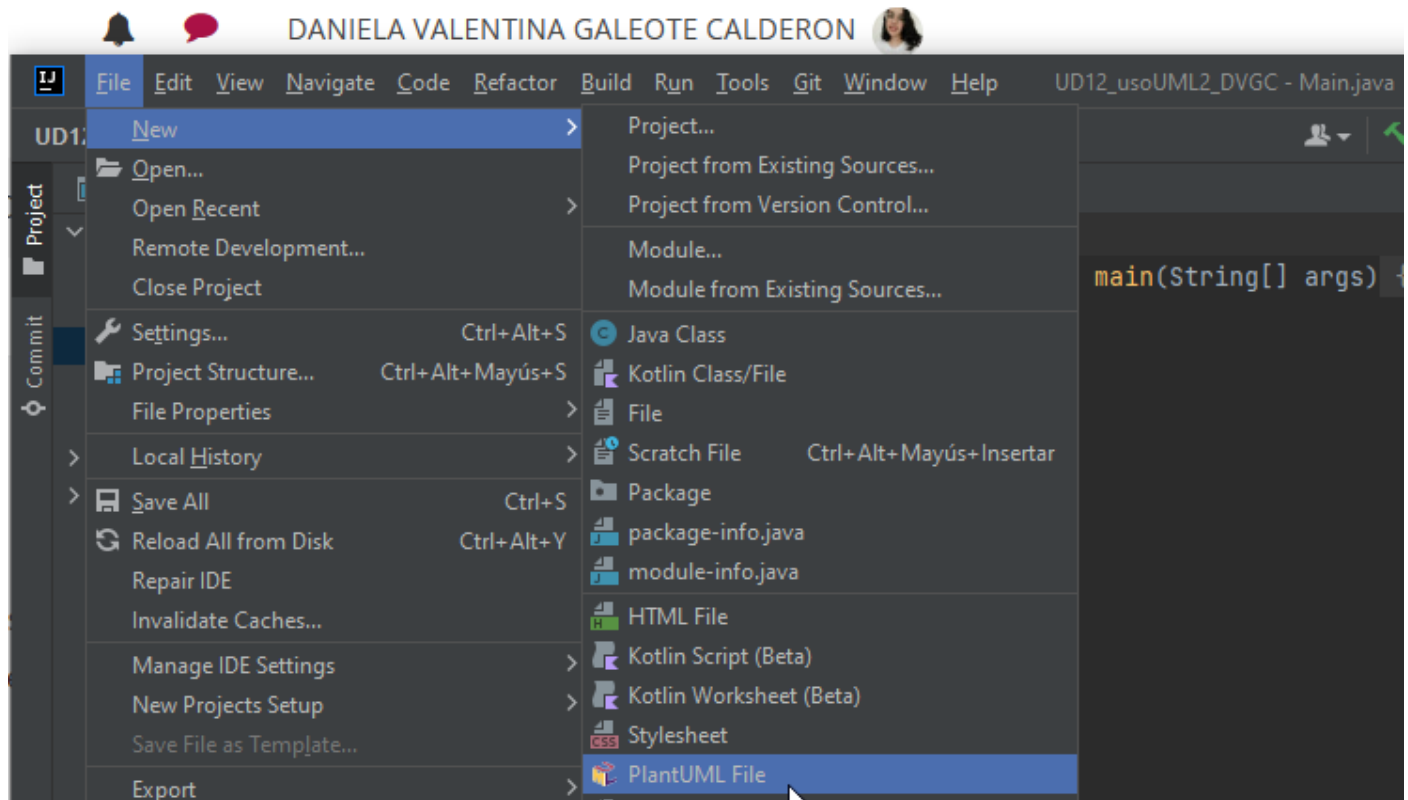


## MEMORIA Práctica clases de uso con PlantUml II.

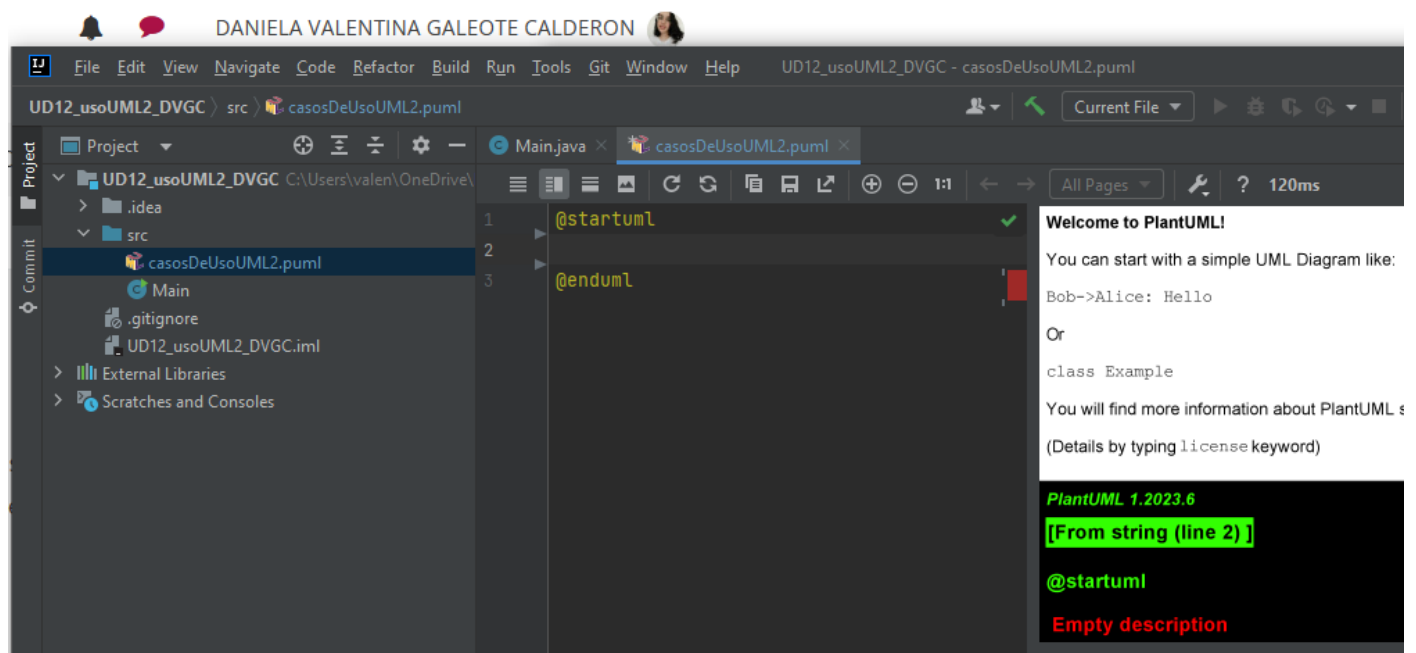
Creo un nuevo archivo PlantUML en IntelliJ:

# 1CFSQ Ent ...



Archivo PlantUML creado:

# 1CFSQ Ent ...



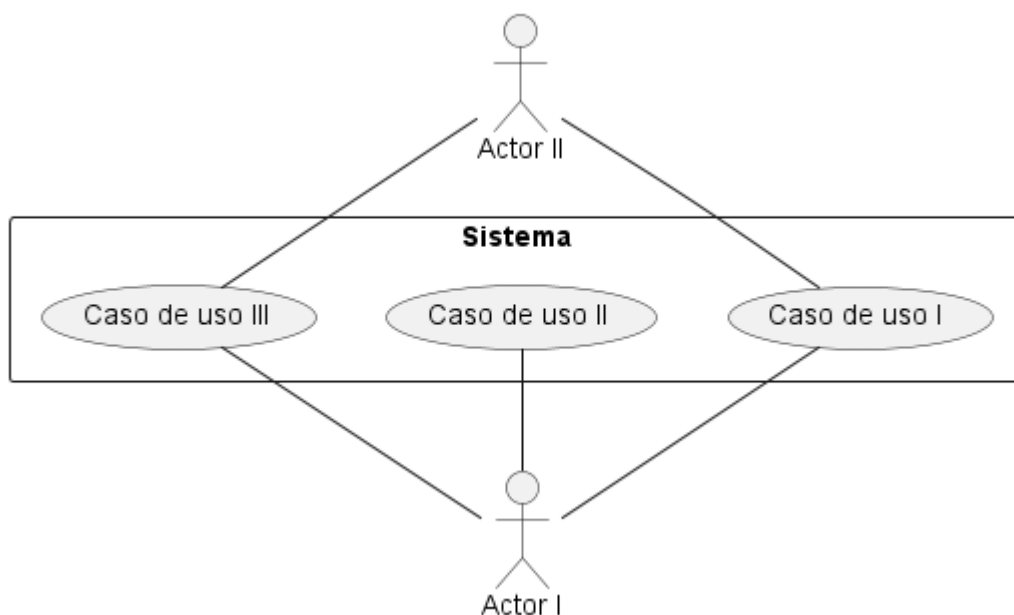
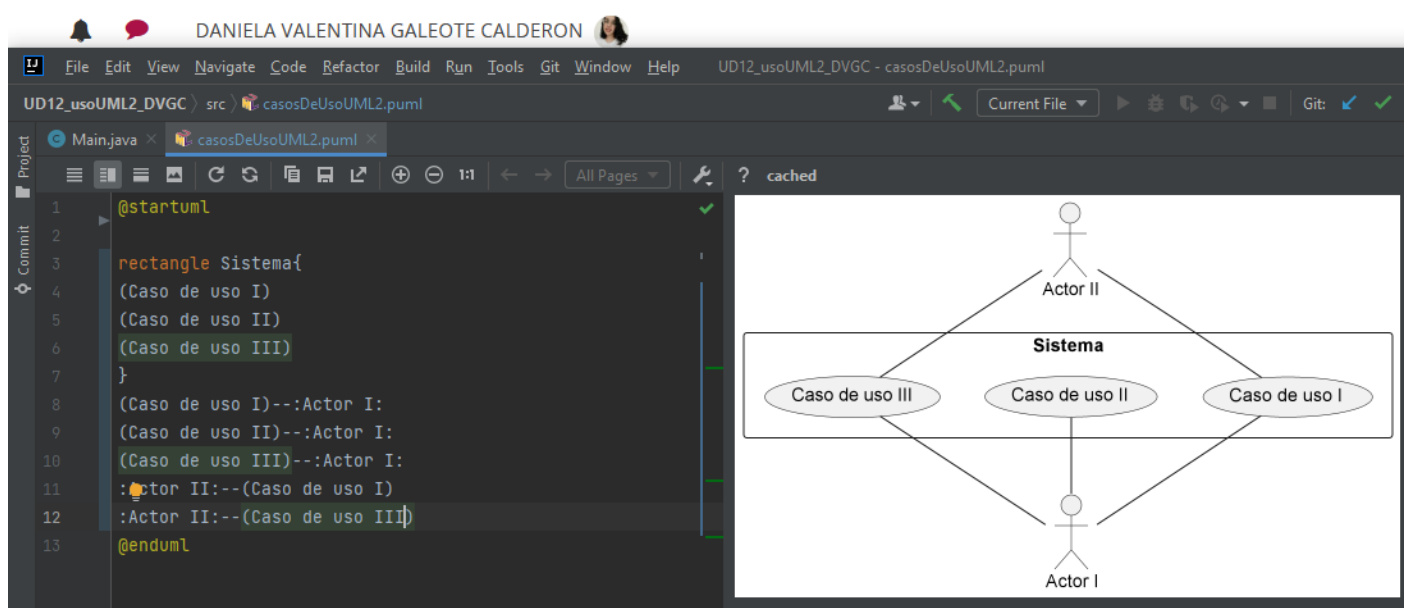
## Caso de uso 1 – Ejemplo de elementos:

Ejemplo de un sistema con tres casos de uso y dos actores externo con sus respectivas comunicaciones.

He usado recangle para el Sistema e incluir los casos de uso dentro y la asociación con dos guiones para comunicarlos con los dos actores externos.

```
rectangle Sistema{
(Caso de uso I)
(Caso de uso II)
(Caso de uso III)
}
(Caso de uso I)--:Actor I:
(Caso de uso II)--:Actor I:
(Caso de uso III)--:Actor I:
:Actor II:--(Caso de uso I)
:Actor II:--(Caso de uso III)
```

# 1CFSQ Ent ...



**CU Biblioteca:**

La misma dinámica que el diagrama anterior con un caso de uso adicional, además le he añadido la línea 'left to right direction' para que se visualice de izquierda a derecha.

```

left to right direction
rectangle Biblioteca{
  (Reservar Libro)
  (Prestar Libro)
  (Devolver Libro)
  (Actualizar Catálogo)
}
(Reservar Libro)--:Socio:
(Prestar Libro)--:Socio:
(Devolver Libro)--:Socio:
:Sociotecario:--(Prestar Libro)
:Sociotecario:--(Actualizar Catálogo)

```

# ICFSQ Ent ...

DANIELA VALENTINA GALEOTE CALDERON

UD12\_usoUML2\_DVGC - Biblioteca.puml

Project: UD12\_usoUML2\_DVGC

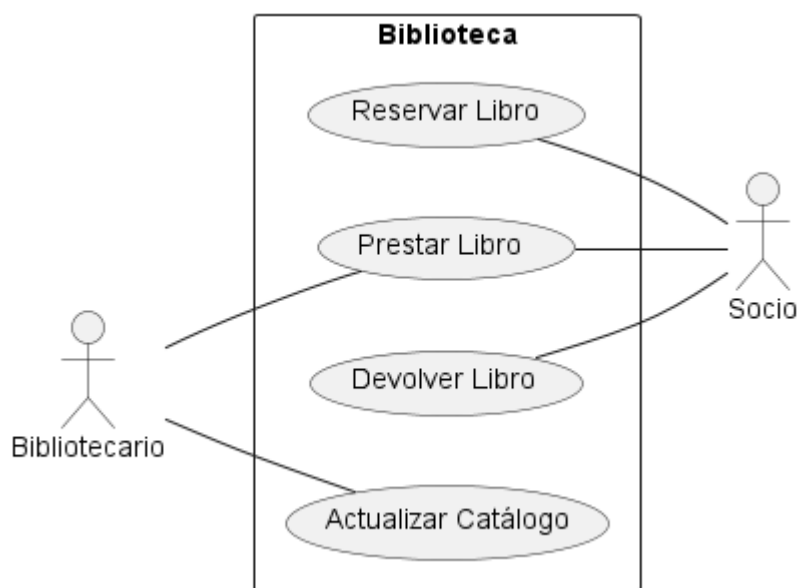
src

- Biblioteca.puml
- casosDeUsoUML2.puml
- Main
- .gitignore
- UD12\_usoUML2\_DVGC.iml
- External Libraries
- Scratches and Consoles

```

1 @startuml
2 left to right direction
3 rectangle Biblioteca{
4   (Reservar Libro)
5   (Prestar Libro)
6   (Devolver Libro)
7   (Actualizar Catálogo)
8 }
9 (Reservar Libro)--:Socio:
10 (Prestar Libro)--:Socio:
11 (Devolver Libro)--:Socio:
12 :Sociotecario:--(Prestar Libro)
13 :Sociotecario:--(Actualizar Catálogo)
14 @enduml

```



## Ejemplo plantilla Caso de Uso:

He realizado la tabla/plantilla con el comando json e incluyendo dentro los campos con “ ” dos puntos para la siguiente celda : y volviendo a colocar el campo con “ ”

Para agregar saltos de línea he puesto una secuencia de escape ‘\n’ al final de la línea

```
json CasoDeUso {
  "Caso de Uso":"Reservar Libro",
  "Actores":"Socio",
  "Resumen":"El socio puede solicitar la reserva de un libro para su posterior\n\n
préstamo, a partir de una fecha determinada",
  "Precondiciones":"El socio no tiene ninguna reserva",
  "Postcondiciones":"El socio tiene un reserva y el libro tiene una nueva reserva a partir
de\n\n
una fecha",
  "Incluye":"- -",
  "Extiende":"- -",
  "Hereda de":"- -",
  "Flujo de eventos":"",
  "Actor":"Sistema",
  "1 El socio socilicita la reserva (codigo libro, fecha)\n\n\n\n\n\n
5. El socio confirma la reserva":"2. El sistema comprueba que el socio no tiene reserva\n\n
3. El sistema compueba que libro está libre para la fecha solicitada\n\n
4. El sistema solicita confirmación de la reserva\n\n\n
6. El sistema realiza la reserva"
```

# 1CFSQ Ent ...

The screenshot shows an IDE with a JSON file on the left and a UML Use Case diagram on the right. The JSON file contains the following content:

```
json CasoDeUso {
  "Caso de Uso":"Reservar Libro",
  "Actores":"Socio",
  "Resumen":"El socio puede solicitar la reserva de un libro para su posterior\n\n
préstamo, a partir de una fecha determinada",
  "Precondiciones":"El socio no tiene ninguna reserva",
  "Postcondiciones":"El socio tiene un reserva y el libro tiene una nueva reserva a partir
de\n\n
una fecha",
  "Incluye":"- -",
  "Extiende":"- -",
  "Hereda de":"- -",
  "Flujo de eventos":"",
  "Actor":"Sistema",
  "1 El socio socilicita la reserva (codigo libro, fecha)\n\n\n\n\n\n
5. El socio confirma la reserva":"2. El sistema comprueba que el socio no tiene reserva\n\n
3. El sistema compueba que libro está libre para la fecha solicitada\n\n
4. El sistema solicita confirmación de la reserva\n\n\n
6. El sistema realiza la reserva"
```

The UML Use Case diagram on the right shows the following details:

CasoDeUso	
Caso de Uso	Reservar Libro
Actores	Socio
Resumen	El socio puede solicitar la reserva de un libro para su posterior préstamo, a partir de una fecha determinada
Precondiciones	El socio no tiene ninguna reserva
Postcondiciones	El socio tiene un 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 socilicita la reserva (codigo libro, fecha)	2. El sistema comprueba que el socio no tiene reserva
	3. El sistema compueba que libro está libre para la fecha solicitada
	4. El sistema solicita confirmación de la reserva
5. El socio confirma la reserva	6. El sistema realiza la reserva

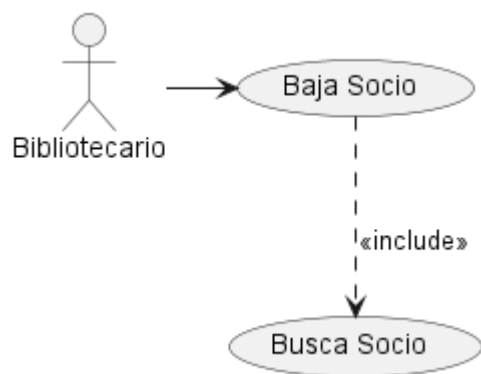
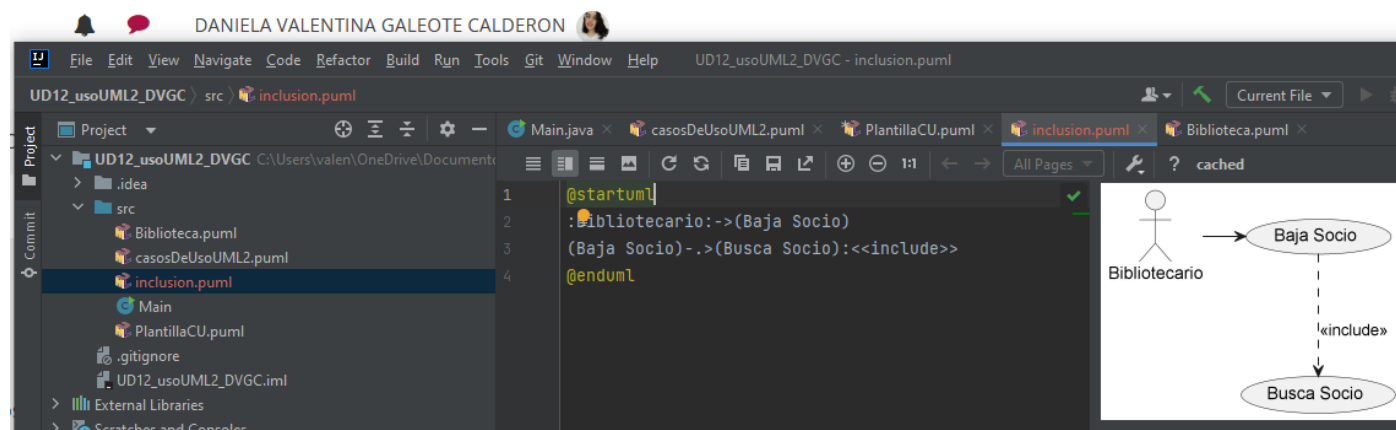
CasoDeUso	
Caso de Uso	Reservar Libro
Actores	Socio
Resumen	El socio puede solicitar la reserva de un libro para su posterior préstamo, a partir de una fecha determinada
Precondiciones	El socio no tiene ninguna reserva
Postcondiciones	El socio tiene un reserva y el libro tiene una nueva reserva a partir de una fecha
Incluye	- -
Extiende	- -
Hereda de	- -
Flujo de eventos	- -
Actor	Sistema
1 El socio socilicita la reserva (codigo libro, fecha)	2. El sistema comprueba que el socio no tiene reserva
	3. El sistema compueba que libro está libre para la fecha solicitada
	4. El sistema solicita confirmación de la reserva
5. El socio confirma la reserva	6. El sistema realiza la reserva

**Inclusión:**

Añado la inclusión a la relación con dos puntos al final : + <<include>>

```
:Bibliotecario:->(Baja Socio)
(Baja Socio)->(Busca Socio):<<include>>
```

# ICFSQ Ent ...



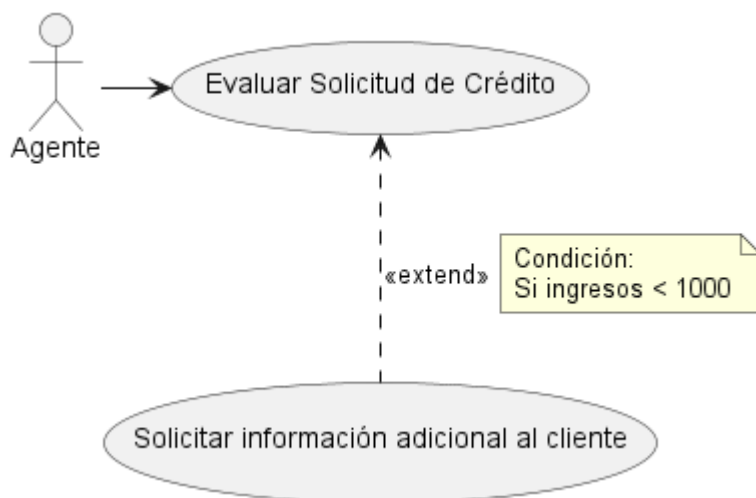
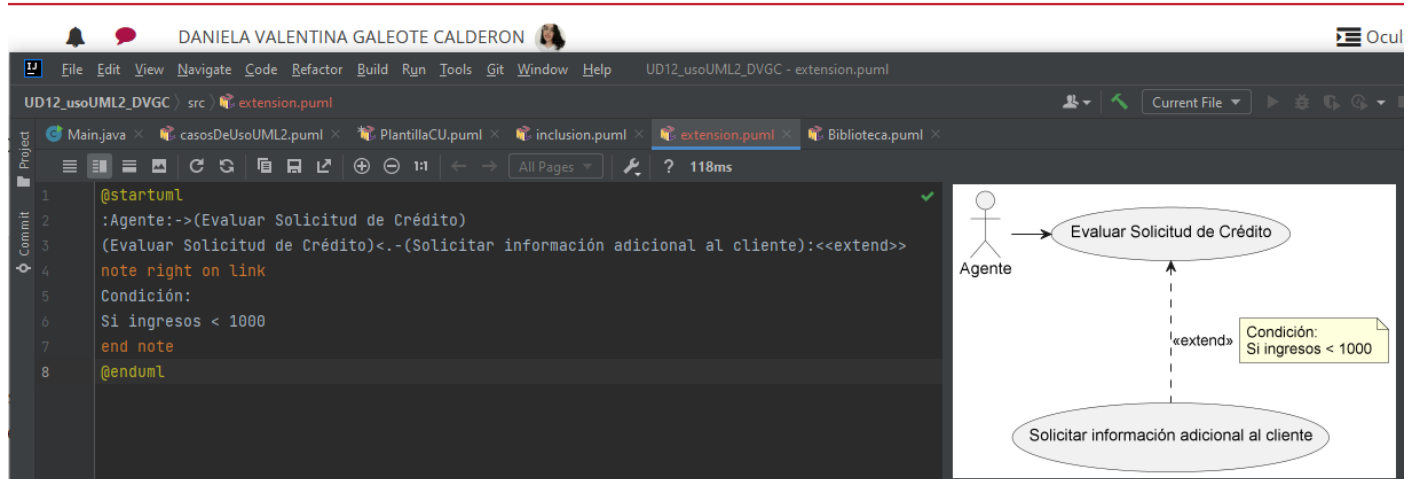
**Extensión:**

Se añade a la relación de igual forma que inclusión, pero se pone <<extend>> al final

Y una nota describiendo la condición para que se ejecute la sucesión de eventos con Note right on link

```
:Agente->(Evaluar Solicitud de Crédito)
(Evaluar Solicitud de Crédito)<.-(Solicitar información adicional al
cliente):<<extend>>
note right on link
Condición:
Si ingresos < 1000
end note
```

# ICFSQ Ent ...



**Herencia:**

Se representa la herencia con una flecha triangular hueca --|>

```
:Cliente Bancario:<|--:Cliente Normal:  
:Cliente Bancario:<|--:Cliente Corporativo:
```

# 1CFSQ Ent ...

DANIELA VALENTINA GALEOTE CALDERON

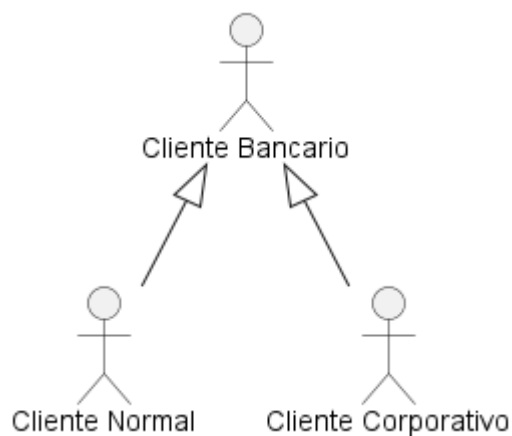
File Edit View Navigate Code Refactor Build Run Tools Git Window Help UD12\_usoUML2\_DVGC - Herencia.puml

UD12\_usoUML2\_DVGC > src > Herencia.puml

Main.java x casosDeUsoUML2.puml x PlantillaCU.puml x inclusion.puml x extension.puml x Herencia.puml

1 @startuml  
2 :Cliente Bancario:<|--:Cliente Normal:  
3 :Cliente Bancario:<|--:Cliente Corporativo:  
4 @enduml

```
graph BT; CN[Cliente Normal] --|> CB[Cliente Bancario]; CC[Cliente Corporativo] --|> CB;
```



**Diagrama inicial: Empresa Pedidos**

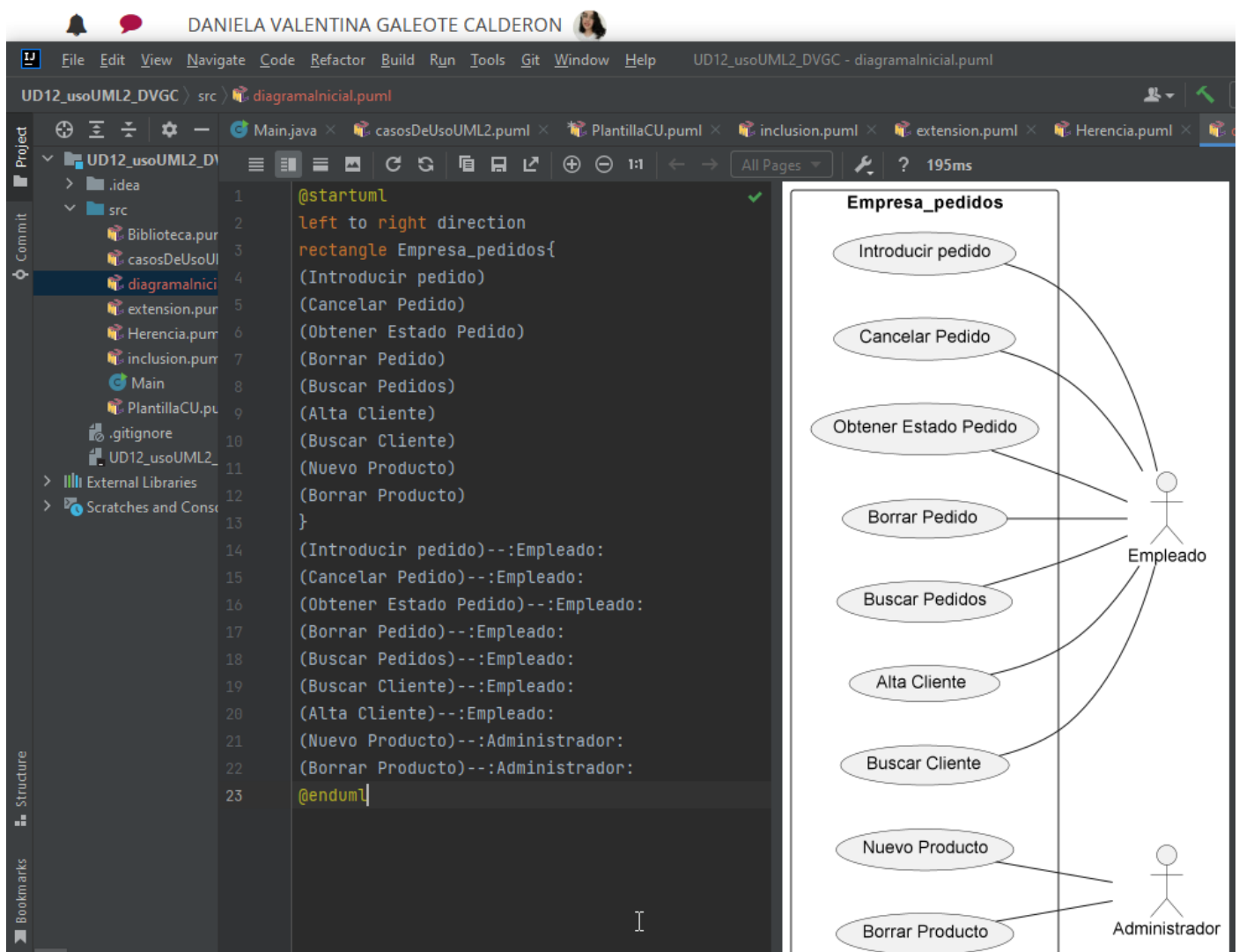
Sistema de la empresa con sus casos de uso y relacionado con los actores externos (Empleado y Administrador)

```

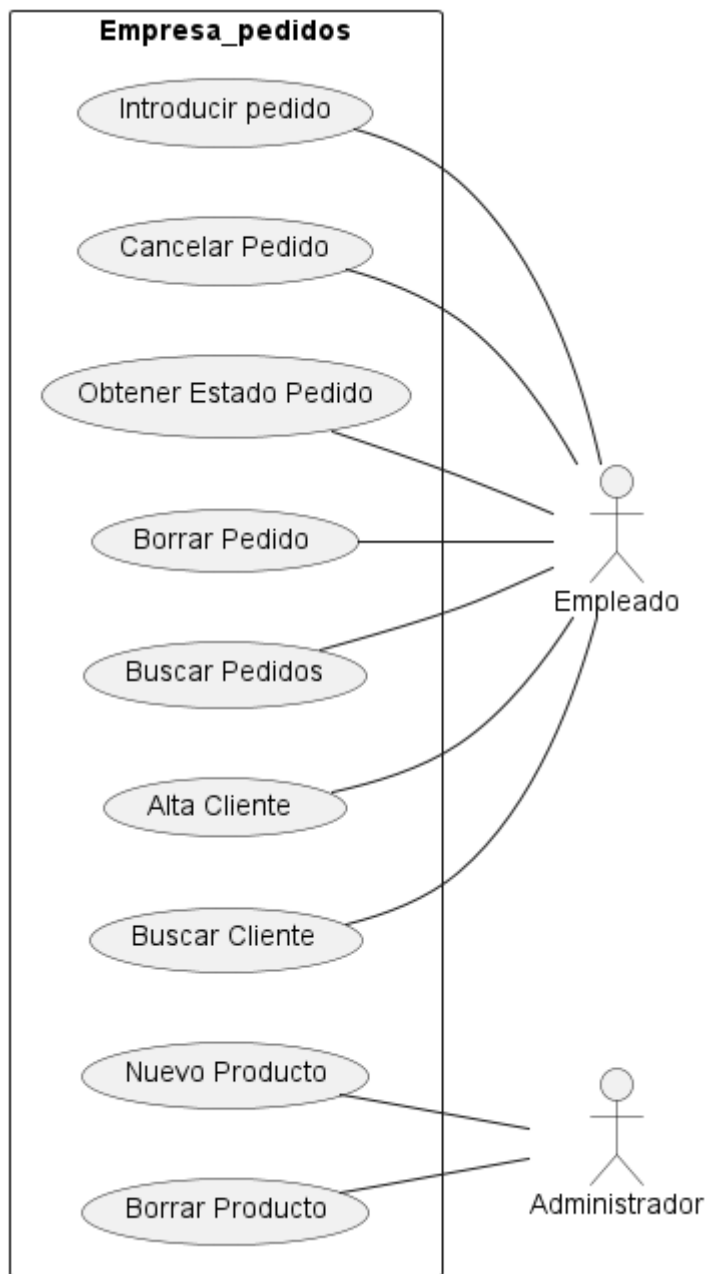
left to right direction
rectangle Empresa_pedidos{
(Introducir pedido)
(Cancelar Pedido)
(Obtener Estado Pedido)
(Borrar Pedido)
(Buscar Pedidos)
(Alta Cliente)
(Buscar Cliente)
(Nuevo Producto)
(Borrar Producto)
}
(Introducir pedido)--:Empleado:
(Cancelar Pedido)--:Empleado:
(Obtener Estado Pedido)--:Empleado:
(Borrar Pedido)--:Empleado:
(Buscar Pedidos)--:Empleado:
(Alta Cliente)--:Empleado:
(Nuevo Producto)--:Administrador:
(Borrar Producto)--:Administrador:

```

# ICFSQ Ent ...







**Diagrama estructurado:**

Similar al inicial pero con relaciones de inclusión y extensión

```

left to right direction
:Empleado:--(Alta Cliente)
:Empleado:--(Introducir pedido)
:Empleado:--(Obtener Estado Pedido)
:Empleado:--(Buscar Pedidos)
:Empleado:--(Cancelar Pedido)
(Alta Cliente)->(Introducir pedido):<<extend>>
(Introducir pedido)->(Buscar Cliente):<<include>>
(Introducir pedido)->(Pago con tarjeta):<<extend>>
(Introducir pedido)->(Pago en cuenta):<<extend>>
(Obtener Estado Pedido)->(Buscar Pedidos):<<include>>
(Cancelar Pedido)->(Buscar Pedidos):<<include>>
  
```

# 1CFSQ Ent ...

