
Diagramas UML Clases

Índice

- 1. Creación del repositorio en GitHub.....3
- 2. Instalación del plugin.....4
- 4. Creación del modelo de datos.....6

La URL del repositorio es:

<https://github.com/mroccast/UMLClasesMaiteRocamora>

1. Creación del repositorio en GitHub

Creamos un repositorio en GitHub. Para ello vamos al icono en la parte superior derecha con el símbolo + y seleccionamos “New repository”, ponemos el nombre de nuestro repositorio y seleccionamos “Create Repository”.

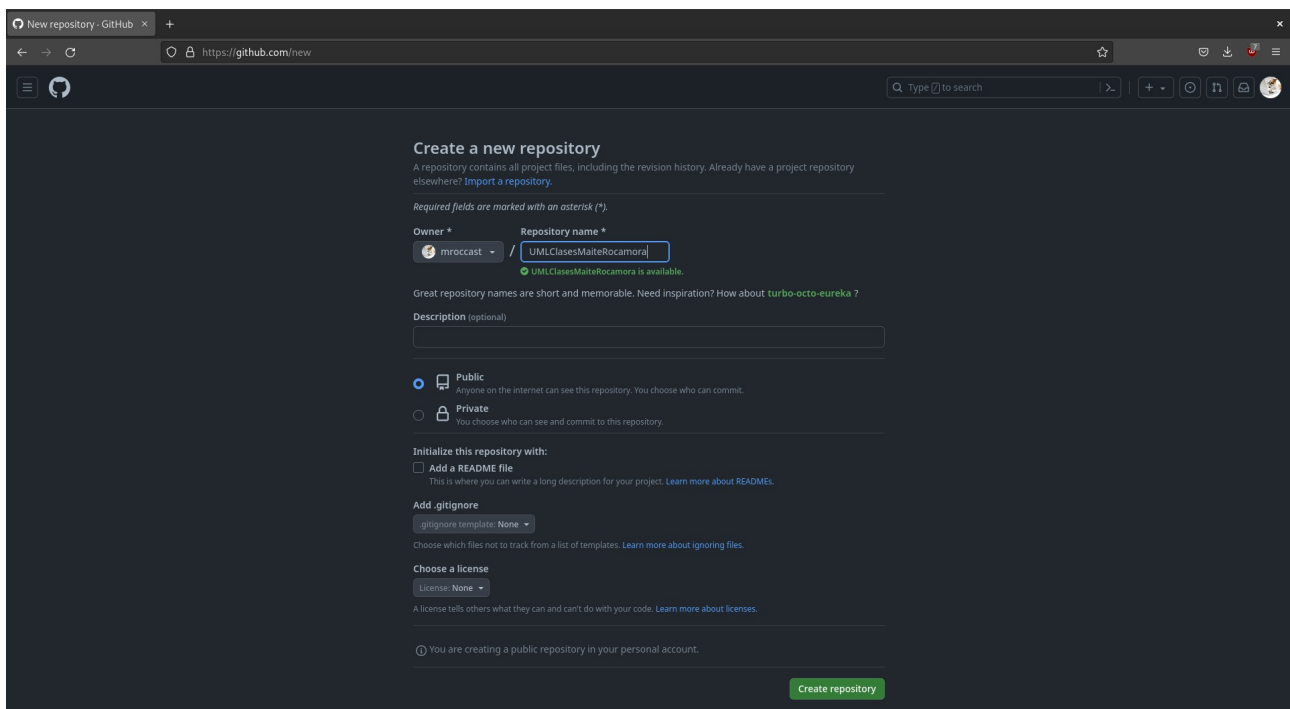


Figura 1: Creación de nuestro repositorio en GitHub

2. Instalación del plugin

Primero vamos a instalar el plugin de PlantUML Integration, vamos a entrar en el menú de IntelliJ, y dentro del menú accedemos a: **File > Settings > Plugins**

Dentro de Plugins, escribimos PlantUML Integration y le damos a “Install” y “OK”

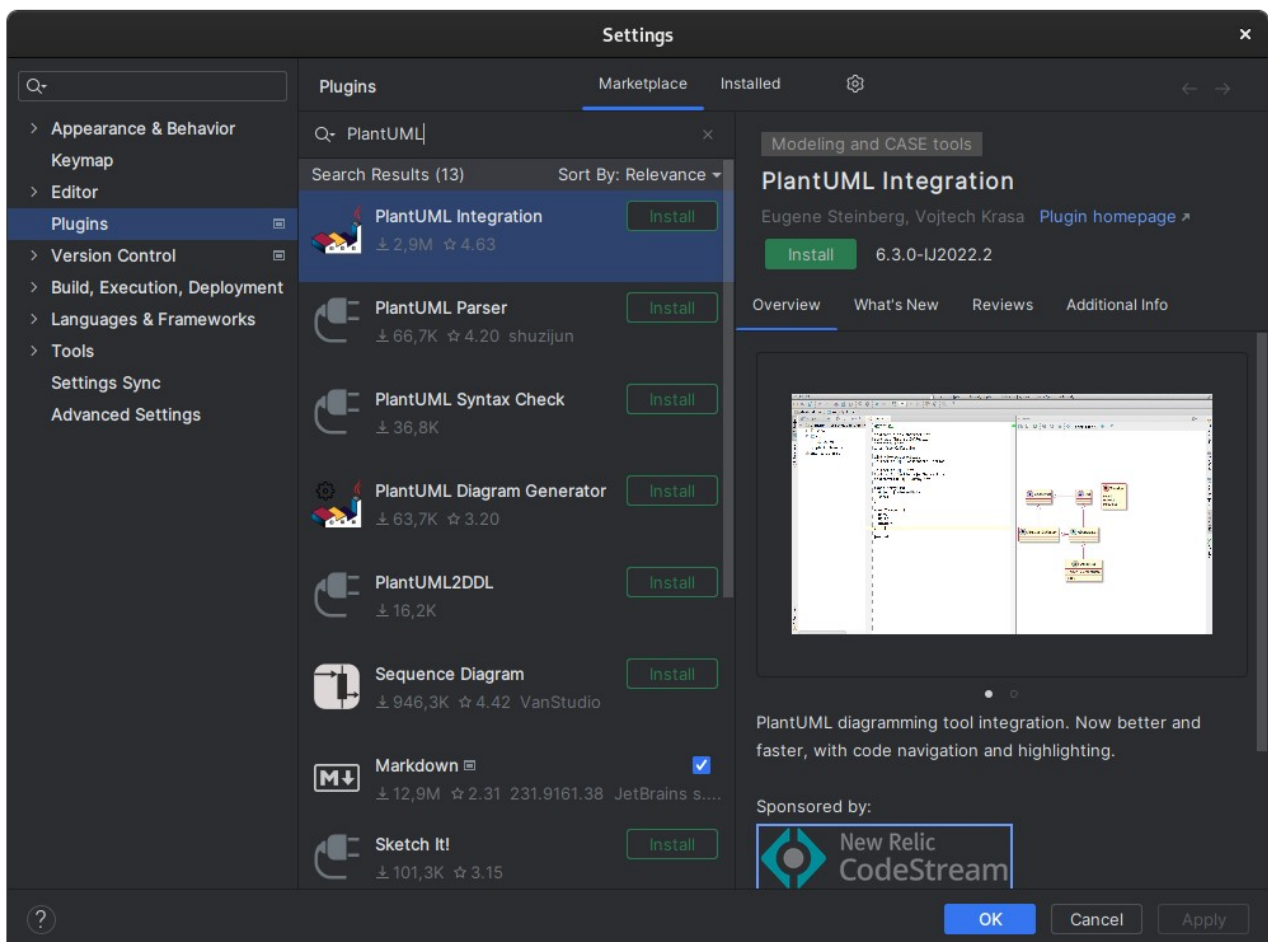
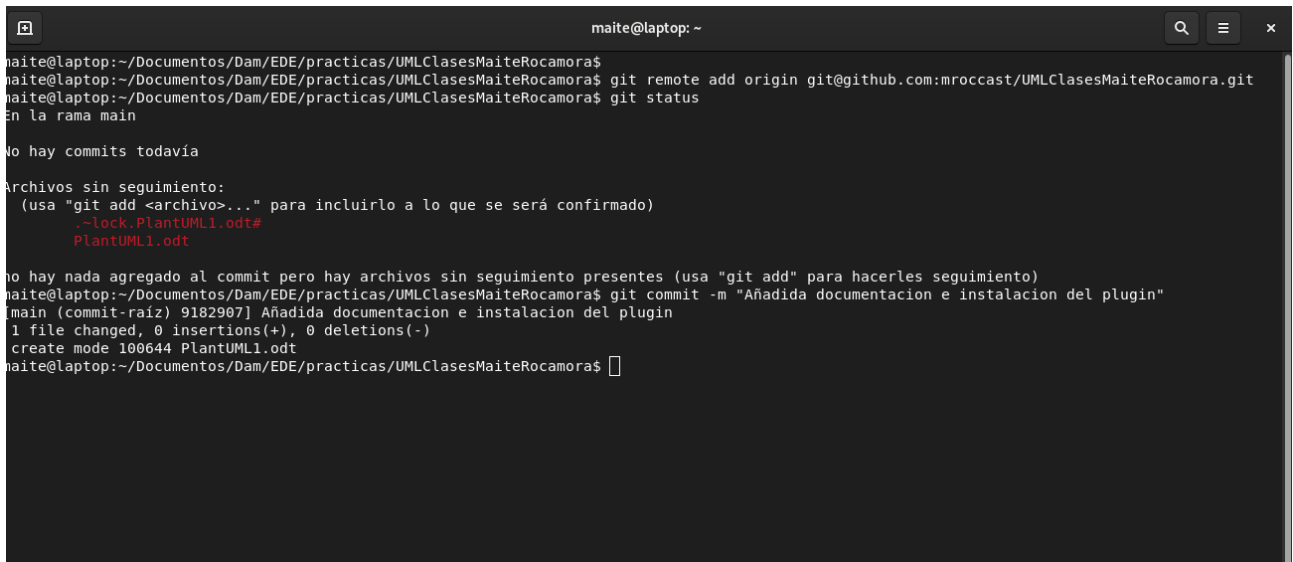


Figura 2: Instalación del plugin PlantUML

En este punto subimos la memoria al repositorio.



```
maite@laptop: ~  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git remote add origin git@github.com:mroccast/UMLClasesMaiteRocamora.git  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git status  
En la rama main  
  
No hay commits todavía  
  
Archivos sin seguimiento:  
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)  
    .-lock.PlantUML1.odt#  
    PlantUML1.odt  
  
No hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git commit -m "Añadida documentacion e instalacion del plugin"  
[main (commit-raiz) 9182907] Añadida documentacion e instalacion del plugin  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 PlantUML1.odt  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$
```

Figura 3: Subida de la memoria al repositorio

Una vez subida la memoria al repositorio vamos a abrir el proyecto con IntelliJ.

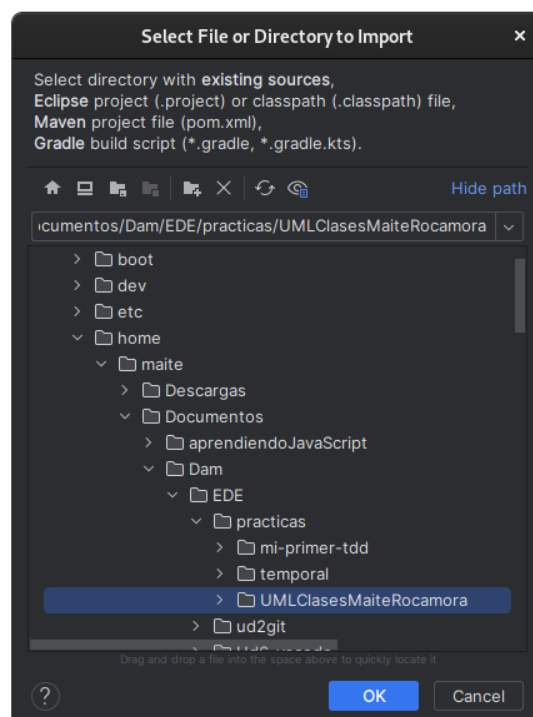


Figura 4: Seleccionamos el directorio del proyecto

4. Creación del modelo de datos

Nos ponemos en contexto con el ejercicio y analizamos el texto.

La **Asociación de Antiguos Alumnos de la UOC** quiere un programa que les permita gestionar a sus asociados, eventos y demás elementos relacionados.

Los **asociados** se pueden dividir en: miembros **numerarios** y en miembros de la **junta directiva**

La **junta directiva** es elegida por votación en una asamblea general cada cuatro años.

Los **miembros** de la junta directiva son convocados a las **reuniones de junta**.

Los demás **miembros** no son convocados.

Las **actividades** que se realizan están abiertas a todos los miembros de la asociación.

La convocatoria de un **evento** se realiza por correo electrónico a todos los miembros activos.

En la **convocatoria**, aparece información sobre el **lugar** que en muchos casos se repite, por lo que quieren almacenar los datos para futuros usos.

En todos los **eventos**, la aceptación de los asistentes se realiza por orden de llegada ya que, se puede dar que el número de asistentes sea limitado, como en las conferencias.

Por tanto, diferenciamos:

- Miembro
- Miembro de la junta directiva
- Evento
- Conferencia
- Reunión de la junta directiva
- Localización

Vemos un modelo de jerarquía de herencia, donde la superclase es Evento que tiene dos subclases, que son las Conferencias y las Reuniones de la junta directiva. Por lo que principalmente el modelo nos quedaría así:

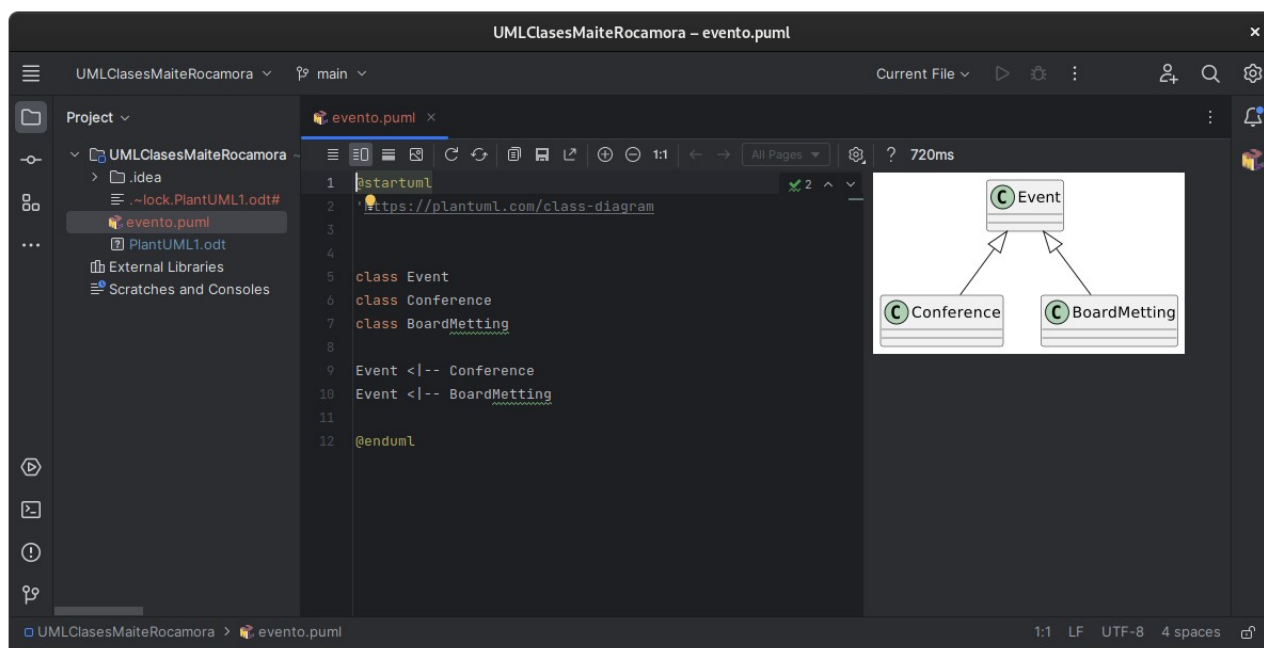


Figura 5: Primer modelo superclase Evento, con dos subclases.

Hacemos un commit para salvar el estado del proyecto

```
maite@laptop: ~  
modificado: PlantUML1.odt  
Archivos sin seguimiento:  
(usa "git add <archivo>..." para incluirlo a lo que se será confirmado)  
.idea/  
.-lock.PlantUML1.odt#  
evento.puml  
sin cambios agregados al commit (usa "git add" y/o "git commit -a")  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git add evento.puml PlantUML1.odt  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git s  
En la rama main  
Cambios a ser confirmados:  
(usa "git restore --staged <archivo>..." para sacar del área de stage)  
modificado: PlantUML1.odt  
nuevo archivo: evento.puml  
Archivos sin seguimiento:  
(usa "git add <archivo>..." para incluirlo a lo que se será confirmado)  
.idea/  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git commit -m "Añadido diagrama de Eventos"  
[main e34c5b4] Añadido diagrama de Eventos  
2 files changed, 12 insertions(+)  
create mode 100644 evento.puml  
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$
```

Figura 6: Commit de nuestro proyecto con PlantUML

Vemos que existe otra jerarquía entre los miembros de la asociación, donde añadimos la clase persona, para identificar a los conferenciantes porque podría darse el caso de que alguno no fuera miembro de la asociación.

Por lo tanto nos quedaría el siguiente modelo:

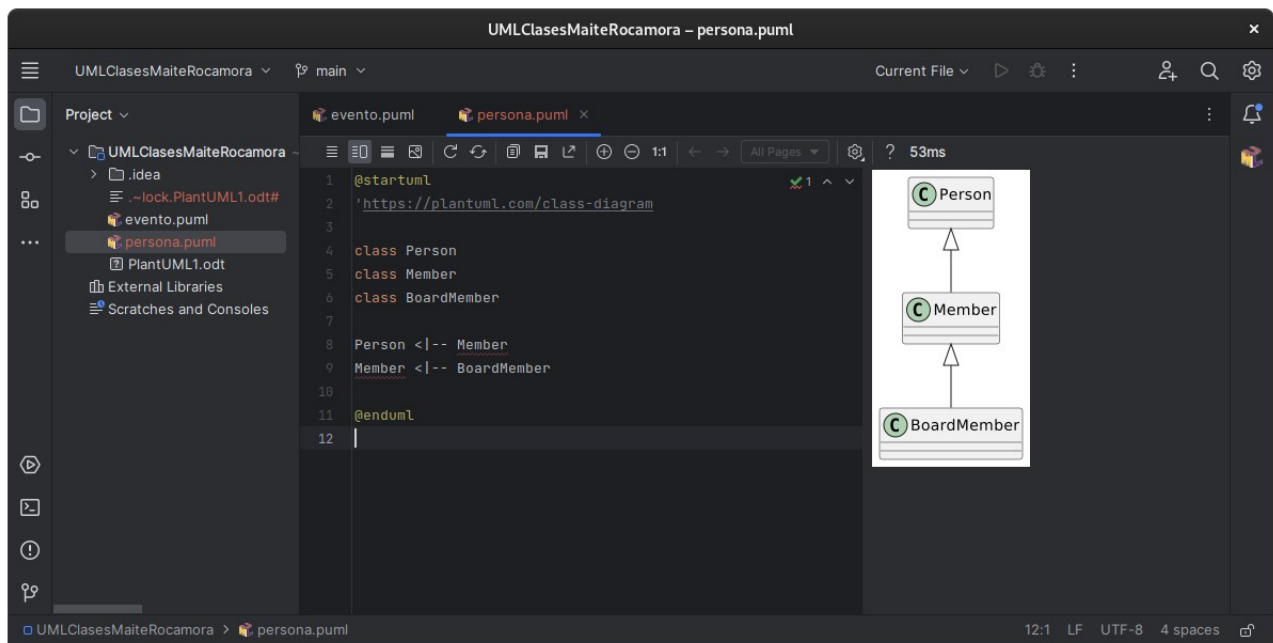
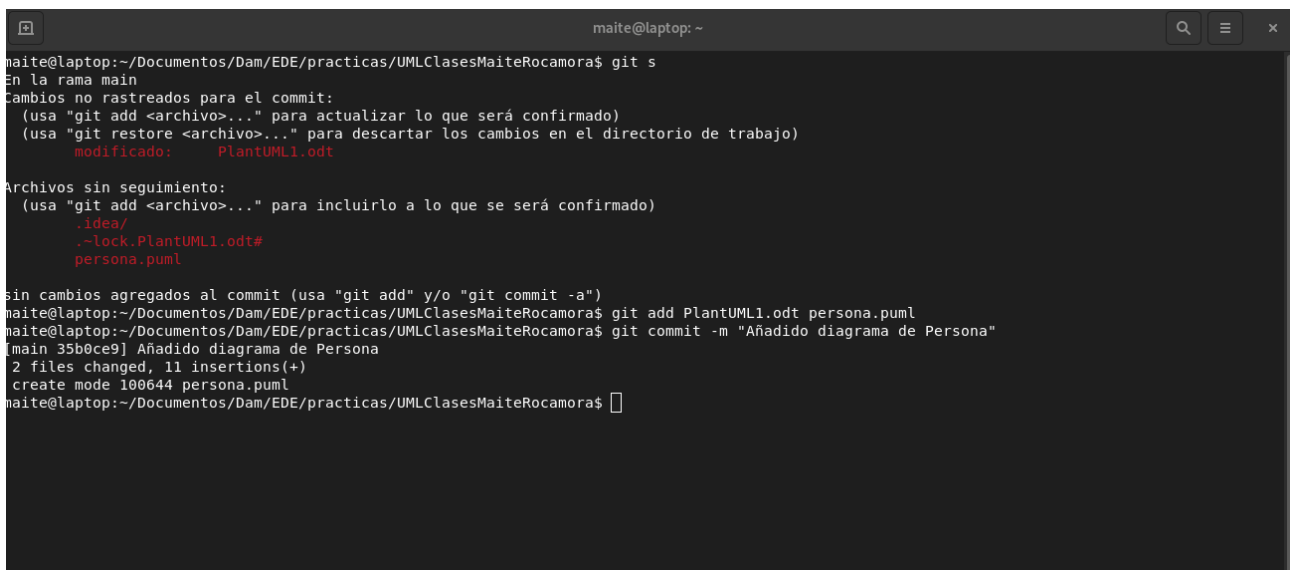


Figura 7: Segundo modelo, jerarquía de clase de miembros

Realizamos un commit y vamos salvando los cambios de nuestro proyecto.

A terminal window titled 'maite@laptop: ~' showing the output of 'git status' and the execution of 'git commit -m "Añadido diagrama de Persona"'. The status output indicates that 'PlantUML1.odt' is modified and 'persona.puml' is a new file. The commit message is 'Añadido diagrama de Persona' and it shows 2 files changed with 11 insertions.

```
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git s
En la rama main
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
      modificado:   PlantUML1.odt

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
      .idea/
      .-lock.PlantUML1.odt#
      persona.puml

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git add PlantUML1.odt persona.puml
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git commit -m "Añadido diagrama de Persona"
[main 35b0ce9] Añadido diagrama de Persona
 2 files changed, 11 insertions(+)
 create mode 100644 persona.puml
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$
```

Figura 8: Commit añadido diagrama de persona

Además, tenemos las clases Localización y Asociación (AAUOC) , que se relacionan con el resto de clases del siguiente modo:

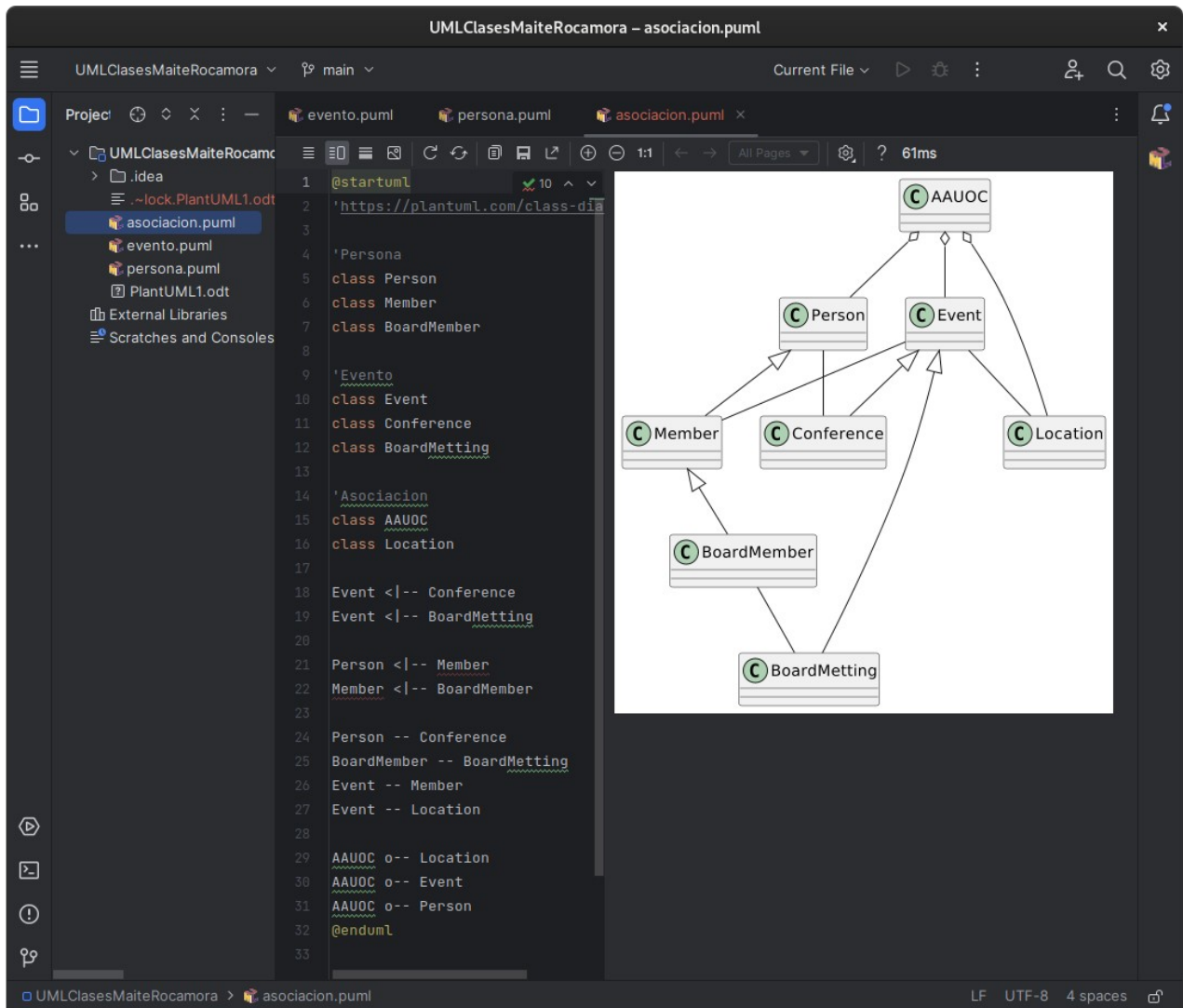


Figura 9: Modelo de la relación de Asociación y Localización con el resto de jerarquías.

Hacemos un commit y guardamos los cambios del proyecto, es muy importante este paso para no perder nada de información.

A terminal window titled 'maite@laptop: ~' showing the execution of 'git s' (git status). The output indicates the current branch is 'main' and lists changes to be confirmed: 'PlantUML1.odt' (modified) and 'asociacion.puml' (new file). It also shows files not under tracking: '.idea/' and '!.lock.PlantUML1.odt#'. The user then runs 'git commit -m "Añadido diagrama de Asociacion"', which results in a commit hash 'c7d58d7' and shows that 2 files were changed with 32 insertions, and a new file 'asociacion.puml' was created.

```
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git s
En la rama main
Cambios a ser confirmados:
  (usa "git restore --staged <archivo>..." para sacar del área de stage)
    modificado:   PlantUML1.odt
    nuevo archivo: asociacion.puml

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
    .idea/
    !.lock.PlantUML1.odt#

maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git commit -m "Añadido diagrama de Asociacion"

[main c7d58d7] Añadido diagrama de Asociacion
 2 files changed, 32 insertions(+)
 create mode 100644 asociacion.puml
maite@laptop:~/Documentos/Dam/EDE/practicas/UMLClasesMaiteRocamora$
```

Figura 10: Commit "Añadido el diagrama de Asociación y Localización"

Vamos a incluir ahora los atributos a las clases.

La Asociación necesita un conjunto de métodos para añadir nuevos eventos, personas y localizaciones al sistema (que son los métodos que llamamos newLocation, newEvent y newPerson) y también un método para informar a los miembros de la convocatoria de un evento (que llamaremos método informEvent).

Los usuarios necesitarán confirmar la asistencia a los eventos (que lo llamaremos método register, que deberá almacenar los asistentes por orden y controlar el número máximo de personas.

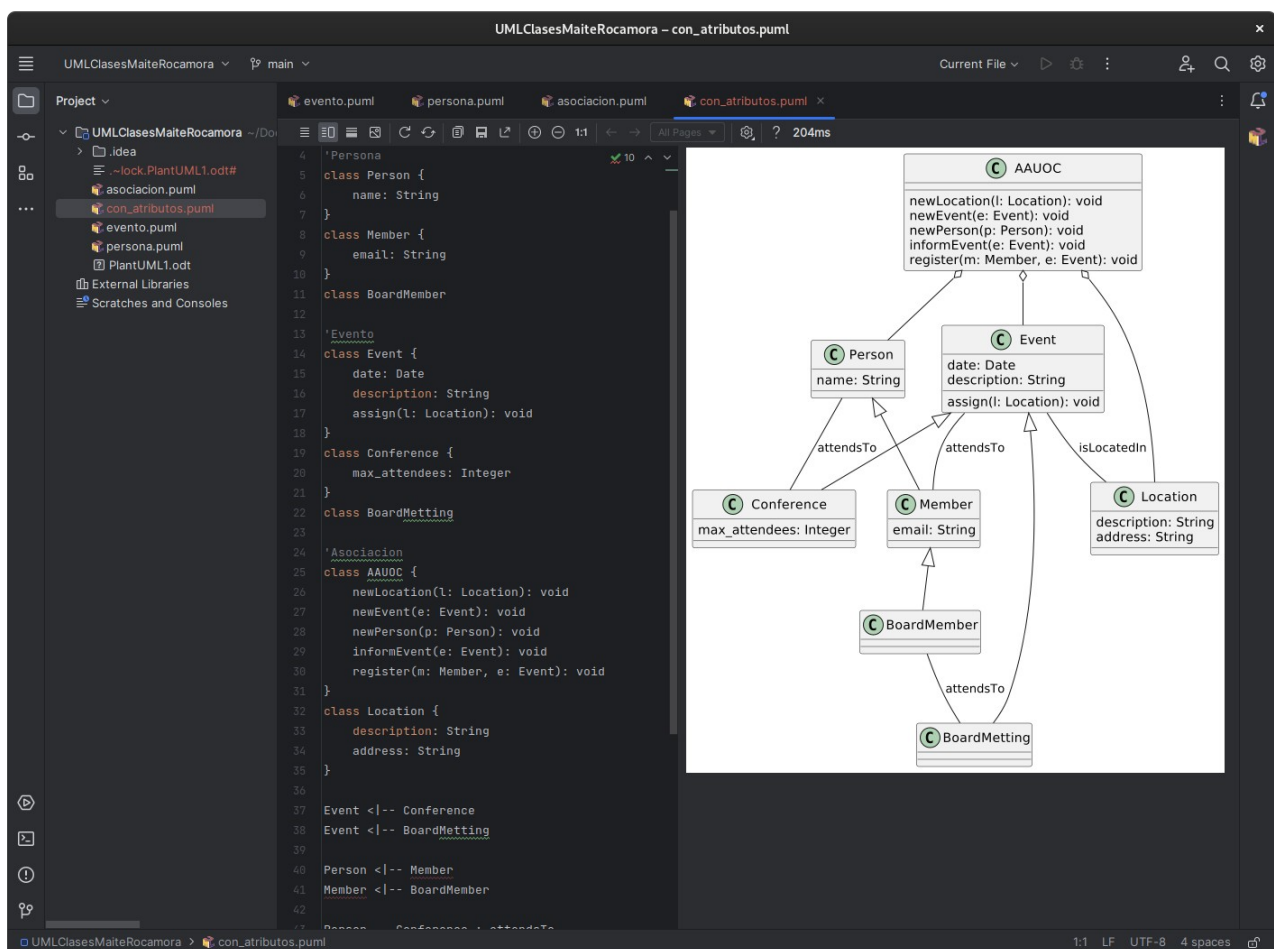


Figura 11: Modelo con atributos y métodos

Continuamos haciendo commits para guardar nuestro proyecto.

```
maite@laptop: ~/Documents/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git add con_atributos.puml PlantUML1.odt
maite@laptop:~/Documents/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git s
En la rama main
Cambios a ser confirmados:
  (usa "git restore --staged <archivo>..." para sacar del área de stage)
    modificado:   PlantUML1.odt
    nuevo archivo: con_atributos.puml

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
    .idea/
    ~/.lock.PlantUML1.odt#

maite@laptop:~/Documents/Dam/EDE/practicas/UMLClasesMaiteRocamora$ git commit -m "Añadido diagrama con atributos"
[main ca879ce] Añadido diagrama con atributos
 2 files changed, 51 insertions(+)
 create mode 100644 con_atributos.puml
maite@laptop:~/Documents/Dam/EDE/practicas/UMLClasesMaiteRocamora$
```

Figura 12: Commit del modelo con atributos y métodos

Incluiremos ahora la cardinalidad y navegabilidad de las relaciones, que son todas bidireccionales porque no se nos han comunicado ningún tipo de relación o acceso a la información de una clase a otra.

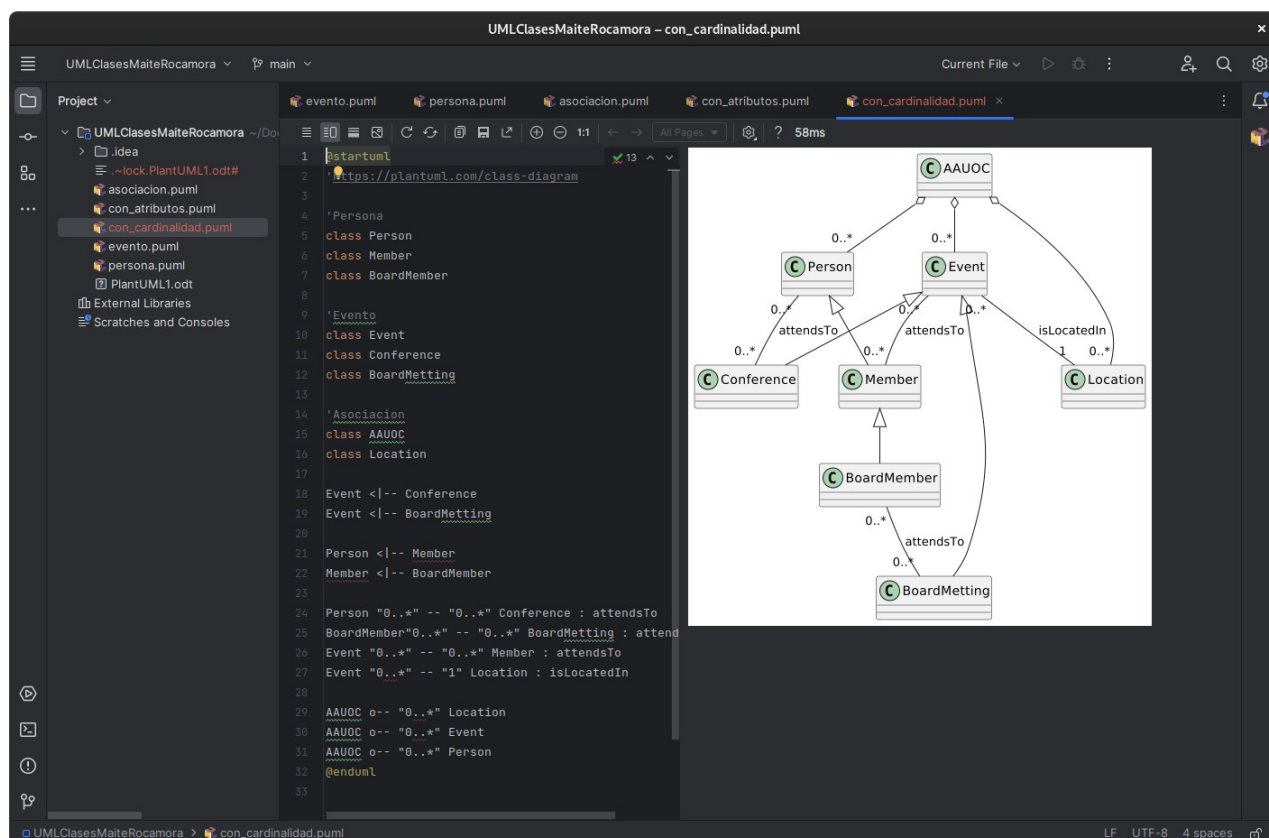


Figura 13: Modelo con las cardinalidades