

ENTORNOS DE DESARROLLO

UD 12: Prácticas Diagramas UML casos de uso

Francisco Salar Pérez 1º S

Práctica clases de uso con PlantUml.

Utilizando como base el archivo que hay al final de este enunciado crea los correspondientes diagramas de casos de uso que hay a lo largo de los apuntes utilizando la herramienta PlantUml.

En la UD anterior puedes encontrar instrucciones precisas de como instalar el plugin PlantUml en IntelliJ, que es el entorno donde debes realizar los diagramas de casos de uso.

Entrega:

Crea un repositorio con dos ramas.

La primera rama debe llamarse "proyecto" y en esta debe incluirse el proyecto creado y los commits del paso a paso del trabajo realizado. En la segunda rama que debe llamarse "memoria" incluye el PDF de la memoria acabada

Adjunta en el lugar destinado para tal fin la URL al repositorio donde se encuentran las dos ramas.

Entrega también en el lugar habilitado para tal fin la memoria con extensión .pdf, es decir el archivo de la memoria que debe estar en la rama llamada "memoria".

Recuerda que toda práctica debe cumplir las condiciones y formas que se pueden leer en el punto.

"Información sobre la evaluación del tema." (¡¡NO FUNCIONA!!)

Recuerda también que para el manejo del CVS puedes utilizar la herramienta que estimes oportuna.

Aquí te dejo un enlace al manual de PlantUml relacionado con la creación de diagramas de casos de uso.

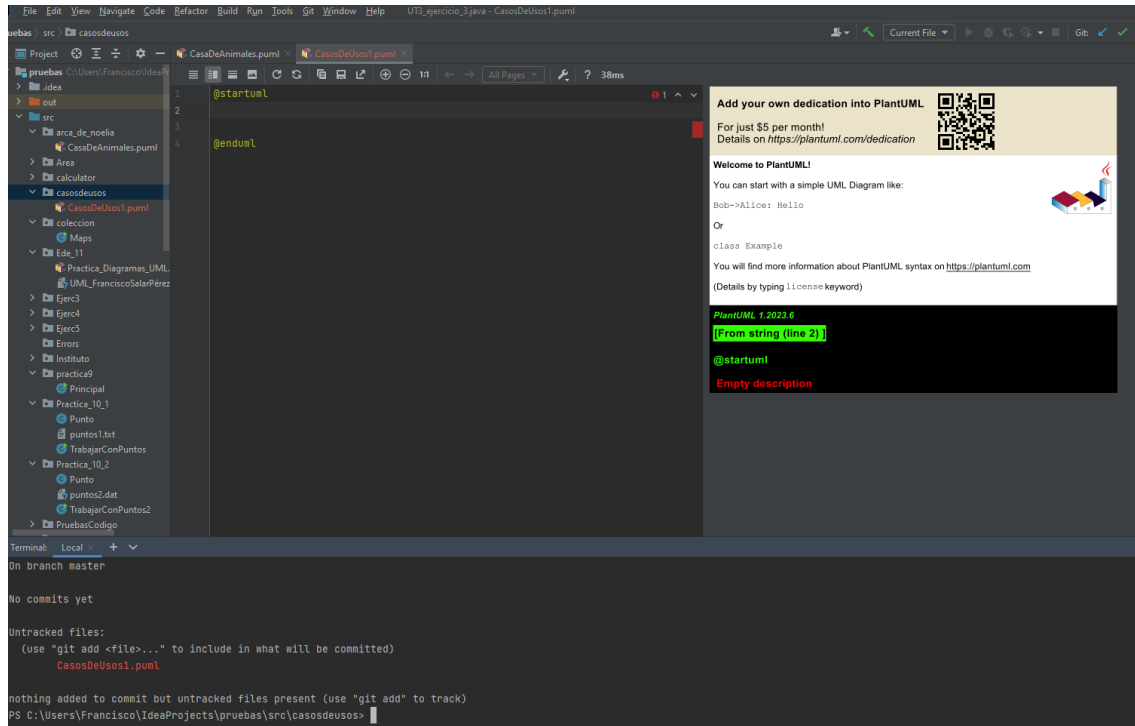
Incluye múltiples ejemplos.

<https://plantuml.com/es/use-case-diagram>

Solución:

Enlace al repositorio en GitHub: <https://github.com/CorazaNegra/CasosDeUsos1.git>

Primero creamos el package “casos_de_usos_1” donde albergaremos el archivo “CasosDeUsos1.puml”, hacemos click secundario en nuestro package y seleccionamos en el nuevo dialogo la opción New PlantUML File, escogemos la opción de “Use Case” y aceptamos.



1. Diagramas de casos de uso

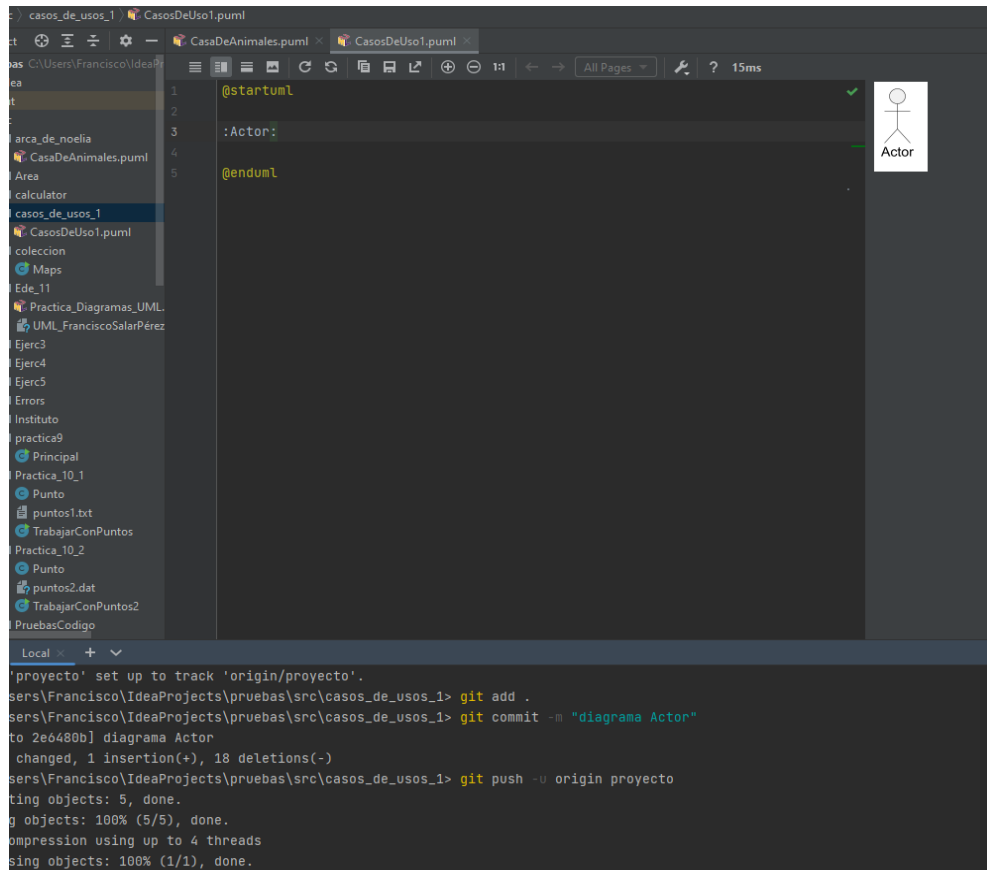
Los diagramas de casos de uso son un tipo de diagrama de comportamiento que sirve para describir lo que debe hacer un sistema desde el punto de vista de quien lo va a utilizar.

Un modelo de casos de uso se construye mediante un proceso iterativo durante las reuniones entre los desarrolladores del sistema y los clientes (y/o los usuarios finales) conduciendo a una especificación de requisitos sobre la que todos coinciden.

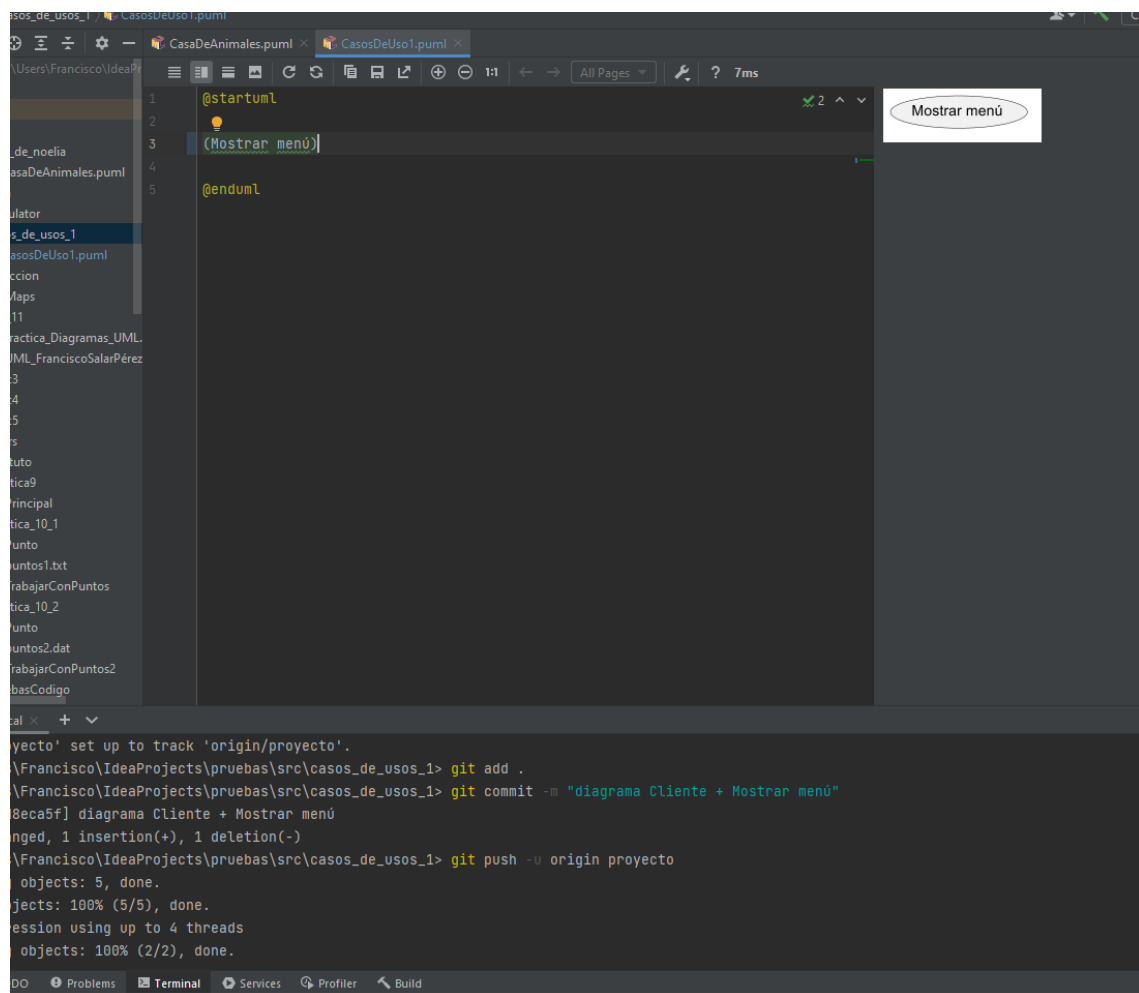
Un caso de uso captura algunas de las acciones y comportamientos del sistema y cómo los actores interactúan con él.

Elementos del diagrama

Actor: Los actores son personas o procesos automáticos que necesitan interactuar con el sistema. Se deben identificar sus papeles en el sistema. En el diagrama, se representan del siguiente modo:



Caso de uso: Los casos de uso se representan mediante elipses y corresponden a acciones generales del sistema. Una forma de reconocerlos es que suelen ser verbos en la descripción del caso de uso.



Asociación: La interacción entre los actores y los casos de uso del sistema se representan por una línea recta que une a ambos.

The screenshot shows an IDE with a UML Use Case diagram and a terminal window. The diagram, titled 'CasosDeUso1.puml', shows an actor named 'Cliente' connected to a use case named 'Mostrar menú'. The code in the background is as follows:

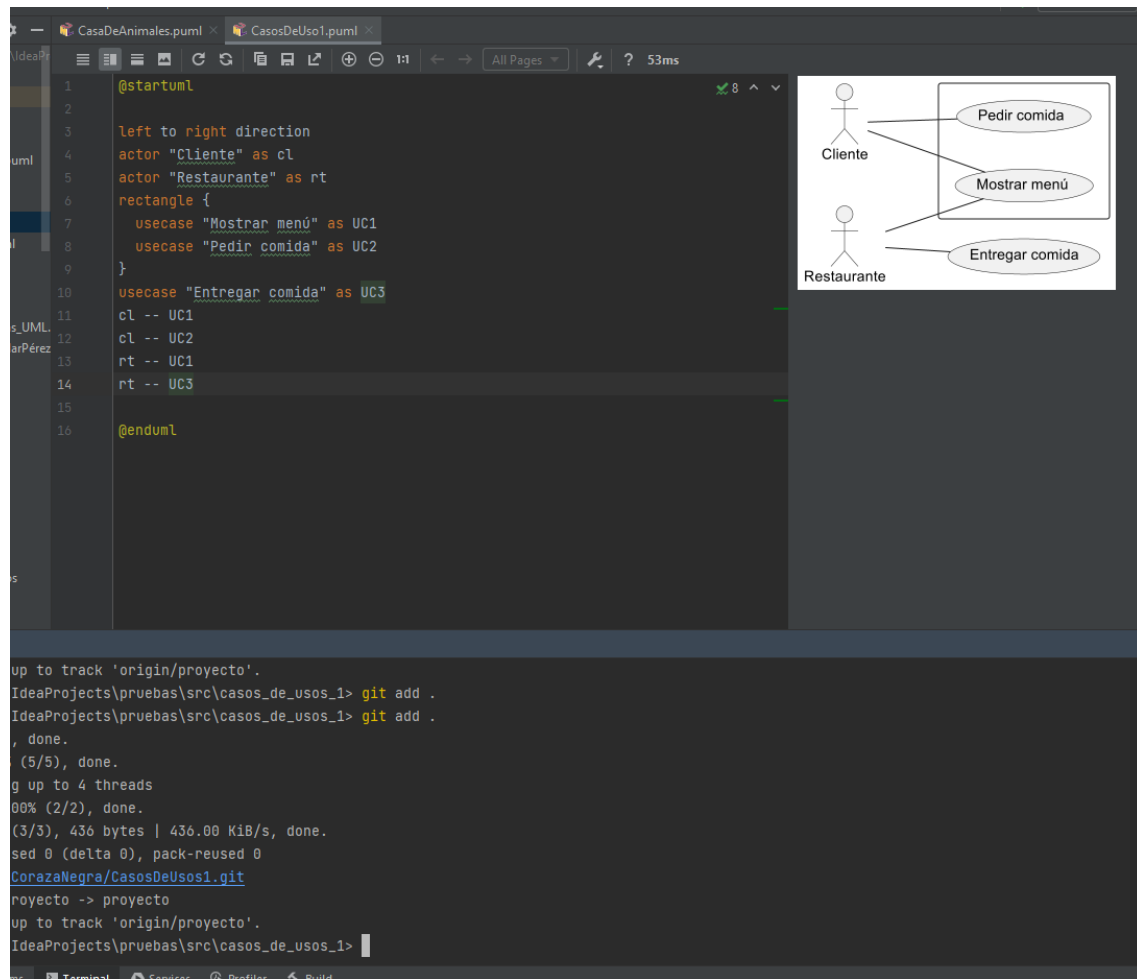
```
@startuml
actor Cliente
usecase U1 as "Mostrar menú"
Cliente --> U1
@enduml
```

The terminal window shows the following commands and output:

```
branch 'proyecto' set up to track 'origin/proyecto'.
PS C:\Users\Francisco\IdeaProjects\pruebas\src\casos_de_usos_1> git add .
PS C:\Users\Francisco\IdeaProjects\pruebas\src\casos_de_usos_1> git commit -m "diagrama Cliente + Mostrar menú"
[proyecto d8eca5f] diagrama Cliente + Mostrar menú
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Francisco\IdeaProjects\pruebas\src\casos_de_usos_1> git push -u origin proyecto
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
```

A message at the bottom of the terminal indicates a VCS root configuration problem: "The directory <Project>\src\casosdeusos is registered as a Git root, but no Git repositories were found there. // <Project>\src\casos_de_usos_1 // Configure... (14 minutes ago)".

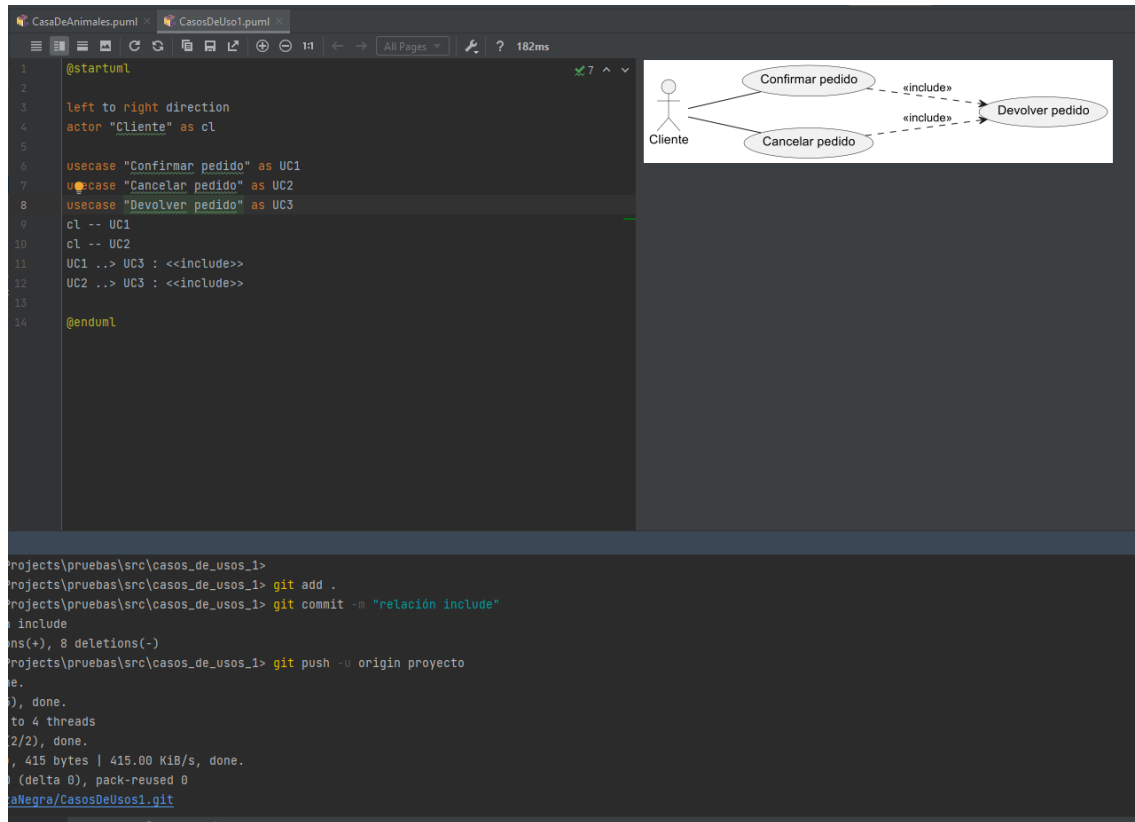
Sistema: El sistema es el software que vamos a desarrollar. Puede ser un pequeño componente cuyos actores son otros componentes, o puede ser una aplicación completa. Se representa como una caja rectangular. Dentro de ella se incluyen los casos de uso soportados por el sistema.



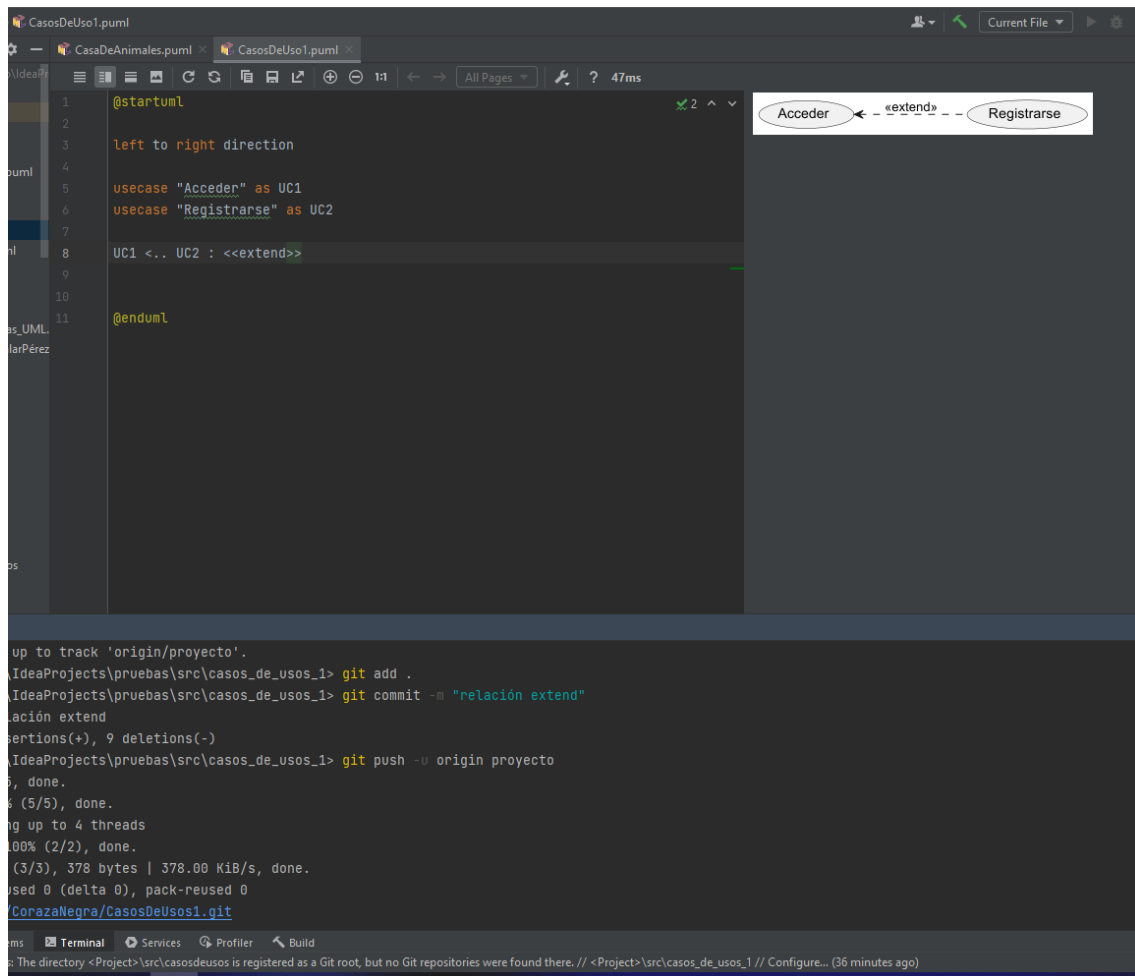
Inclusión: se utiliza cuando el comportamiento de un caso de uso se incluye dentro del comportamiento de otro. Se representa con una flecha de trazo discontinuo desde el caso que incluye hasta el caso incluido, con el estereotipo «include» o «use»

Los casos de uso incluidos se pueden compartir, así evitamos repetirlos.

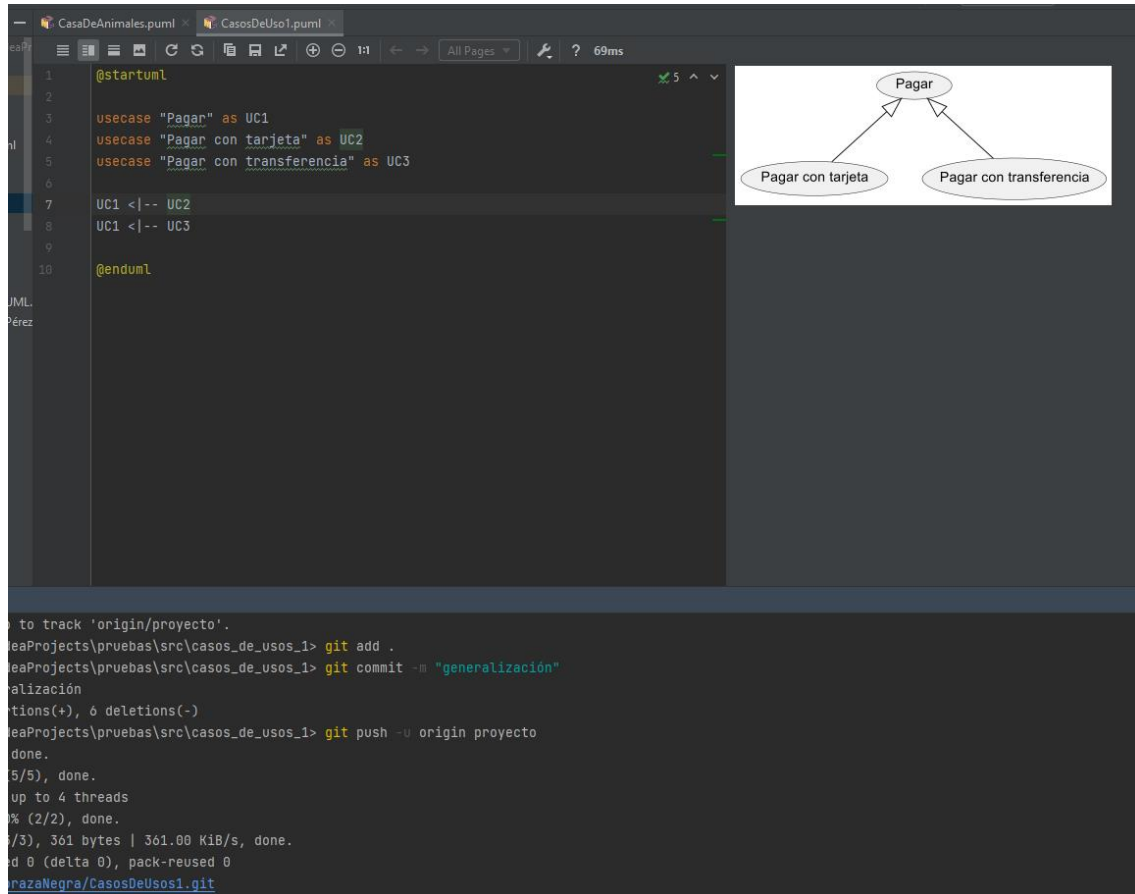
También se pueden utilizar para estructurar el diagrama en varios niveles de detalle, pero no conviene abusar de ellos (ver recomendaciones más abajo).



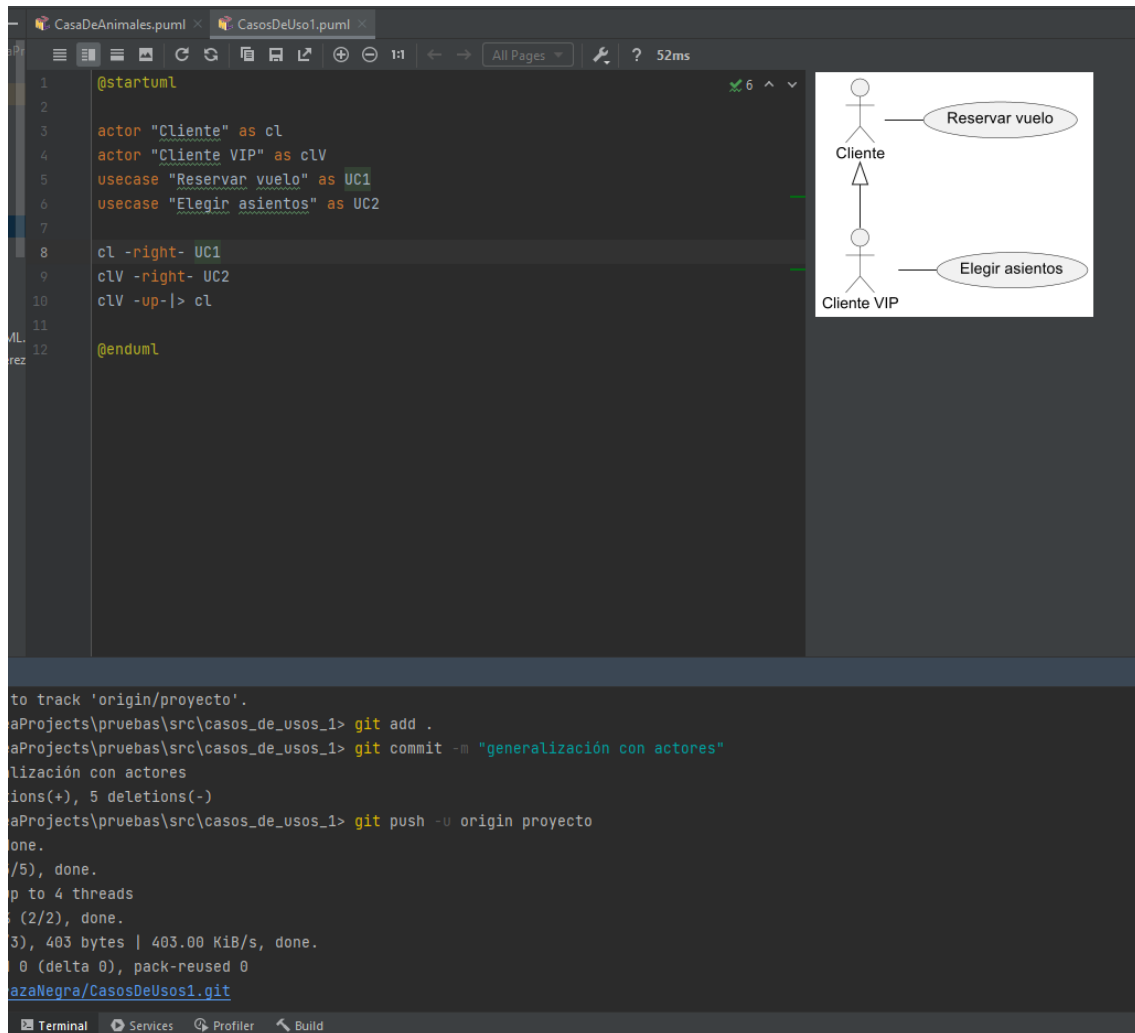
Extensión: se utiliza cuando un caso además aporta un comportamiento adicional en determinadas circunstancias o cuando se cumple cierta condición. Se representa con una flecha de trazo discontinuo que apunta al caso que queremos extender, y el estereotipo «extend».



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. Se representa como una flecha continua acabada en punta triangular hueca que apunta al caso más general.



También se puede utilizar con actores:



¿Cómo distinguir las relaciones entre casos?

A veces no está del todo claro cuándo debemos usar la extensión, generalización o inclusión. En caso de duda, podemos guiarnos por las siguientes reglas:

- La inclusión equivale a “copiar y pegar” un caso de uso dentro de otro. Si imaginamos los casos de uso como líneas de un programa, sería equivalente a incluir el código de un caso dentro del otro.
- La extensión se utiliza cuando un caso añade funcionalidad a otro dependiendo de alguna condición. Si imaginamos los casos de uso como líneas de un programa, la relación “B extiende a A” (flecha apuntando de B a A) equivaldría a introducir una condición if en algún punto de A, de forma que si se cumple la condición, se ejecuta B antes de proseguir con A.
- La generalización sirve para indicar que varios casos tienen el mismo comportamiento pero lo llevan a cabo de distinto modo. Equivale a la herencia en programación orientada a objetos, donde las clases hijas pueden rescribir parte del código de la clase padre.

Recomendaciones

Algunas recomendaciones a la hora de elaborar casos de uso:

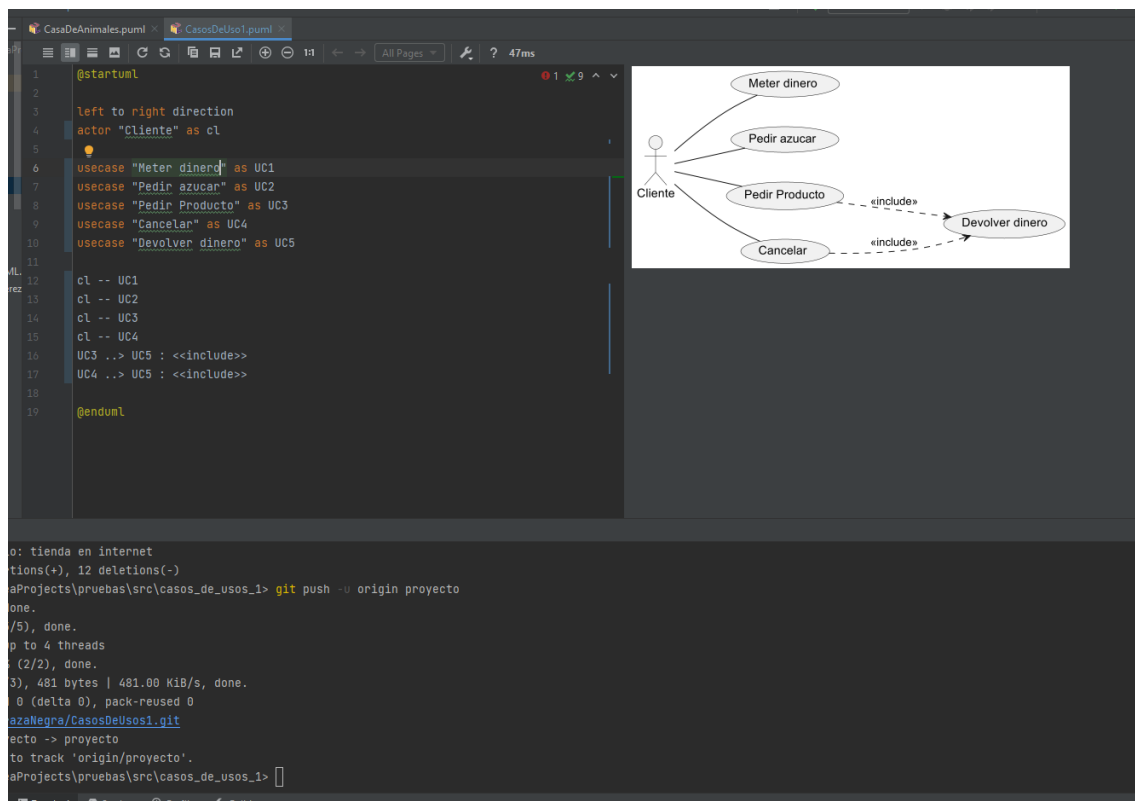
- Los casos de uso describen las interacciones más importantes con el sistema, no su funcionamiento interno.
- Conviene mantener los casos de uso lo más simples posibles. Todos los implicados en el proyecto deben ser capaces de entender el diagrama de casos de uso. Si se complica demasiado, pierde su utilidad.
- Hay que tener en cuenta que el diagrama de casos de uso se elabora durante la fase de especificación de requisitos y que éstos son cambiantes por naturaleza. Por ello no conviene ahondar demasiado en detalles ni descomponer cada caso en sub-casos, ya que si éstos cambian habremos malgastado el tiempo.
- En cualquier caso, es preferible acompañar al diagrama de una buena descripción narrativa del caso de uso.

Ejemplo: La máquina de café

Supongamos que se requiere desarrollar el control de una máquina de entrega de café automática.

La máquina debe permitir a una persona introducir dinero, escoger uno de los productos de acuerdo a su precio, escoger un nivel de azúcar y entregar el producto y las vueltas.

El usuario puede en cualquier momento antes de escoger el azúcar cancelar la operación, mediante un botón existente para este objetivo.



El diagrama hace uso de la relación «include» para reutilizar el caso de uso “Devolver dinero”:

Ejemplo: Tienda en Internet

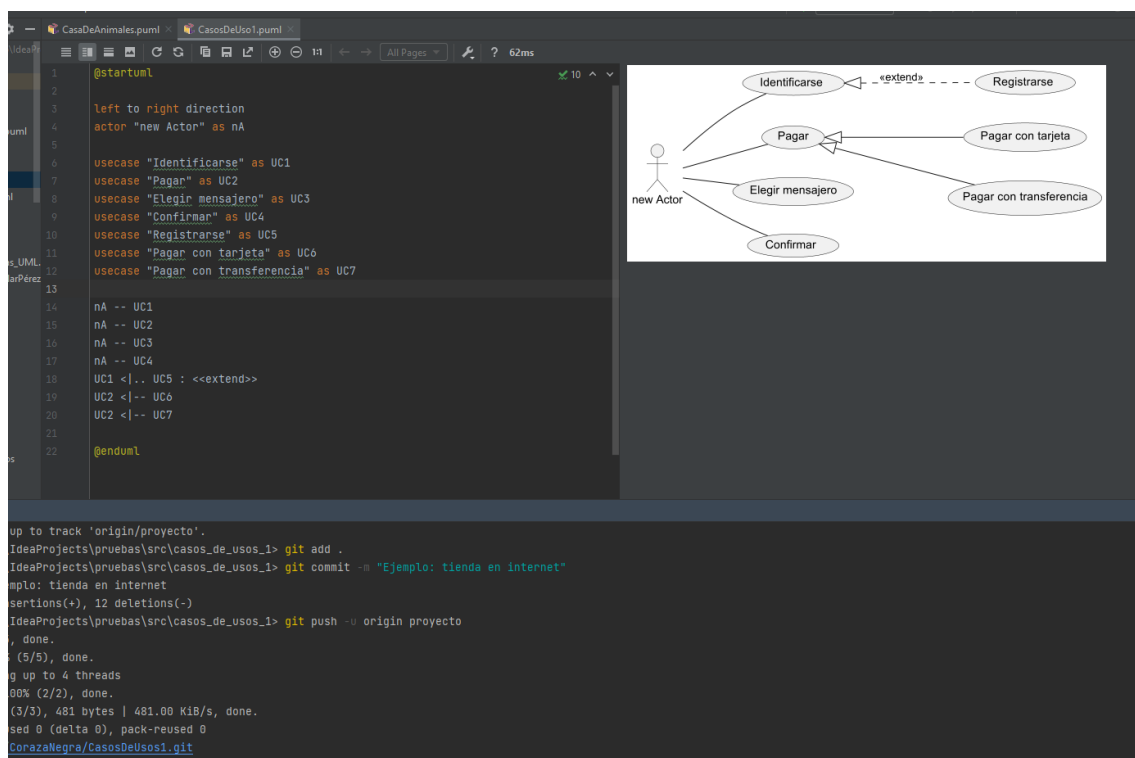
Queremos modelar el sistema de pago en una tienda web.

El cliente debe identificarse mediante su dirección de correo. Si es un nuevo cliente se le debe registrar en el sistema previamente, pidiéndole los datos personales.

Una vez identificado al cliente, éste podrá elegir el medio de pago: por transferencia bancaria o con tarjeta de crédito. Según el medio de pago se le solicitarán unos datos u otros.

El cliente también deberá elegir el método de envío.

Finalmente, se le mostrarán todos los datos del pedido para pedirle que confirme.



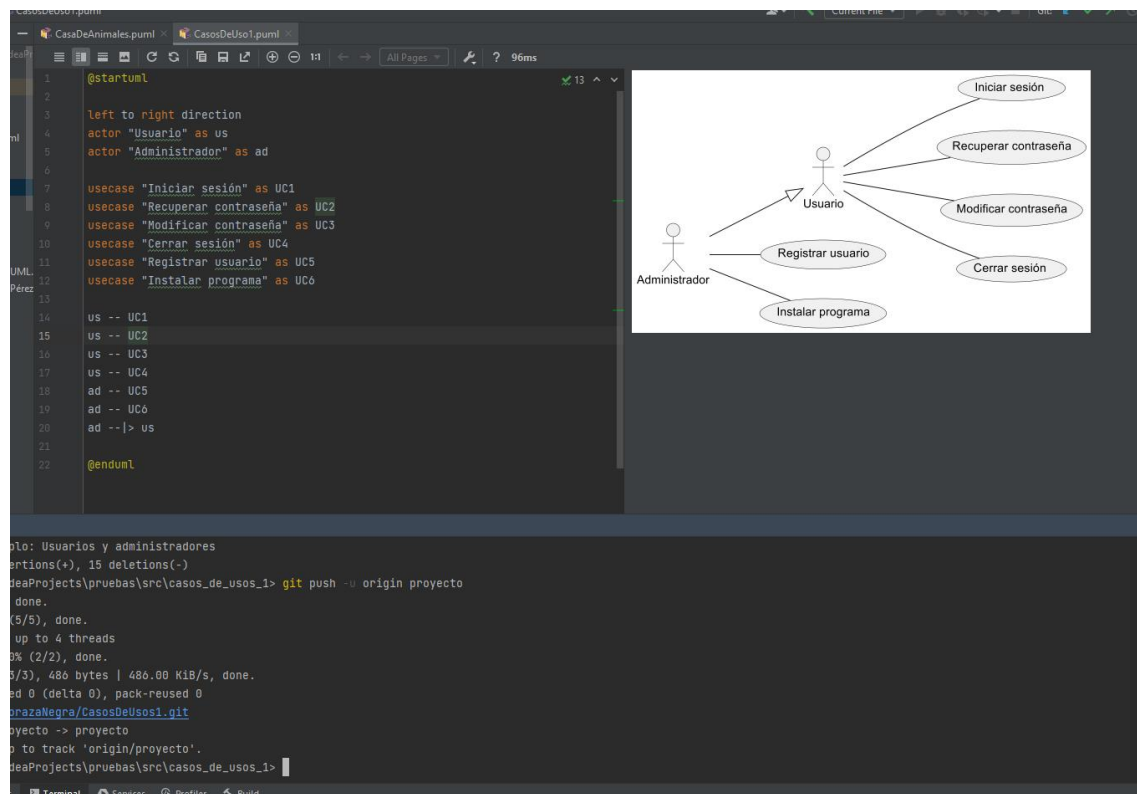
En este diagrama se puede observar la descomposición del caso general “Pagar” en los casos específicos “Pagar con tarjeta” y “Pagar con transferencia” mediante una generalización.

El caso “Registrarse” extiende a “Identificarse” porque está sujeto a la condición “si es un nuevo cliente”.

Ejemplo: Usuarios y administradores

Queremos modelar un sistema en el que hay usuarios. Los usuarios pueden iniciar sesión, modificar su contraseña, recuperar su contraseña y cerrar sesión.

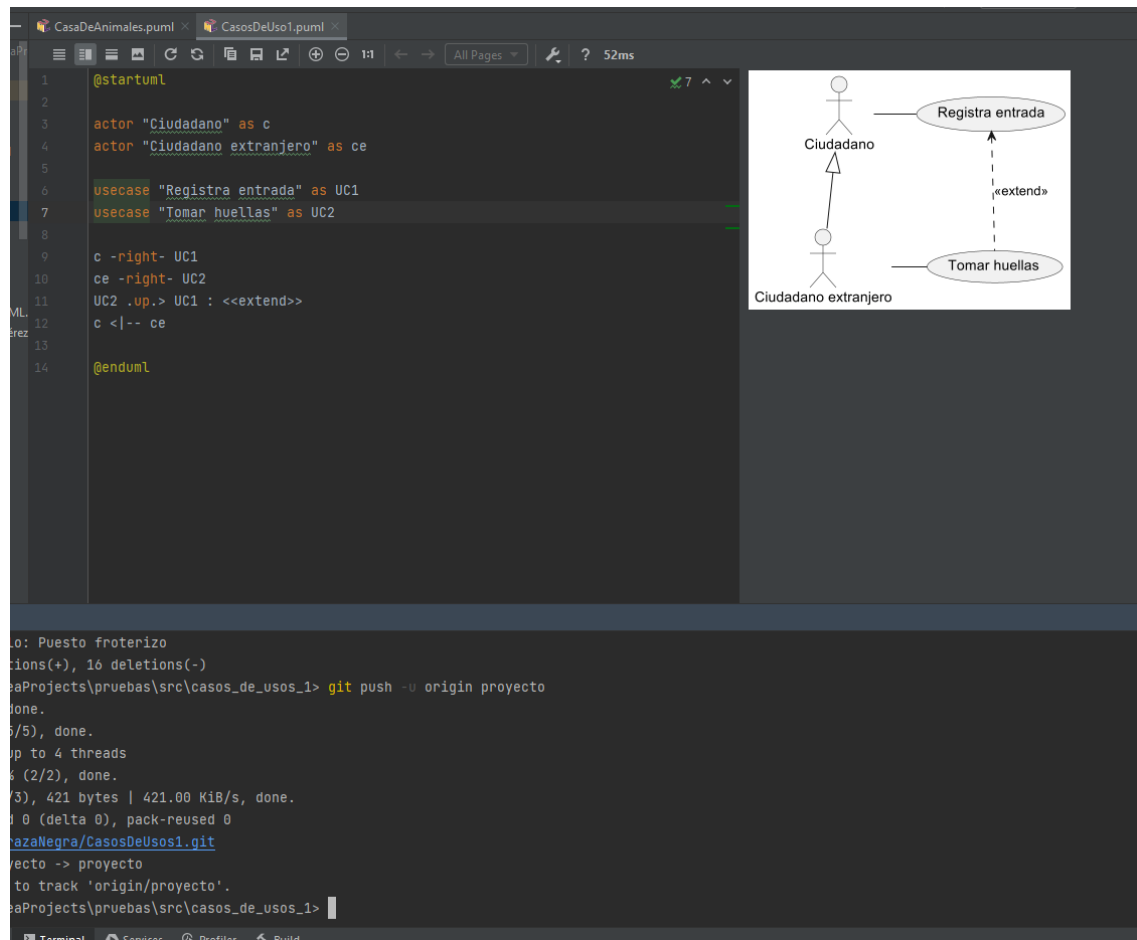
Los administradores tienen los mismos permisos que los usuarios, pero además, pueden registrar usuarios e instalar programas.



En este ejemplo nuevamente se utiliza la generalización, esta vez entre actores, para indicar que los administradores son un tipo específico de usuario.

Ejemplo: Puesto fronterizo

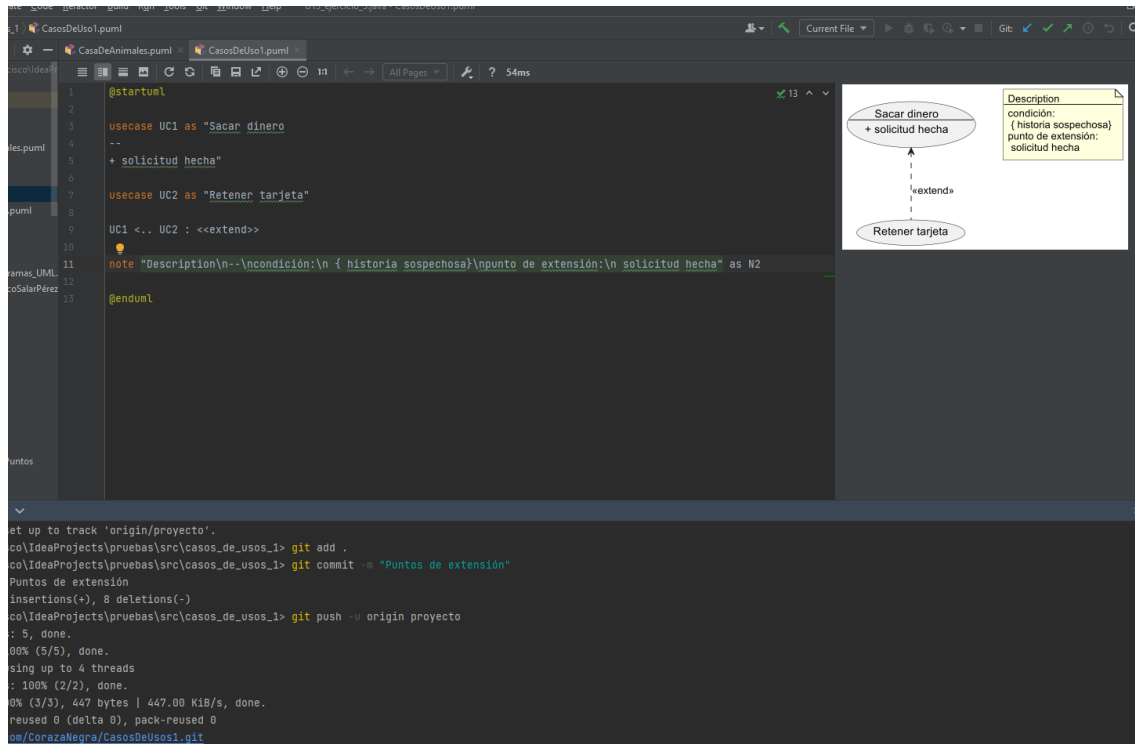
En la frontera de un país se registran todos los ciudadanos que entran. Además, en caso de que el ciudadano sea extranjero, se le toma la huella dactilar.



Nuevamente utilizamos la relación «extend» para indicar que la toma de huellas se realiza como parte del registro de entrada cuando se da la condición de que el ciudadano es extranjero, lo que expresamos añadiendo el actor "ciudadano extranjero" que interactúa con este caso de uso.

Puntos de extensión

Los puntos de extensión 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. Veámoslo con un ejemplo:



En el caso base (Sacar dinero) hemos añadido un punto de extensión solicitud hecha, que es el momento en el que se inserta el comportamiento de Retener tarjeta, pero sólo cuando se dé la condición historia sospechosa.

En Modelio, los puntos de extensión están en el apartado Nodes → Extension Point. A la flecha le hemos añadido una nota (Common → Note) indicando la condición entre llaves y el punto de extensión.