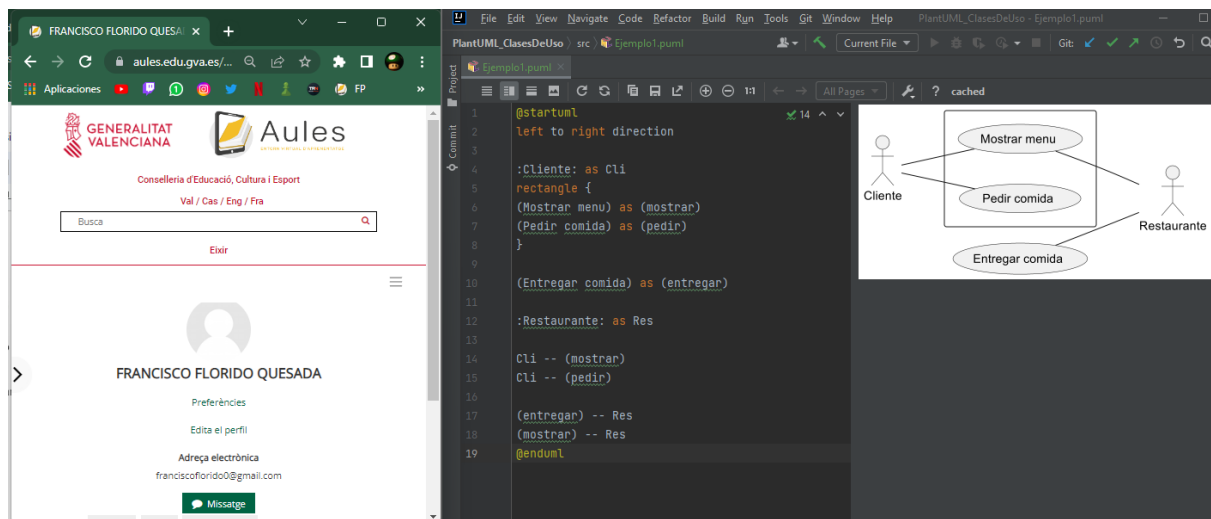


# Casos de uso en PlantUML

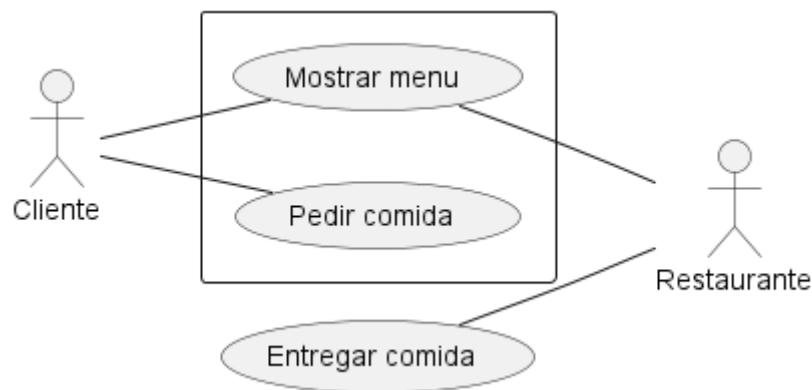
## PRIMER EJEMPLO

Lo primero será crear el primer ejemplo. Para ello tenemos que saber crear casos de uso, actores, asociaciones y sistemas.

Para crear este primer ejemplo he creado el siguiente código:



El diagrama nos quedaría de esta forma:



Como podemos ver en el código, hemos creado dos actores, Cliente y Restaurante, con el siguiente código:

```
:Cliente: as Cli
:Restaurante: as Res
```

Después de crear los actores hemos creado los casos de uso con el siguiente código:

```
(Mostrar menu) as (mostrar)
(Pedir comida) as (pedir)
(Entregar comida) as (entregar)
```

Y después de esto hemos metido las asociaciones “Mostrar menú” y “Pedir Comida” en un sistema utilizando el siguiente código:

```
rectangle {
(Mostrar menu) as (mostrar)
(Pedir comida) as (pedir)
}
```

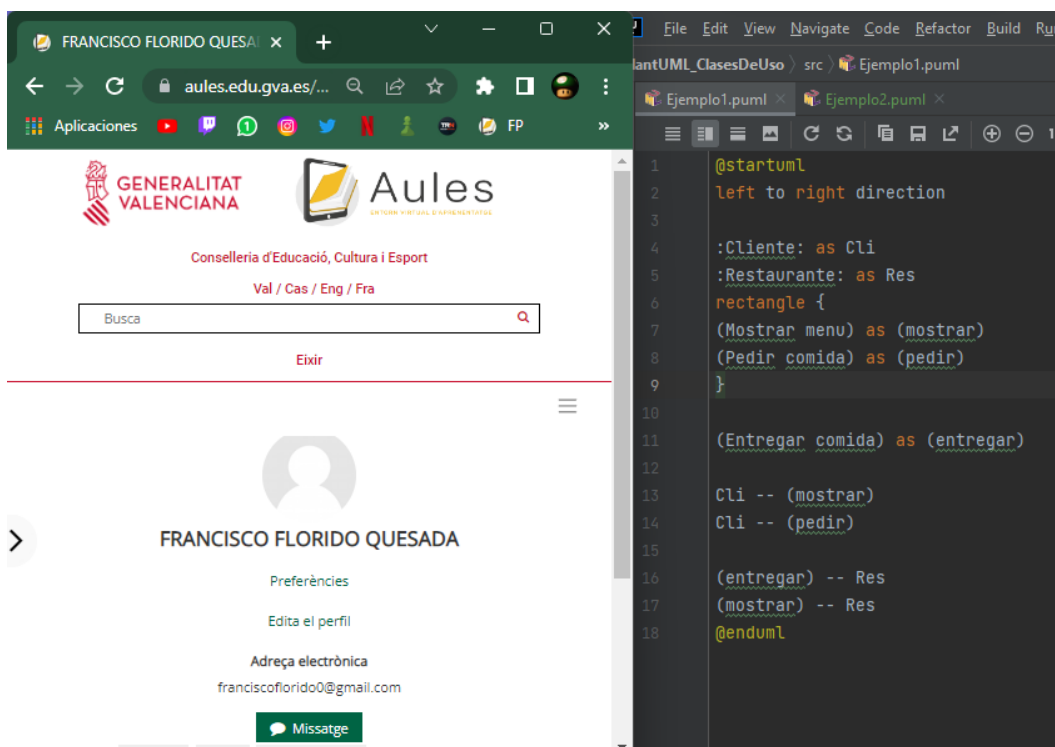
Para finalizar este primer ejemplo hemos creado las asociaciones con el código que vemos a continuación:

```
Cli -- (mostrar)
Cli -- (pedir)

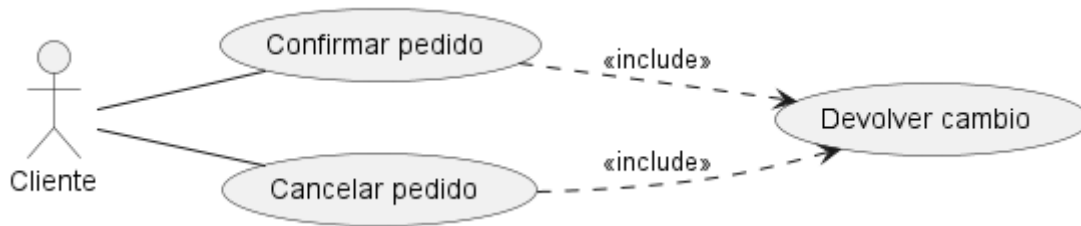
(entregar) -- Res
(mostrar) -- Res
```

## SEGUNDO EJEMPLO

Ahora pasamos con el segundo ejemplo, para realizarlo he creado el siguiente código:



Y nos queda el diagrama de esta forma:



Para ello hemos creado un actor llamado “Cliente”, tres casos de uso: “Confirmar pedido”, “Cancelar pedido” y “Devolver cambio”, y después los hemos asociado entre sí.

Para hacer que una asociación aparezca con las rayitas y con el nombre encima lo podemos hacer de esta forma:

```

(confirmar) ..> (cambio): << include >>
(cancelar) ..> (cambio): << include >>
  
```

Poniendo <<include>> en este relación indicamos que un caso de uso incluye a otro.

### TERCER EJEMPLO

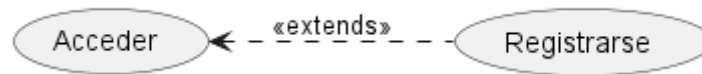
Para realizar este tercer ejemplo he creado el siguiente código:

The screenshot shows a web browser on the left displaying the Aules portal (aules.edu.gva.es) with the user profile of FRANCISCO FLORIDO QUESADA. On the right, an IDE window shows the UML code for Ejemplo3.puml:

```

@startuml
left to right direction
3
4 (Acceder)
5 (Registrarse)
6
7 (Acceder) <.. (Registrarse): <<extends>>
8
@enduml
  
```

Y el diagrama me ha quedado de esta forma:



Para este diagrama hemos utilizado lo que hemos aprendido en los dos anteriores ejemplos solo que cambiando el nombre de la asociación para que un caso de uso sea una extensión de otro.

### CUARTO EJEMPLO

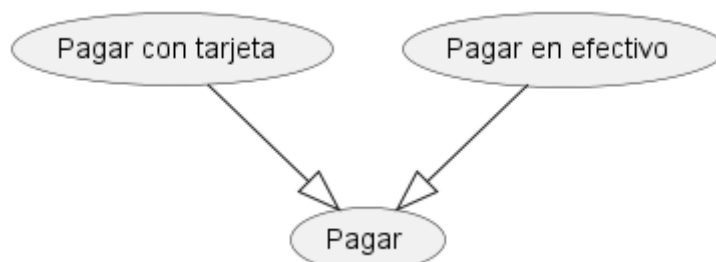
Para este diagrama he creado este pequeño código:

The screenshot shows a web browser on the left displaying the Aules portal (Conselleria d'Educació, Cultura i Esport) with a user profile for FRANCISCO FLORIDO QUESADA. On the right, a code editor shows the following UML code:

```

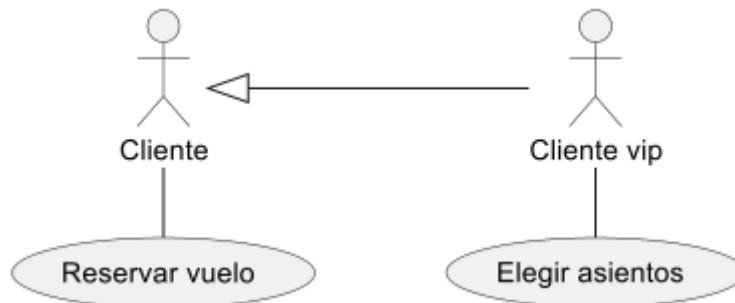
1 @startuml
2 (Pagar)
3 (Pagar con tarjeta) --|> (Pagar)
4 (Pagar en efectivo) --|> (Pagar)
5 @enduml
  
```

Y el diagrama me ha quedado de esta forma:



Para este diagrama hemos utilizado un nuevo tipo de asociación llamado generalización, este tipo de asociación 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 utilizar con actores:



```

:Cliente: as cli
:Cliente vip: as cliv

cli <|-- cliv
cli | (Reservar vuelo)
cliv - (Elegir asientos)
  
```

## EJEMPLO MAQUINA DE CAFE

Para este diagrama he creado el siguiente código:

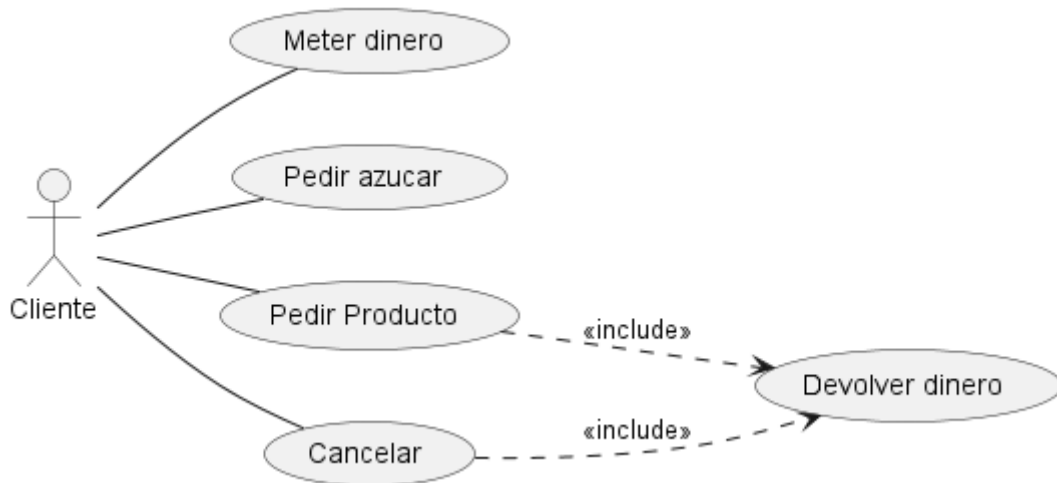
The screenshot shows a web browser on the left and a code editor on the right. The browser displays the 'Aules' website, which is part of the 'GENERALITAT VALENCIANA' system, specifically the 'Conselleria d'Educació, Cultura i Esport'. The code editor shows a PlantUML script for a coffee machine use case.

```

@startuml
left to right direction
:Cliente: as cli

cli -- (Meter dinero)
cli -- (Pedir azucar)
cli -- (Pedir Producto)
cli -- (Cancelar)
(Pedir Producto) ..> (Devolver dinero): <<include>>
(Cancelar) ..> (Devolver dinero): <<include>>
@enduml
  
```

Y el diagrama me queda de esta forma:



Para este diagrama he utilizado los conocimientos adquiridos en los ejemplos anteriores ya que no hay nada nuevo.

## EJEMPLO TIENDA INTERNET

Para crear este diagrama he creado el siguiente código:

The screenshot shows a web browser on the left displaying the Aules website (Conselleria d'Educació, Cultura i Esport) with a search bar and user profile for FRANCISCO FLORIDO QUESADA. On the right, a code editor shows the following UML code:

```

@startuml
left to right direction

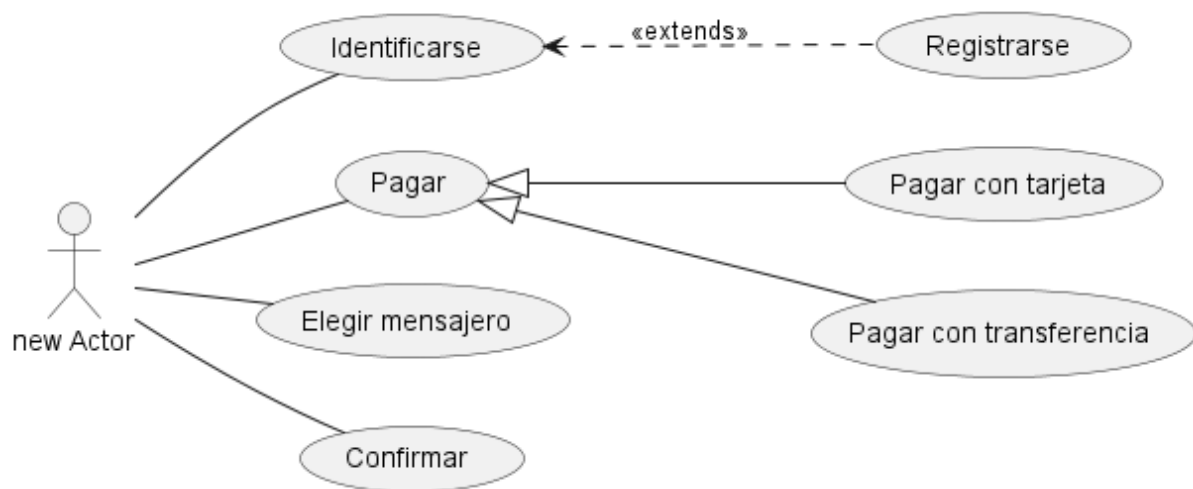
:new Actor: as act

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

(Identificarse) <.. (Registrarse): <<extends>>
(Pagar) <|-- (Pagar con tarjeta)
(Pagar) <|-- (Pagar con transferencia)

@enduml
  
```

Y el diagrama me ha quedado de esta forma:



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”.

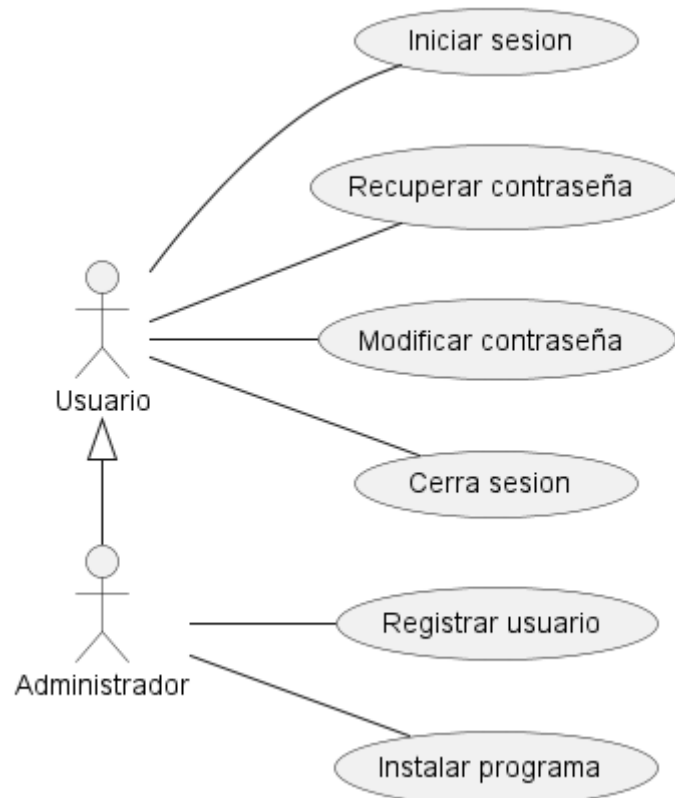
## EJEMPLO USUARIOS Y ADMINISTRADORES

Para este diagrama he creado el siguiente código:

```

1  @startuml
2  left to right direction
3
4  :Usuario: as usu
5  :Administrador: as admin
6
7  usu -- (Iniciar sesion)
8  usu -- (Recuperar contraseña)
9  usu -- (Modificar contraseña)
10 usu -- (Cerra sesion)
11 usu <- admin
12
13 admin -- (Registrar usuario)
14 admin -- (Instalar programa)
15 @enduml
  
```

Y el diagrama me ha quedado de la siguiente manera:



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

Para crear este diagrama he creado el siguiente código:

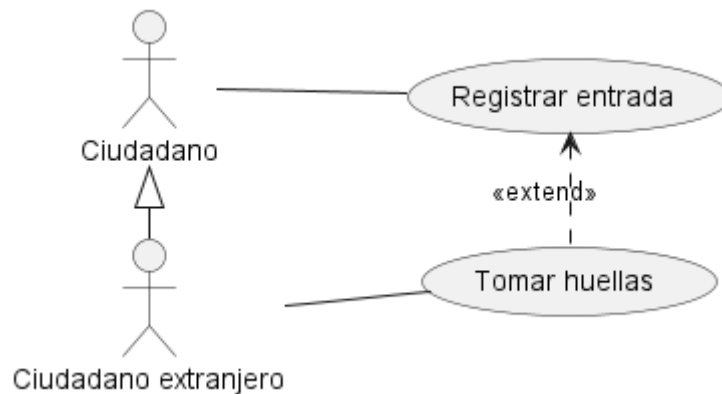
The screenshot shows a web browser on the left displaying the Aules portal (Conselleria d'Educació, Cultura i Esport) with a search bar and user profile for FRANCISCO FLORIDO QUESADA. On the right, an IDE window shows the following UML code:

```

1 @startuml
2 left to right direction
3 :Ciudadano: as ciu
4 :Ciudadano extranjero: as ext
5
6 ciu -- (Registrar entrada)
7 ciu <|- ext
8
9 ext -- (Tomar huellas)
10
11 (Registrar entrada) <. (Tomar huellas): <<extend>>
12 @enduml
  
```



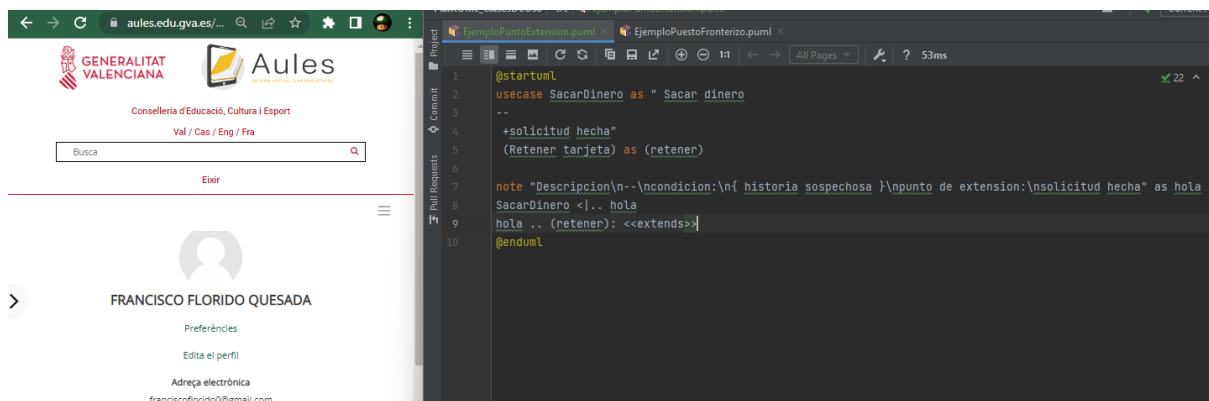
Y el diagrama se me queda de la siguiente forma:



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

Para recrear este diagrama he creado el siguiente código:



Y me queda de esta forma:

Para que un caso de uso tenga esa línea en medio debemos crearlo con la sentencia que vemos en el código proporcionado.

