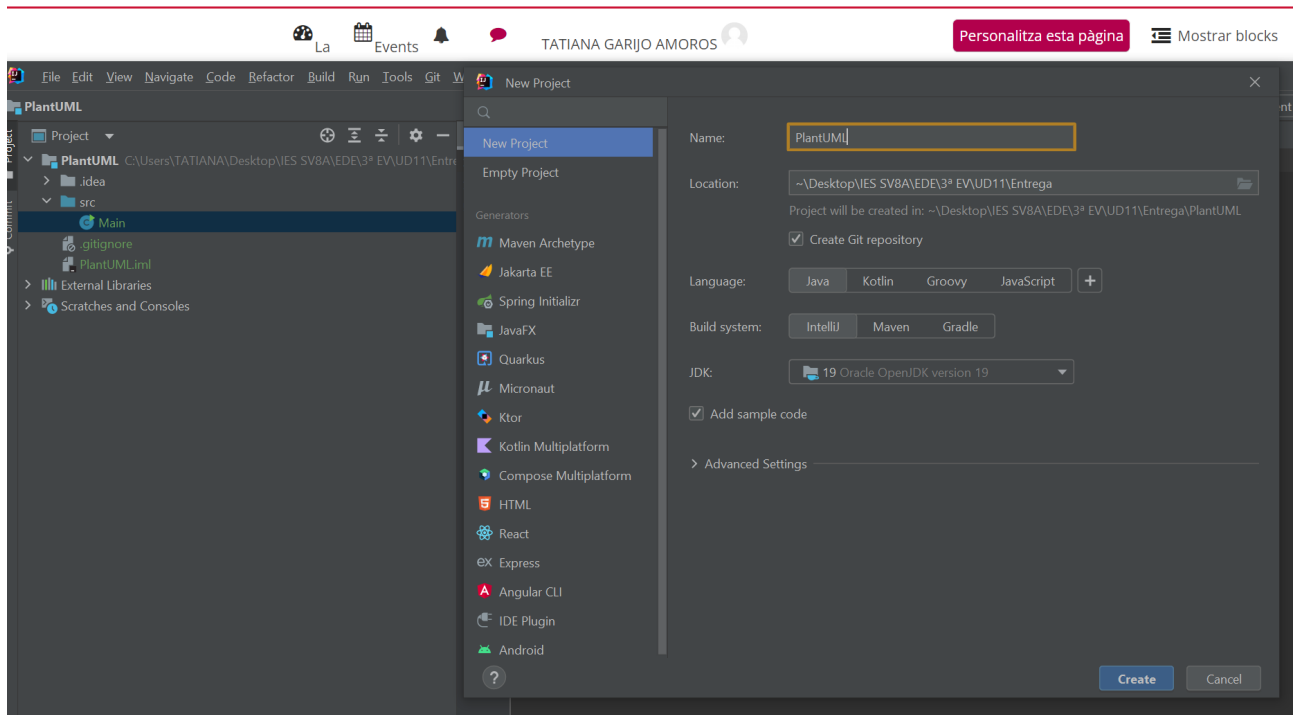


PRÁCTICA Diagramas UML Clases de uso II

Repositorio GitHub

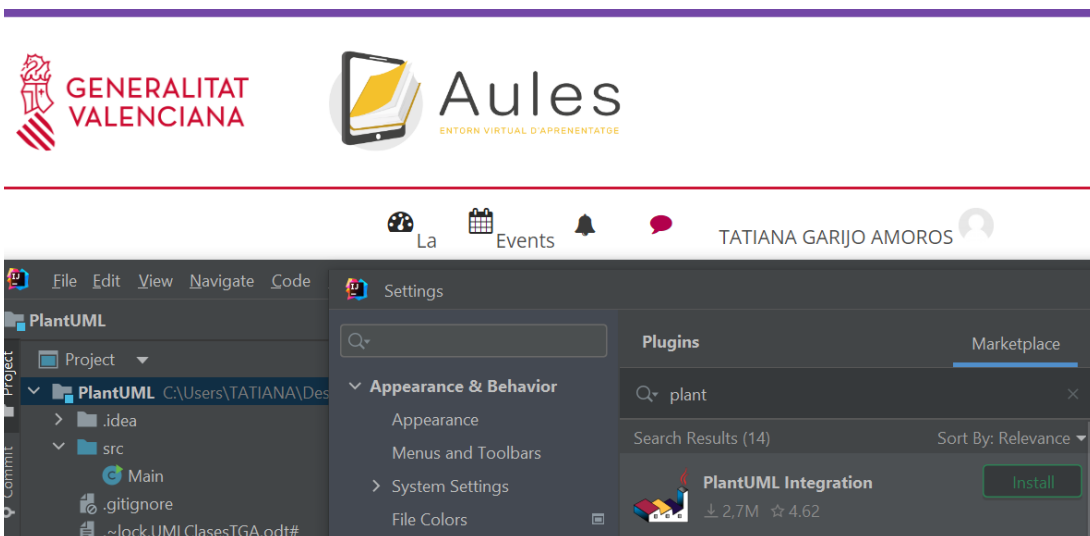
<https://github.com/tatianagarijo/UMLCasosUsoTatianaGarijo.git>



- New Project IntelliJ
- Abrir IntelliJ
- Seleccionar la opción New Project
- Configurar las características del proyecto
 - En mi caso he seleccionado
 - Nombre del proyecto
 - Lenguaje: Java
 - Build System: IntelliJ
 - version 19 JDK
 - Create Git repository

• PlantUML

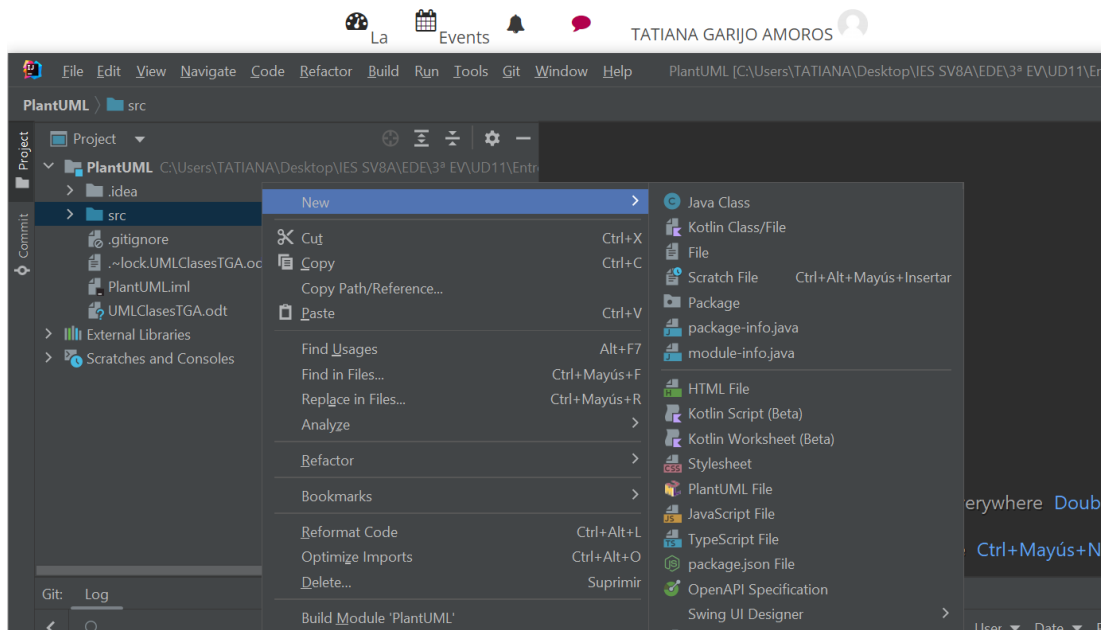
- PlantUML es una herramienta que nos permite crear diagramas tipo UML.
- Instalamos el plugin PlantUML en IntelliJ.
 - En la web oficial <https://plantuml.com/es/> podemos encontrar toda la información del plugin.
 - Navegamos por el menú File --> Settings --> Plugins
 - Buscamos el plugin PlantUML haciendo uso del buscador, una vez localizado hacemos clic sobre el botón **install**.



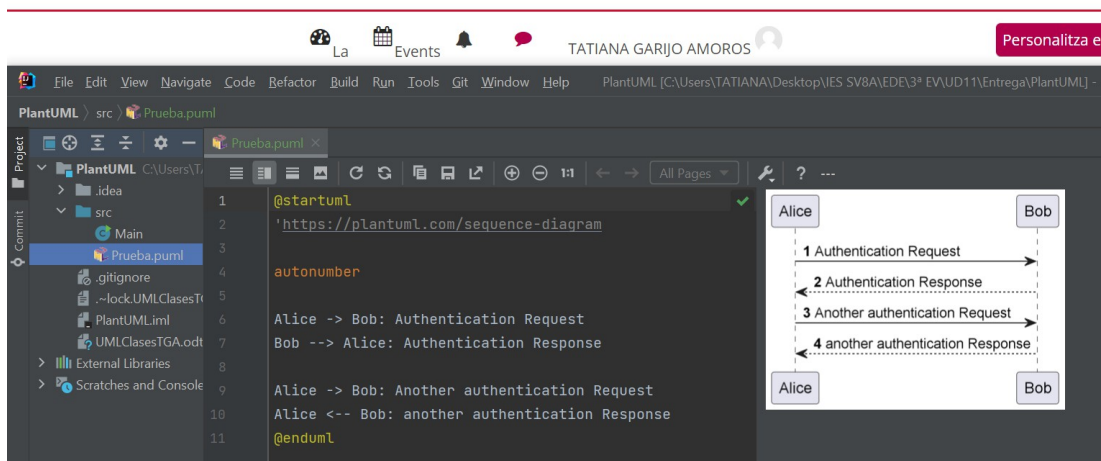
- El IDE nos pide reiniciar para cargar el plugin, hacemos clic sobre el botón **Restart IDE**



- Hacemos clic botón derecho sobre la carpeta src → new → PlantUML File → Use Case



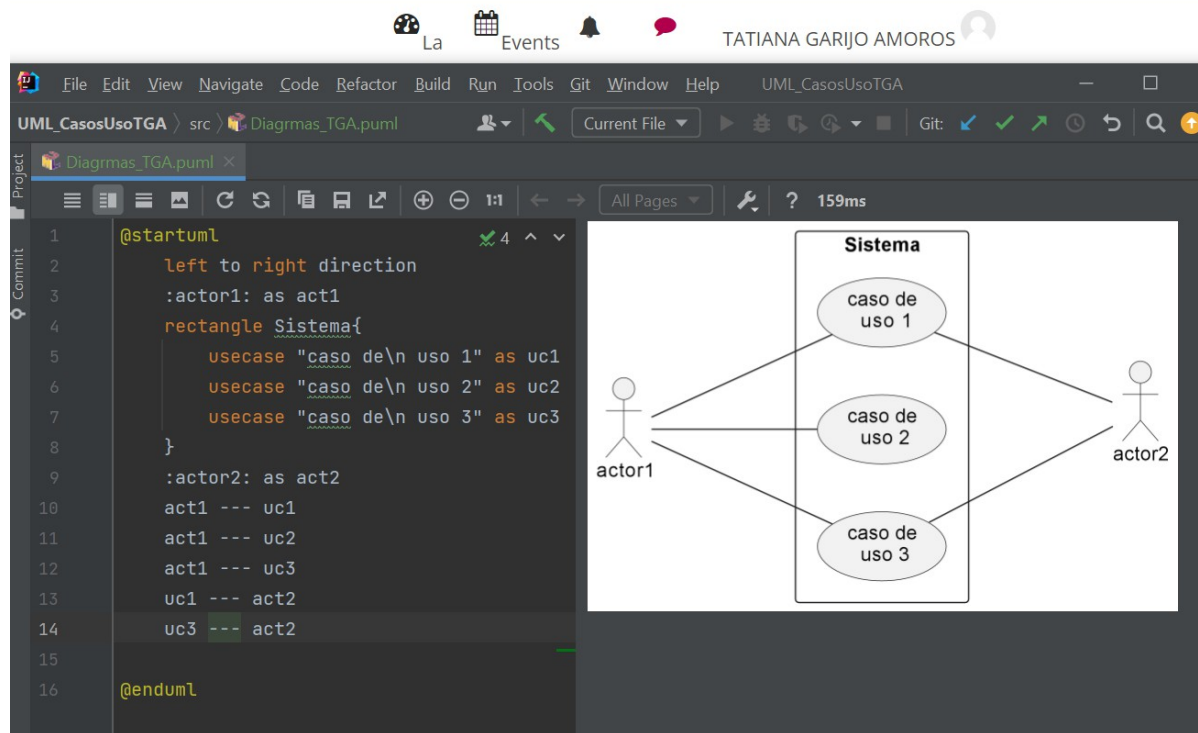
- Le damos el nombre de nuestra elección y nos aparece la siguiente ventana



- En el panel central entre las instrucciones `@startuml` `@enduml` iremos escribiendo el código necesario para crear el diagrama, en el panel derecho irá apareciendo el diagrama según el código que vayamos generando.
- Para crear los diagramas hacemos uso de la información que nos facilita la web de PlantUML (<https://plantuml.com/es/class-diagram>), copiamos los fragmentos de código necesarios para crear el diagrama requerido.

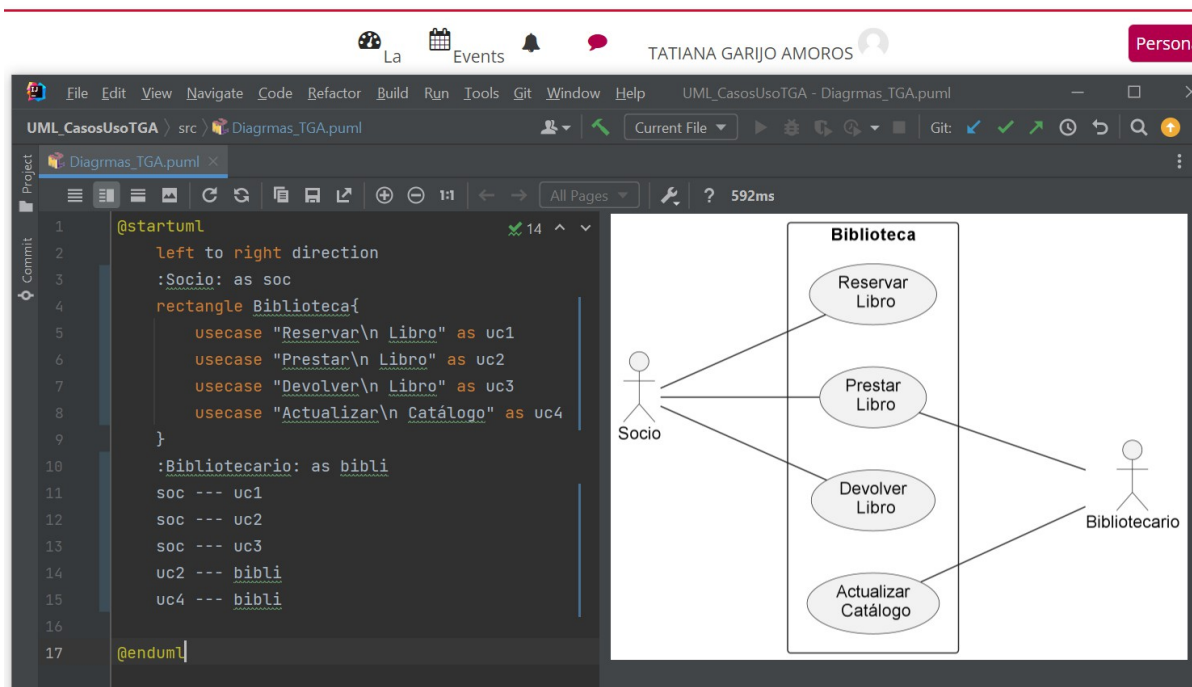
- Replicando las fases del ejemplo resuelto

- Ejemplo 1:



- left to right direction → se usa para darle al diagrama la direccionalidad que deseamos
- :actor: → es la sintaxis para nombrar los actores
- rectangle nombreSistema → se utiliza para empaquetar elementos del sistema
- usecase nombreCasodeUso → sintaxis para representar casos de uso
- as → es la sintaxis para crear un alias del elemento que le precede
- se utilizan las flechas para representar la comunicación normal, inclusión, extensión, etc
 - ..
 - --
 - -->
 - --|>
 - El sentido de las flechas se puede invertir, el trazo se puede hacer más largo añadiendo – o .

- Ejemplo 2:



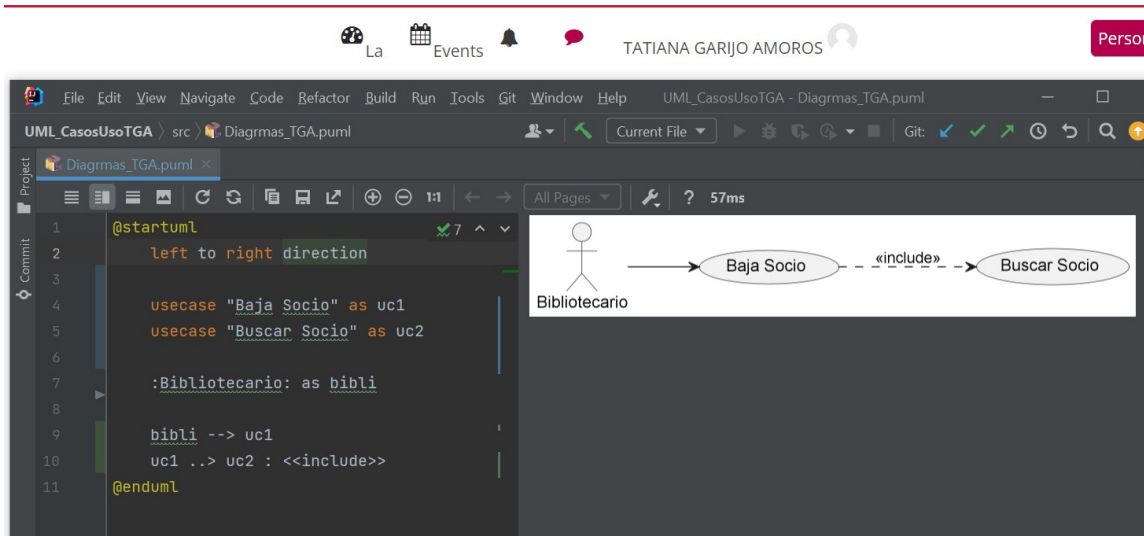
- Plantilla:

Casos de uso	
Actores	
Resumen	
Precondiciones	
Postcondiciones	
Incluye	
Extiende	
Hereda de	
Flujo de eventos	
Actor	Sistema

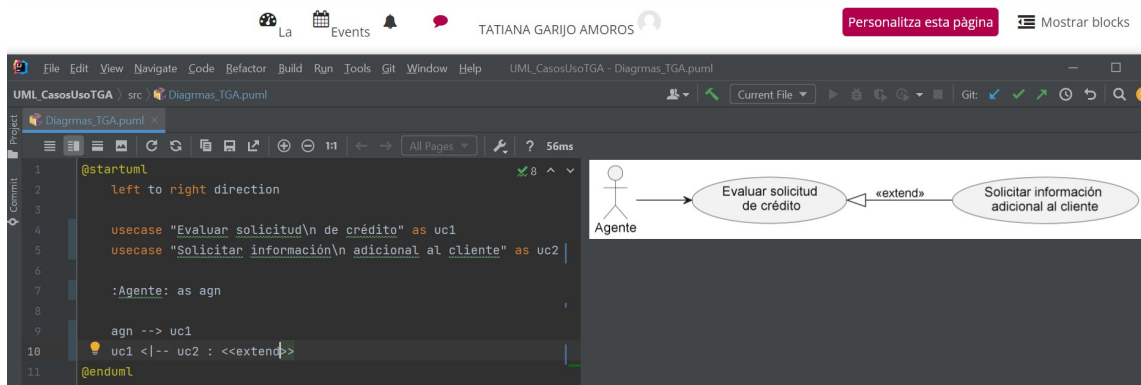
Casos de uso	Reservar libro
Actores	Socio
Resumen	El socio puede solicitar la reserva de un

	libro...
Precondiciones	El socio no tiene ninguna reserva
Postcondiciones	El socio tiene una reserva y el libro tiene una nueva reserva a partir de una fecha
Incluye	..
Extiende	..
Hereda de	..
Flujo de eventos	
Actor	Sistema
1.El socio solicita la reserba (código, libro..)	2, El sistema comprueba ... 3. El sistema comprueba ... 4. El sistema solicita ...
5,El socio confirma la reserva	6. El sistema realiza la reserva

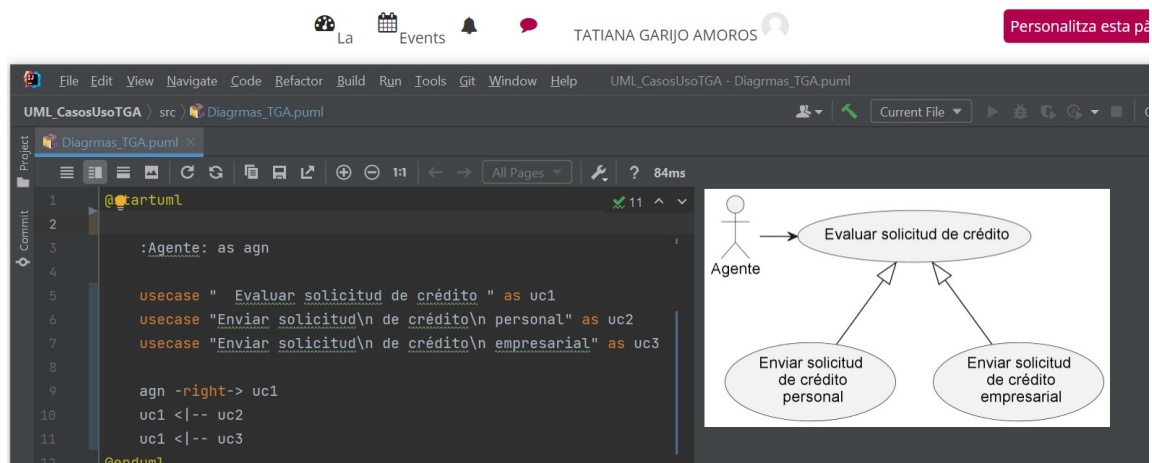
- Ejemplo3:
 - Inclusión → sintaxis .> <.



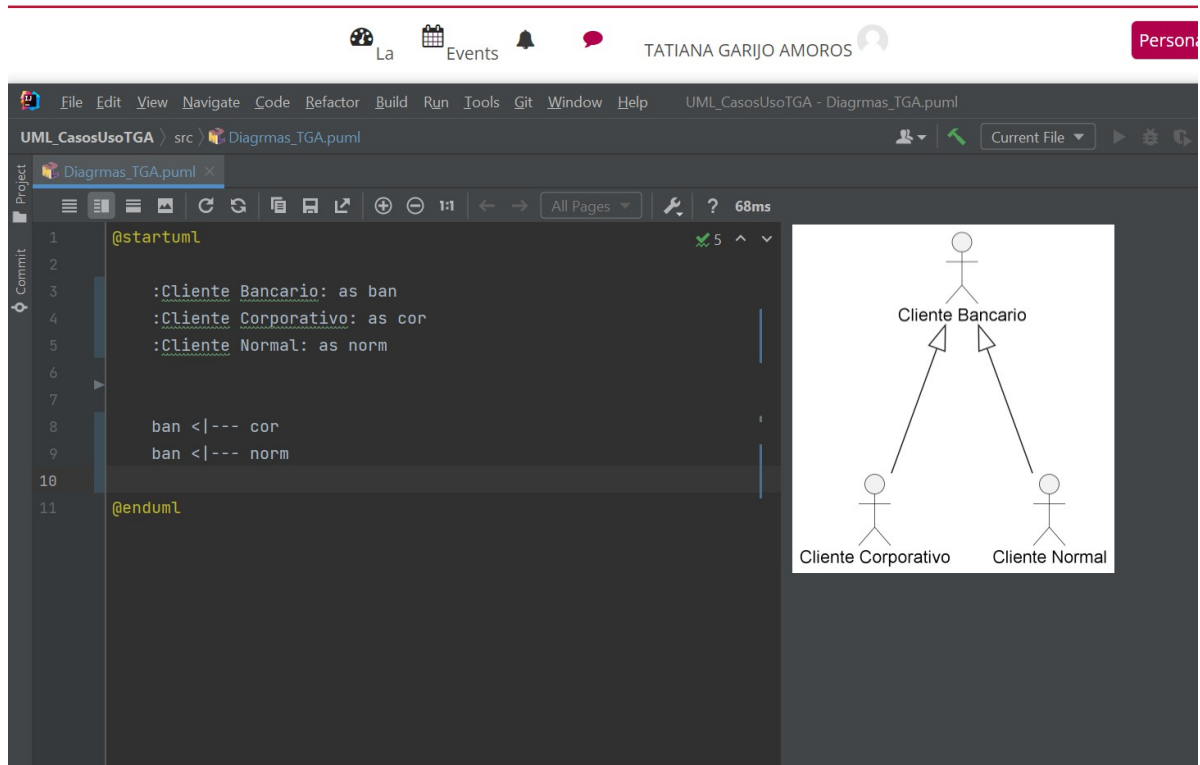
- Ejemplo4:
 - Extensión → sintaxis --|> <|--
 - ** aunque por el vídeo se podría interpretar a representación de las flechas ..> <..



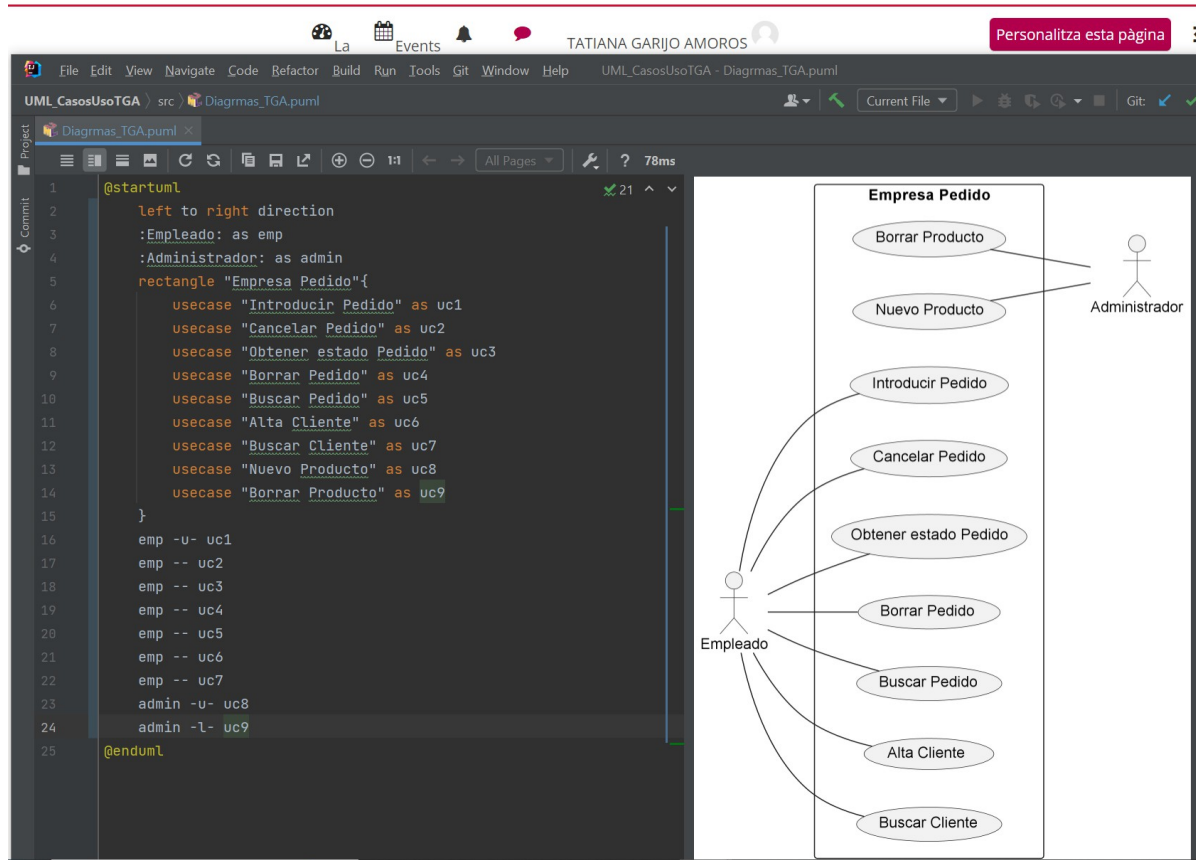
- Ejemplo5:
 - Herencia → sintaxis --|> <|-- (representación según el vídeo)



- Ejemplo6:
 - Herencia (actores)

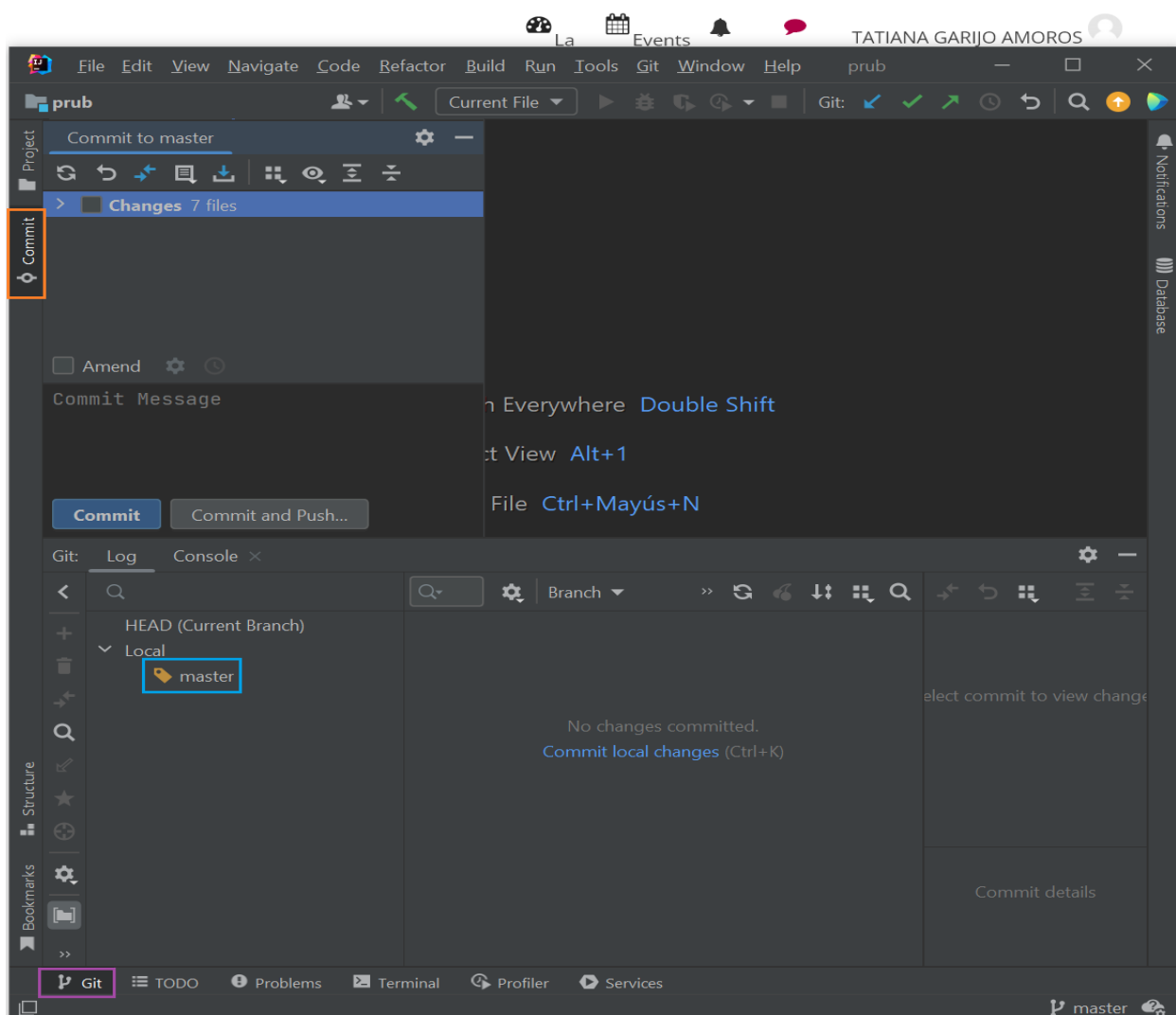


- Ejemplo7:

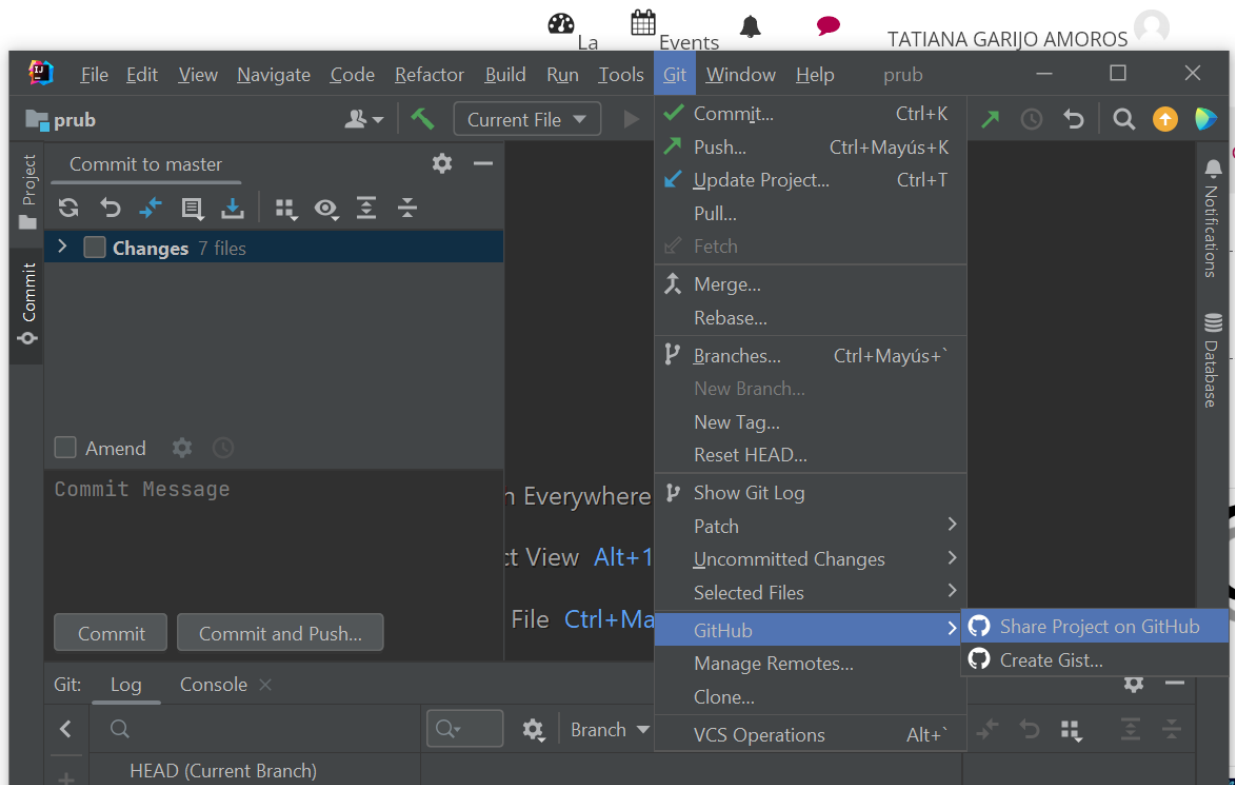


- VSC

- Al crear el proyecto he seleccionado la opción de crear un repositorio Git
- Durante el proceso realizar commit, haciendo uso del **panel lateral izquierdo 'commit'**
 - Seleccionar los archivos modificados y los que se vayan a incluir en el commit
 - Añadir mensaje del commit
 - Hacer clic sobre el botón commit
-



- Subir repositorio a Github



- Hacer clic sobre el menú superior 'Git' → GitHub → Share Project on GitHub

- Dar nombre al repositorio y hacer clic sobre el botón 'Share'

