

# **Entornos de Desarrollo**

## **Unidad 11 - Práctica Diagramas UML Clases**

<https://github.com/kande0/UMLClasesCarlosBlascoEspada.git>

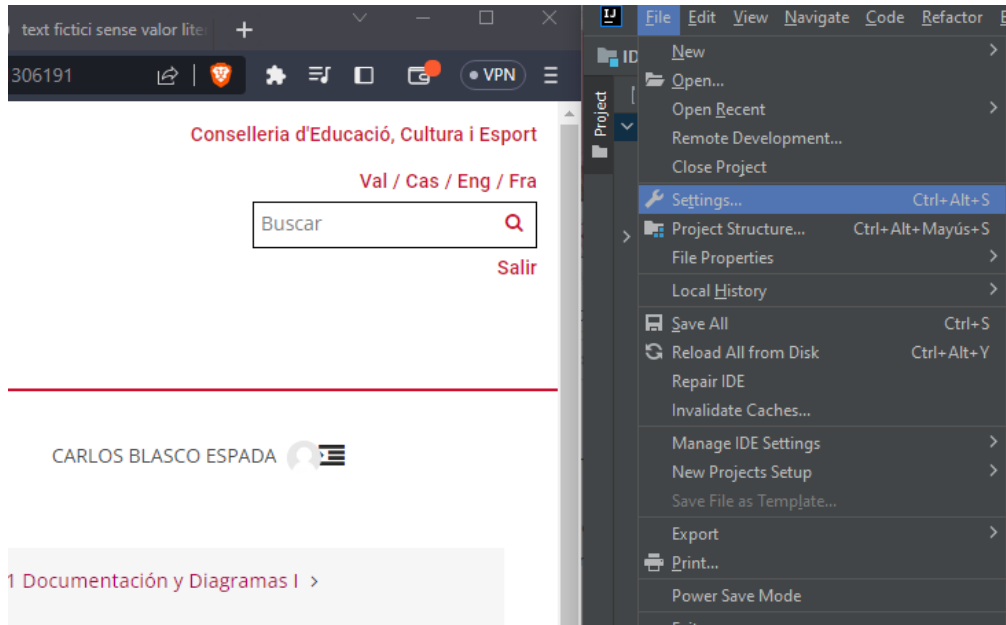
**Carlos Blasco Espada**  
**1º Q - DAM**

## Pasos previos:

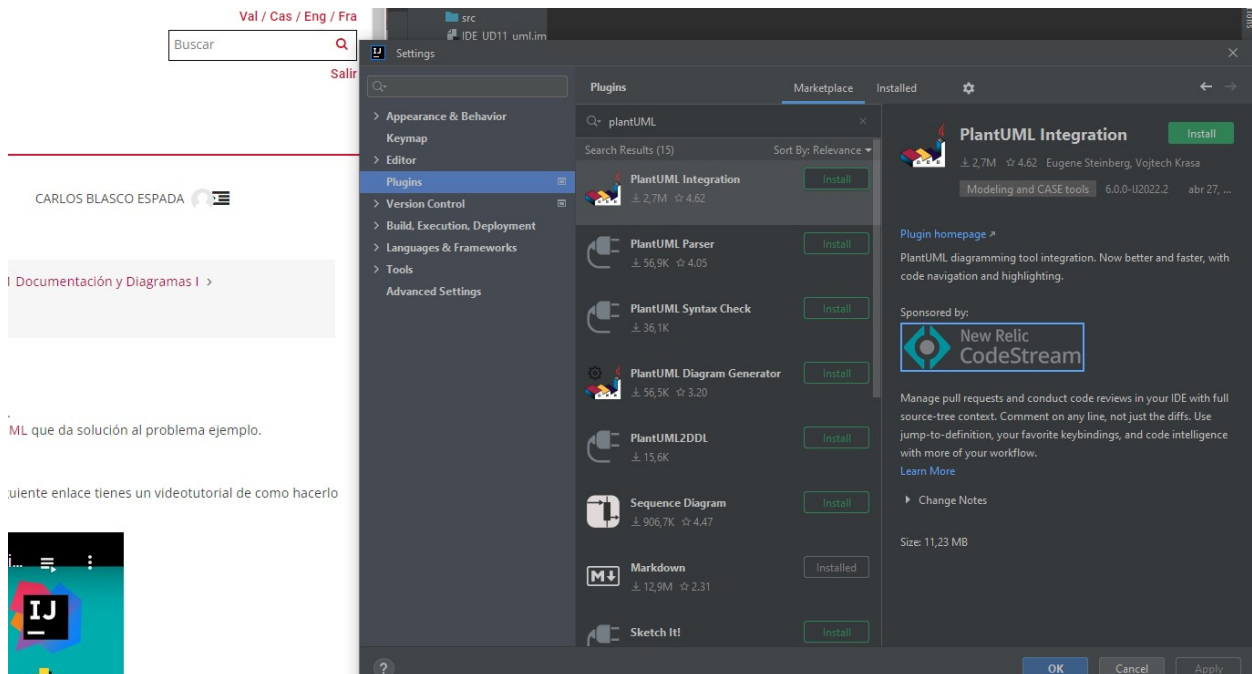
- Necesitamos el entorno de desarrollo IntelliJ IDEA.
- Crear el proyecto.

## Pasos a seguir:

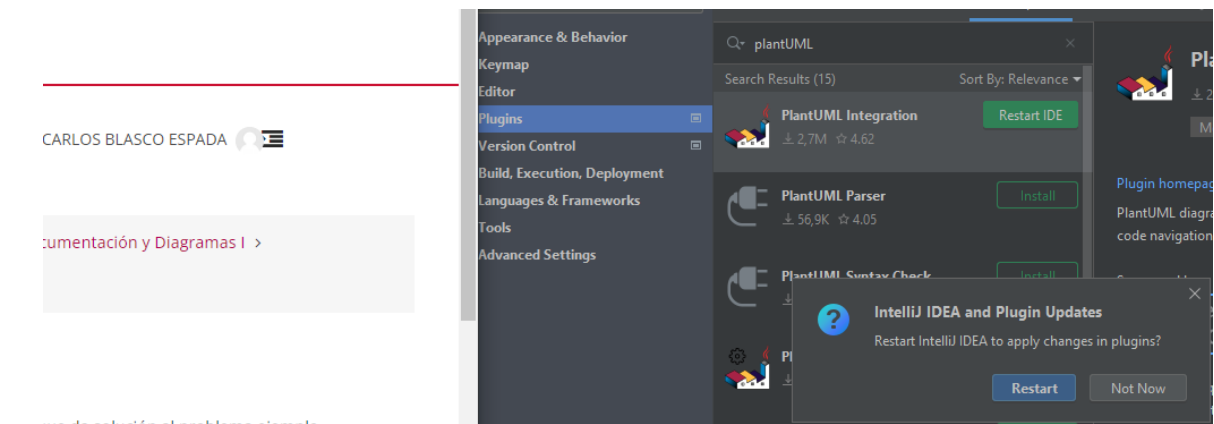
1. Instalaremos el plugin PlantUML. El primer paso es ir a File > Settings.



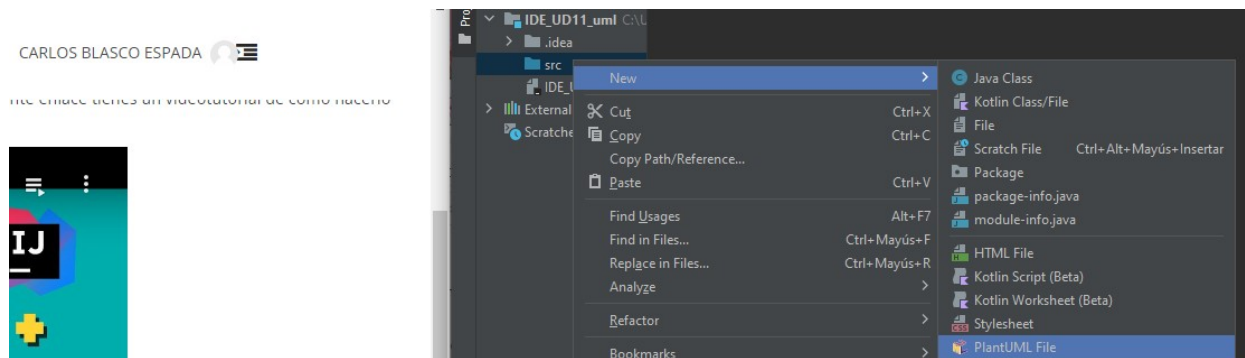
2. Hacemos clic en Plugins en el menú izquierdo y buscamos plantUML. Hacemos clic en "Install" en el plugin PlantUML Integration-



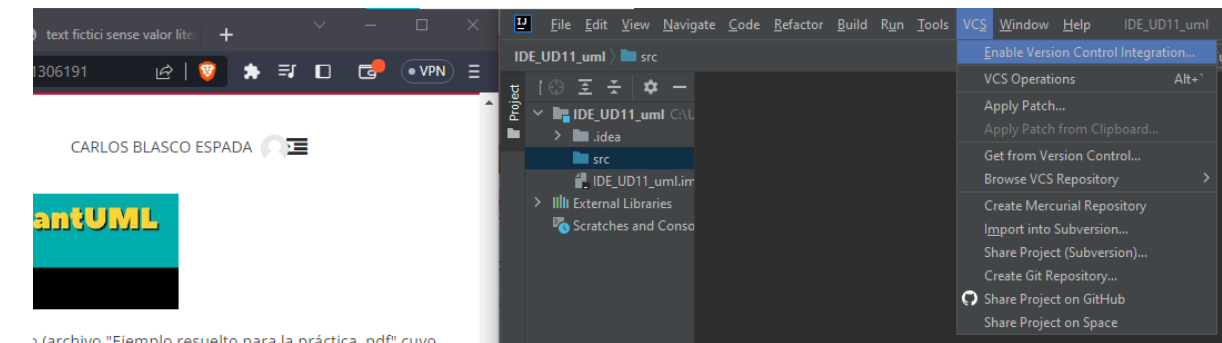
3. Para finalizar la instalación del plugin debemos reiniciar IntelliJ IDEA haciendo clic en "Restart IDE".



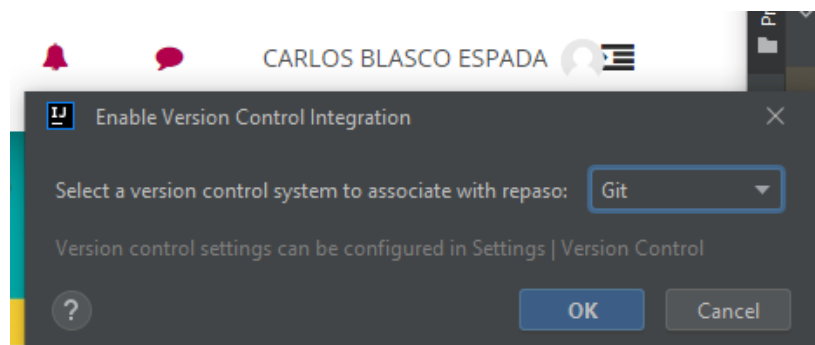
4. Tras el reinicio podremos crear un modelo UML, pero antes habilitaremos el VCS.



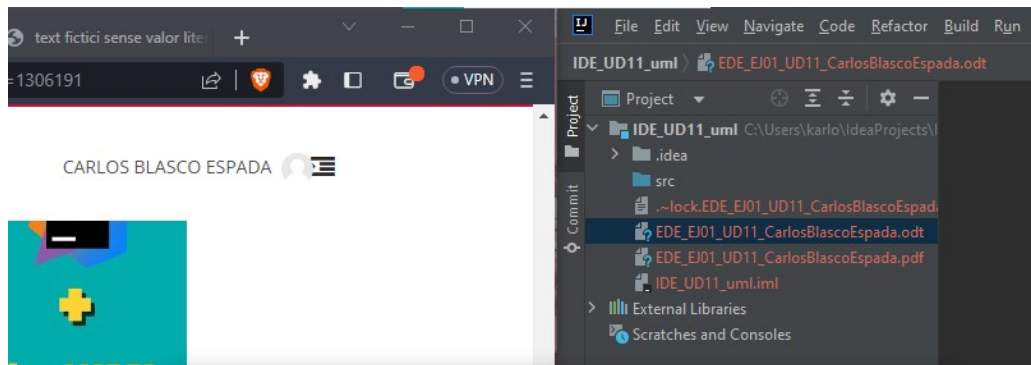
5. En la barra superior hacemos clic en VCS > Enable Version Control Integration.



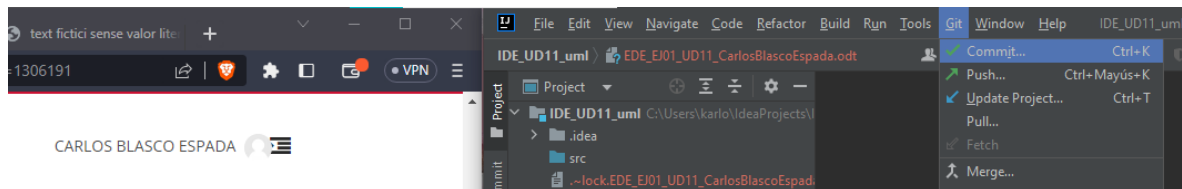
6. El VCS será Git. Hacemos clic en OK y ya estará activado.



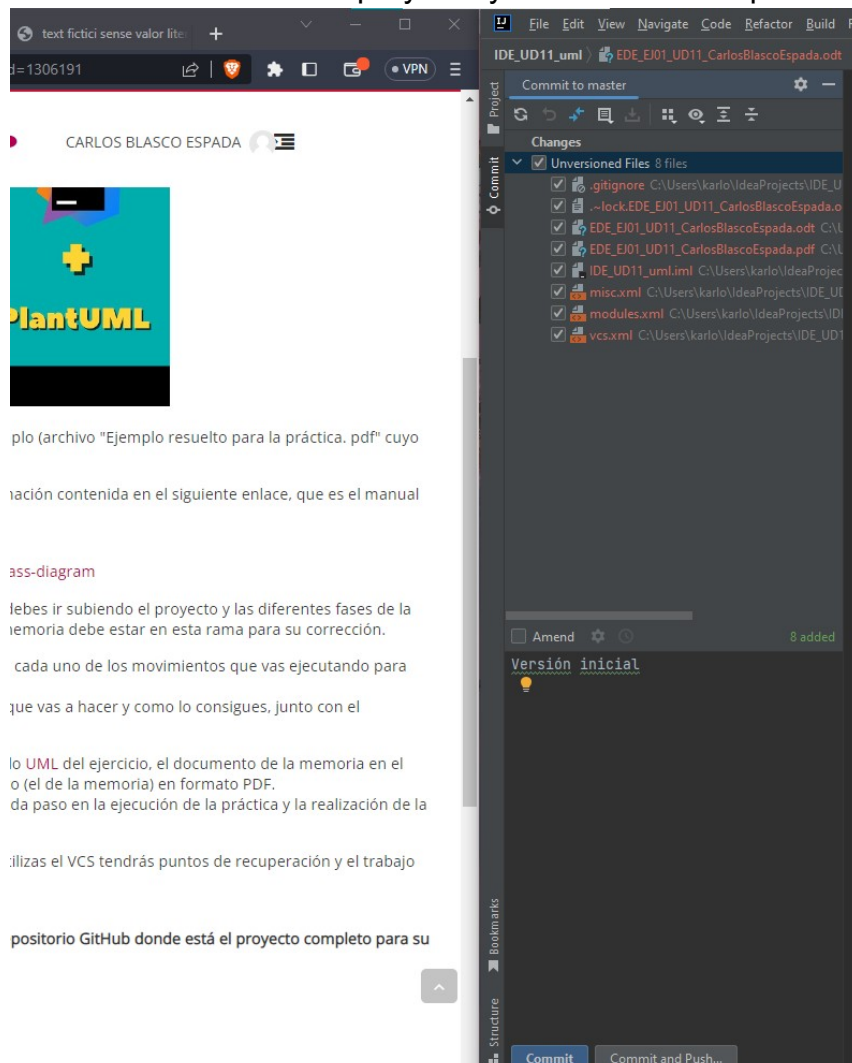
7. Añadimos el archivo .odt y su .pdf correspondiente al repositorio. Podemos hacer este paso arrastrando los archivos al proyecto o mediante el explorador de archivos.



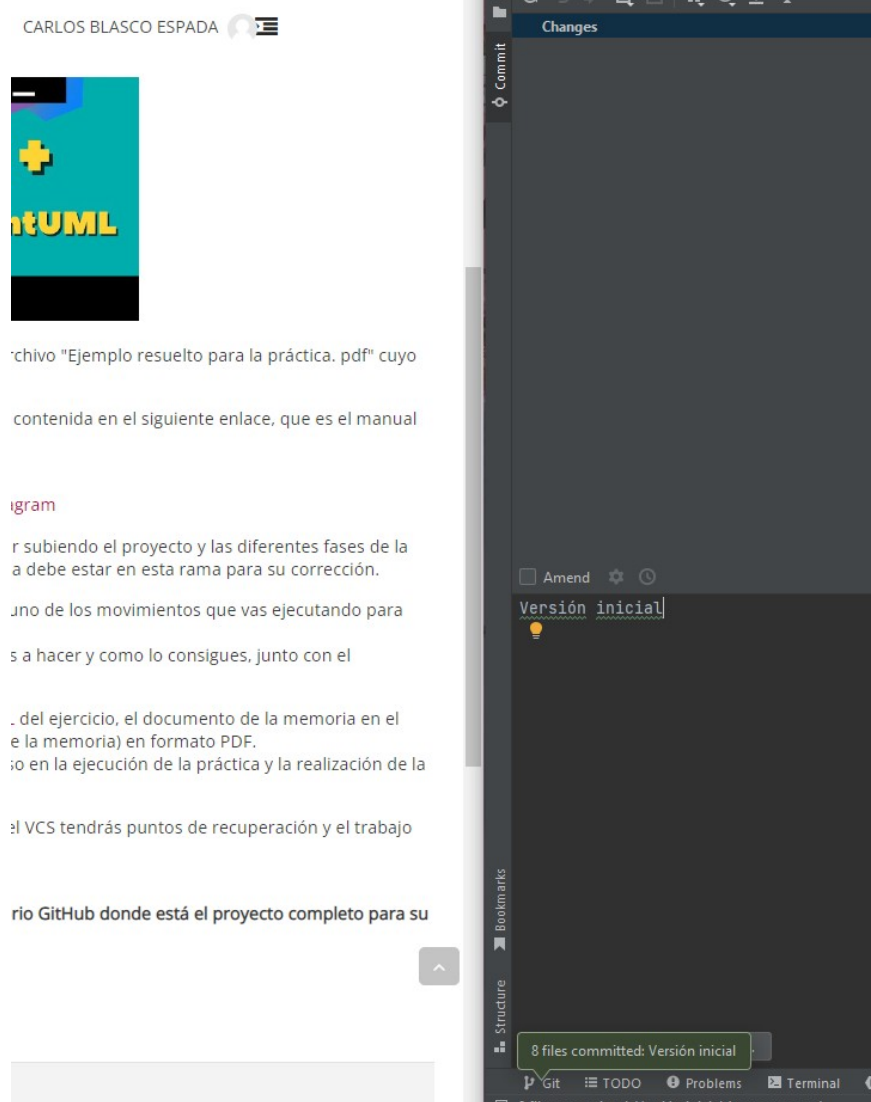
8. Realizamos un commit de los pasos realizados hasta ahora. Git > Commit.



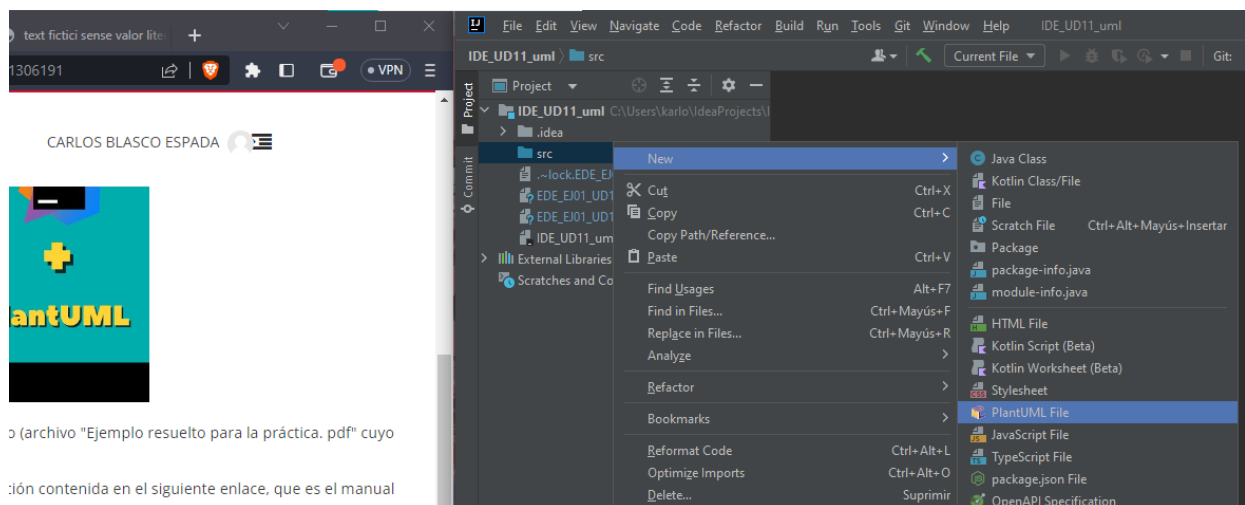
9. Añadimos todos los archivos del proyecto y una breve descripción.



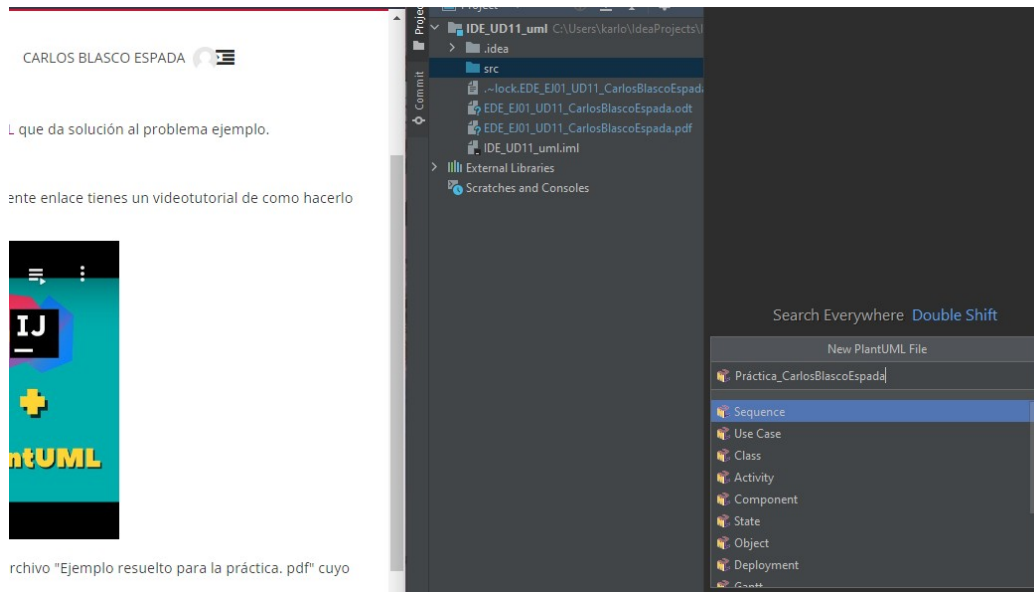
## 10. Pulsamos en Commit y ya tendremos nuestra primera versión del proyecto.



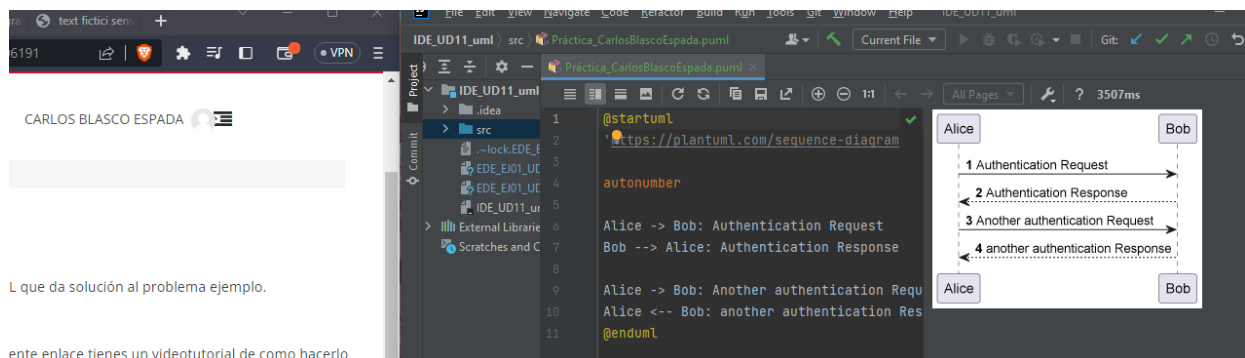
## 11. Creamos el modelo UML. Clic derecho sobre "src" > New > PlantUML File.



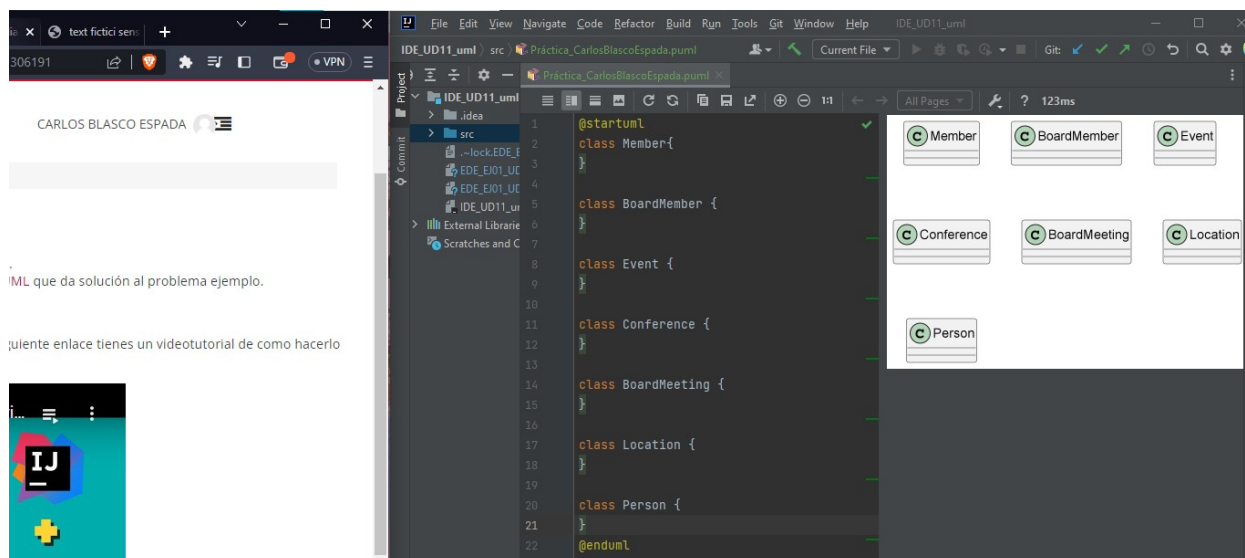
## 12. Establecemos un nombre para el archivo y pulsamos Enter.



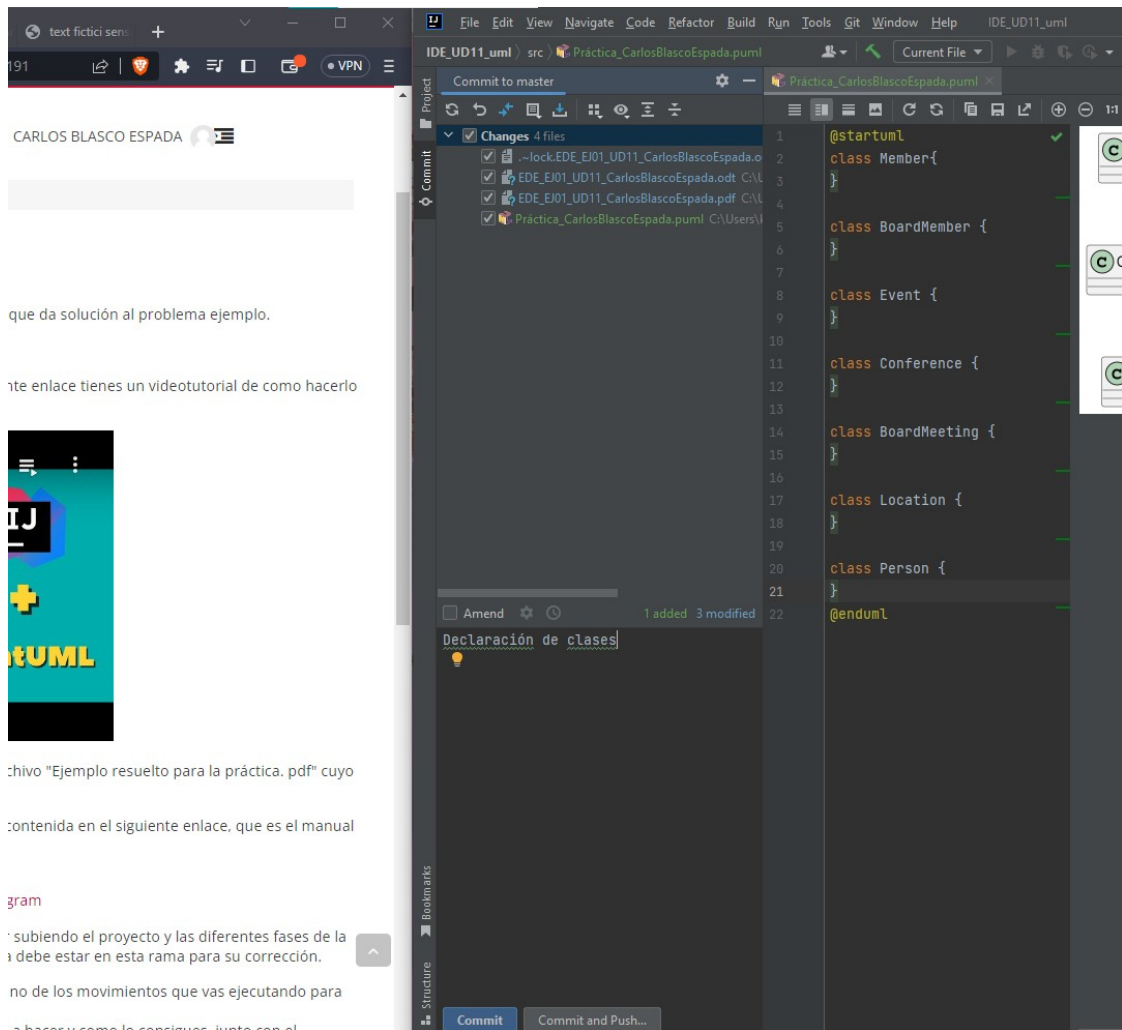
## 13. Tendremos un ejemplo por defecto.



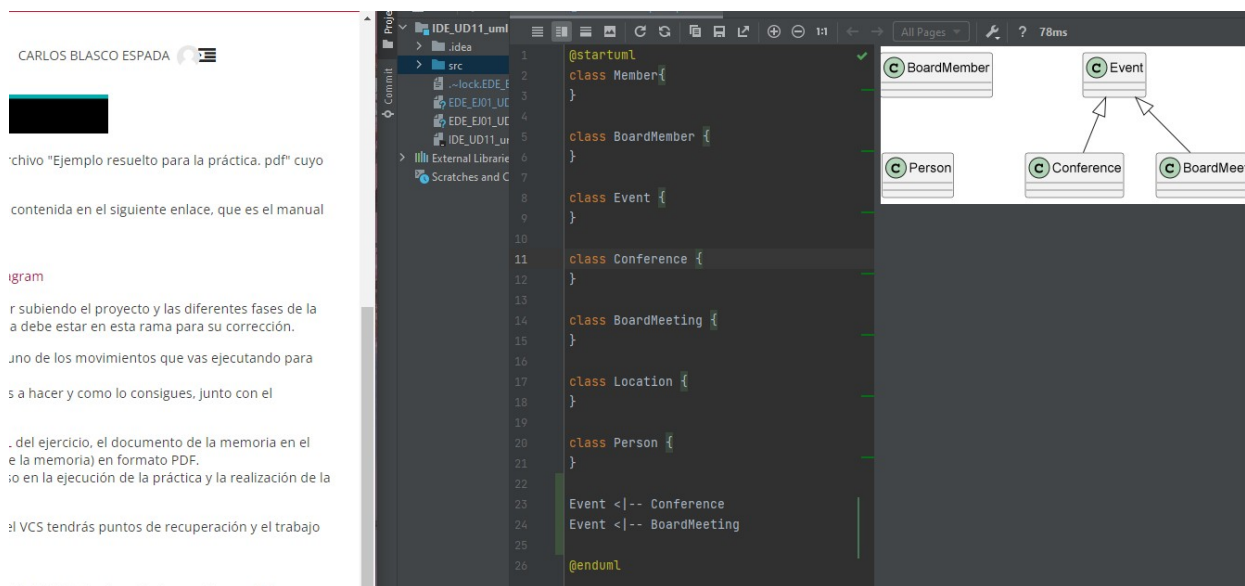
## 14. Borramos el ejemplo y hacemos la declaración de las clases necesarias, siguiendo la sintaxis de PlantUML.



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

Changes 3 files

✓ lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.o

✓ EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt C:\U

✓ Práctica\_CarlosBlascoEspada.puml C:\Users\I

Amend

3 modified

Añadimos jerarquía de evento

Commit

Commit and Push...

1 @startuml

2 class Member{

3 }

4

5 class BoardMember {

6 }

7

8 class Event {

9 }

10

11 class Conference {

12 }

13

14 class BoardMeeting {

15 }

16

17 class Location {

18 }

19

20 class Person {

21 }

22

23 Event <|-- Conference

24 Event <|-- BoardMeeting

25

26 @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

IDE\_UD11\_uml

src

lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.o

EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt

IDE\_UD11\_uml

External Libraries

Scratches and C

1 @startuml

2 class Member{

3 }

4

5 class BoardMember {

6 }

7

8 class Event {

9 }

10

11 class Conference {

12 }

13

14 class BoardMeeting {

15 }

16

17 class Location {

18 }

19

20 class Person {

21 }

22

23 Event <|-- Conference

24 Event <|-- BoardMeeting

25

26 Person <|-- Member

27 Member <|-- BoardMember

28

29 @enduml

Person

Member

BoardMember

Event

Conference

BoardMeeting

Location



## 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 'amas 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

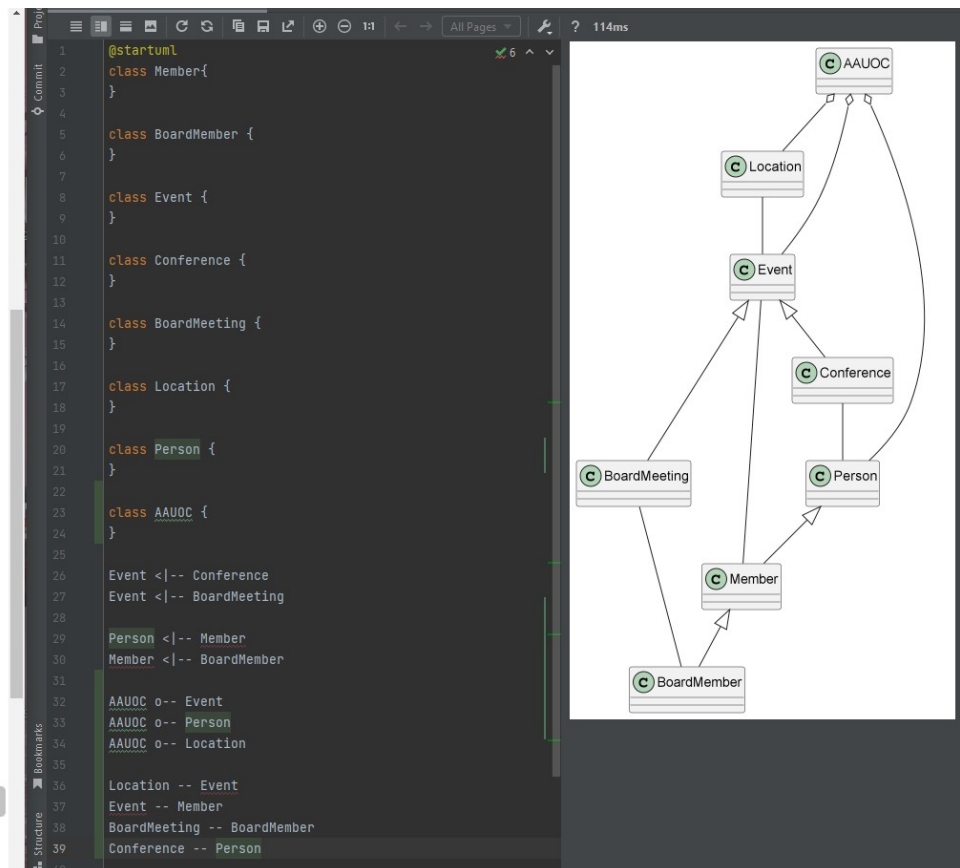
modelo UML del ejercicio, el documento  
noria y el mismo documento (el de la

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  
esta explicación.

información contenida en el siguiente  
mas de Clases.

gram

nde debes ir subiendo el proyecto y las  
amos la rama Master y la memoria

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

modelo UML del ejercicio, el documento  
ria 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 lists three files: `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 on the right shows the following classes and relationships:

```
@startuml
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. Incluimos los métodos y atributos correspondientes dentro de cada clase. También se añaden las etiquetas de las relaciones usando ":" seguido del texto de la etiqueta.

CARLOS BLASCO ESPADA



el ejemplo (archivo "Ejemplo resuelto de esta explicación.

información contenida en el siguiente ramas de Clases.

agram

londe debes ir subiendo el proyecto y las aremos la rama Master y la memoria

rios de cada uno de los movimientos que

de lo que vas a hacer y como lo o.

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

lizar cada paso en la ejecución de la

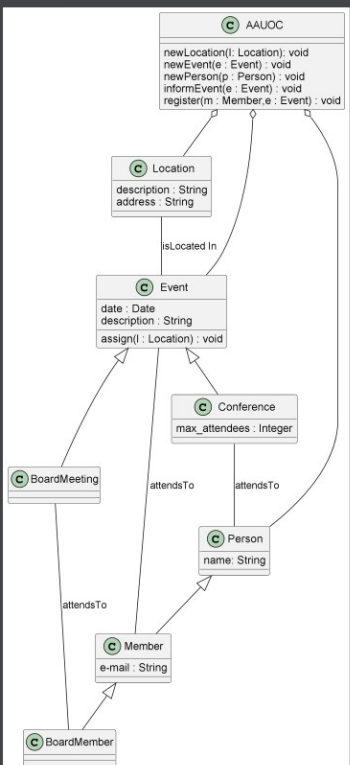
ar. Si utilizas el VCS tendrás puntos de

ce al repositorio GitHub donde está el


```

1 @startuml
2 class Member{
3     e-mail : String
4 }
5
6 class BoardMember {
7 }
8
9 class Event {
10     date : Date
11     description : String
12     assign(l : Location) : void
13 }
14
15 class Conference {
16     max_attendees : Integer
17 }
18
19 class BoardMeeting {
20 }
21
22 class Location {
23     description : String
24     address : String
25 }
26
27 class Person {
28     name: String
29 }
30
31 class AAUOC {
32     newLocation(I: Location): void
33     newEvent(e : Event) : void
34     newPerson(p : Person) : void
35     informEvent(e : Event) : void
36     register(m : Member,e : Event) : void
37 }
38
39 Event <|-- Conference
40 Event <|-- BoardMeeting
41
42 Person <|-- Member
43 Member <|-- BoardMember
44
45 AAUOC o-- Event
46 AAUOC o-- Person
47 AAUOC o-- Location
48
49 Location -- Event : isLocated In
50 Event -- Member : attendsTo
51 BoardMeeting -- BoardMember : attendsTo
52 Conference -- Person : attendsTo
53
54 @enduml

```



CARLOS BLASCO ESPADA



Clases.

ama UML.

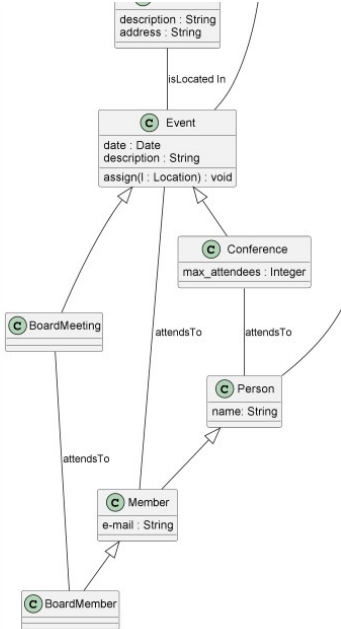
modelo UML que da solución al

v. En el siguiente enlace tienes un os de dos minutos.

```

28     name: String
29 }
30
31 class AAUOC {
32     newLocation(I: Location): void
33     newEvent(e : Event) : void
34     newPerson(p : Person) : void
35     informEvent(e : Event) : void
36     register(m : Member,e : Event) : void
37 }
38
39 Event <|-- Conference
40 Event <|-- BoardMeeting
41
42 Person <|-- Member
43 Member <|-- BoardMember
44
45 AAUOC o-- Event
46 AAUOC o-- Person
47 AAUOC o-- Location
48
49 Location -- Event : isLocated In
50 Event -- Member : attendsTo
51 BoardMeeting -- BoardMember : attendsTo
52 Conference -- Person : attendsTo
53
54 @enduml

```



## 23. Nuevamente realizamos un commit.

Conselleria d'Educació, Cultura  
i Esport

Val / Cas / Eng / Fra

Buscar

Salir

CARLOS BLASCO ESPADA

>

lases.

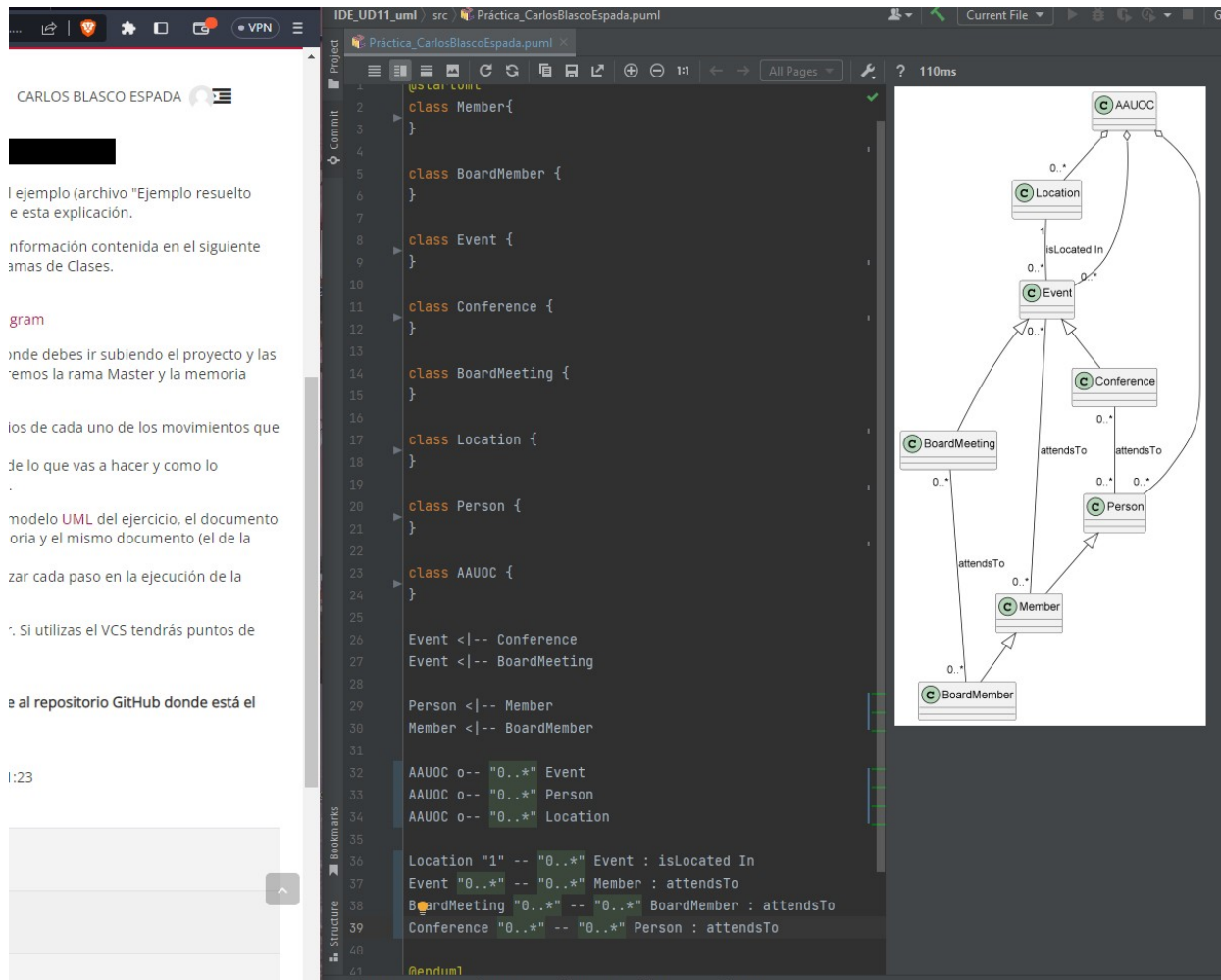
na UML.  
odelo UML que da solución al

En el siguiente enlace tienes un  
s de dos minutos.



```
Práctica_CarlosBlascoEspada.puml C:\Users\karlo\IdeaP
20 }
21
22 class Location {
23     description : String
24     address : String
25 }
26
27 class Person {
28     name: String
29 }
30
31 class AAUOC {
32     newLocation(I: Location): void
33     newEvent(e : Event) : void
34     newPerson(p : Person) : void
35     informEvent(e : Event) : void
36     register(m : Member,e : Event) :
37 }
38
39 Event <|-- Conference
40 Event <|-- BoardMeeting
41
42 Person <|-- Member
43 Member <|-- BoardMember
44
45 AAUOC o-- Event
46 AAUOC o-- Person
47 AAUOC o-- Location
48
49 Location -- Event : isLocated In
50 Event -- Member : attendsTo
51 BoardMeeting -- BoardMember : attend
52 Conference -- Person : attendsTo
53
54 @enduml
```


24. Para el último punto de PlantUML borramos los métodos y atributos y añadimos la cardinalidad de las relaciones. Pondremos "0..\*" a cada lado de la relación ya que no se especifica el tipo de relación o acceso (excepto de Location el cual es "1").





## 25. Realizamos un último commit de los cambios.

CARLOS BLASCO ESPADA



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

información contenida en el siguiente diagrama de Clases.

Diagrama

Entonces debes ir subiendo el proyecto y las clases a la rama Master y la memoria local.

Después de cada uno de los movimientos que hagas, lo que vas a hacer y como lo vas a hacer.

El modelo UML del ejercicio, el documento de la explicación y el mismo documento (el de la explicación) se van a ir ejecutando en la ejecución de la explicación.

Si utilizas el VCS tendrás puntos de control en el repositorio GitHub donde está el proyecto.

Commit to master

Changes 3 files

- lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt# C:\Users\karlo\OneDrive\lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt
- EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt C:\Users\karlo\OneDrive\lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt
- Práctica\_CarlosBlascoEspada.puml C:\Users\karlo\OneDrive\lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.puml

Unversioned Files 2 files

Amend 2 added 3 modified

Borramos los métodos y atributos, y añadimos cardinalidad

```
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-- "0..*" Event
AAUOC o-- "0..*" Person
AAUOC o-- "0..*" Location

Location "1" -- "0..*" Event : isLoc
Event "0..*" -- "0..*" Member : atte
BoardMeeting "0..*" -- "0..*" BoardM
Conference "0..*" -- "0..*" Person :
```

## 26. Añadimos enlace GitHub en odt/pdf y realizamos commit.

CARLOS BLASCO ESPADA



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

información contenida en el siguiente diagrama de Clases.

Diagrama

Entonces debes ir subiendo el proyecto y las clases a la rama Master y la memoria local.

Después de cada uno de los movimientos que hagas, lo que vas a hacer y como lo vas a hacer.

El modelo UML del ejercicio, el documento de la explicación y el mismo documento (el de la explicación) se van a ir ejecutando en la ejecución de la explicación.

Commit to master

Changes 3 files

- lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt# C:\Users\karlo\OneDrive\lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt
- EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt C:\Users\karlo\OneDrive\lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.odt
- EDE\_EJ01\_UD11\_CarlosBlascoEspada.pdf C:\Users\karlo\OneDrive\lock.EDE\_EJ01\_UD11\_CarlosBlascoEspada.pdf

Amend 3 modified

Incluimos enlace GitHub