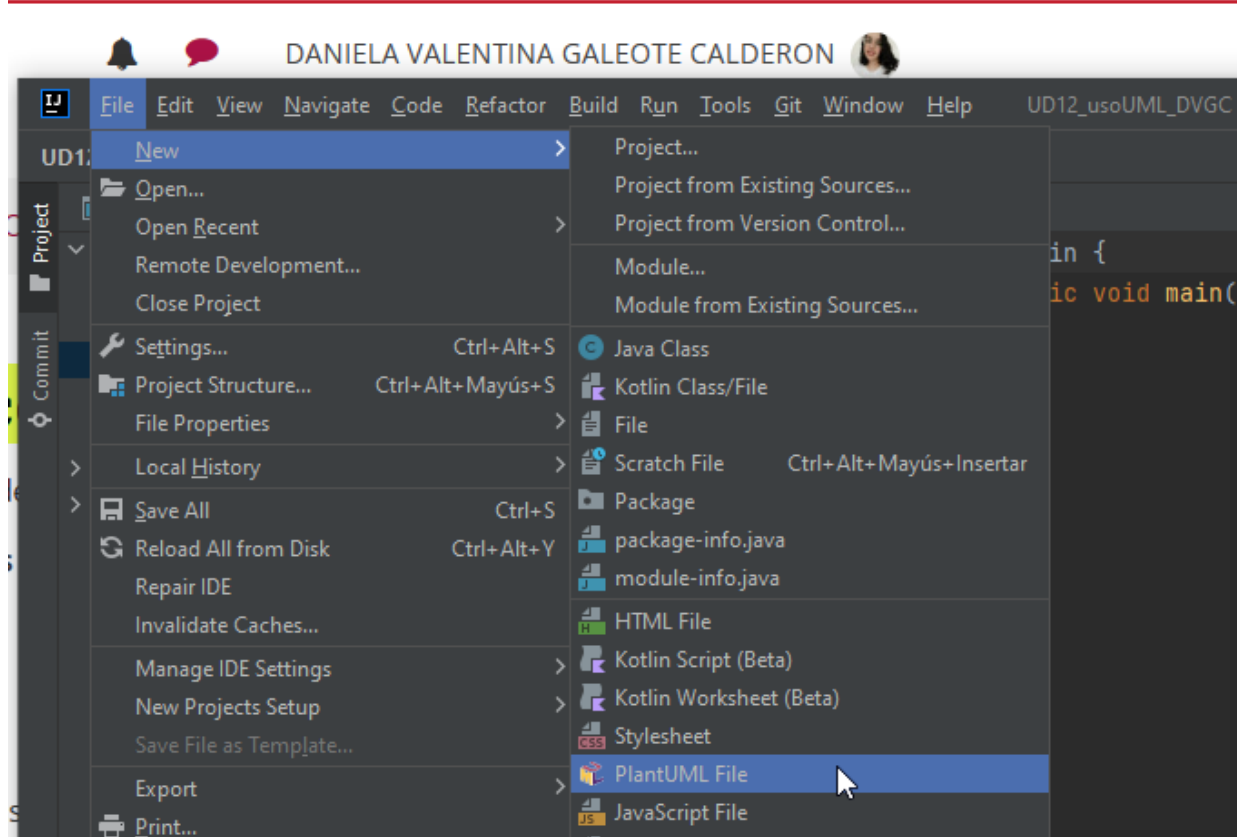


## Práctica clases de uso con PlantUml.

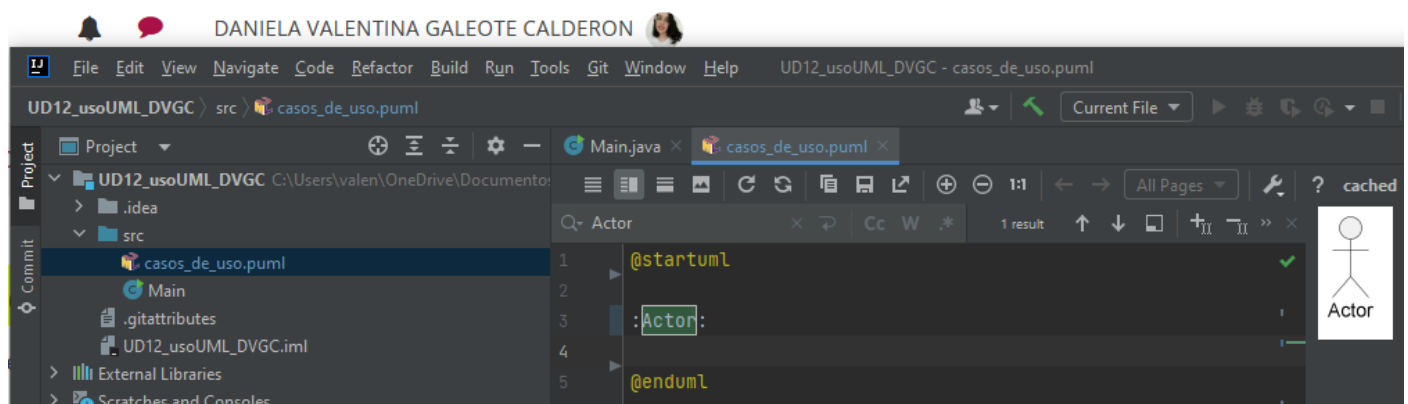
Creo un nuevo archivo PlantUML en IntelliJ:

# ICFSQ Ent ...



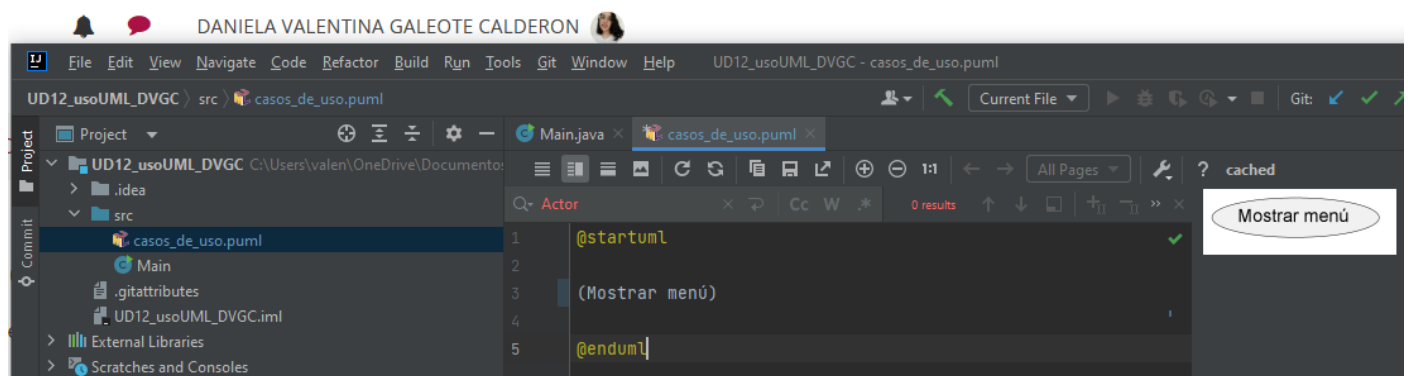
**Actor:** Empiezo creado el primer caso de uso que es un actor, lo implemento con doble punto “:Actor:”

# ICFSQ Ent ...



**Caso de uso:** añadido un caso de uso poniendo el texto entre paréntesis “(Mostrar menú)”

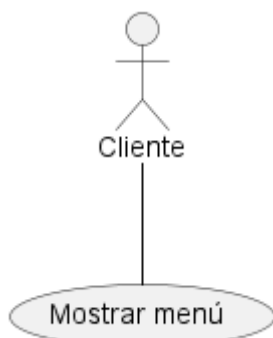
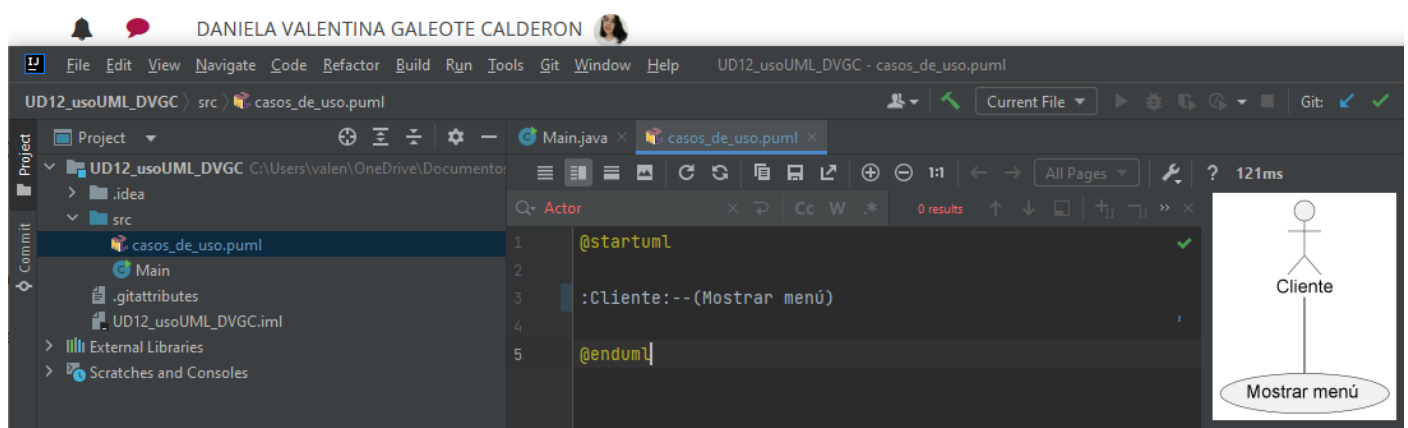
## 1CFSQ Ent ...



Mostrar menú

**Asociación:** Para representar la interacción entre los actores y los casos de uso se unen con una línea recta utilizando dos guiones “:Cliente:--(Mostrar menú)”

## 1CFSQ Ent ...



**Sistema:** Para representar el sistema como caja rectangular e incluir dentro de ella los casos soportados se utiliza la palabra 'rectangle' seguido de una llave que incluirá los casos de uso y sus alias. Después puedo representar las relaciones con los actores mediante el alias.

```
rectangle {  
  (mostrar menú) as CU1  
  (Pedir comida) as CU2  
}  
:Cliente:--CU1  
:Cliente:--CU2  
:Restaurante:--CU1  
:Restaurante:--(Entregar comida)
```

## ICFSQ Ent ...

DANIELA VALENTINA GALEOTE CALDERON

UD12\_usoUML\_DVGC - casos\_de\_uso.puml

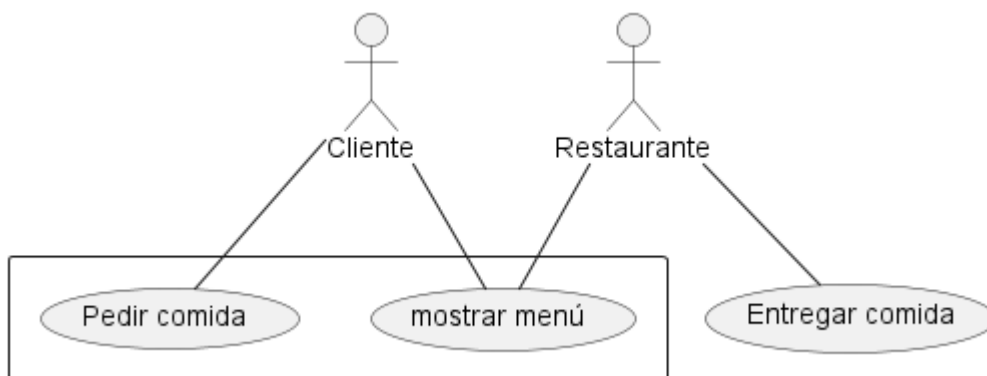
UD12\_usoUML\_DVGC > src > casos\_de\_uso.puml

Project: Main.java x casos\_de\_uso.puml x

Q- Actor 0 results

```
1 @startuml  
2  
3 rectangle {  
4   (mostrar menú) as CU1  
5   (Pedir comida) as CU2  
6 }  
7 :Cliente:--CU1  
8 :Cliente:--CU2  
9 :Restaurante:--CU1  
10 :Restaurante:--(Entregar comida)  
11  
12 @enduml
```

```
graph TD
    Cliente((Cliente)) --- CU1((mostrar menú))
    Cliente --- CU2((Pedir comida))
    Restaurante((Restaurante)) --- CU1
    Restaurante --- CU3((Entregar comida))
    subgraph System
        CU1
        CU2
    end
```

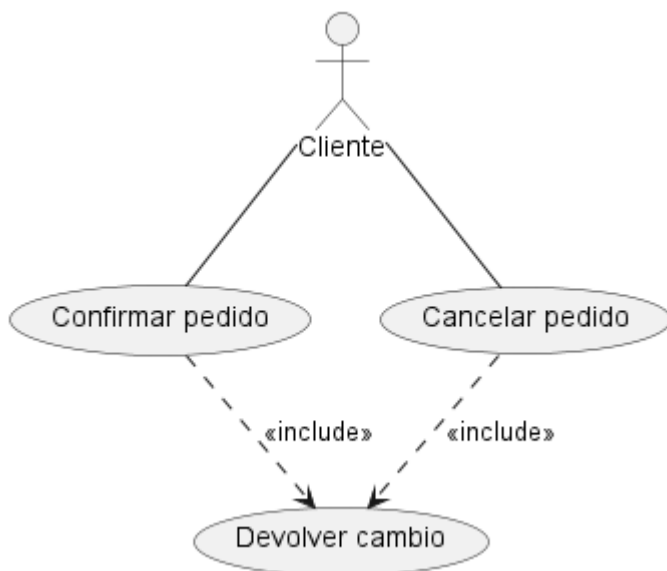
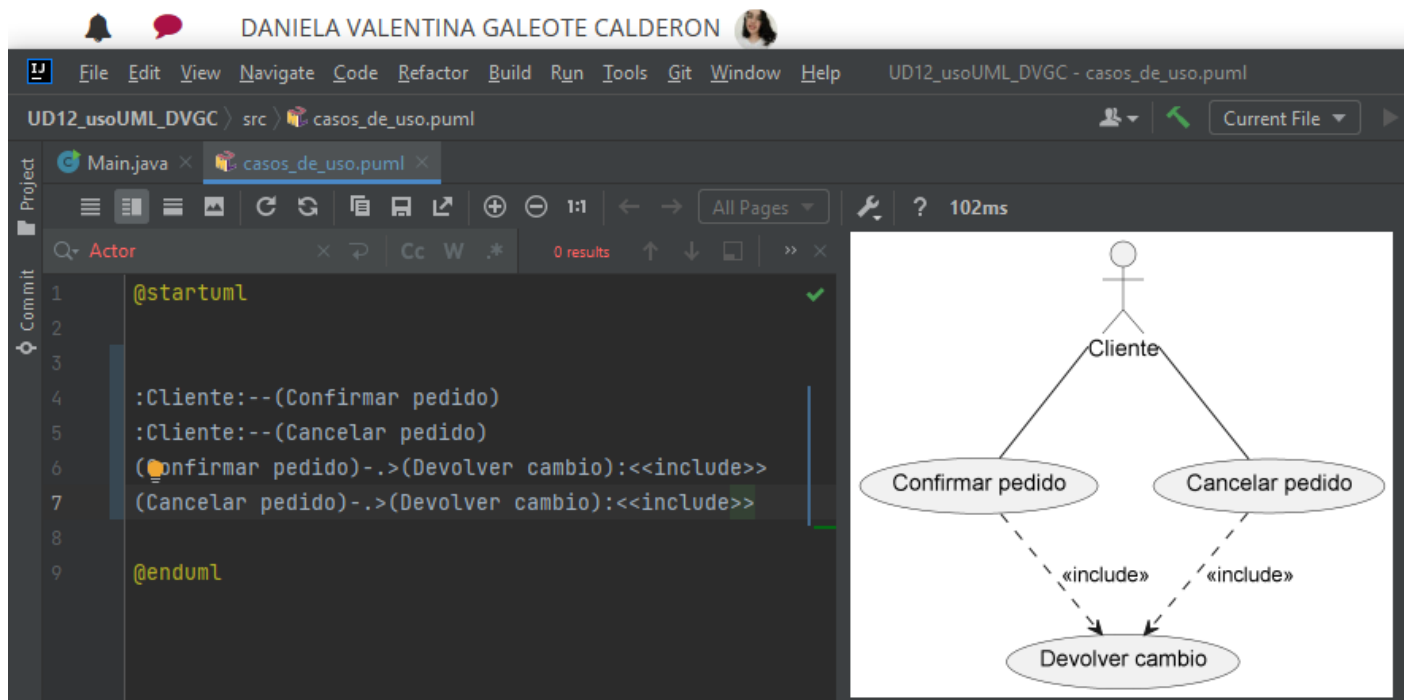


**Inclusión:** Para representar la inclusión de un comportamiento dentro de otro utilizamos la línea discontinua con el estereotipo <<include>>

La línea discontinua se codifica con .> ó -.> y al final de la relación se añade dos puntos para poner el estereotipo :<<include>>

```
:Cliente:--(Confirmar pedido)
:Cliente:--(Cancelar pedido)
(Confirmar pedido)-.>(Devolver cambio):<<include>>
(Cancelar pedido)-.>(Devolver cambio):<<include>>
```

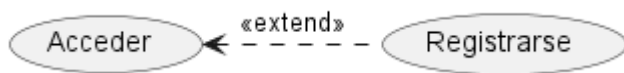
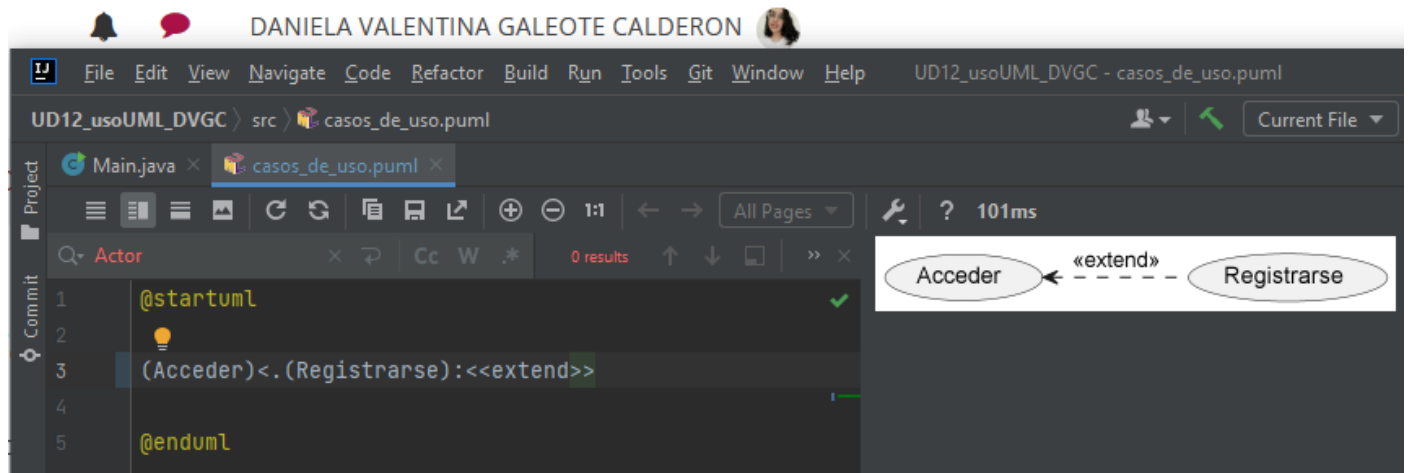
# 1CFSQ Ent ...



**Extensión:** Se representa con flecha de línea discontinua y el estereotipo <<extend>>

```
(Acceder)<.(Registrarse):<<extend>>
```

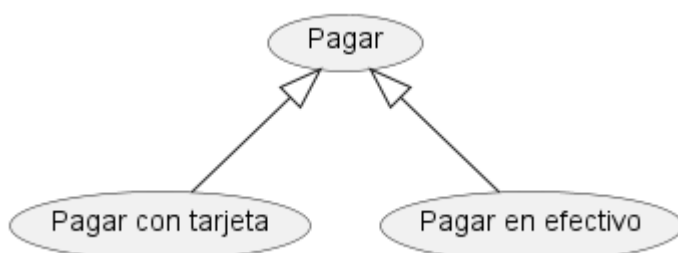
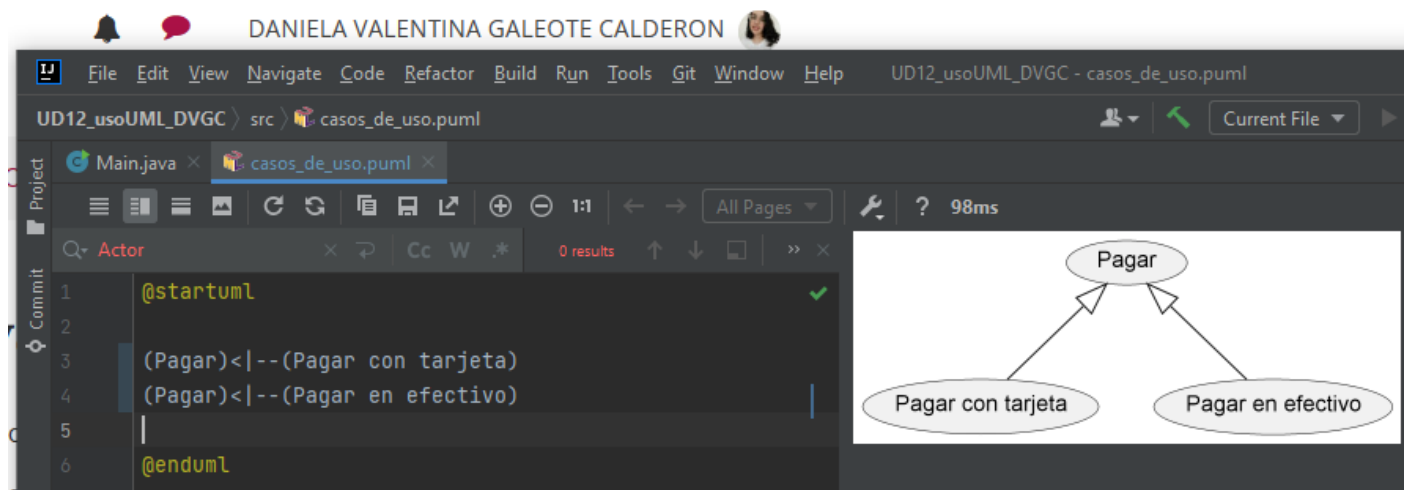
# 1CFSQ Ent ...



**Generalización:** Se representa con una flecha triangular hueca --|>

```
(Pagar)<|--(Pagar con tarjeta)
(Pagar)<|--(Pagar en efectivo)
```

# 1CFSQ Ent ...



También se puede hacer con actores:

```
:Cliente:-(Reservar vuelo)
:Cliente VIP:-(Elegir asientos)
:Cliente:<|--:Cliente VIP:
```

# 1CFSQ Ent ...

DANIELA VALENTINA GALEOTE CALDERON

UD12\_usoUML\_DVGC - casos\_de\_uso.puml

UD12\_usoUML\_DVGC > src > casos\_de\_uso.puml

Project: Main.java x casos\_de\_uso.puml x

Commit: 0 results

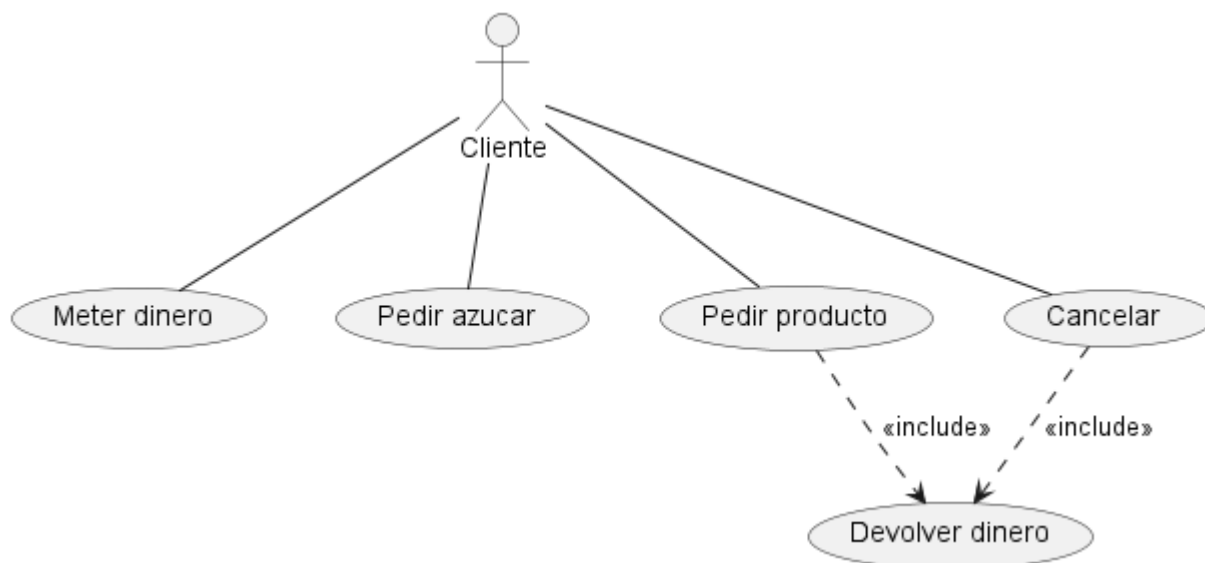
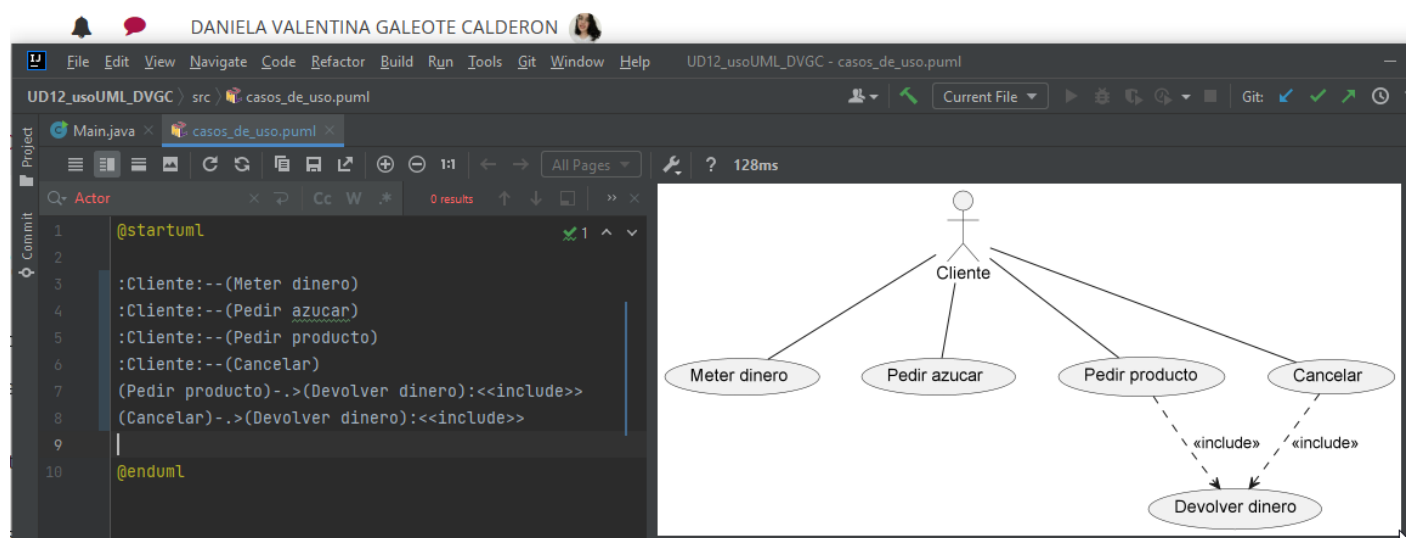
```
1 @startuml
2
3 :Cliente:-(Reservar vuelo)
4 :Cliente VIP:-(Elegir asientos)
5 :Cliente:<|--:Cliente VIP:
6
7 @enduml
```

```
graph TD
    Cliente((Cliente)) --- UC1([Reservar vuelo])
    ClienteVIP((Cliente VIP)) --- UC2([Elegir asientos])
    ClienteVIP --|> Cliente
```



**Ejemplos:****La máquina de café:**

```
:Cliente:--(Meter dinero)
:Cliente:--(Pedir azucar)
:Cliente:--(Pedir producto)
:Cliente:--(Cancelar)
(Pedir producto)->(Devolver dinero):<<include>>
(Cancelar)->(Devolver dinero):<<include>>
```

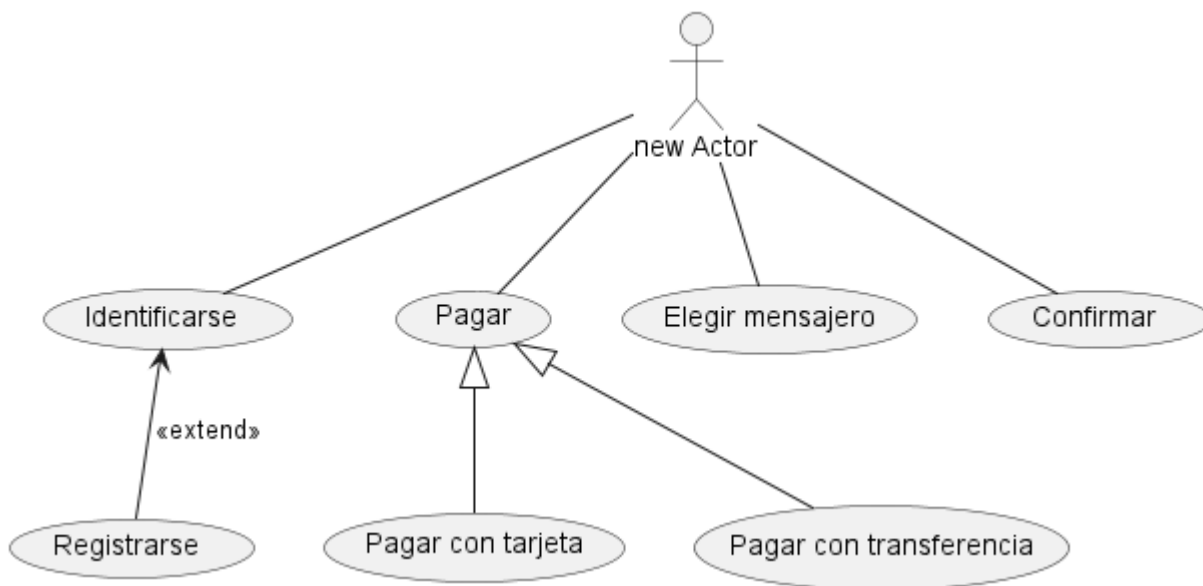
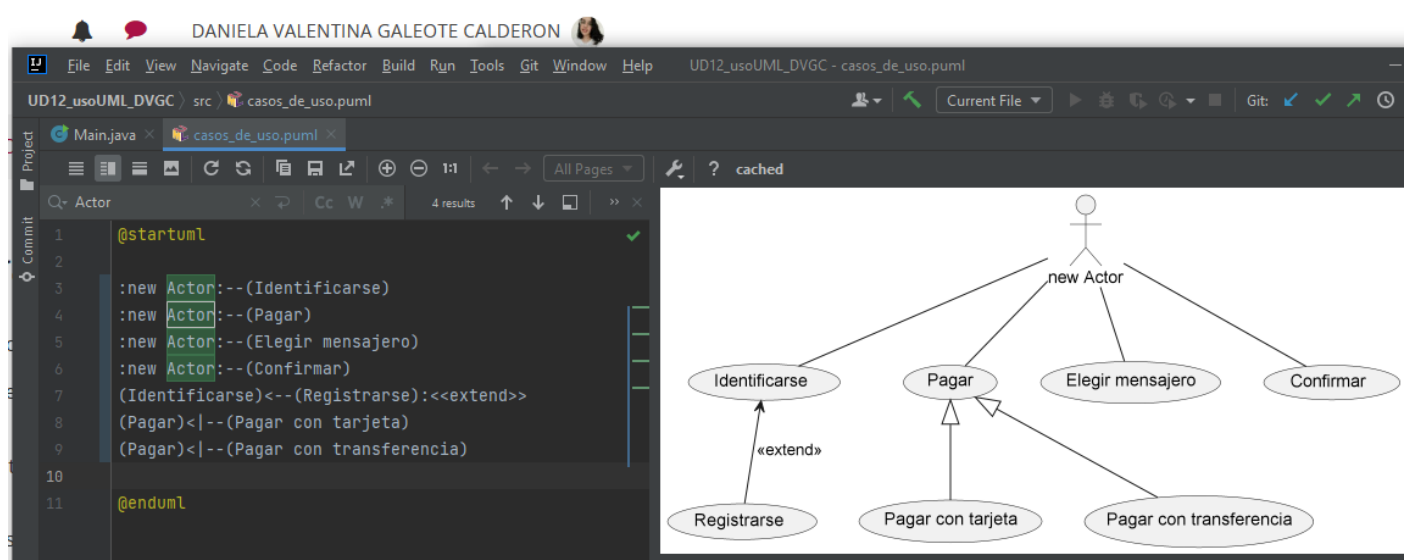
**ICFSQ Ent ...**

**Tienda en Internet:**

```

:new Actor:--(Identificarse)
:new Actor:--(Pagar)
:new Actor:--(Elegir mensajero)
:new Actor:--(Confirmar)
(Identificarse)<--(Registrarse):<<extend>>
(Pagar)<|--(Pagar con tarjeta)
(Pagar)<|--(Pagar con transferencia)

```

**UMLFSQ Ent ...**



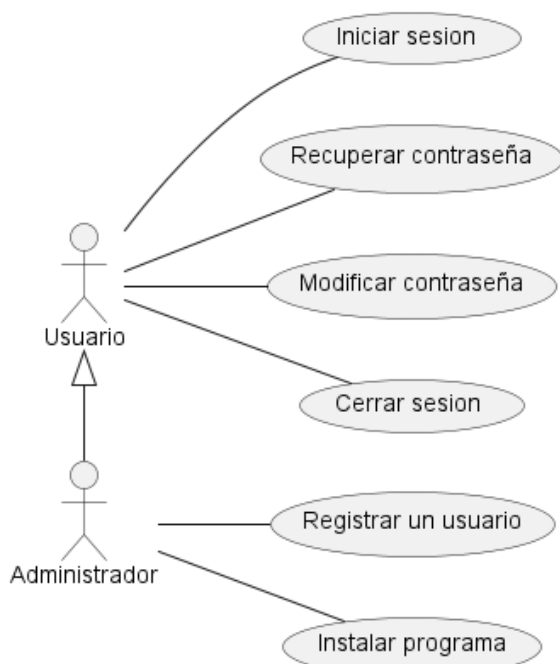
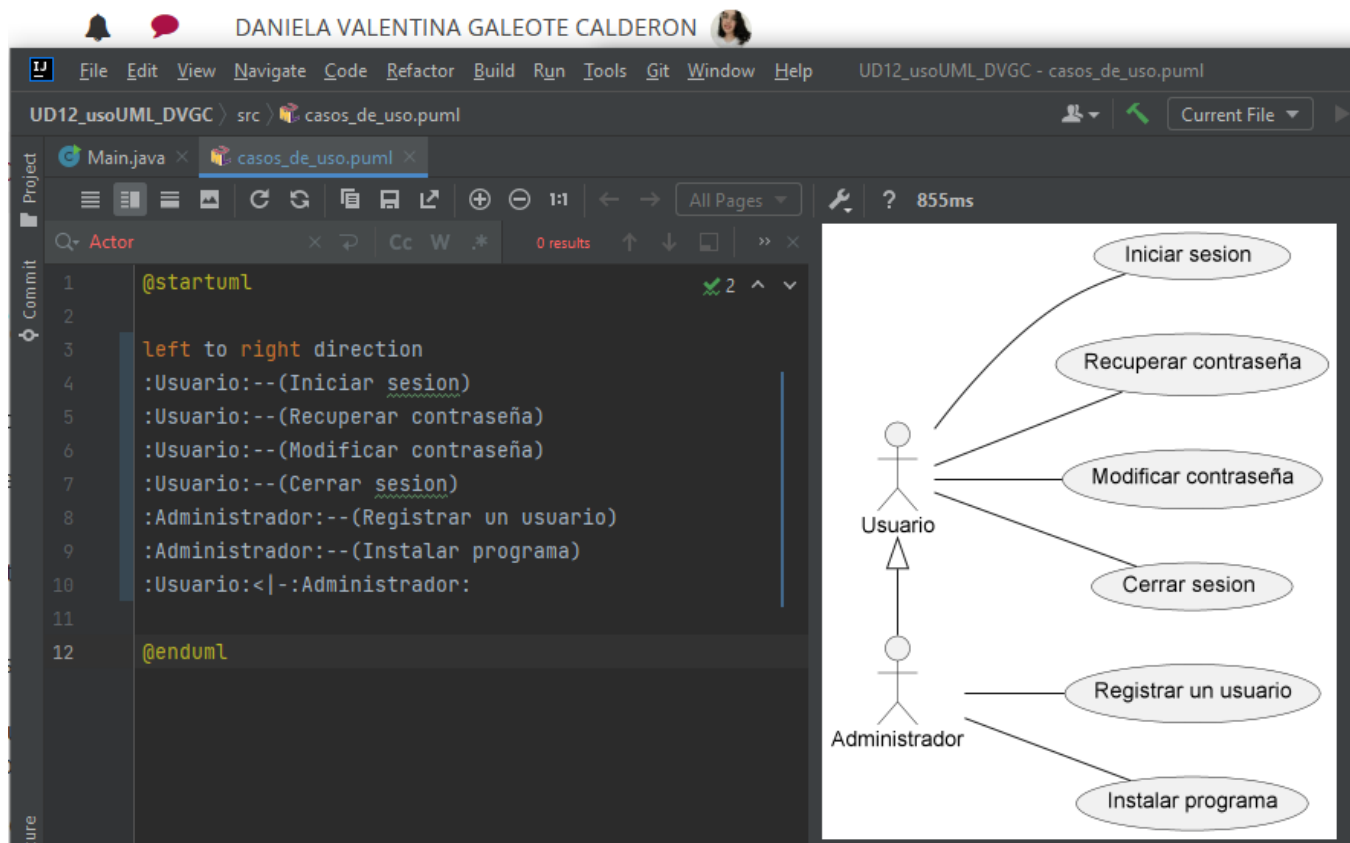
**Ejemplo: Usuarios y administradores:**

He añadido la línea "left to right direction" para que el diagrama se visualice de izquierda a derecha

```

left to right direction
:Usuario:--(Iniciar sesion)
:Usuario:--(Recuperar contraseña)
:Usuario:--(Modificar contraseña)
:Usuario:--(Cerrar sesion)
:Administrador:--(Registrar un usuario)
:Administrador:--(Instalar programa)
:Usuario:<|:-:Administrador:
  
```

# ICFSQ Ent ...



**Ejemplo: Puesto fronterizo**

```

:Ciudadano:-(Registrar entrada)
:Ciudadano extranjero:-(Tomar huellas)
:Ciudadano:<|--:Ciudadano extranjero:
(Registrar entrada)<--.(Tomar huellas):<<extend>>

```

# 1CFSQ Ent ...



DANIELA VALENTINA GALEOTE CALDERON



UD12\_usoUML\_DVGC - casos\_de\_uso.puml

UD12\_usoUML\_DVGC > src > casos\_de\_uso.puml

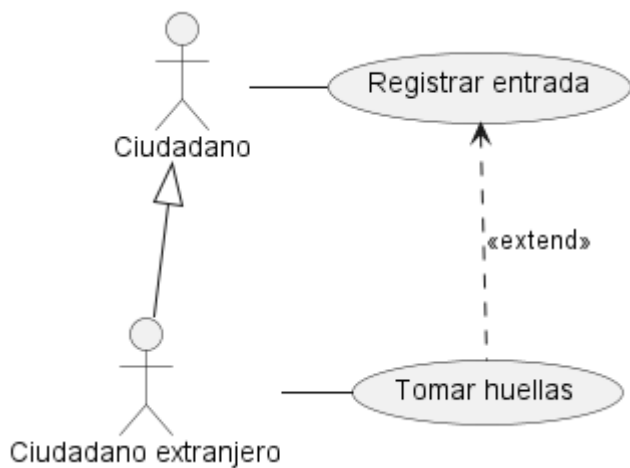
Project: Main.java x casos\_de\_uso.puml x

Search: Actor 0 results

```

1 @startuml
2
3 :Ciudadano:-(Registrar entrada)
4 :Ciudadano extranjero:-(Tomar huellas)
5 :Ciudadano:<|--:Ciudadano extranjero:
6 (Registrar entrada)<--.(Tomar huellas):<<extend>>
7
8 @enduml

```



## Punto de extensión

-Modifico el aspecto de los casos de uso con skinparam

-BackgroundColor para el color de fondo y BorderColor para el color del borde

-También modifiko el color de las flechas con skinparam ArrowColor

-Para añadir una línea en mitad del caso de uso, lo utilizo en el formato alias con Usecase UC1 as "" y añadido un salto de línea más dos guiones para marcar que ahí se visualizará una división.

-Finalmente para añadir la nota a el punto de extensión la coloco a la izquierda de la relación con "note left on link", añadido una división de la misma forma con dos guiones y termino la nota con "end note"

```
skinparam usecase {
BackgroundColor Technology
BorderColor DarkCyan
}

skinparam arrow {
ArrowColor DarkCyan
}

Usecase UC1 as "Sacar dinero
--
+solicitud hecha"
Usecase UC2 as "Retener tarjeta

"
UC1 <- UC2:<<Extend>>
note left on link
Descripción
--
Condición:
{historia sospechosa}
punto de extensión:
solicitud hecha
end note
```

## UMLFSQ Ent ...

UD12\_usoUML\_DVGC - casos\_de\_uso.puml

UD12\_usoUML\_DVGC > src > casos\_de\_uso.puml

Project: Main.java x casos\_de\_uso.puml x

Q: Actor 0 results

```
1 @startuml
2   skinparam usecase {
3     BackgroundColor Technology
4     BorderColor DarkCyan
5   }
6
7   skinparam arrow {
8     ArrowColor DarkCyan
9   }
10  Usecase UC1 as "Sacar dinero
11  --
12  +solicitud hecha"
13  Usecase UC2 as "Retener tarjeta"
14
15  "
16  UC1 <- UC2:<<Extend>>
17  note left on link
18  Descripción
19  --
20  Condición:
21  {historia sospechosa}
22  punto de extensión:
23  solicitud hecha
24  end note
25 @enduml
```

Descripción  
Condición:  
{historia sospechosa}  
punto de extensión:  
solicitud hecha

«Extend»

Sacar dinero  
+solicitud hecha

Retener tarjeta

