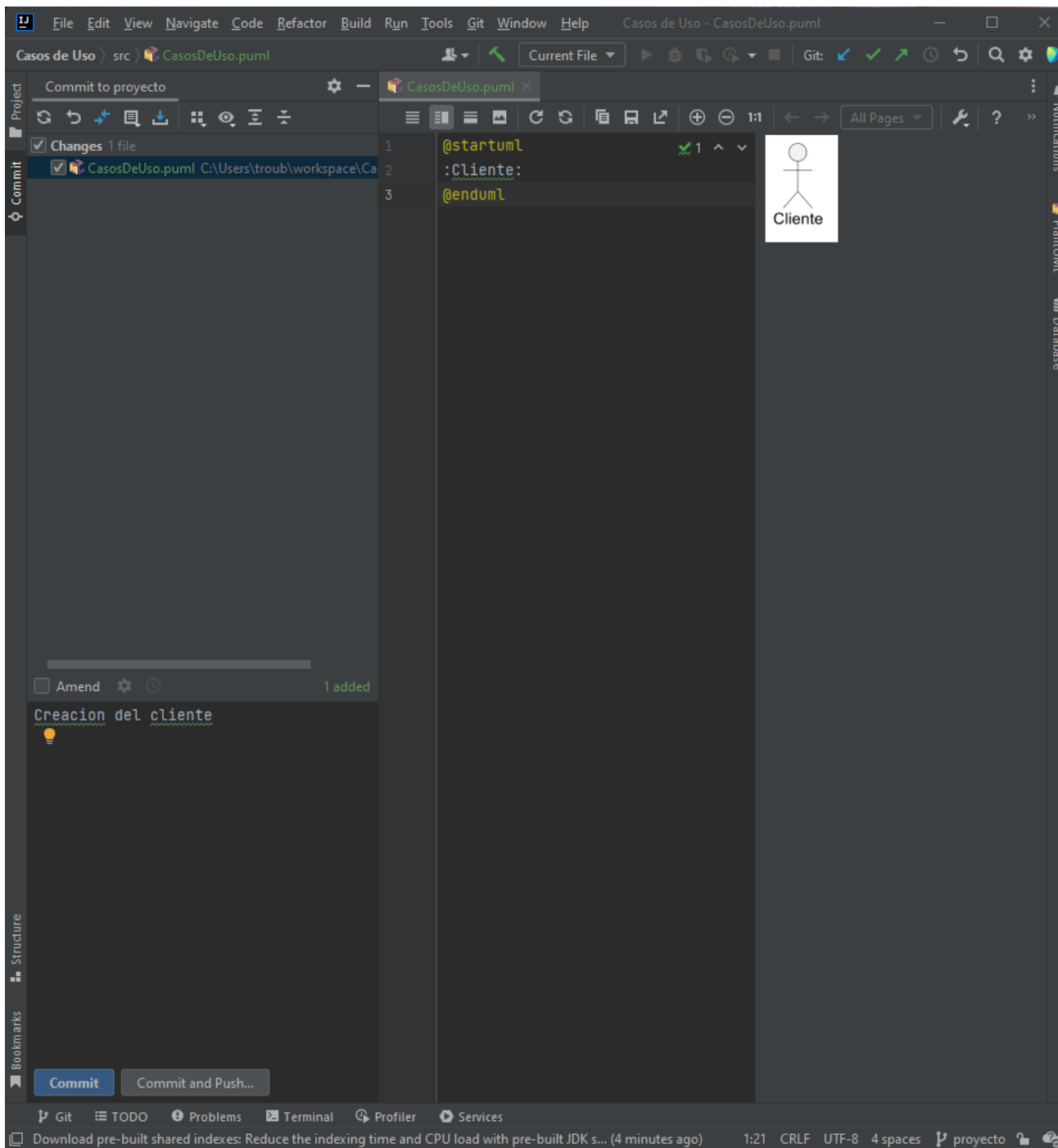
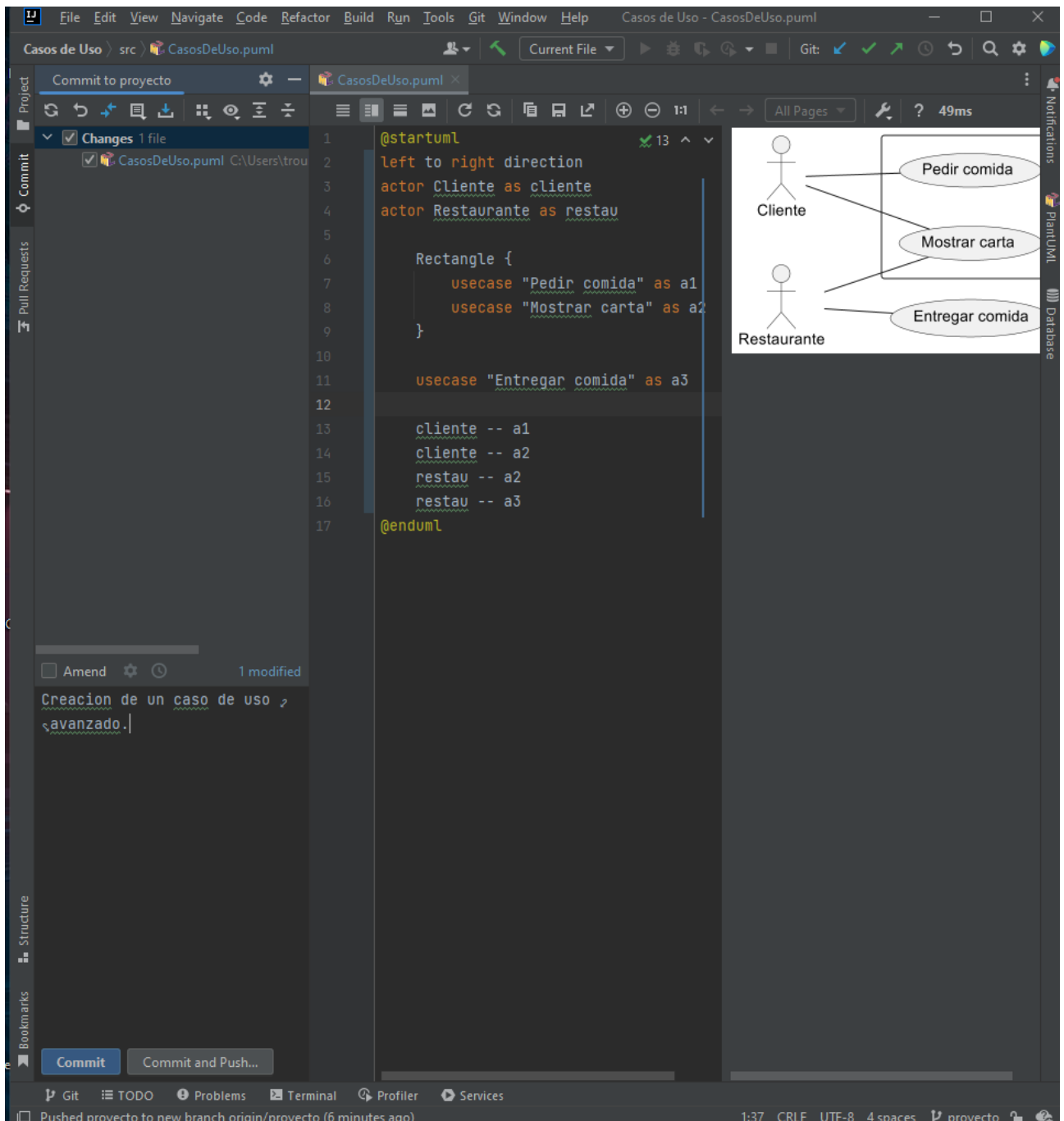


Actor: Los actores son personas o procesos automáticos que necesitan interactuar con el sistema. Se deben identificar sus papeles en el sistema. En el diagrama, se representan del siguiente modo

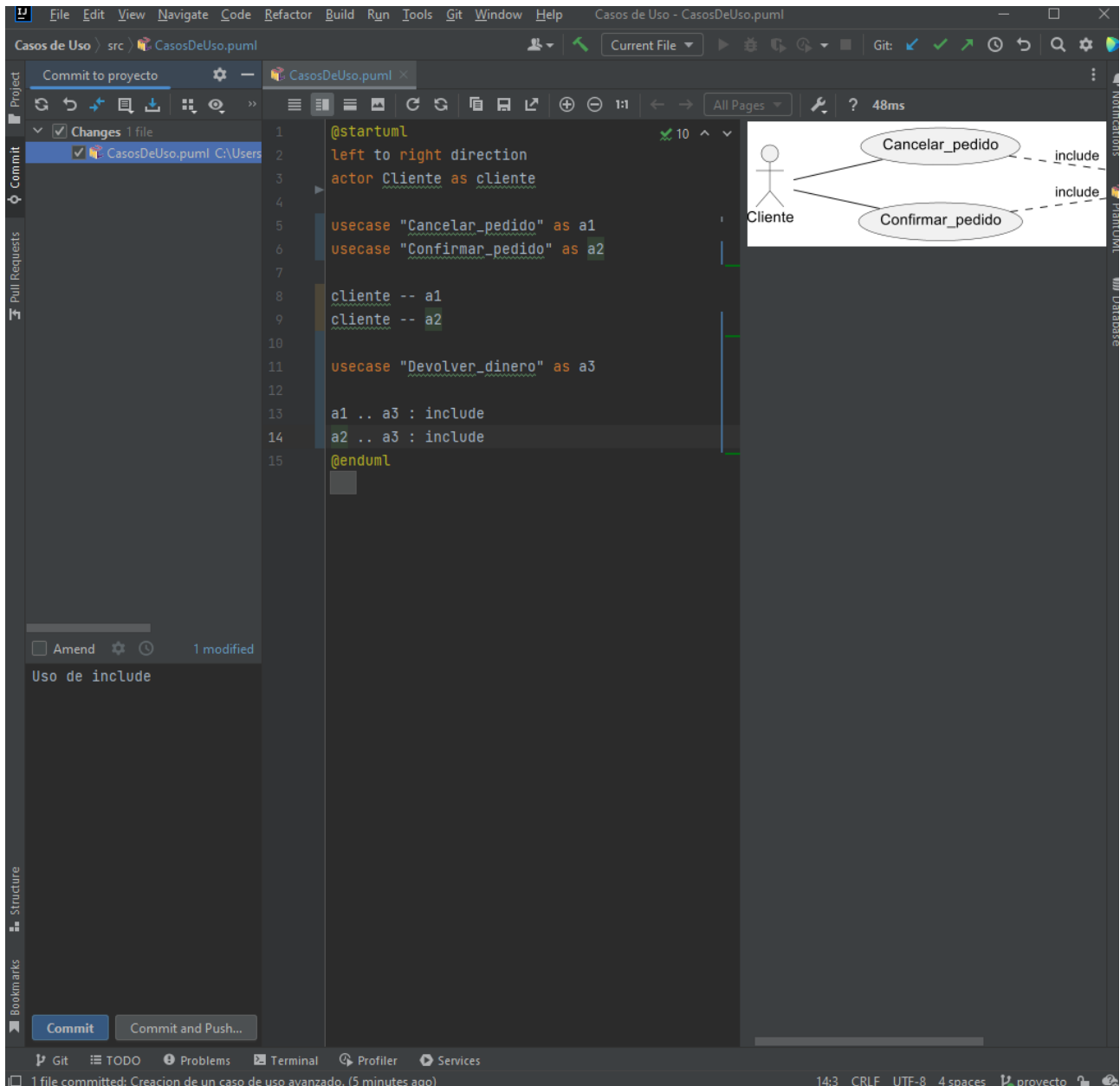


Caso de uso: Los casos de uso se representan mediante elipses y corresponden a acciones generales del sistema. Una forma de reconocerlos es que suelen ser verbos en la descripción del caso de uso.
Asociación: La interacción entre los actores y los casos de uso del sistema se representan por una línea recta que une a ambos.

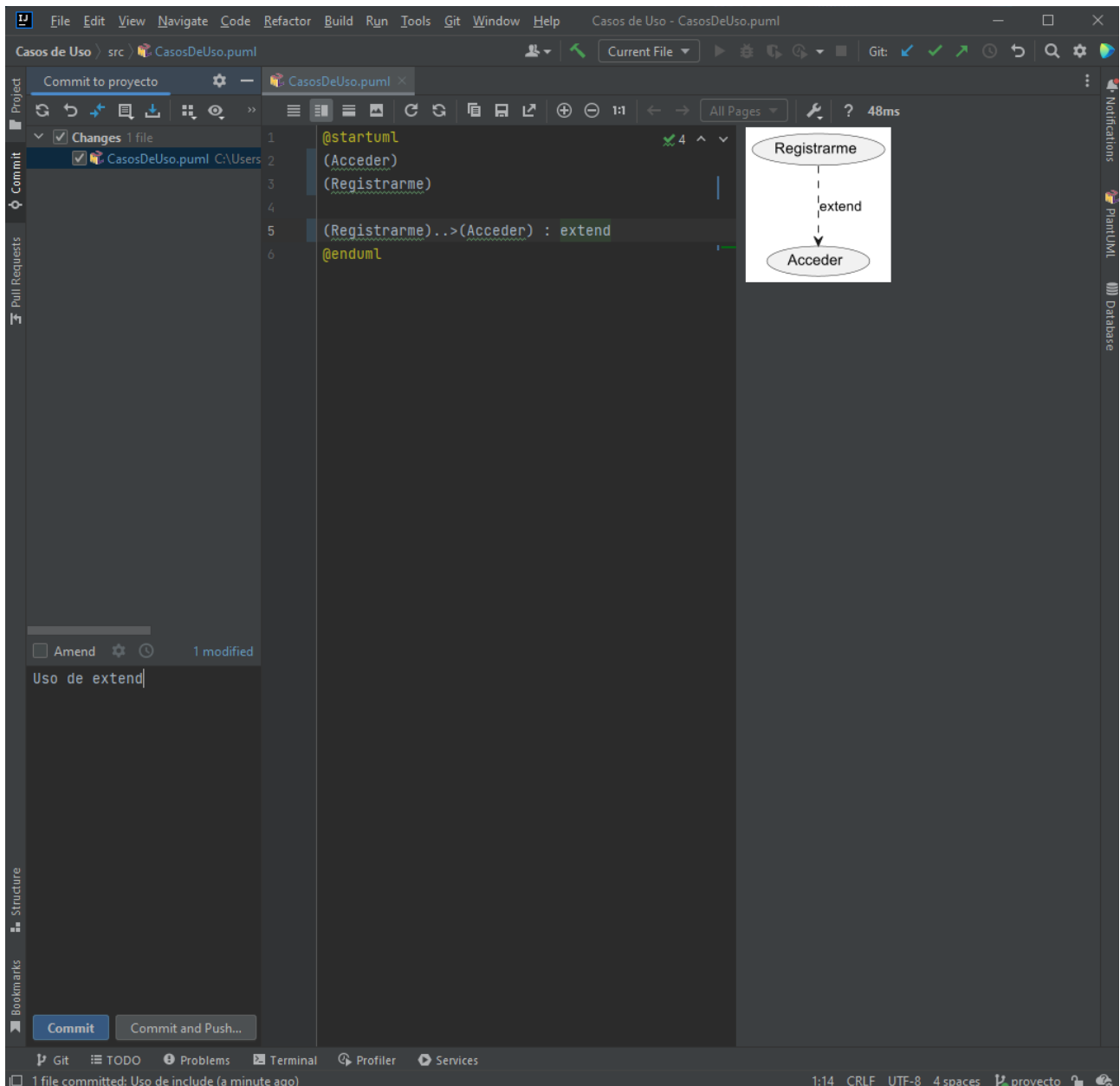


Sistema: El sistema es el software que vamos a desarrollar. Puede ser un pequeño componente cuyos actores son otros componentes, o puede ser una aplicación completa. Se representa como una caja rectangular. Dentro de ella se incluyen los casos de uso soportados por el sistema

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.

Casos de Uso - CasosDeUso.puml

Commit to proyecto

Changes 1 file

- CasosDeUso.puml C:\Users\...

Amend 1 modified

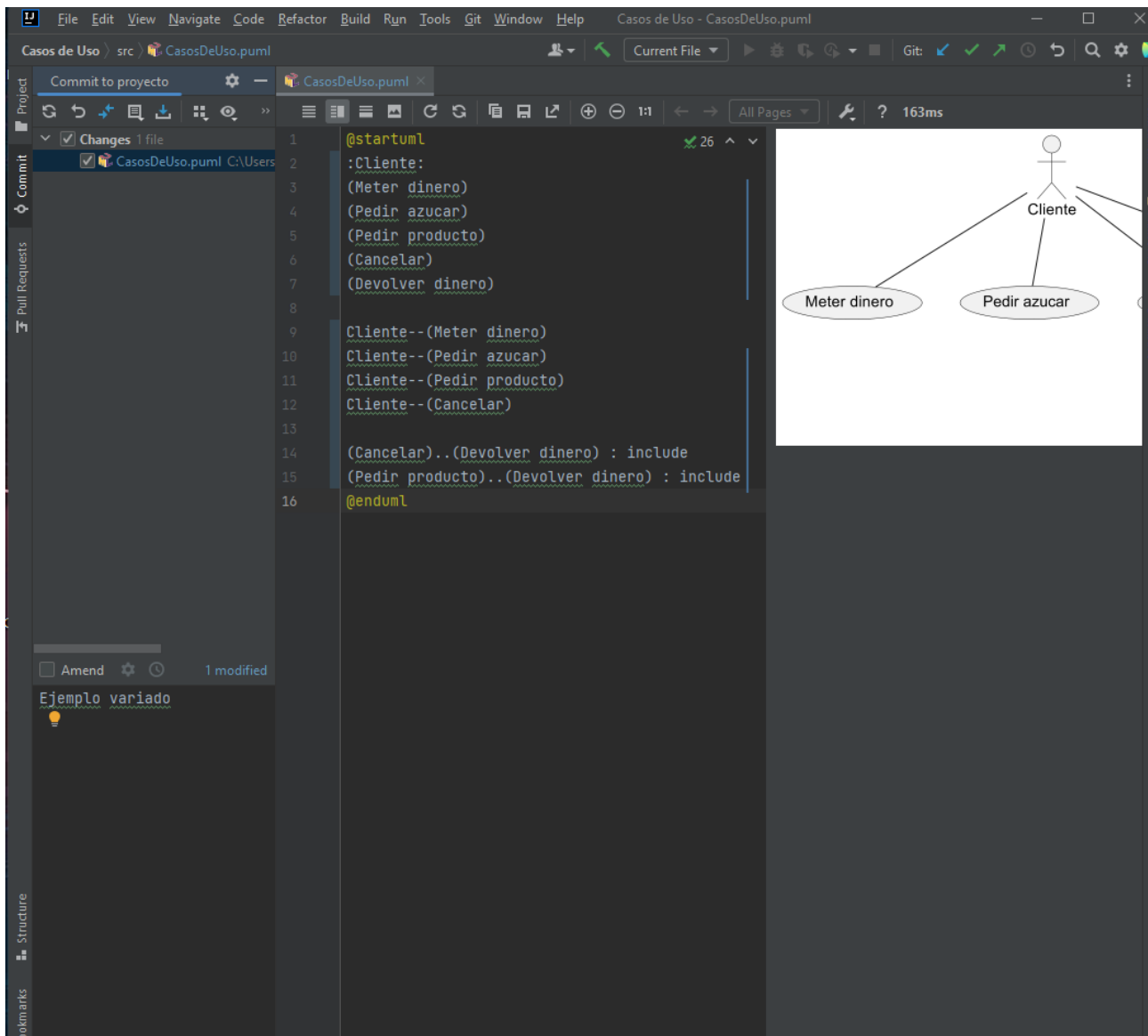
Uso de generalizacion

```
@startuml
(Pagar con efectivo)
(Pagar con tarjeta)
(Pagar)

(Pagar con tarjeta)--|>(Pagar)
(Pagar con efectivo)--|>(Pagar)
@enduml
```

```
graph TD
    A([Pagar con efectivo]) --> C([Pagar])
    B([Pagar con tarjeta]) --> C
```

Ejemplo: La máquina de café



Ejemplo: Tienda en Internet

The screenshot displays an IDE with the following components:

- Top Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Git, Window, Help. The active file is `Casos de Uso - CasosDeUso.puml`.
- Left Panel:** Project view showing `CasosDeUso.puml` and `CasosDeUso.puml C:\Users\tro`. Below it, a commit message `Ejemplo variado 2` is visible.
- Editor:** Contains UML use case diagram code for `CasosDeUso.puml`. The code includes:

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

Actor--(Identificarse)
Actor--(Pagar)
Actor--(Elegir mensajero)
Actor--(Confirmar)

(Registrarse)..> (Identificarse) : extend
(Pagar con tarjeta) --> (Pagar)
(Pagar con transferencia) --> (Pagar)

@enduml
```
- Right Panel:** A preview of the UML diagram showing an actor connected to use cases `Elegir mensajero` and `Confirmar`.
- Bottom Panel:** Git Log view showing a list of commits. The current branch is `HEAD (Current Branch)`. The log includes:

Commit	Author	Date
Ejemplo variado	dpketes	12 minutes ago
Uso de generalizacion	dpketes	17 minutes ago
Uso de extend	dpketes	19 minutes ago
Uso de include	dpketes	20 minutes ago
Creacion de un caso de uso avanzado.	dpketes	26 minutes ago
Creacion y asociacion de un caso c	origin/proyecto	34 minutes ago
Creacion del cliente	dpketes	37 minutes ago
1st commit	dpketes	40 minutes ago

Ejemplo: Usuarios y administradores

The image shows a screenshot of an IDE (IntelliJ IDEA) with a UML Use Case diagram and a Git log.

UML Use Case Diagram:

```
@startuml
:usuario:
:Administrador:

(Iniciar sesion)
(Recuperar contra)
(Modificar contra)
(Cerrar sesion)
(Registrar usuario)
(Instalar programa)

usuario--(Iniciar sesion)
usuario--(Recuperar contra)
usuario--(Modificar contra)
usuario--(Cerrar sesion)

Administrador -->|> usuario

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

@enduml
```

The diagram shows two actors: **usuario** and **Administrador**. The **usuario** actor is associated with five use cases: **(Iniciar sesion)**, **(Recuperar contra)**, **(Modificar contra)**, **(Cerrar sesion)**, and **(Registrar usuario)**. The **Administrador** actor is associated with two use cases: **(Registrar usuario)** and **(Instalar programa)**. There is a directed association from **Administrador** to **usuario** with the label **-->|> usuario**.

Git Log:

Commit	Author	Date
HEAD (Current Branch)	origin & proyecto	dpketes 4 minutes ago
Local		
proyecto		dpketes 16 minutes ago
Remote		
Ejemplo variado 2	dpketes	21 minutes ago
Ejemplo variado	dpketes	23 minutes ago
Uso de generalizacion	dpketes	25 minutes ago
Uso de extend	dpketes	31 minutes ago
Uso de include	dpketes	38 minutes ago
Creacion de un caso de uso avanzado.	dpketes	41 minutes ago
Creacion y asociacion de un caso de uso con el cliente	dpketes	44 minutes ago
1st commit	dpketes	

Ejemplo: Puesto fronterizo

Visual Studio Code editor showing a UML Use Case diagram for a border checkpoint system.

The diagram illustrates the following use cases and relationships:

- ciudadano2** (Actor) is associated with **Tomar huellas** (Use Case).
- ciudadano** (Actor) is associated with **Registrar entrada** (Use Case).
- Tomar huellas** (Use Case) is associated with **Registrar entrada** (Use Case) via an **extend** relationship.

The diagram is displayed in the right-hand pane of the editor. The left-hand pane shows the project structure and the file `CasosDeUso.puml` is open. The bottom pane shows the Git log, indicating the current branch is `HEAD (Current Branch)` and the project is `proyecto`.

```
@startuml
    actor ciudadano2
    actor ciudadano
    usecase "Tomar huellas" as UC1
    usecase "Registrar entrada" as UC2
    ciudadano2 --> UC1
    ciudadano --> UC2
    UC1 -.-> UC2 : extend
@enduml
```