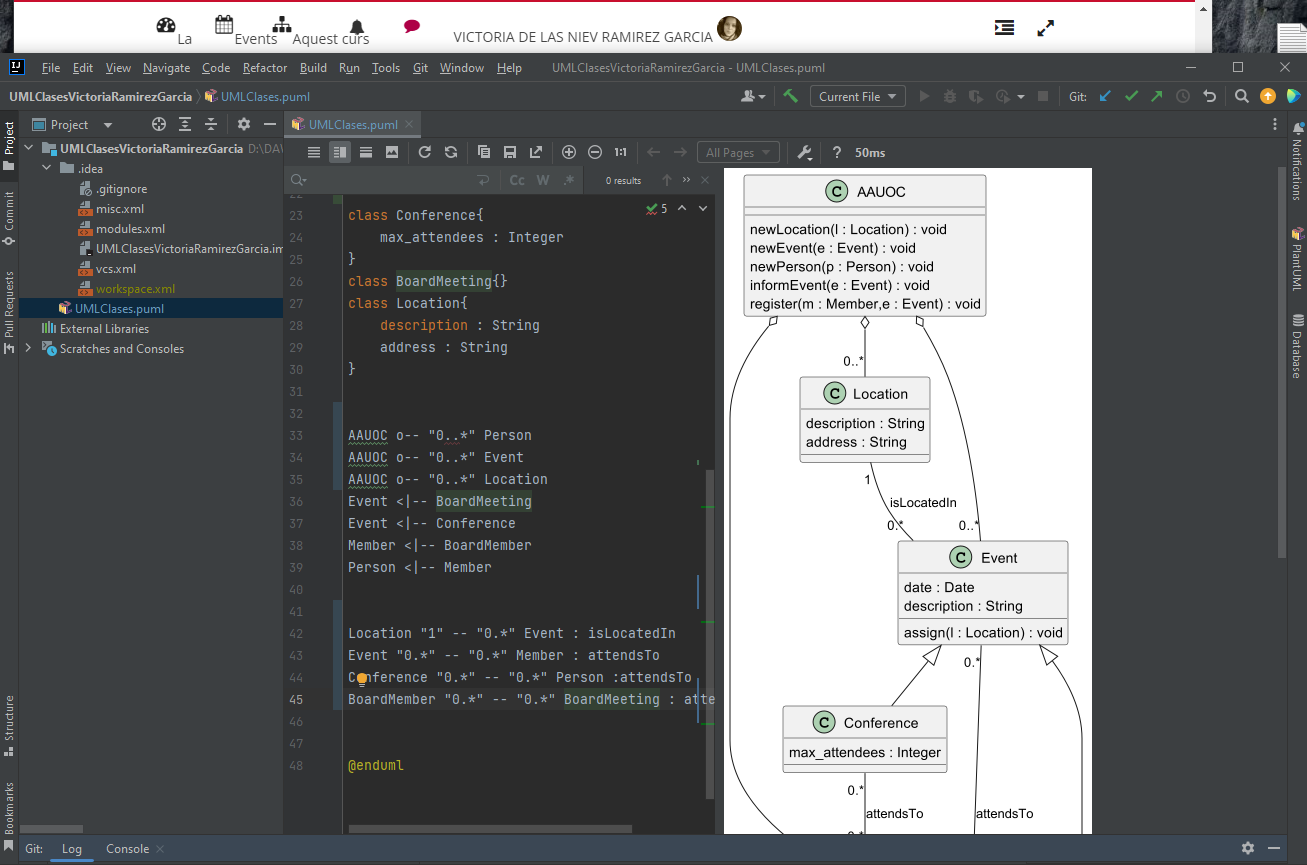


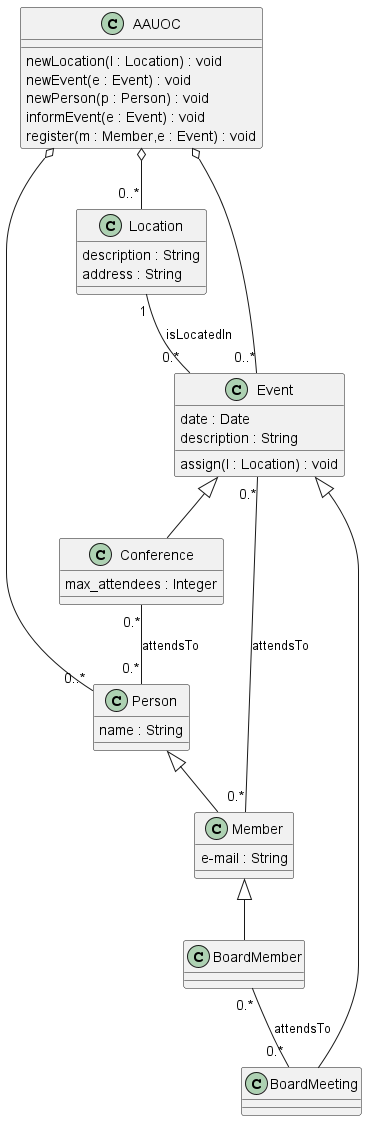
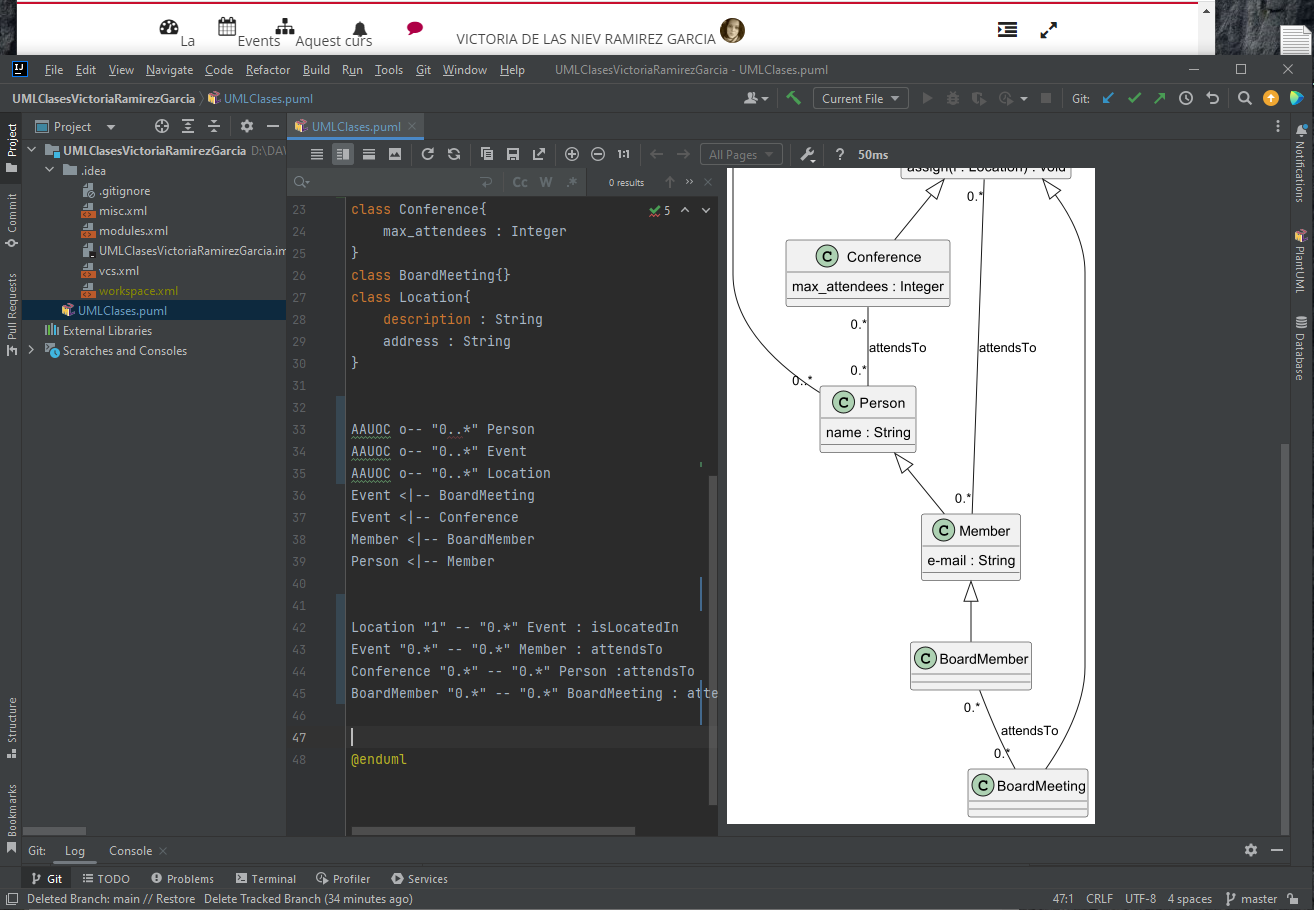
Añadimos las variables y los métodos de las clases, y hacemos un commit and push.





Ahora añadimos las cardinalidades, todas bidireccionales ya que no se ha establecido ningún tipo de relación de una clase a otra:





Para finalizar realizamos el último commit and push para actualizar el repositorio remoto, con el mensaje “Proyecto finalizado”.

