

**Entornos de Desarrollo**  
**Unidad 11 - Práctica Diagramas UML Clases**

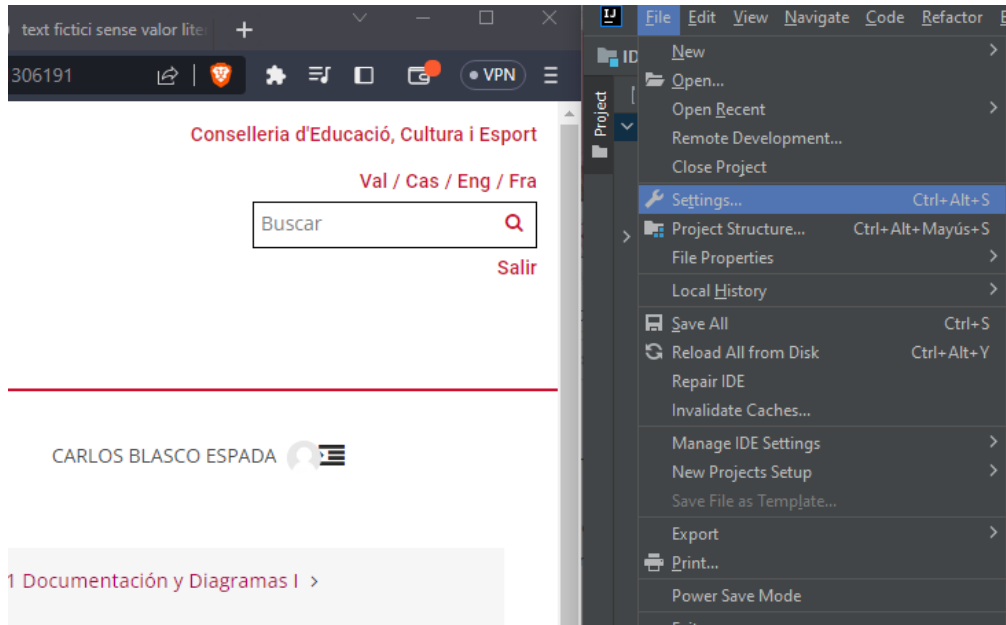
**Carlos Blasco Espada**  
**1º Q - DAM**

## Pasos previos:

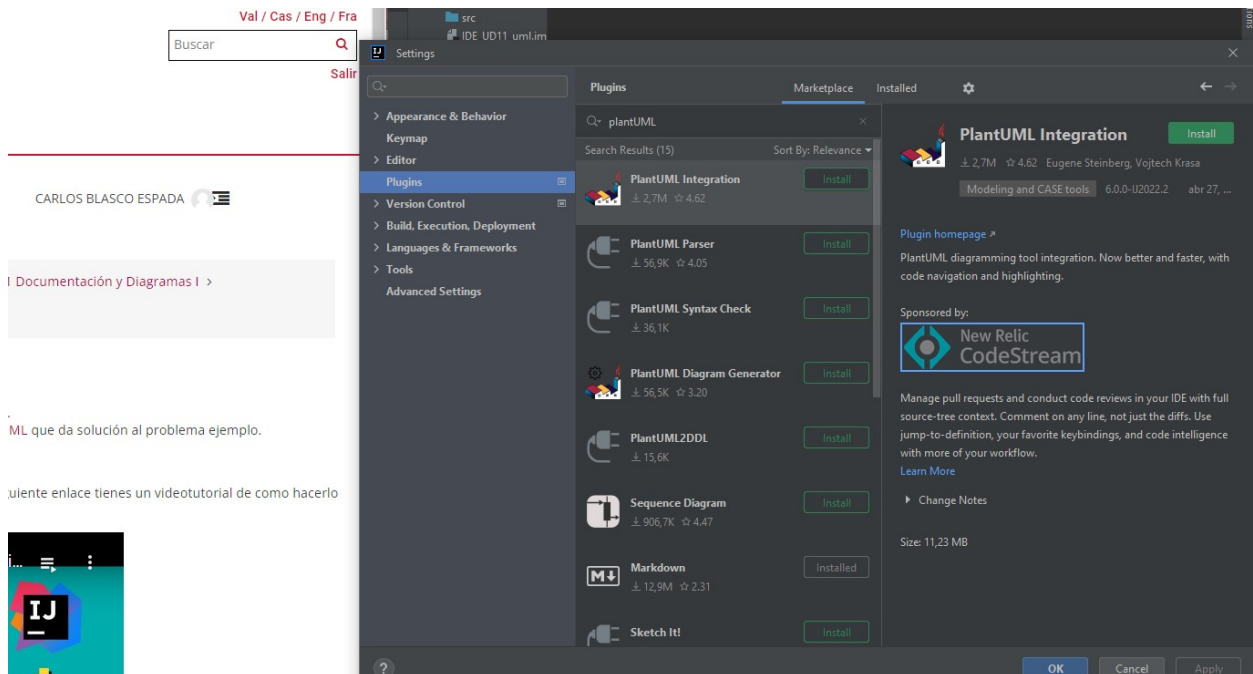
- Necesitamos el entorno de desarrollo IntelliJ IDEA.
- Crear el proyecto.

## Pasos a seguir:

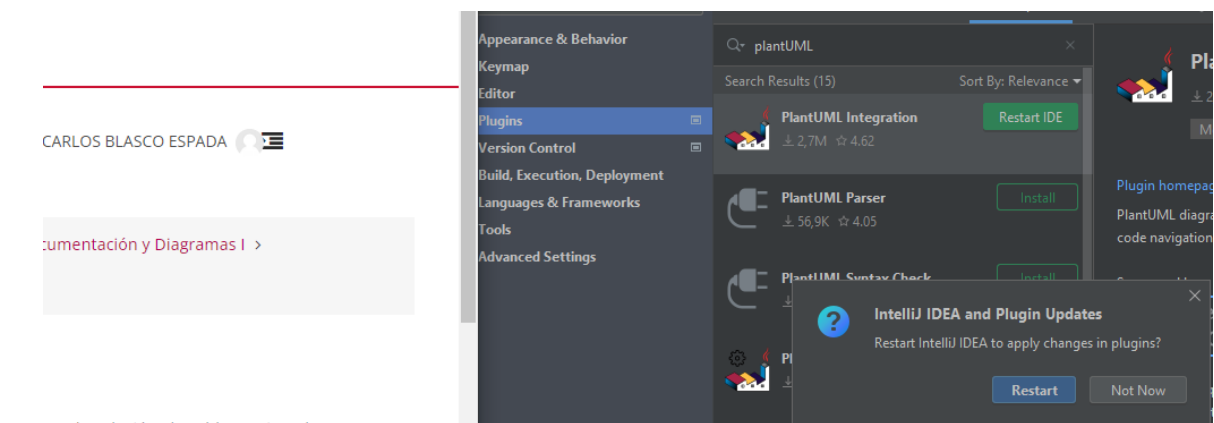
1. Instalaremos el plugin PlantUML. El primer paso es ir a File > Settings.



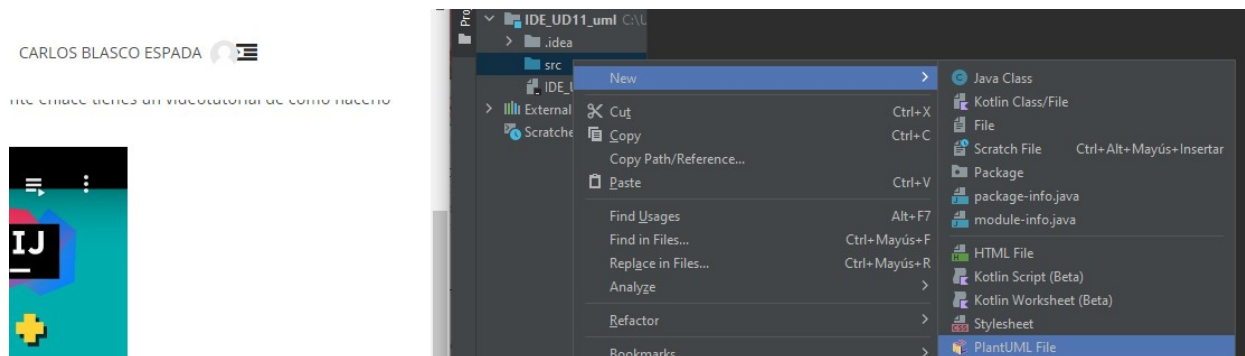
2. Hacemos clic en Plugins en el menú izquierdo y buscamos plantUML. Hacemos clic en "Install" en el plugin PlantUML Integration-



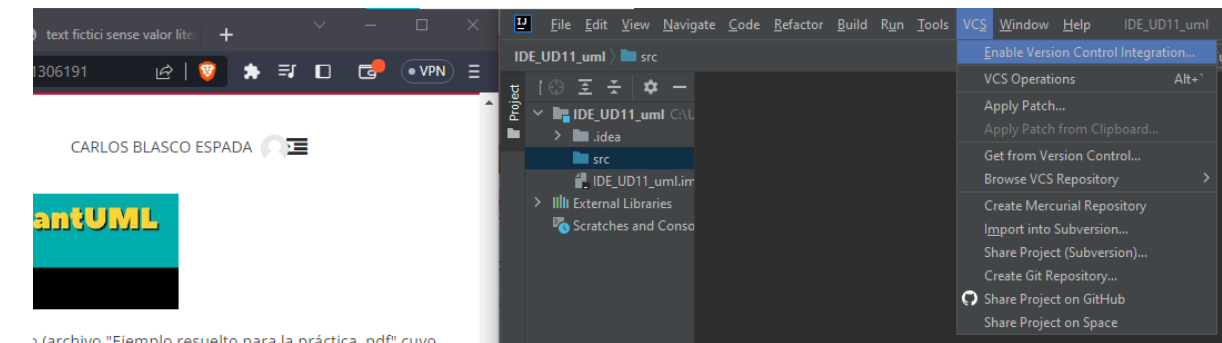
3. Para finalizar la instalación del plugin debemos reiniciar IntelliJ IDEA haciendo clic en "Restart IDE".



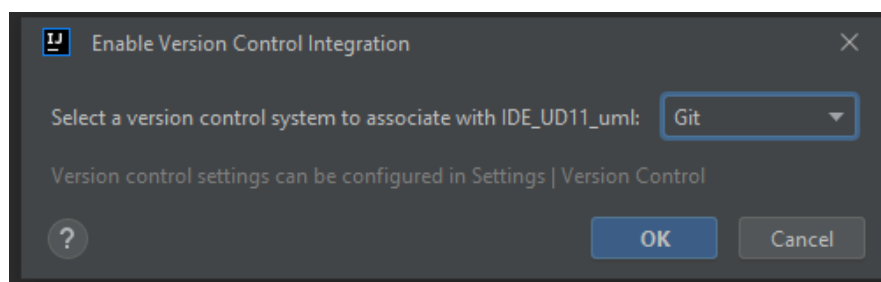
4. Tras el reinicio podremos crear un modelo UML, pero antes habilitaremos el VCS.



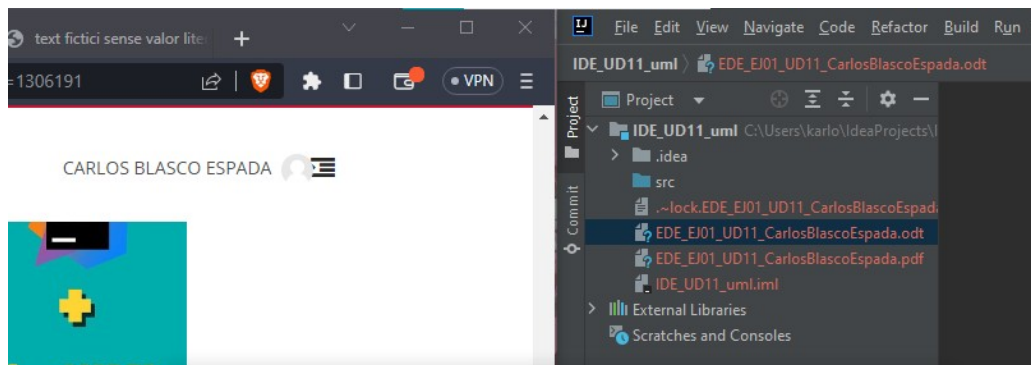
5. En la barra superior hacemos clic en VCS > Enable Version Control Integration.



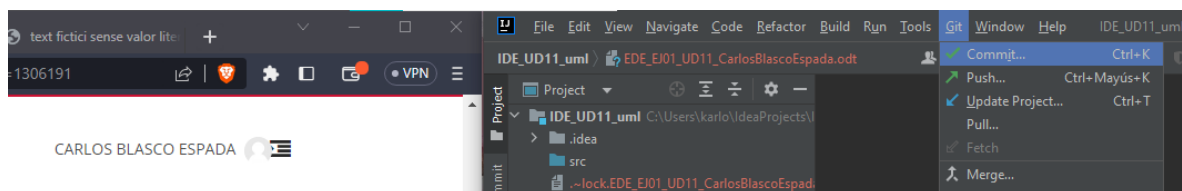
6. El VCS será Git. Hacemos clic en OK y ya estará activado.



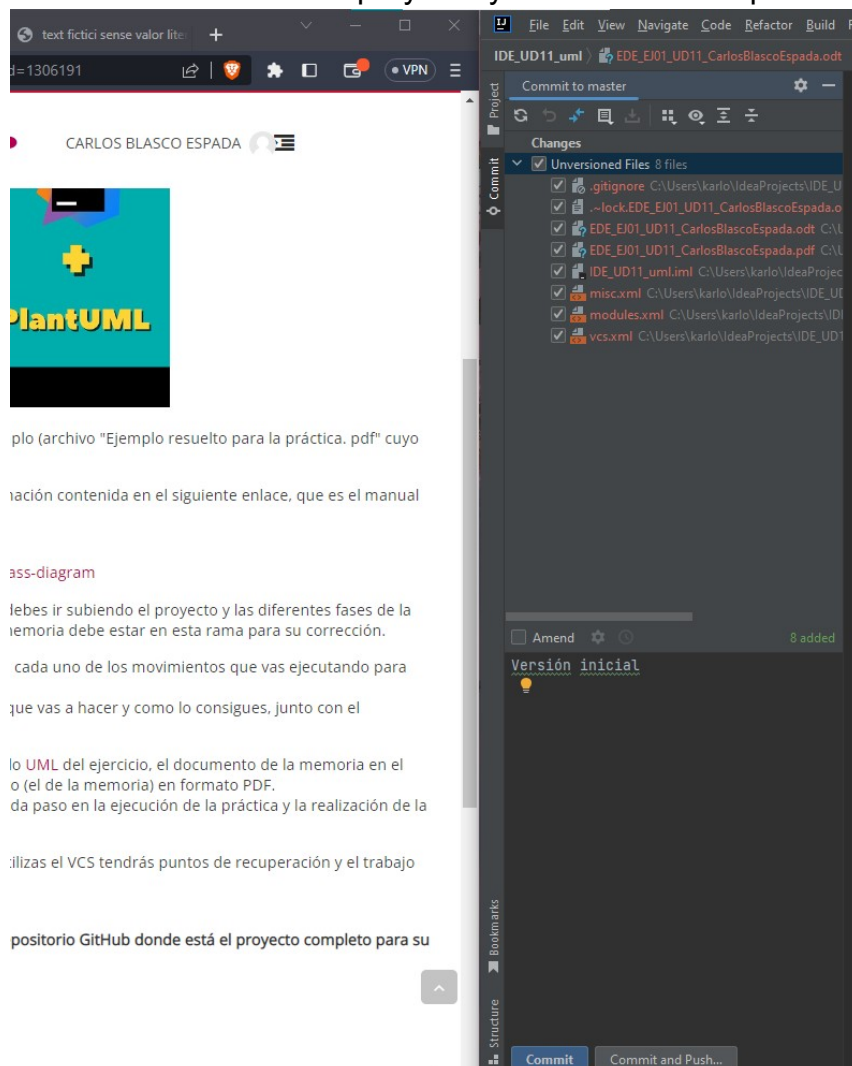
7. Añadimos el archivo .odt y su .pdf correspondiente al repositorio. Podemos hacer este paso arrastrando los archivos al proyecto o mediante el explorador de archivos.



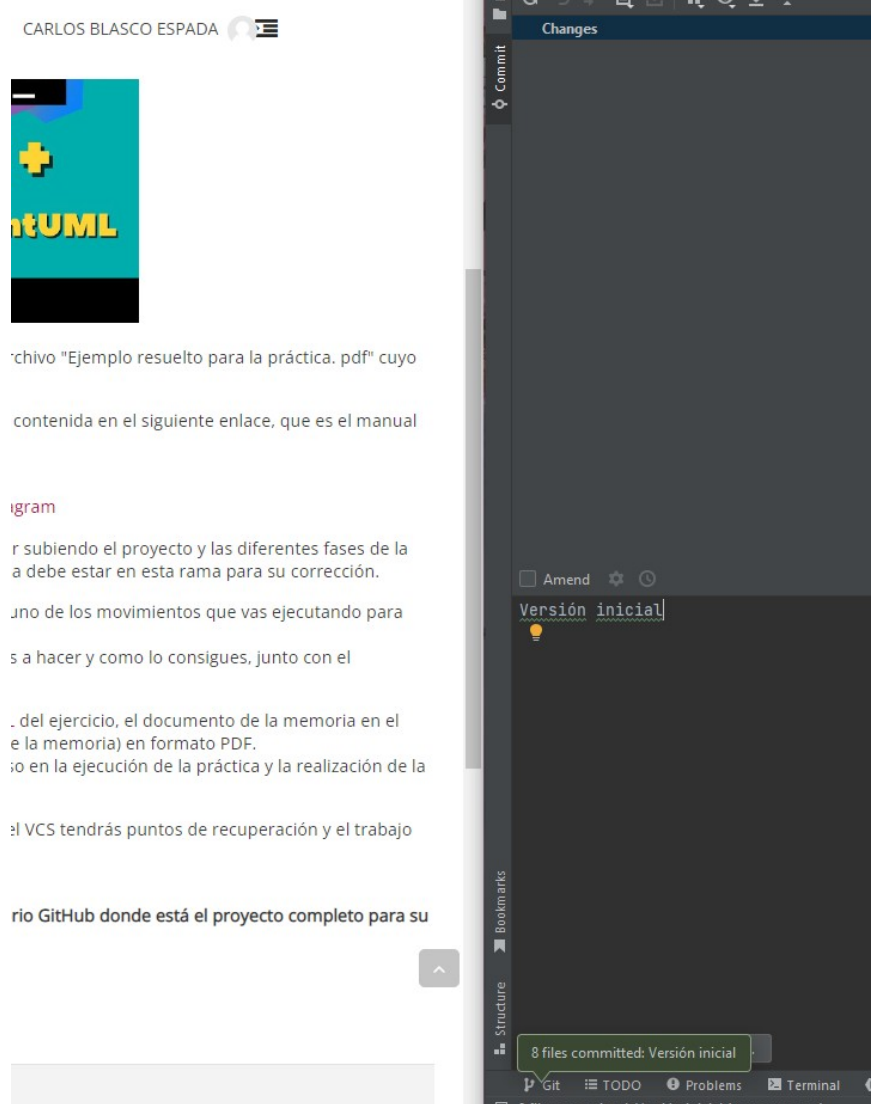
8. Realizamos un commit de los pasos realizados hasta ahora. Git > Commit.



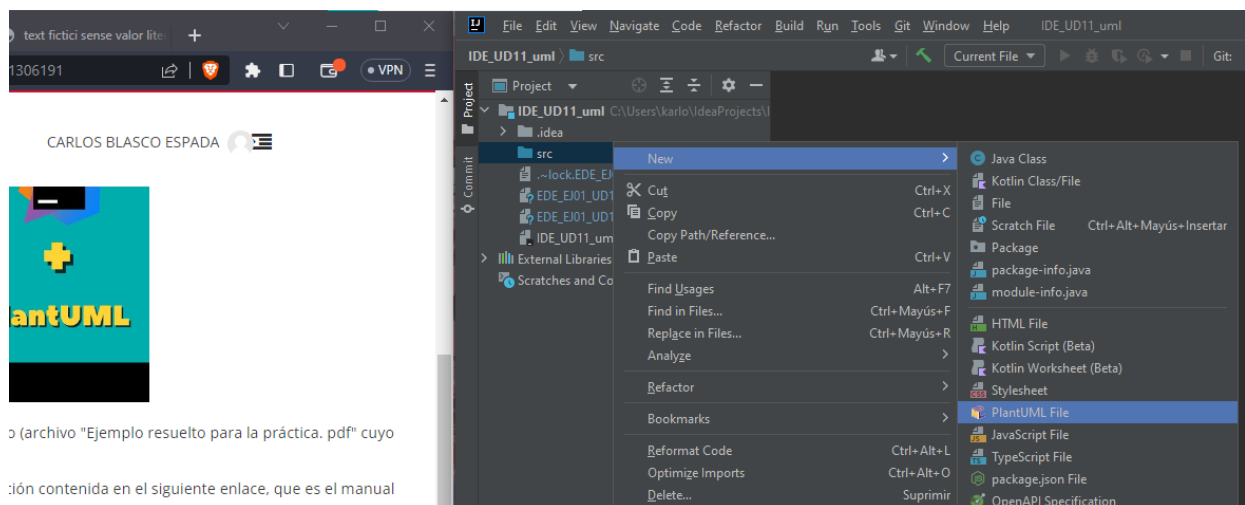
9. Añadimos todos los archivos del proyecto y una breve descripción.



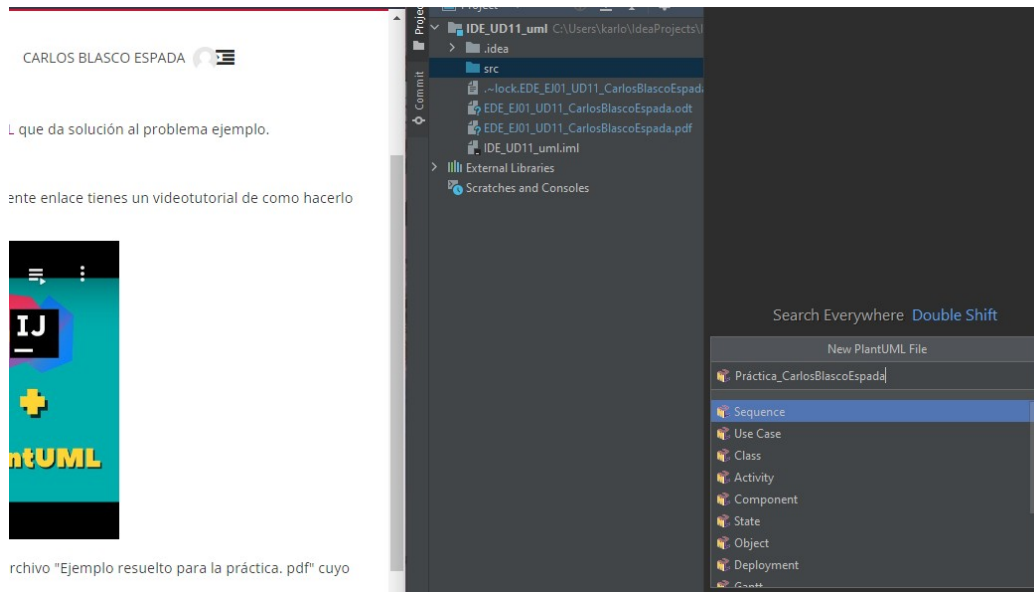
## 10. Pulsamos en Commit y ya tendremos nuestra primera versión del proyecto.



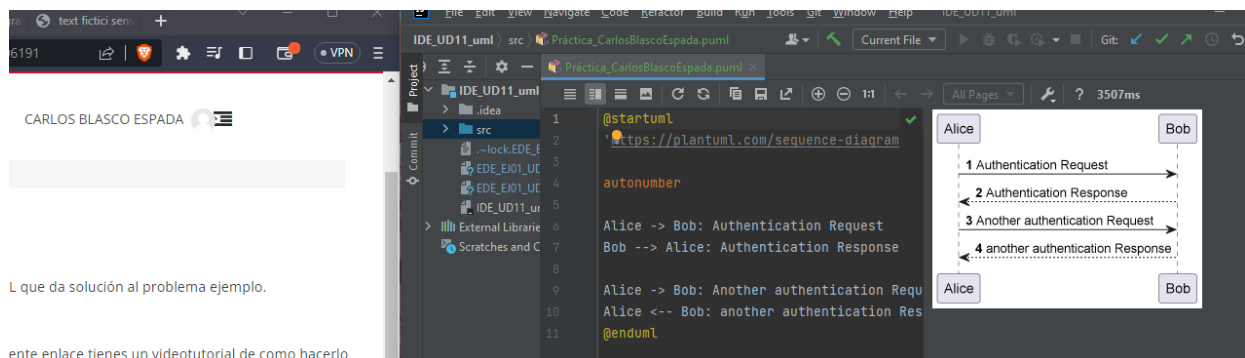
## 11. Creamos el modelo UML. Clic derecho sobre "src" > New > PlantUML File.



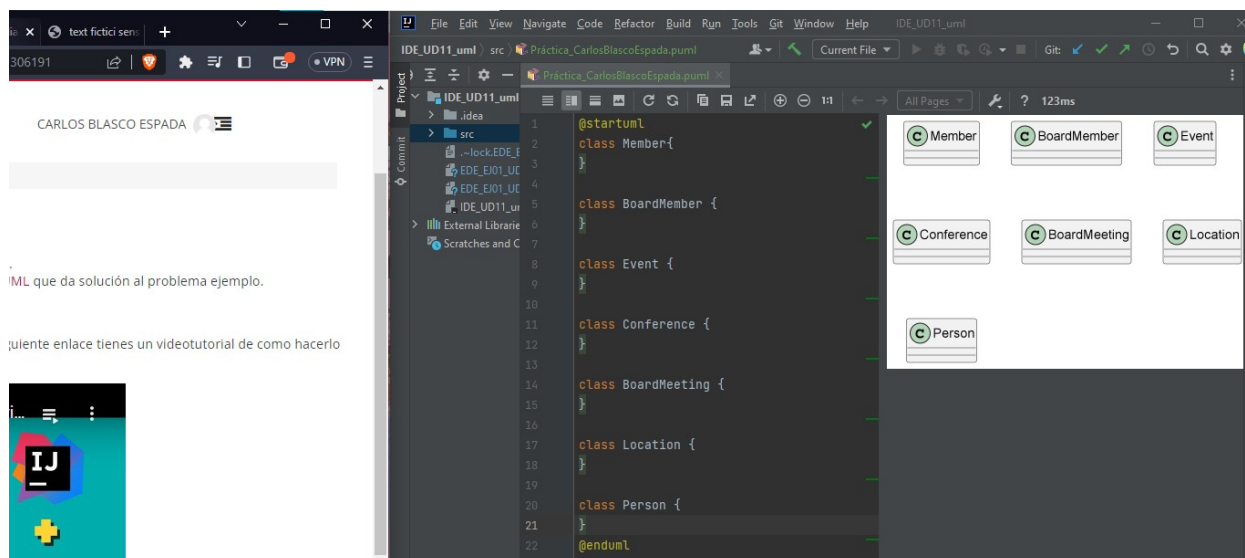
## 12. Establecemos un nombre para el archivo y pulsamos Enter.



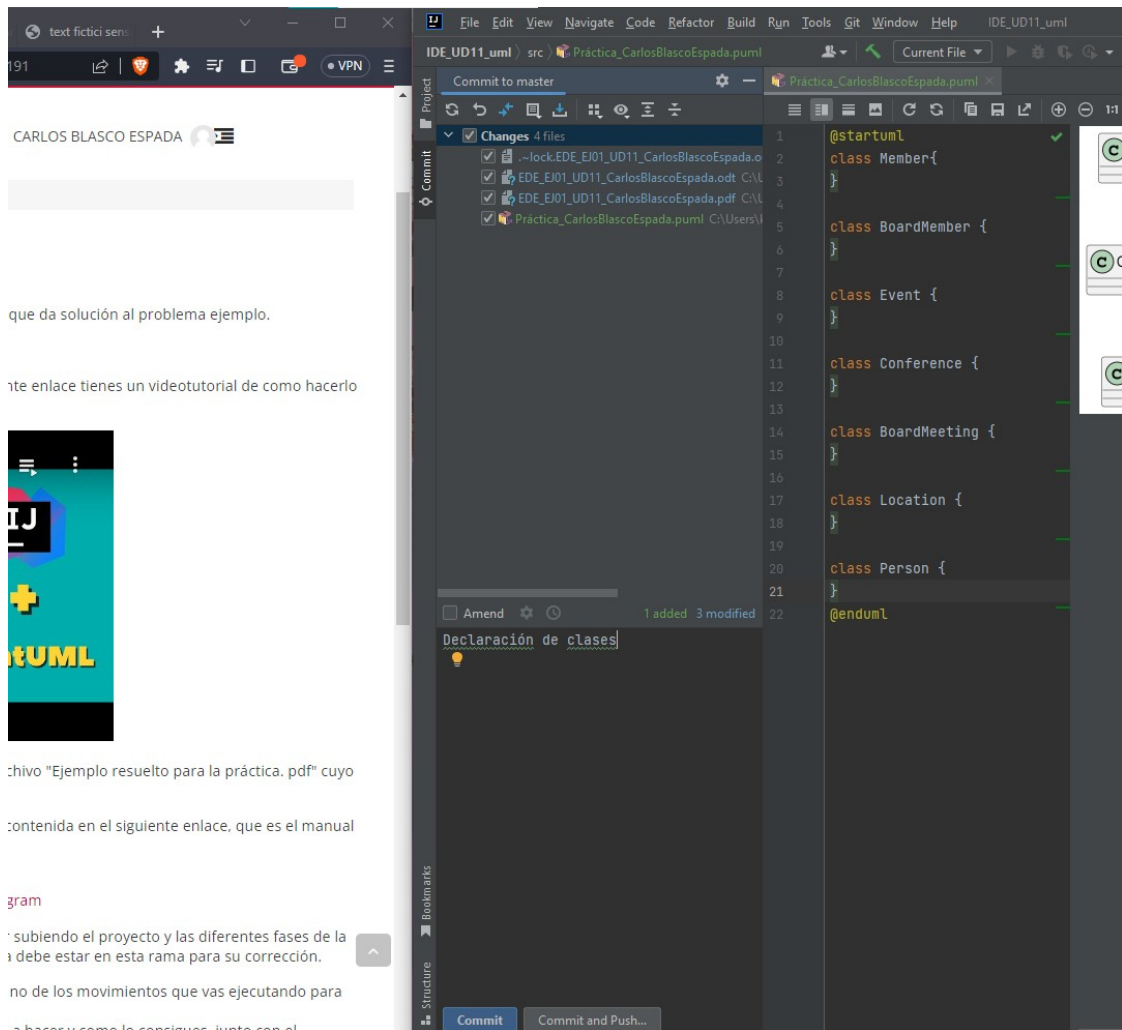
## 13. Tendremos un ejemplo por defecto.



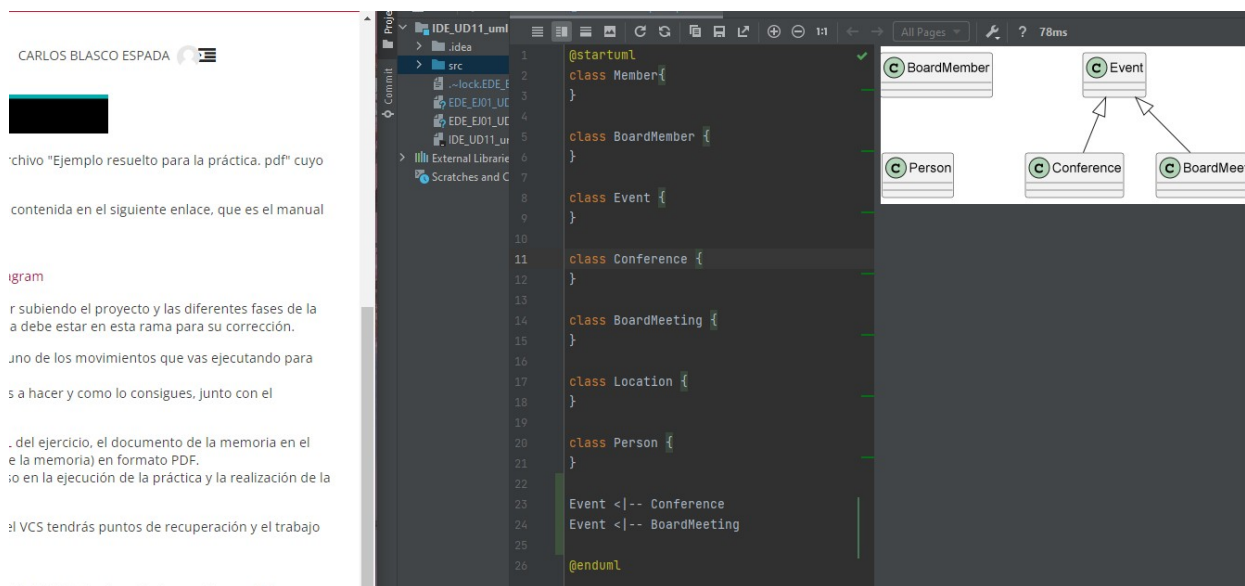
## 14. Borramos el ejemplo y hacemos la declaración de las clases necesarias, siguiendo la sintaxis de PlantUML.



15. Realizamos un commit con los cambios siguiendo el proceso mencionado anteriormente (Git > Commit > seleccionar archivos y poner descripción > Commit).



16. Definimos la clase Event como superclase con las clases Conference y BoardMeeting como subclases. Esto se indica mediante la sintaxis "<|--".





## 17. Realizamos un commit con los cambios.

CARLOS BLASCO ESPADA

rchivo "Ejemplo resuelto para la práctica. pdf" cuyo  
contenida en el siguiente enlace, que es el manual  
agram  
ir subiendo el proyecto y las diferentes fases de la  
ia debe estar en esta rama para su corrección.  
uno de los movimientos que vas ejecutando para  
s a hacer y como lo consigues, junto con el  
L del ejercicio, el documento de la memoria en el  
le la memoria) en formato PDF.  
so en la ejecución de la práctica y la realización de la  
el VCS tendrás puntos de recuperación y el trabajo  
rio GitHub donde está el proyecto completo para su

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

## 18. Añadimos la jerarquía entre miembros, donde BoardMember es la subclase de Member, que a su vez es la subclase de Person.

CARLOS BLASCO ESPADA

rchivo "Ejemplo resuelto para la práctica. pdf" cuyo  
contenida en el siguiente enlace, que es el manual  
agram  
ir subiendo el proyecto y las diferentes fases de la  
ia debe estar en esta rama para su corrección.  
uno de los movimientos que vas ejecutando para  
s a hacer y como lo consigues, junto con el  
L del ejercicio, el documento de la memoria en el  
le la memoria) en formato PDF.  
so en la ejecución de la práctica y la realización de la  
el VCS tendrás puntos de recuperación y el trabajo  
rio GitHub donde está el proyecto completo para su

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



## 19. Realizamos un commit con los cambios.

CARLOS BLASCO ESPADA

archivo "Ejemplo resuelto para la práctica. pdf" cuyo

contenida en el siguiente enlace, que es el manual

agram

ir subiendo el proyecto y las diferentes fases de la

uno de los movimientos que vas ejecutando para

s a hacer y como lo consigues, junto con el

L del ejercicio, el documento de la memoria en el  
e la memoria) en formato PDF.  
so en la ejecución de la práctica y la realización de la

el VCS tendrás puntos de recuperación y el trabajo

itorio GitHub donde está el proyecto completo para su

The screenshot shows a code editor with a commit message "Añadimos jerarquía de miembros" and a UML class diagram code snippet. The code defines classes Member, BoardMember, Event, Conference, BoardMeeting, Location, and Person, along with their relationships and generalizations.

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

## 20. Añadimos la clase AAUOC y sus relaciones de agregación con Location, Event y Person. También pondremos cómo se relacionan el resto de clases mediante "--".

CARLOS BLASCO ESPADA

El ejemplo (archivo "Ejemplo resuelto  
de esta explicación.

información contenida en el siguiente  
temas de Clases.

agram

onde debes ir subiendo el proyecto y las  
iremos la rama Master y la memoria

rios de cada uno de los movimientos que

de lo que vas a hacer y como lo  
o.

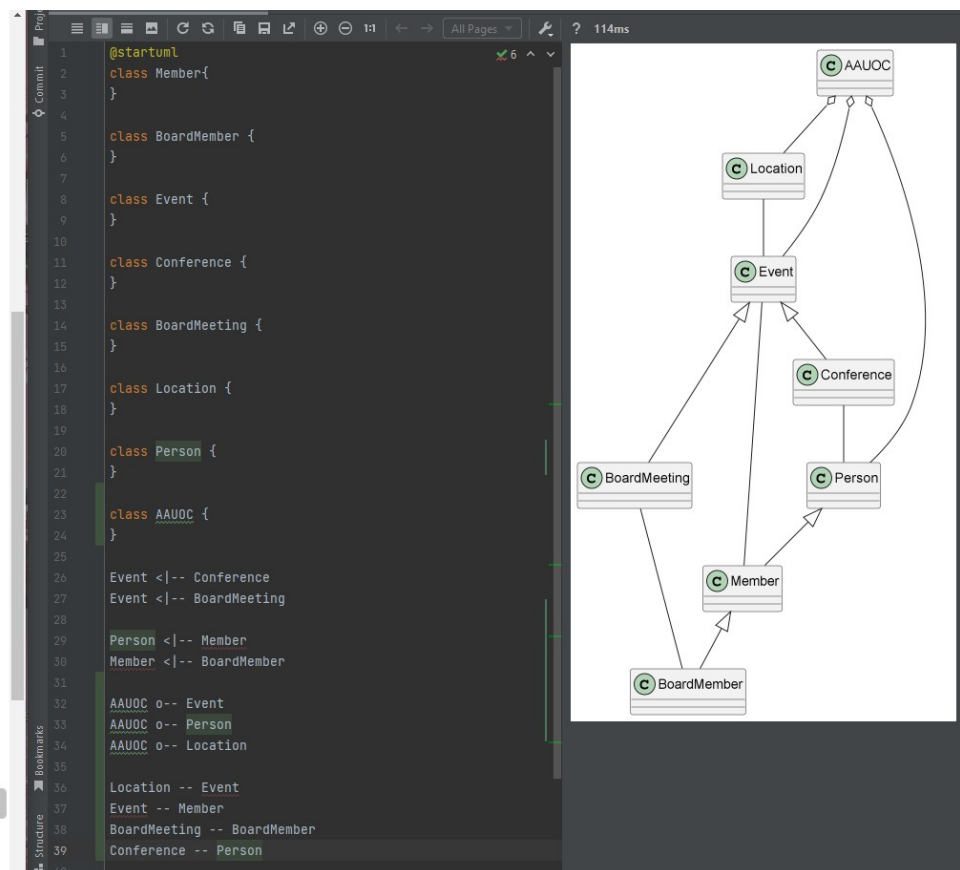
modelo UML del ejercicio, el documento  
noria y el mismo documento (el de la

lizar cada paso en la ejecución de la

ir. Si utilizas el VCS tendrás puntos de

re al repositorio GitHub donde está el

1:23



21. Realizamos un nuevo commit con estos cambios. Se puede observar en este commit y los anteriores que no sale el pdf, esto es porque una vez realizada la captura de abajo se escribe en el odt, se guarda el pdf y se hace el commit con la memoria actualizada.

CARLOS BLASCO ESPADA



ejemplo (archivo "Ejemplo resuelto" de esta explicación).

información contenida en el siguiente mas de Clases.

gram

nde debes ir subiendo el proyecto y las ramas la Master y la memoria

os de cada uno de los movimientos que se lo que vas a hacer y como lo

modelo UML del ejercicio, el documento seria y el mismo documento (el de la

ar cada paso en la ejecución de la

.Si utilizas el VCS tendrás puntos de

al repositorio GitHub donde está el

The screenshot shows an IDE with a commit window on the left and a UML class diagram on the right. The commit window shows three files being committed: `lock.EDE_EJ01_UD11_CarlosBlascoEspada.odt`, `EDE_EJ01_UD11_CarlosBlascoEspada.odt`, and `Práctica_CarlosBlascoEspada.puml`. The commit message is "Añadimos clase AAUOC y resto de relaciones". The UML class diagram shows the following classes and relationships:

```

classDiagram
    class Member {
    }
    class BoardMember {
    }
    class Event {
    }
    class Conference {
    }
    class BoardMeeting {
    }
    class Location {
    }
    class Person {
    }
    class AAUOC {
    }
    Event <|-- Conference
    Event <|-- BoardMeeting
    Person <|-- Member
    Member <|-- BoardMember
    AAUOC o-- Event
    AAUOC o-- Person
    AAUOC o-- Location
    Location -- Event
    Event -- Member
    BoardMeeting -- BoardMember
    Conference -- Person
  
```

22.0