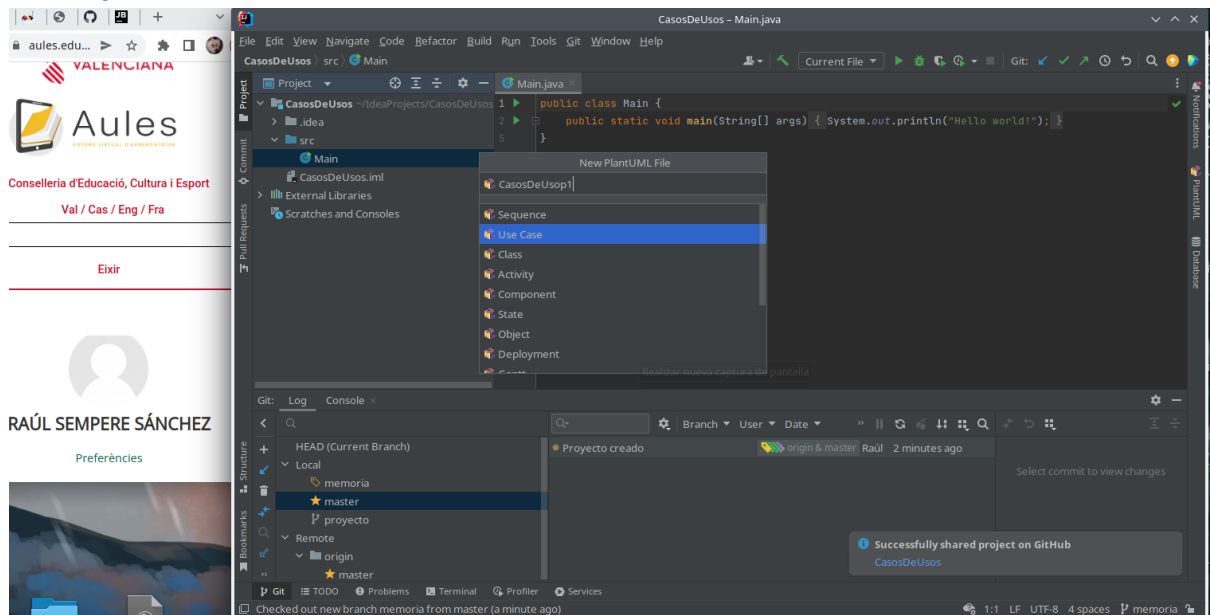


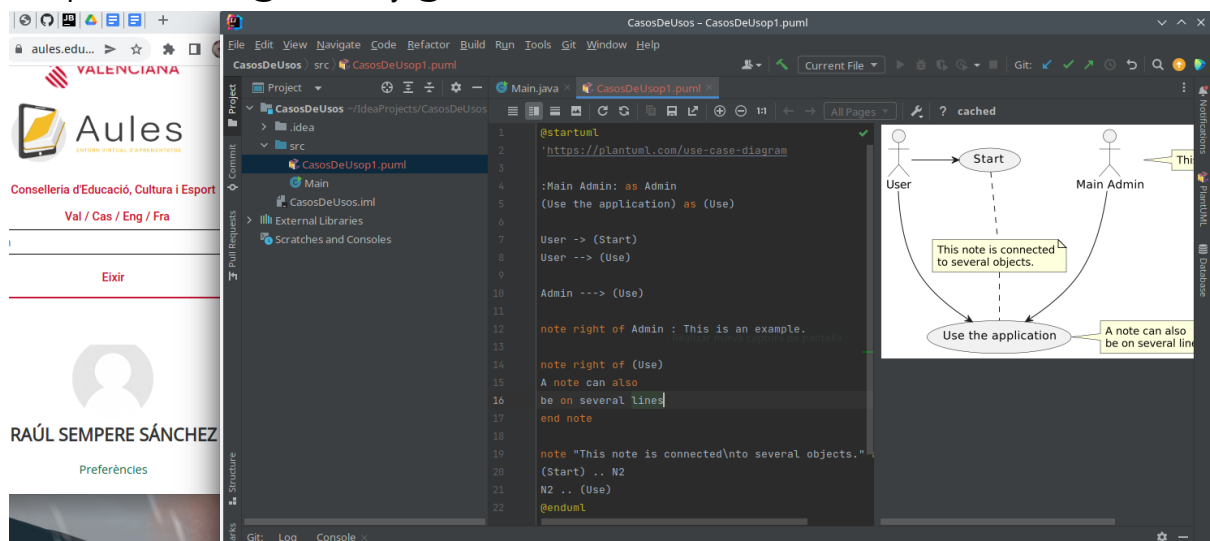
Memoria de práctica: Casos de uso

Repositorio: <https://github.com/raulico661/CasosDeUsos>

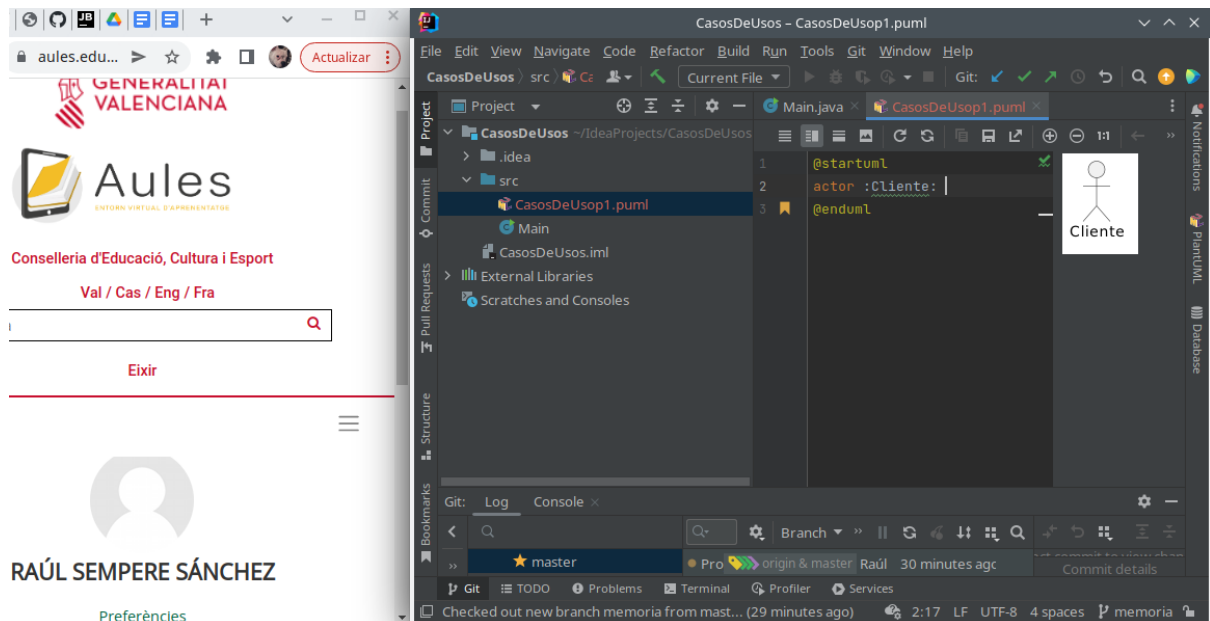
Creamos nuestro proyecto y primero vamos a crear un nuevo fichero PlantUML pero esta vez elegimos la opción "Use Case".



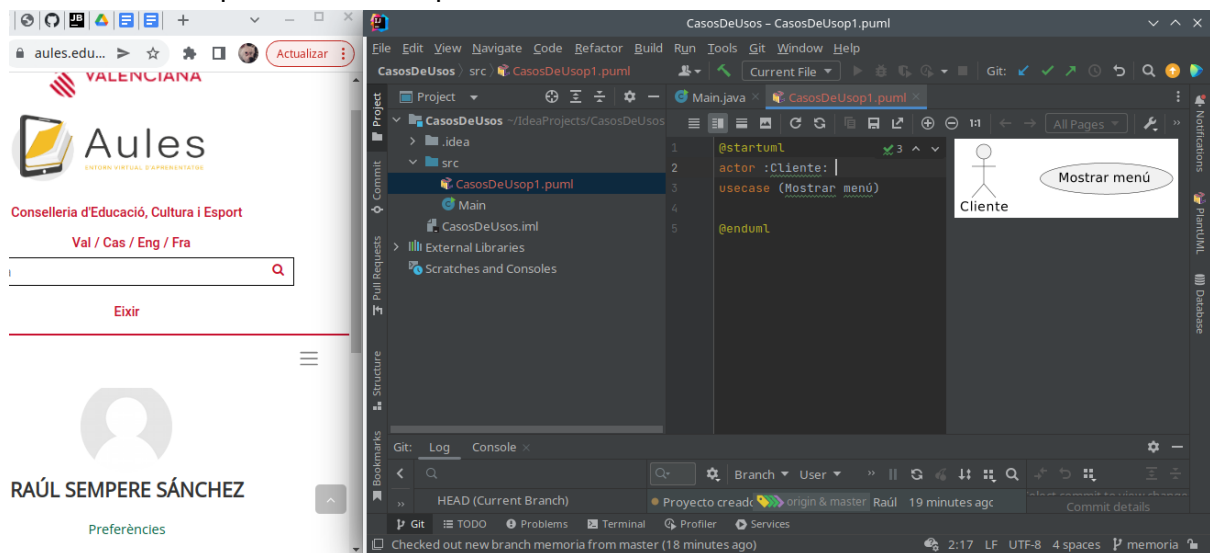
Aparecerá un código predeterminado de ejemplo una vez creado el fichero, eliminamos todo excepto las líneas `@startuml` y `@enduml`.



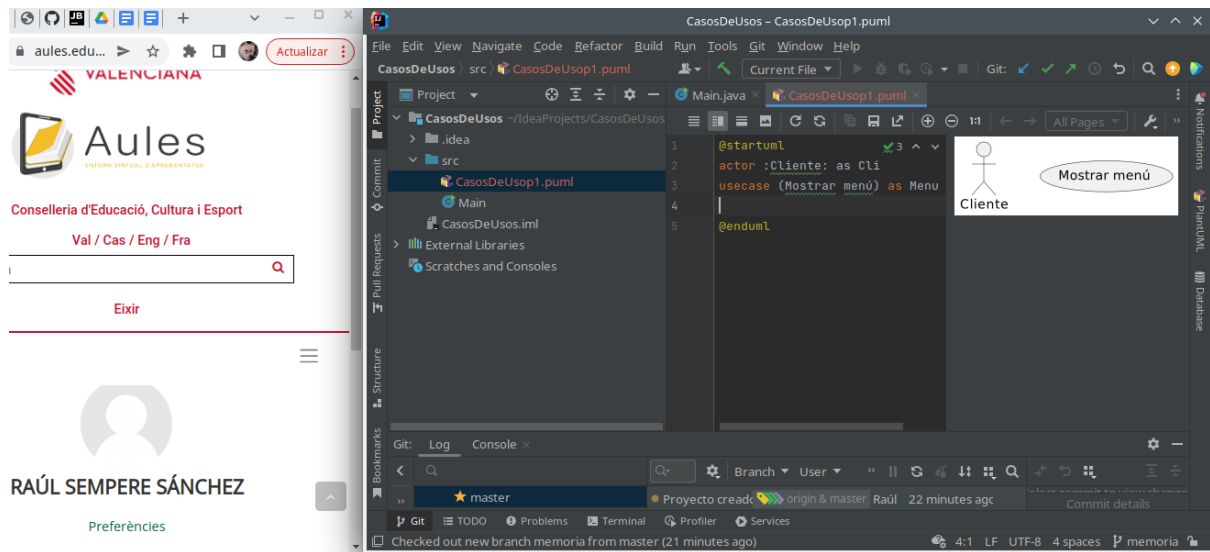
Para empezar creando nuestro diagrama primero necesitaremos un actor que son las personas o los procesos que necesitan interactuar con el sistema. Se escribe “actor :nombreActor:” pero la palabra reservada “actor” es opcional.



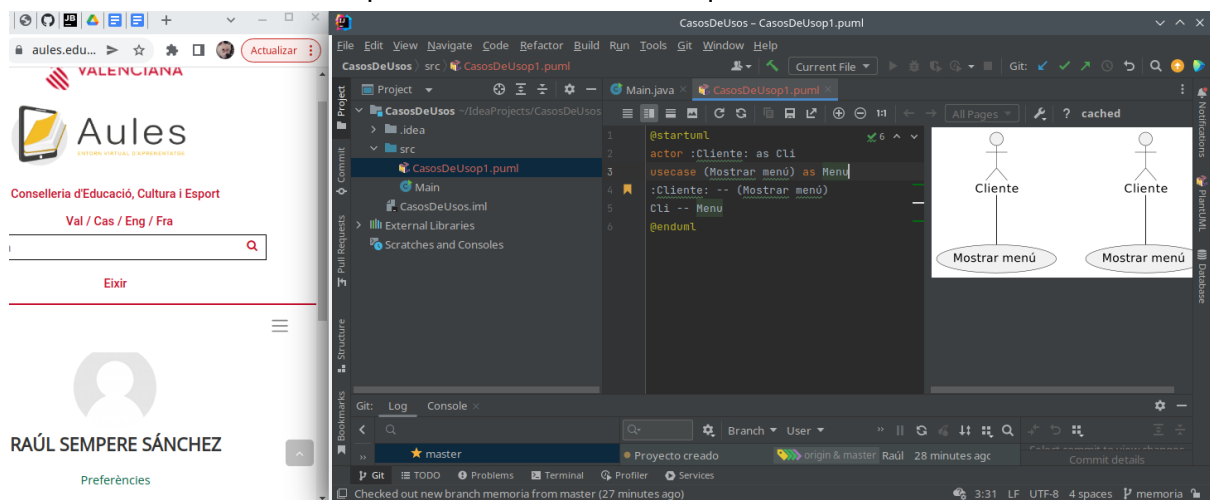
Vamos a necesitar también un caso de uso, que representan las acciones generales del sistema. Se escriben de la siguiente forma “usecase (El nombre del caso)”, la palabra reservada “usecase” es opcional, se puede escribir solo en paréntesis o en el caso de que la utilizamos los paréntesis son opcionales.



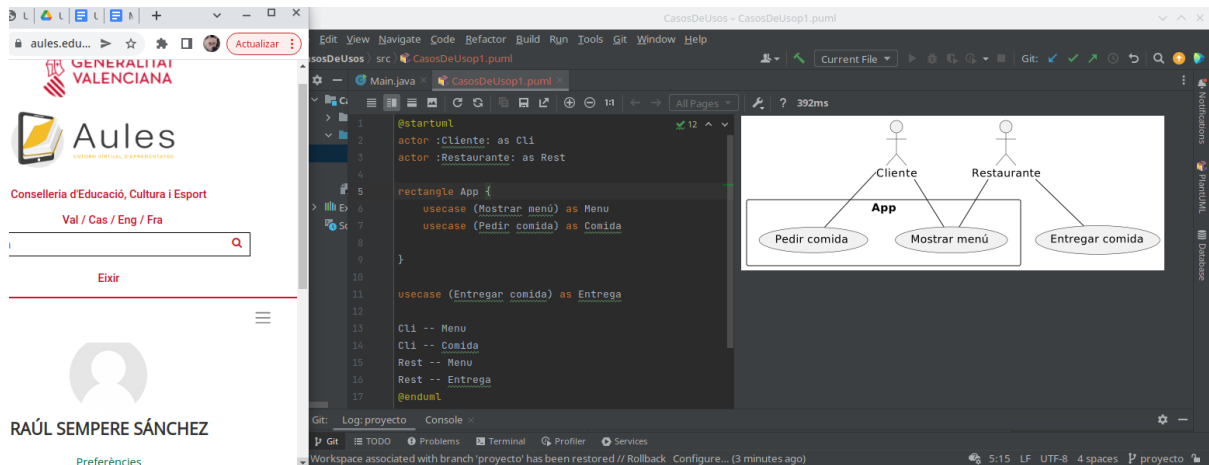
Podemos utilizar alias para acortar los nombres de los actores o casos de uso utilizando detrás de ellos la palabra “as” seguido del alias que queremos utilizar para poder utilizarlos más tarde.



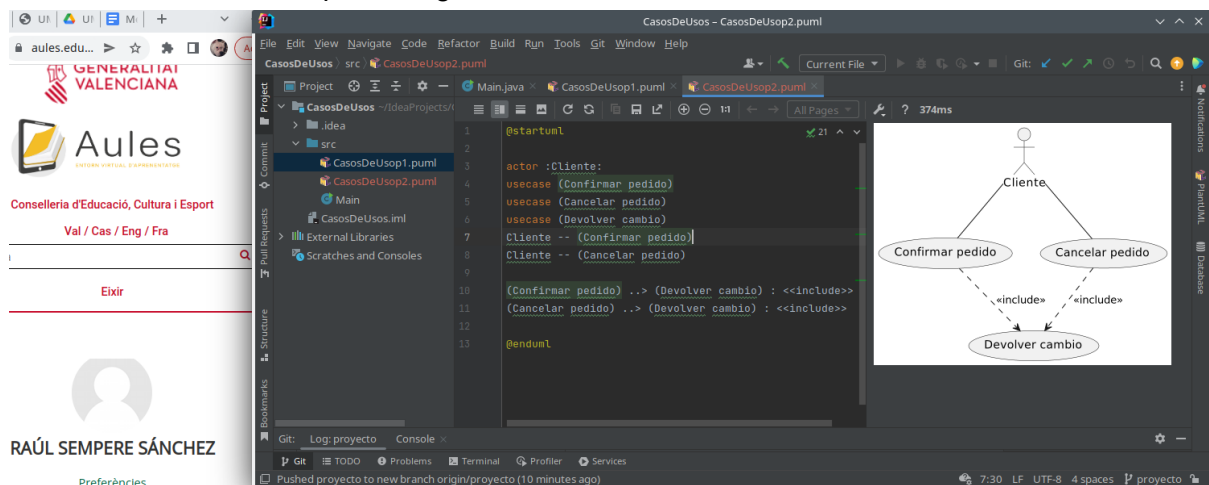
Para establecer la asociación existente entre un caso y un actor, la interacción entres estos dos, podemos utilizar tanto los alias como los nombres pero estos últimos deben ir acompañados de sus “:” o “()” según sean actores o casos respectivamente. Se escriben de la siguiente forma “:actor: - - (caso)” o “aliasActor - - aliasCaso” en caso de haber utilizado alias. En nuestro caso nos quedaremos con la forma que utiliza los alias.



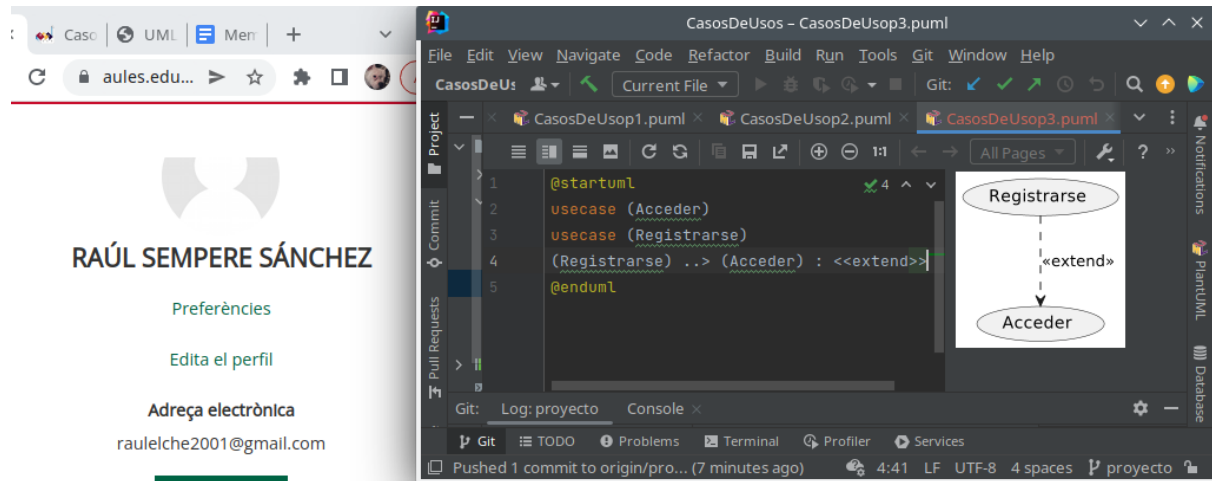
Ahora vamos a añadir el sistema al diagrama, es decir, el software que vamos a desarrollar, para ello hemos añadido más actores y casos de uso. Escribiremos “rectangle nombre {}” el nombre es opcional y dentro de los corchetes tendremos que declarar los casos de uso de nuestro diagrama como se puede ver en la captura.



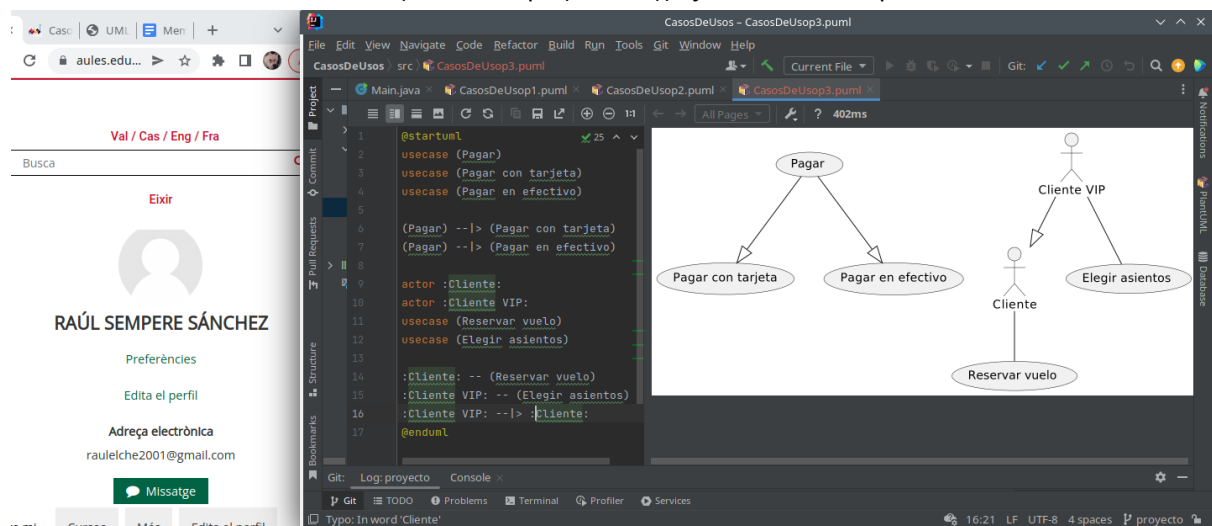
Cuando el comportamiento de un caso de uso se incluye dentro del comportamiento de otro utilizamos la inclusión. En este ejemplo, hemos relacionado a un cliente con dos acciones cancelar o confirmar el pedido, estas dos opciones incluyen la acción devolver el cambio del producto, por lo que estas dos acciones incluyen a esta última y se escribe así “(Caso) ..> (CasoIncluido)”. Para poner una etiqueta en la relación como hemos hecho en la captura escribimos “: nombreEtiqueta” seguido de la inclusión.



La extensión se utiliza cuando un caso tiene un comportamiento adicional en determinadas ocasiones. Esta se representa de la misma forma que la inclusión “(Caso1) ..> (Caso2)” y también se le puede añadir una etiqueta.



La generalización se utiliza para expresar que un caso de uso especializado es una forma particular de conseguir los objetivos de otro caso de uso más general, también se puede usar con actores. Se escribe “(Caso1 -|> (Caso2))” y “:Actor1: -|> :Actor2:”.



Habiendo explicado todo esto podemos realizar los ejemplos de la teoría, este es el primer ejemplo

The screenshot shows an IDE with a project named 'CasosDeUsos - Ejemplo1.puml'. On the left, there is a sidebar with a user profile for 'RAÚL SEMPERE SÁNCHEZ' and a list of use cases: 'Meter dinero', 'Pedir azúcar', 'Pedir producto', 'Cancelar', and 'Devolver dinero'. The main editor displays the UML Use Case diagram for 'Cliente'. The diagram shows a central actor 'Cliente' connected to five use cases: 'Meter dinero', 'Pedir azúcar', 'Pedir producto', 'Cancelar', and 'Devolver dinero'. There are also include relationships: 'Pedir producto' includes 'Devolver dinero', and 'Cancelar' includes 'Devolver dinero'. The code editor on the right shows the corresponding Java code for the diagram.

```
@startuml
actor :Cliente:
usecase (Meter dinero)
usecase (Pedir azúcar)
usecase (Pedir producto)
usecase (Cancelar)
usecase (Devolver dinero)

:Cliente -- (Meter dinero)
:Cliente -- (Pedir azúcar)
:Cliente -- (Pedir producto)
:Cliente -- (Cancelar)

(Pedir producto) ..> (Devolver dinero) : <<include>>
(Cancelar) ..> (Devolver dinero) : <<include>>

@enduml
```

El segundo ejemplo.

The screenshot shows an IDE with a project named 'CasosDeUsos - Ejemplo2.puml'. On the left, there is a sidebar with a user profile for 'RAÚL SEMPERE SÁNCHEZ' and a list of use cases: 'Elegir mensajero', 'Confirmar', 'Identificarse', 'Pagar', 'Registrarse', 'Pagar con tarjeta', and 'Pagar con transferencia'. The main editor displays the UML Use Case diagram for 'new Actor'. The diagram shows a central actor 'new Actor' connected to seven use cases: 'Elegir mensajero', 'Confirmar', 'Identificarse', 'Pagar', 'Registrarse', 'Pagar con tarjeta', and 'Pagar con transferencia'. There are also include relationships: 'Registrarse' includes 'Identificarse', 'Pagar con tarjeta' includes 'Pagar', and 'Pagar con transferencia' includes 'Pagar'. The code editor on the right shows the corresponding Java code for the diagram.

```
@startuml
actor :new Actor:
usecase (Elegir mensajero)
usecase (Confirmar)
usecase (Identificarse)
usecase (Pagar)
usecase (Registrarse)
usecase (Pagar con tarjeta)
usecase (Pagar con transferencia)

:new Actor -- (Elegir mensajero)
:new Actor -- (Confirmar)
:new Actor -- (Identificarse)
:new Actor -- (Pagar)
:new Actor -- (Registrarse)
:new Actor -- (Pagar con tarjeta)
:new Actor -- (Pagar con transferencia)

(Registrarse) ..> (Identificarse) : <<include>>
(Pagar con tarjeta) ..> (Pagar) : <<include>>
(Pagar con transferencia) ..> (Pagar) : <<include>>

@enduml
```

El tercer ejemplo.

The screenshot shows an IDE with a project named 'CasosDeUsos - Ejemplo3.puml'. On the left, there is a sidebar with a user profile for 'RAÚL SEMPERE SÁNCHEZ' and a list of use cases: 'Iniciar sesión', 'Recuperar contraseña', 'Modificar contraseña', 'Cerrar sesión', 'Registrar usuario', and 'Instalar programa'. The main editor displays the UML Use Case diagram for 'Administrador' and 'Usuario'. The diagram shows two actors: 'Administrador' and 'Usuario'. 'Administrador' is connected to 'Registrar usuario' and 'Instalar programa'. 'Usuario' is connected to 'Iniciar sesión', 'Recuperar contraseña', 'Modificar contraseña', and 'Cerrar sesión'. There are also include relationships: 'Registrar usuario' includes 'Iniciar sesión', 'Recuperar contraseña', 'Modificar contraseña', and 'Cerrar sesión'. The code editor on the right shows the corresponding Java code for the diagram.

```
@startuml
actor :Administrador:
actor :Usuario:
usecase (Iniciar sesión)
usecase (Recuperar contraseña)
usecase (Modificar contraseña)
usecase (Cerrar sesión)
usecase (Registrar usuario)
usecase (Instalar programa)

:Administrador -- (Registrar usuario)
:Administrador -- (Instalar programa)

:Usuario -- (Iniciar sesión)
:Usuario -- (Recuperar contraseña)
:Usuario -- (Modificar contraseña)
:Usuario -- (Cerrar sesión)

(Registrar usuario) ..> (Iniciar sesión) : <<include>>
(Registrar usuario) ..> (Recuperar contraseña) : <<include>>
(Registrar usuario) ..> (Modificar contraseña) : <<include>>
(Registrar usuario) ..> (Cerrar sesión) : <<include>>

@enduml
```

Y el cuarto y último ejemplo.

The screenshot shows a web browser on the left displaying a user profile for 'RAÚL SEMPÈRE SÁNCHEZ' with options like 'Preferències', 'Edita el perfil', and 'Adreça electrònica'. On the right, a UML editor window titled 'CasosDeUsos - Ejemplo4.puml' shows a use case diagram and its corresponding Puml code. The diagram features two actors, 'Ciudadano' and 'Ciudadano extranjero', both connected to 'Registrar entrada'. 'Ciudadano extranjero' is also connected to 'Tomar huellas', which in turn connects to 'Registrar entrada' via an '«extend»' relationship. The Puml code defines these actors, use cases, and relationships, including an '«extend»' block for the 'Tomar huellas' use case.

Por último vamos a crear puntos de extensión que se utilizan en las relaciones de extensión para indicar en qué punto del caso base se inserta el comportamiento del caso extendido, y bajo qué condición. Crearemos los casos de uso, si insertamos un as podemos escribir más texto en la burbuja, con skinparam podremos añadirle colores y editar su apariencia y añadiremos una nota que explique la condición que debe ocurrir para que esto se produzca. Para ello creamos una nota con note, escribimos la condición en ella y la conectamos como si fuera un actor o un caso de uso más entre los casos de uso como vemos en la captura.

The screenshot shows a UML editor window titled 'CasosDeUsos - CasosDeUso4.puml'. The diagram features two use cases, 'Sacar dinero + solicitud hecha' and 'Retener tarjeta', connected by an '«extend»' relationship. A note box is connected to the '«extend»' relationship, containing the text: 'condición: {historia sospechosa} punto de extensión: solicitud hecha'. The Puml code defines these use cases, a skinparam for the 'Sacar dinero' use case, and the '«extend»' relationship with a note.