

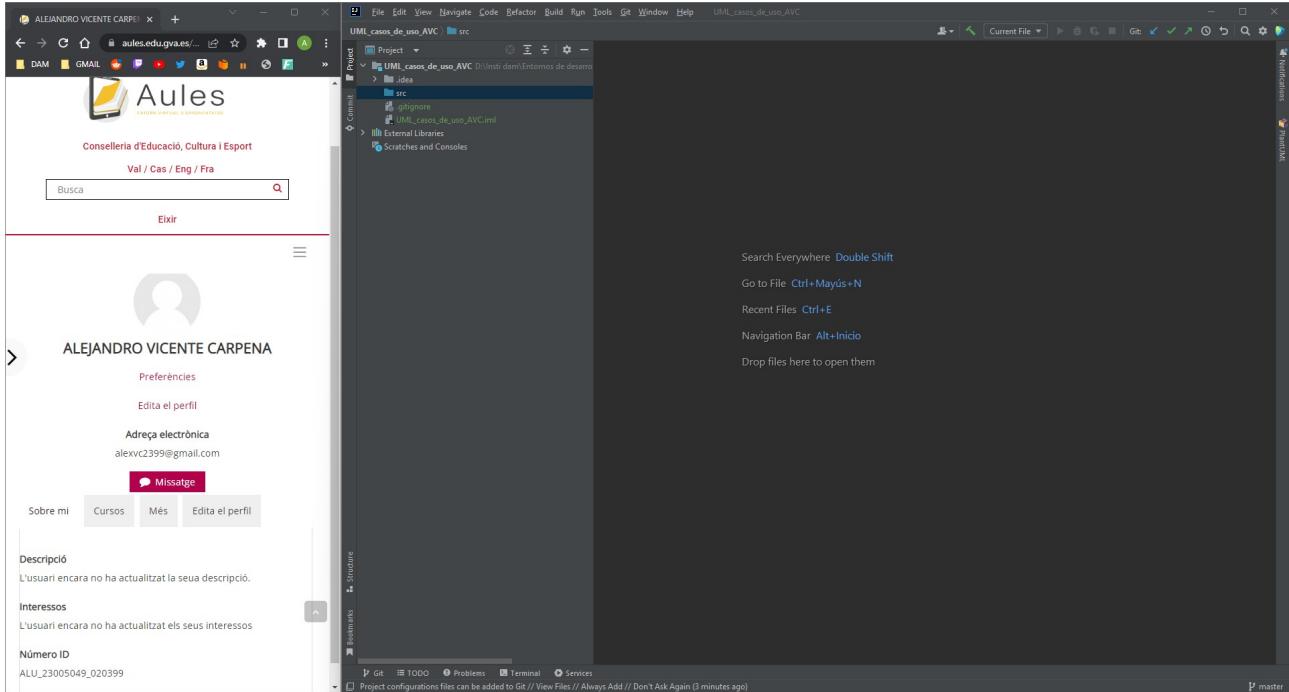
Memoria Clases de uso con PlantUML

Alejandro Vicente Carpena

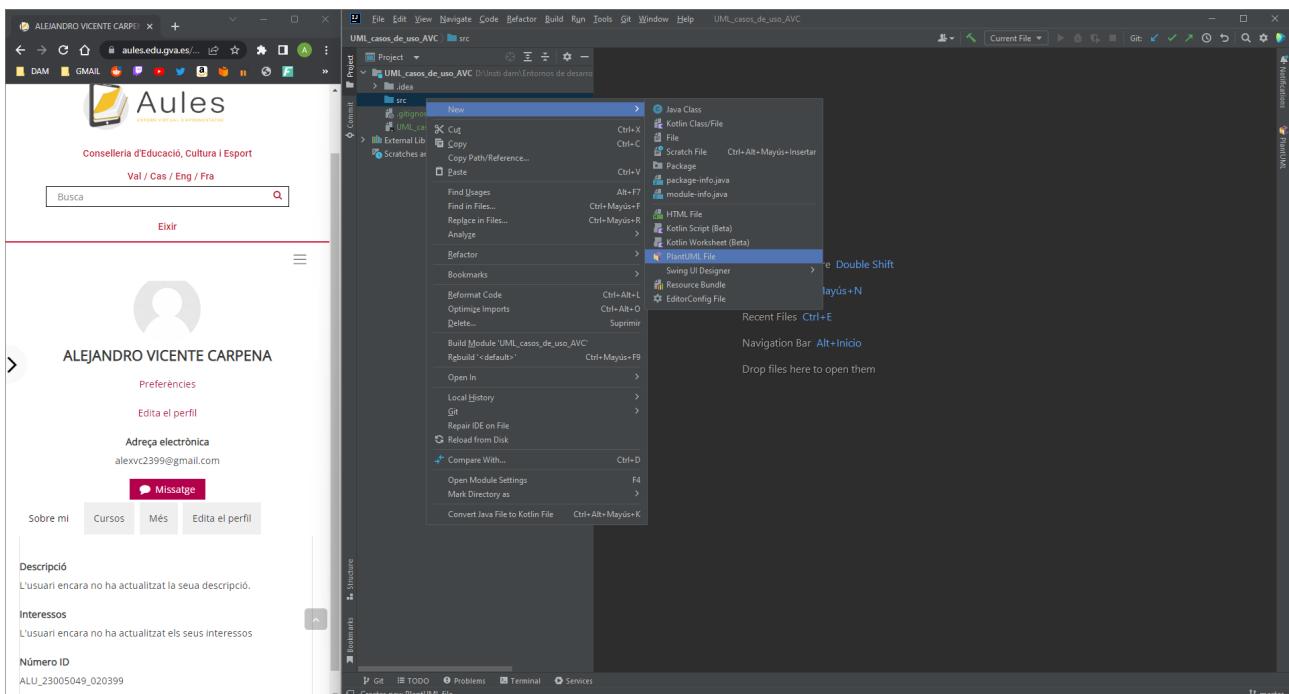
ENLACE AL REPOSITORIO:

https://github.com/alexvc991/1--UML_clases_de_uso_AVC/tree/Memoria

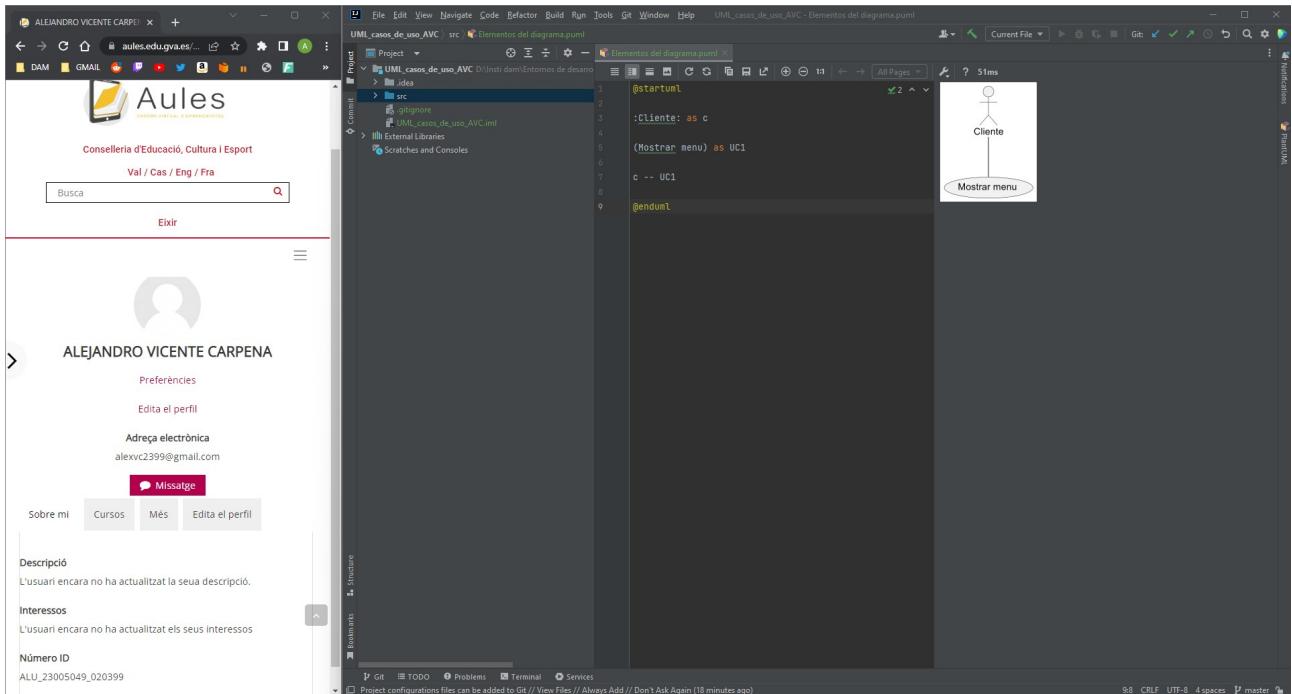
El primer paso de la practica seria crear un nuevo proyecto de java en intelliJ y crear un repositorio del proyecto



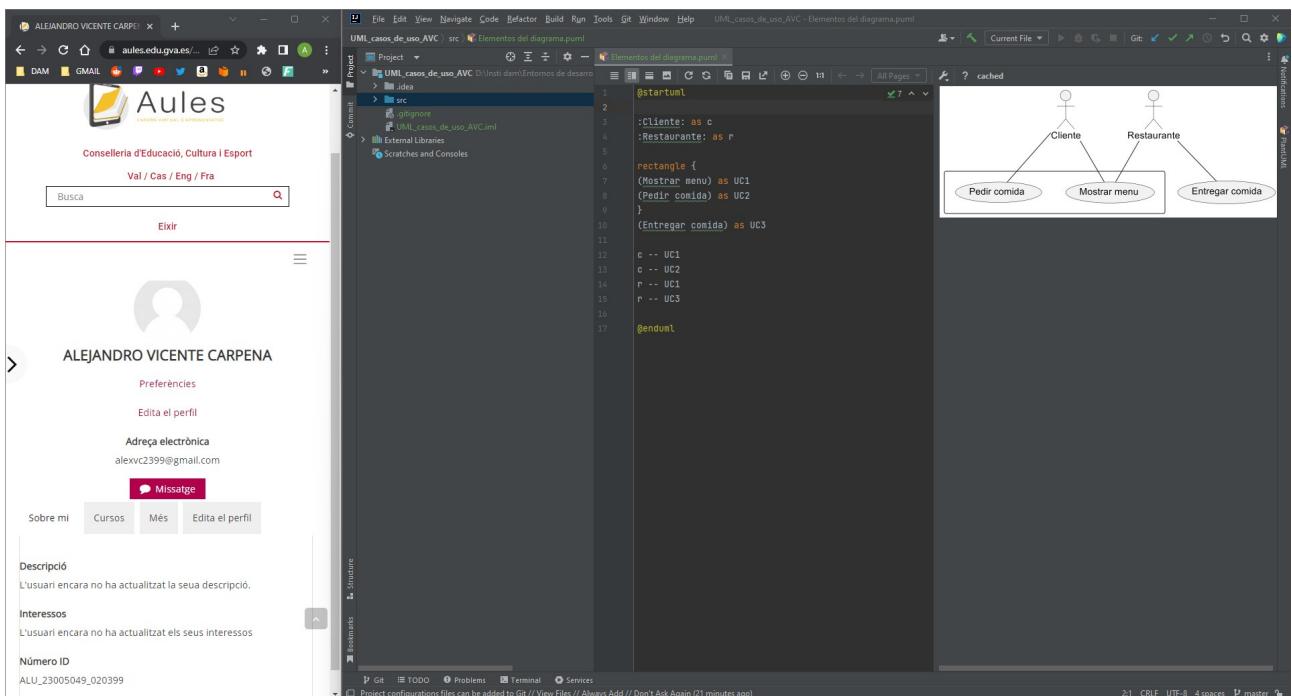
Una vez tenemos el proyecto, creamos un archivo plantuml



Este seria el resultado del primer diagrama.



Como el segundo diagrama simplemente amplia al primero, he pensado mantenerlos como si fuese uno. Así quedaría el primer diagrama actualizado



Y realizo el primer commit

The screenshot shows a dual-pane interface. On the left is a web browser displaying a user profile for 'ALEJANDRO VICENTE CARPENA'. On the right is an IDE window titled 'UML_casos_de_uso_AVC' showing a UML class diagram and its Puml code.

UML Class Diagram:

```
graph TD; Cliente --> Pedir_comida; Cliente --> Mostrar_menu; Restaurante --> Entregar_comida;
```

Puml Code:

```
@startuml  
:Cliente: as c  
:Restaurante: as r  
  
rectangle {  
    (Mostrar menu) as UC1  
    (Pedir comida) as UC2  
}  
(Entregar comida) as UC3  
  
c -- UC1  
c -- UC2  
r -- UC1  
r -- UC3  
  
@endum
```

Despues de haber realizado el primer commit, he visto una linea interesante (left to right direction) que nos dejaria el diagrama mas parecido al diagrama del ejemplo

The screenshot shows a dual-pane interface. On the left is a web browser displaying a user profile for 'ALEJANDRO VICENTE CARPENA'. On the right is an IDE window titled 'UML_casos_de_uso_AVC' showing a UML class diagram and its Puml code. A red box highlights the 'left to right direction' annotation in the code.

UML Class Diagram:

```
graph LR; Cliente --> Pedir_comida; Cliente --> Mostrar_menu; Restaurante --> Entregar_comida;
```

Puml Code:

```
@startuml  
left to right direction  
:Cliente: as c  
:Restaurante: as r  
  
rectangle {  
    (Mostrar menu) as UC1  
    (Pedir comida) as UC2  
}  
(Entregar comida) as UC3  
  
c -- UC1  
c -- UC2  
r -- UC1  
r -- UC3  
  
@endum
```

El resultado del segundo diagrama seria el siguiente

The screenshot shows a dual-pane interface. On the left is a web browser window displaying a user profile for 'ALEJANDRO VICENTE CARPENA'. The profile includes a photo, name, email (alexvc2399@gmail.com), and a 'Message' button. Below the profile are tabs for 'Sobre mi', 'Cursos', 'Més', and 'Edita el perfil'. On the right is an IDE window titled 'UML_casos_de_uso_AVC - Segundo diagrama.puml'. It displays a UML class diagram with a 'Cliente' actor at the top. Three classes are shown: 'Confirmar pedido' (UC1), 'Cancelar pedido' (UC2), and 'Devolver cambio' (UC3). UC1 and UC2 have dashed arrows pointing to UC3, labeled 'include'. A note in the code indicates 'left to right direction'.

```
graph TD; Actor[Cliente] --> UC1[Confirmar pedido]; Actor --> UC2[Cancelar pedido]; UC1 -- include --> UC3[Devolver cambio]; UC2 -- include --> UC3;
```

```
@startuml
left to right direction
Cliente as c
UC1 as UC1
UC2 as UC2
UC3 as UC3
UC1 ..> UC3 :<<include>>
UC2 ..> UC3 :<<include>>
@enduml
```

Así quedaría el tercer diagrama

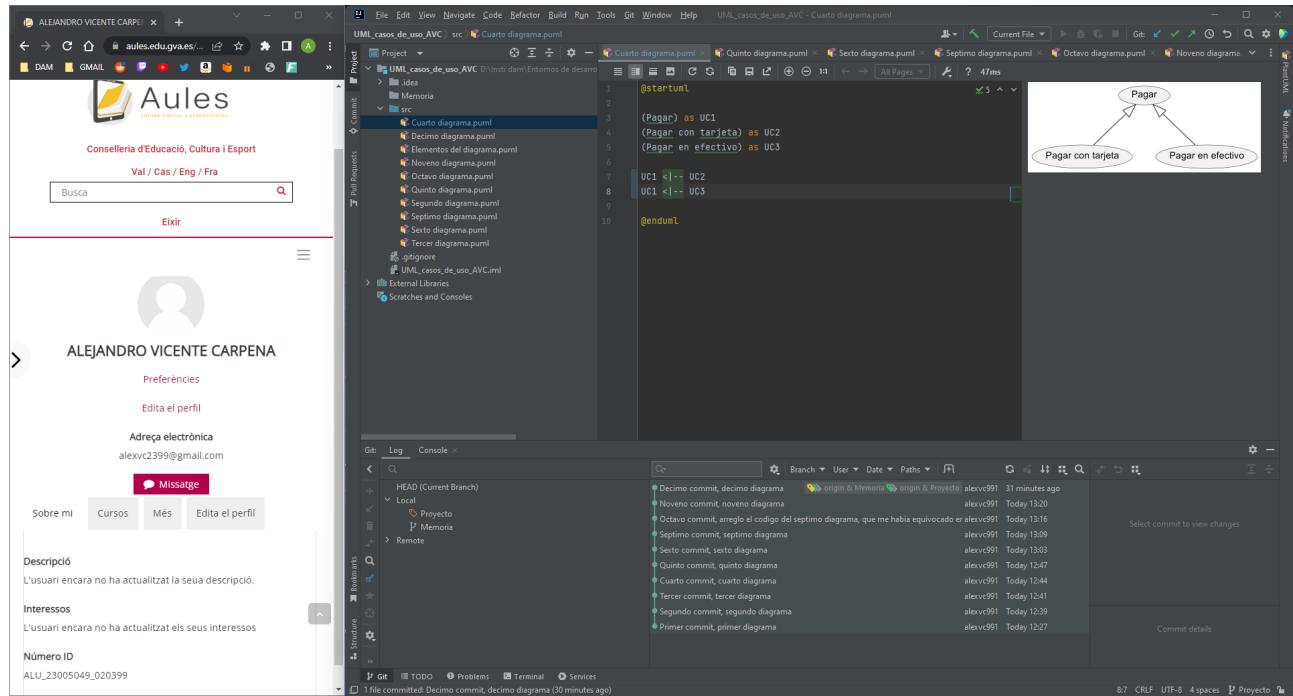
The screenshot shows a dual-pane interface. On the left is a web browser window displaying a user profile for 'ALEJANDRO VICENTE CARPENA'. The profile includes a photo, name, email (alexvc2399@gmail.com), and a 'Message' button. Below the profile are tabs for 'Sobre mi', 'Cursos', 'Més', and 'Edita el perfil'. On the right is an IDE window titled 'UML_casos_de_uso_AVC - Tercer diagrama.puml'. It displays a UML class diagram with two classes: 'Acceder' (UC1) and 'Registrarse' (UC2). UC1 has a dashed arrow pointing to UC2, labeled 'extends'. A note in the code indicates 'left to right direction'.

```
graph TD; UC1[Acceder] -- extends --> UC2[Registrarse];
```

```
@startuml
left to right direction
UC1 as UC1
UC2 as UC2
UC1 ..> UC2 :<<extend>>
@enduml
```

Este seria el resultado del cuarto diagrama.

En este me habia equivocado en la practica, ya que no me habia fijado que la relacion entre pagar y pagar con tarjeta y efectivo .



```
graph TD; Pagar --> PagarConTarjeta; Pagar --> PagarEnEfectivo;
```

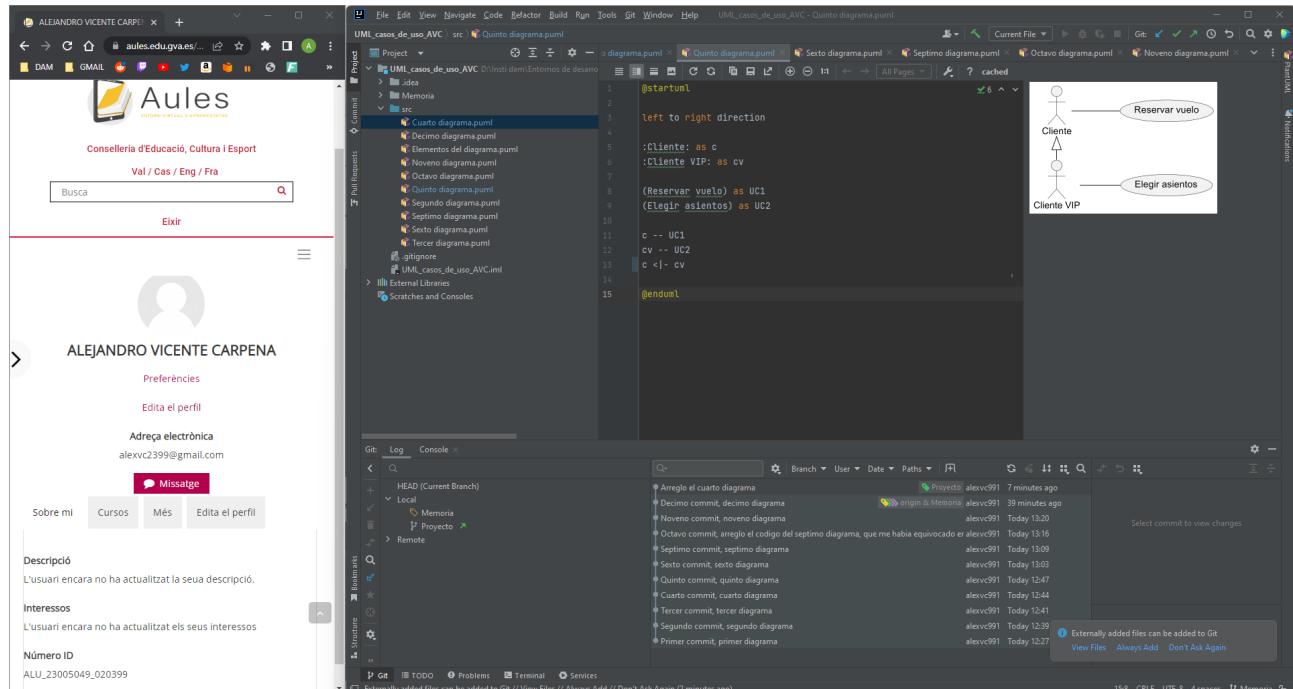
AULES
ALEJANDRO VICENTE CARPENA
Preferències
Edita el perfil
Adreça electrònica
alexvc239@gmail.com
Message
Sobre mi Cursos Més Edita el perfil
Descripció
L'usuari encara no ha actualitzat la seua descripció.
Interessos
L'usuari encara no ha actualitzat els seus interessos
Número ID
ALU_23005049_020399

```
File Edit View Navigate Code Behavior Build Run Tools Git Window Help UML_casos_de_uso_AVC - Cuarto diagrama.puml  
Project src Cuarto diagrama.puml  
1 @startuml  
2  
3 (Pagar) as UC1  
4 (Pagar con tarjeta) as UC2  
5 (Pagar en efectivo) as UC3  
6  
7 UC1 <|-- UC2  
8 UC1 <|-- UC3  
9  
10 @enduml
```

HEAD (Current Branch)
Local
Project
Remote
Structure
Git Log Console
Select commit to view changes
Commit details
87 CRLF UTF-8 4 spaces Project

Este seria el quinto.

En este tambien me he equivocado con la flecha



```
graph LR; Cliente -- "left to right direction" --> ReservarVuelo; Cliente --> ElegirAsientos;
```

AULES
ALEJANDRO VICENTE CARPENA
Preferències
Edita el perfil
Adreça electrònica
alexvc239@gmail.com
Message
Sobre mi Cursos Més Edita el perfil
Descripció
L'usuari encara no ha actualitzat la seua descripció.
Interessos
L'usuari encara no ha actualitzat els seus interessos
Número ID
ALU_23005049_020399

```
File Edit View Navigate Code Behavior Build Run Tools Git Window Help UML_casos_de_uso_AVC - Quinto diagrama.puml  
Project src Quinto diagrama.puml  
1 @startuml  
2  
3 :left to right direction  
4  
5 :Cliente: as c  
6 :Cliente VIP: as cv  
7  
8 (Reservar vuelo) as UC1  
9 (Elegir asientos) as UC2  
10  
11 c -- UC1  
12 cv -- UC2  
13 c <--> cv  
14  
15 @enduml
```

HEAD (Current Branch)
Local
Project
Remote
Structure
Git Log Console
Select commit to view changes
Externally added files can be added to Git
View Files Always Add Don't Ask Again
Externally added files can be added to View Files // Always Add // Don't Ask Again (2 minutes ago)
Externally added files can be added to View Files // Always Add // Don't Ask Again
158 CRLF UTF-8 4 spaces Project

El sexto diagrama

The screenshot shows a dual-pane interface. On the left is a web browser displaying the Aules website profile page for 'ALEJANDRO VICENTE CARPENA'. On the right is a UML modeling tool showing the 'Sexto diagrama.puml' file and its sequence diagram.

Sexto diagrama.puml:

```
1 @startuml
2 left to right direction
3
4 :Cliente: as c
5 (Meter dinero) as UC1
6 (Pedir azucar) as UC2
7 (Pedir producto) as UC3
8 (Cancelar) as UC4
9 (Devolver dinero) as UC5
10
11 c -- UC1
12 c -- UC2
13 c -- UC3
14 c -- UC4
15
16 UC3 ..> UC5 :<<include>>
17 UC4 ..> UC5 :<<include>>
18
19 @enduml
```

Sequence Diagram:

```
sequenceDiagram
    actor Cliente
    Cliente->>Object: Meter dinero
    Cliente->>Object: Pedir azucar
    Cliente->>Object: Pedir producto
    Note over Cliente: 
        UC1  
UC2  
UC3  
UC4
    Object-->>Object: Cancelar
    Note over Object: 
        UC5
    Cliente->>Object: Devolver dinero
    Cliente->>Object: Cancelar
    Note over Object: 
        UC5
```

The sequence diagram illustrates the interactions between a 'Cliente' actor and an object, showing four user cases (UC1-UC4) leading to a 'Cancelar' state, which then includes two more user cases (UC5).

El séptimo diagrama.

En este diagrama tambien me habia equivocado al igual que en el cuarto

The screenshot shows a dual-pane interface. On the left is a web browser displaying the Aules website profile page for 'ALEJANDRO VICENTE CARPENA'. On the right is a UML modeling tool showing the 'Septimo diagrama.puml' file and its corresponding sequence diagram.

Septimo diagrama.puml:

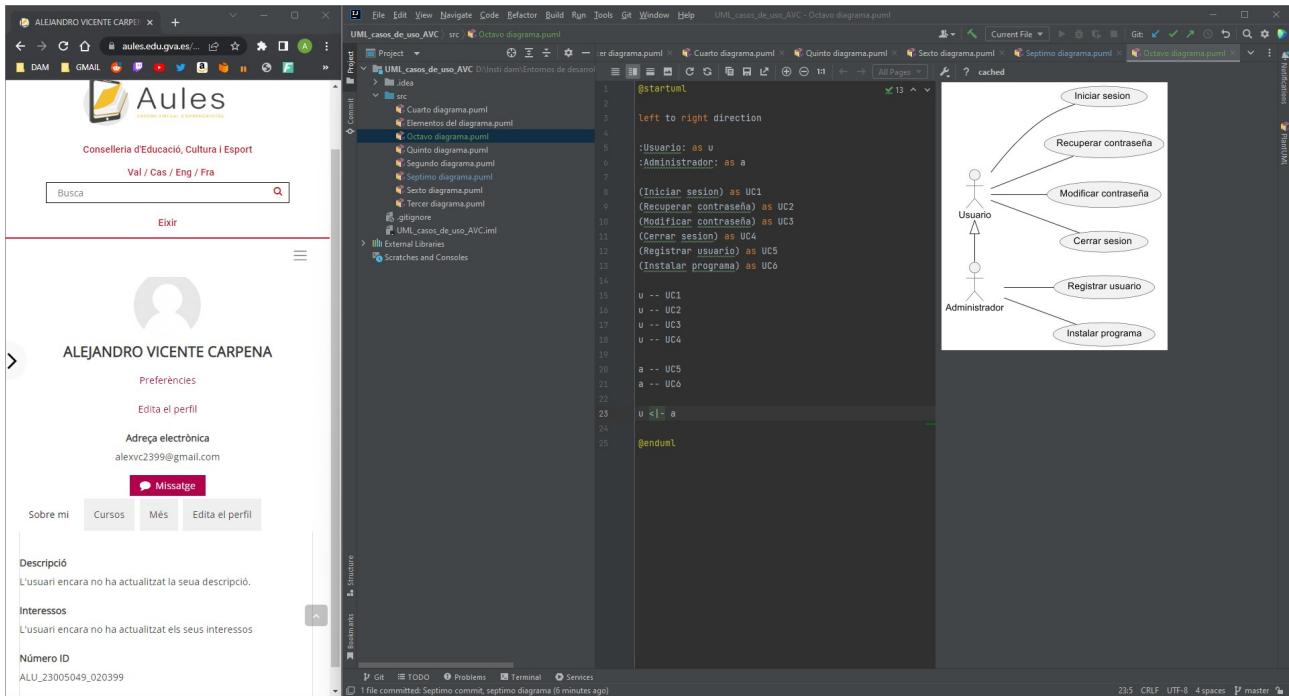
```
1 @startuml
2 left to right direction
3
4 :new Actor: as a
5 (Identificarse) as UC1
6 (Pagar) as UC2
7 (Elegir mensajero) as UC3
8 (Confirmar) as UC4
9 (Registrarse) as UC5
10 (Pagar con tarjeta) as UC6
11 (Pagar con transferencia) as UC7
12
13 a -- UC1
14 a -- UC2
15 a -- UC3
16 a -- UC4
17 a -- UC5
18 a -- UC6
19 a -- UC7
20 UC1 ..> UC5 :<<extend>>
21 UC2 |-- UC6
22 UC2 |-- UC7
23
24 @enduml
```

Sequence Diagram:

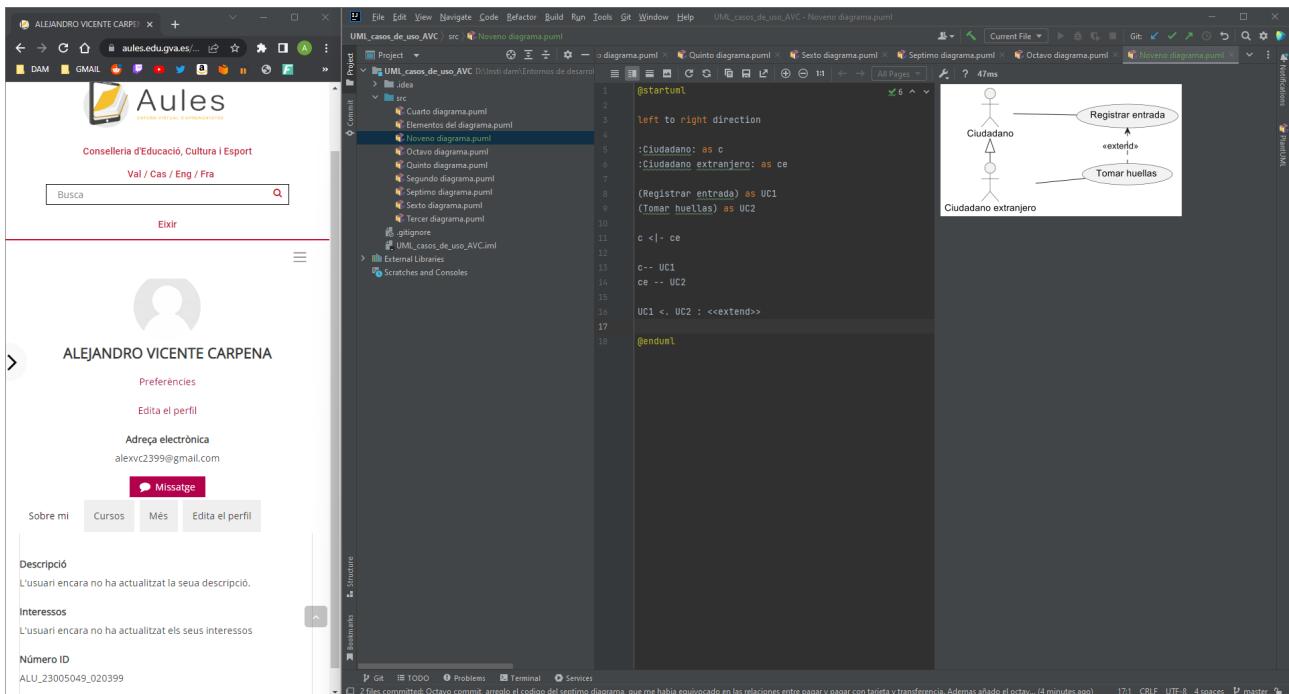
```
sequenceDiagram
    actor new Actor
    new Actor->>Object: Identificarse
    new Actor->>Object: Pagar
    new Actor->>Object: Elegir mensajero
    Note over new Actor: 
        UC1  
UC2  
UC3  
UC4  
UC5  
UC6  
UC7
    Object-->>Object: Confirmar
    Note over Object: 
        UC5
    new Actor->>Object: Pagar con tarjeta
    new Actor->>Object: Pagar con transferencia
    Note over Object: 
        UC6  
UC7
```

The sequence diagram illustrates the interactions between a 'new Actor' and an object, showing seven user cases (UC1-UC7). 'UC1' extends 'UC5', and 'UC2' generalizes both 'UC6' and 'UC7'.

Resultado del octavo diagrama



Resultado del noveno diagrama



Y el resultado del decimo y ultimo diagrama.

En este caso no he logrado encontrar la forma de enlazar la nota con el enlace entre sacar dinero y retener tarjeta, asi que lo he dejado lo mas cercano posible

The screenshot shows a dual-pane interface. On the left is a web browser displaying a user profile for 'ALEJANDRO VICENTE CARPENA' on a platform called 'Aules'. On the right is a UML editor showing the 'Decimo diagrama.puml' file and its graphical representation.

UML Editor Content:

```
@startuml
usecase UC1 #palegreen;line:green; as *-->Sacar dinero
+solicitud hecha
UC1 -- UC2 #green;line.dotted; : <<Extend>>
note top of UC2 #lightyellow;line:black;
Description
condición:
{historia sospechosa }
punto de extensión:
solicitud hecha
end note
@enduml
```

Diagram Representation:

```
graph TD
    UC1([Sacar dinero +solicitud hecha]) -- "UC1 -- UC2 #green;line.dotted; : <<Extend>>" --> UC2([Retener tarjeta])
    UC2 --- Note1{Historia sospechosa}
    UC2 --- Note2{Punto de extensión: solicitud hecha}
```

The diagram consists of two rounded rectangles. The top one is labeled 'Sacar dinero +solicitud hecha'. An arrow labeled 'UC1 -- UC2 #green;line.dotted; : <<Extend>>' points from it to the bottom one, which is labeled 'Retener tarjeta'. A callout box labeled 'Description' contains the condition '{historia sospechosa}' and the extension point 'punto de extensión: solicitud hecha'.