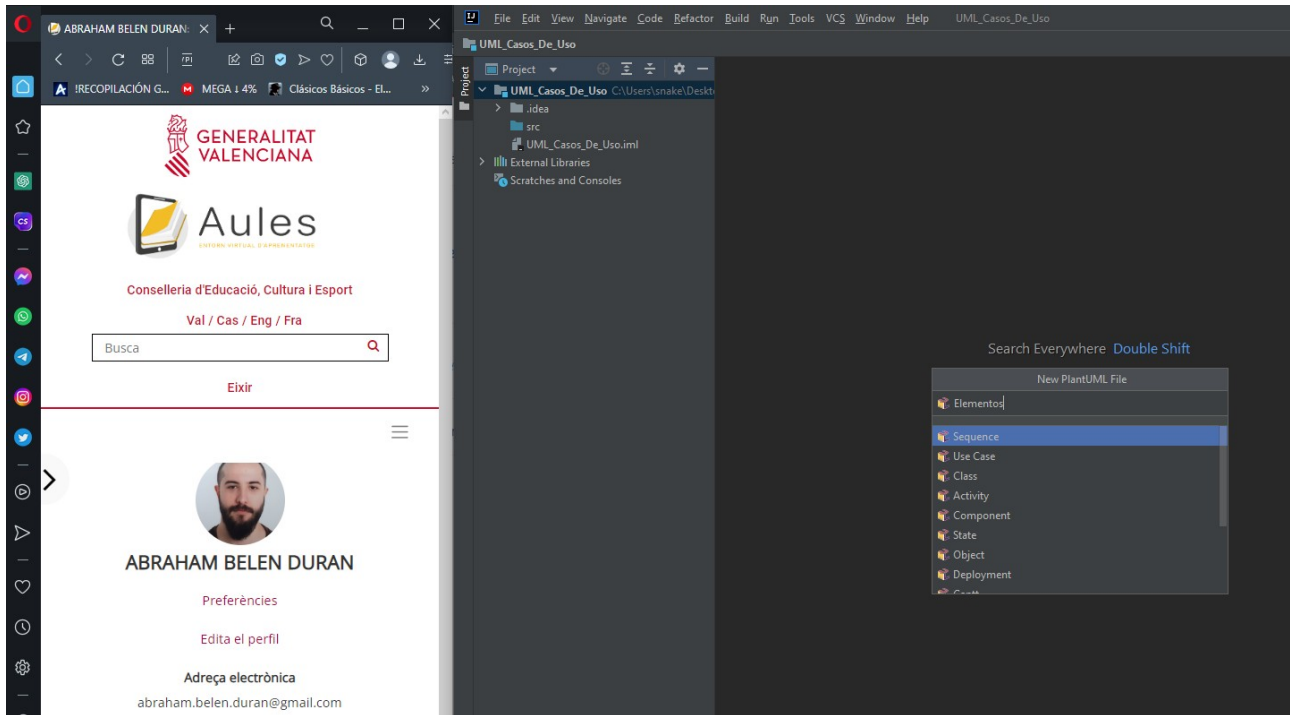


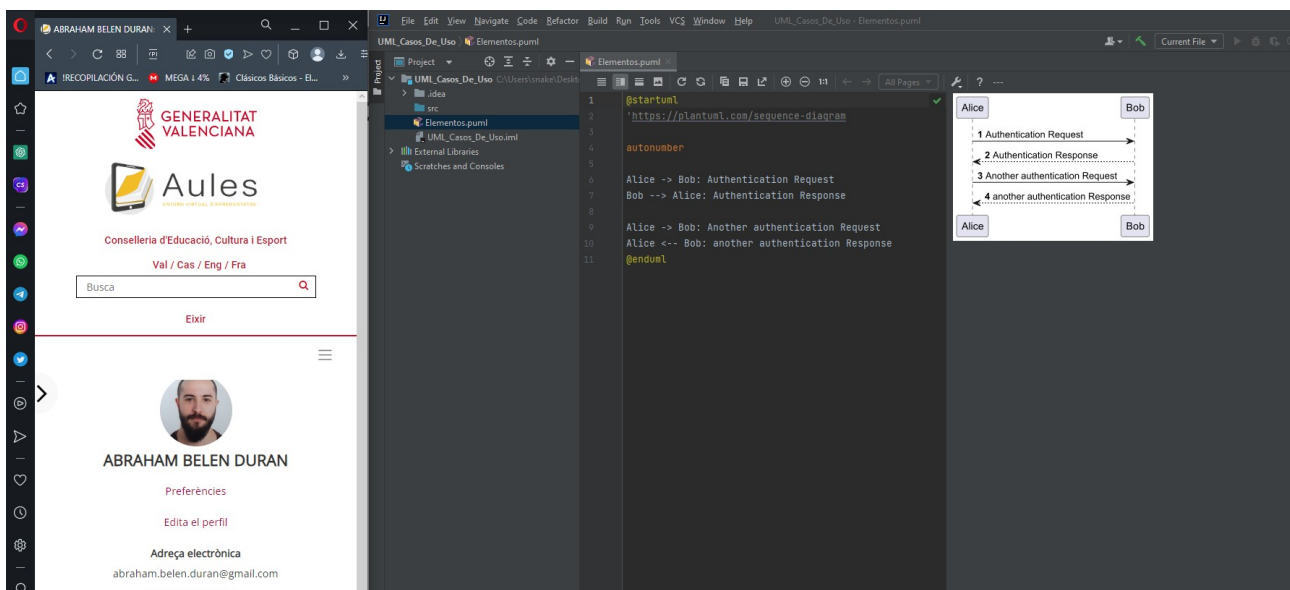
Entornos de Desarrollo

U.D 12: Diagramas II.
Práctica clases de uso con
PlantUML

Lo primero que haremos será crear un nuevo proyecto llamado `UML_Casos_De_Uso` dentro del cual crearemos un archivo de formato PlantUML. Como en la anterior práctica ya mostré como se creaba un archivo de esta clase y como se instalaba la extensión necesaria para la práctica, pasaremos directamente a lo que pide el ejercicio, no sin antes mostrar la captura del nombre que le he proporcionado al fichero UML, que en mi caso ha sido `Elementos`.

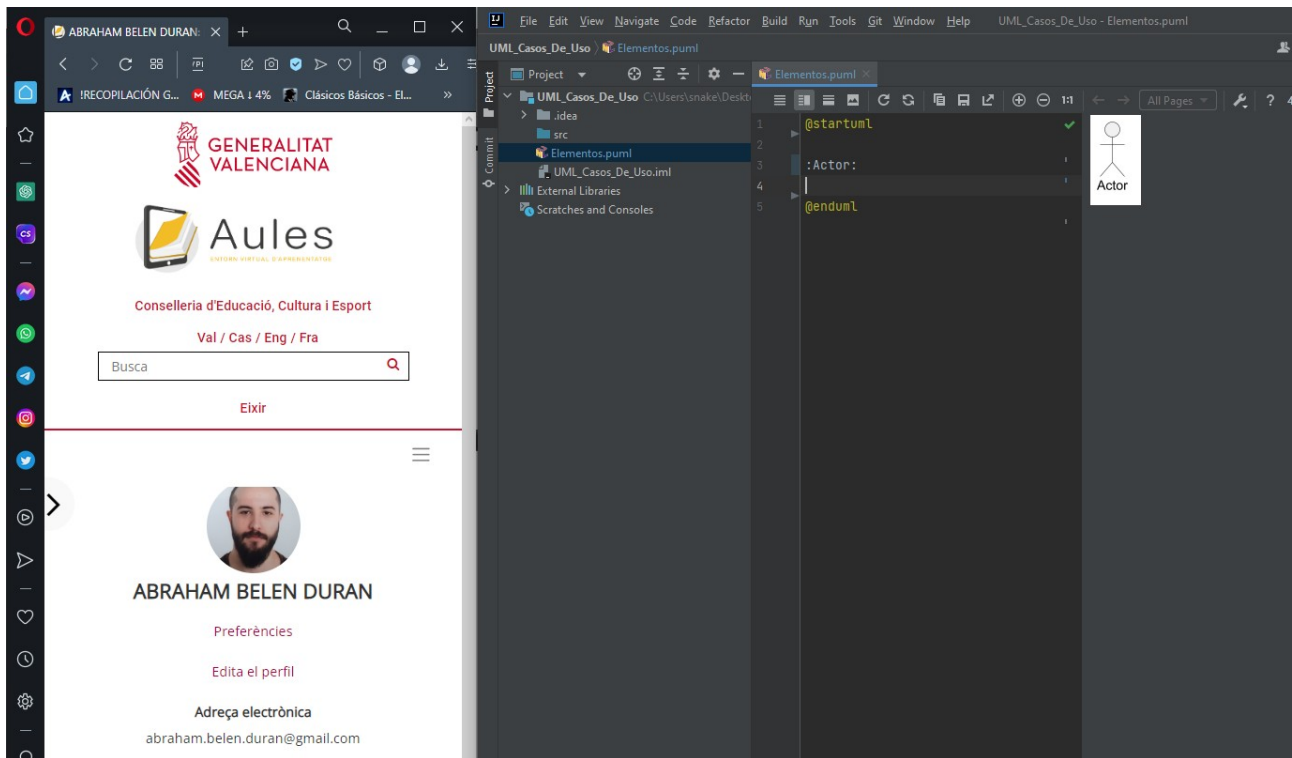


Bien una vez hecho esto se nos quedaría así nuestra ventana de proyecto.



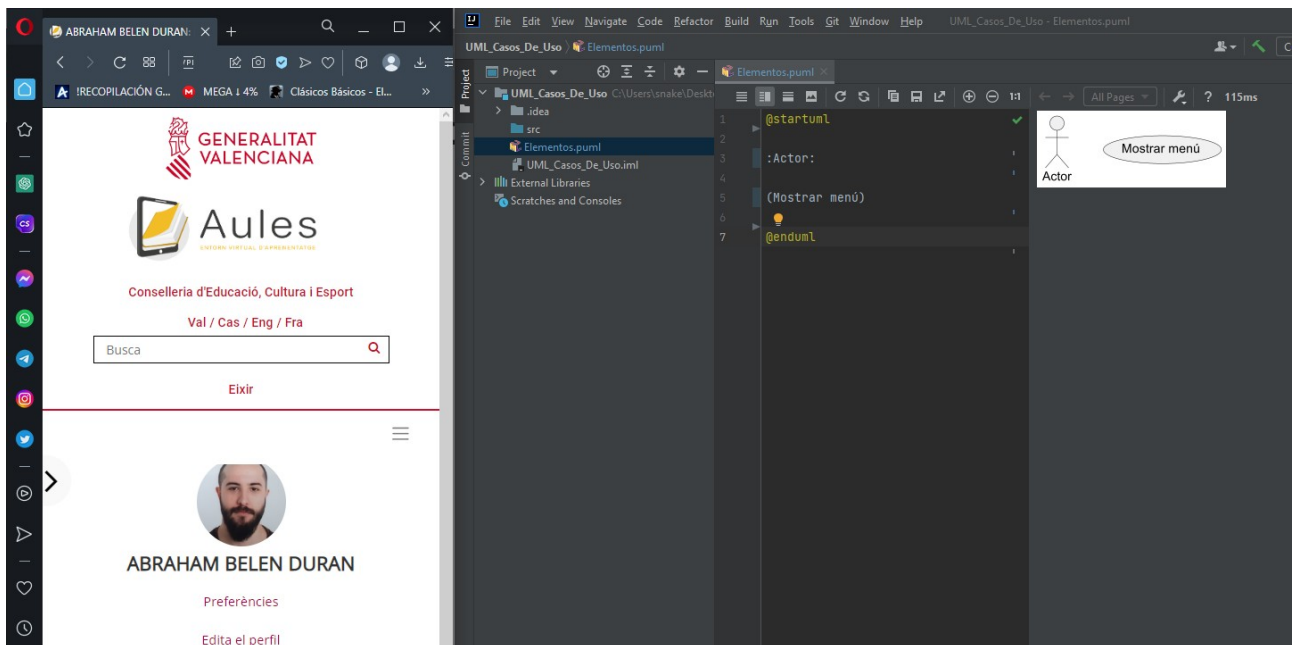
Así que ya podemos continuar con la práctica.

Bien, para empezar ya con lo que pide el ejercicio deberemos crear un 'Actor'. No tiene mayor complicación, tal y como muestro en la siguiente imagen.

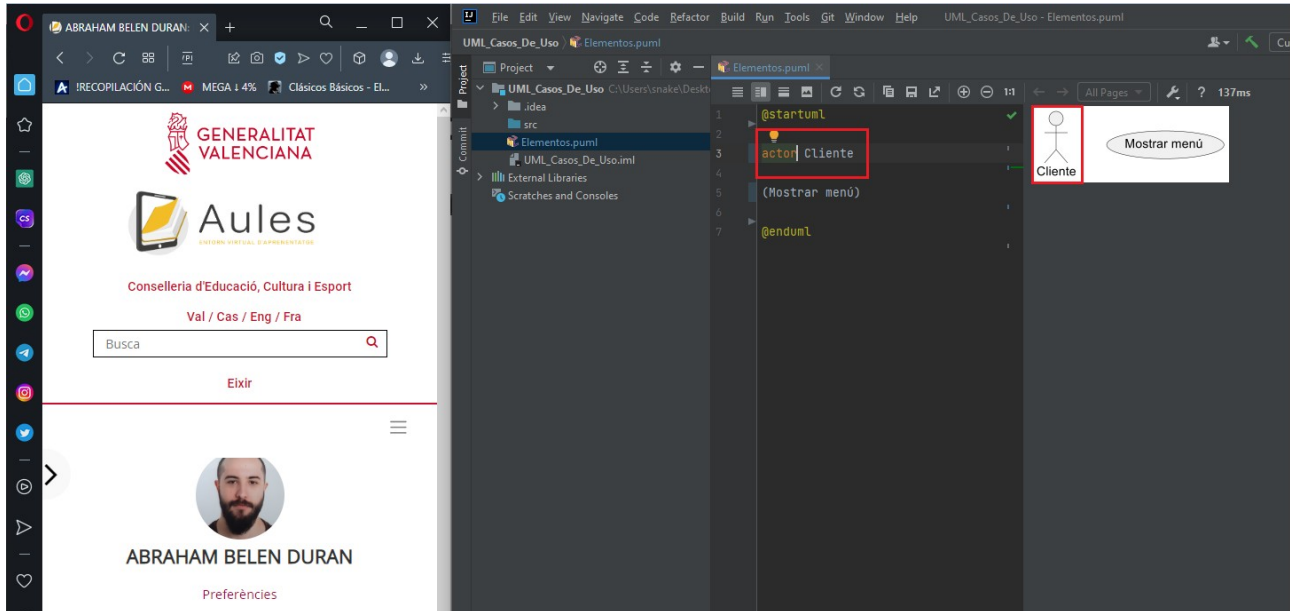


‘Los actores son personas o procesos automáticos que necesitan interactuar con el sistema. Se deben identificar sus papeles en el sistema.’

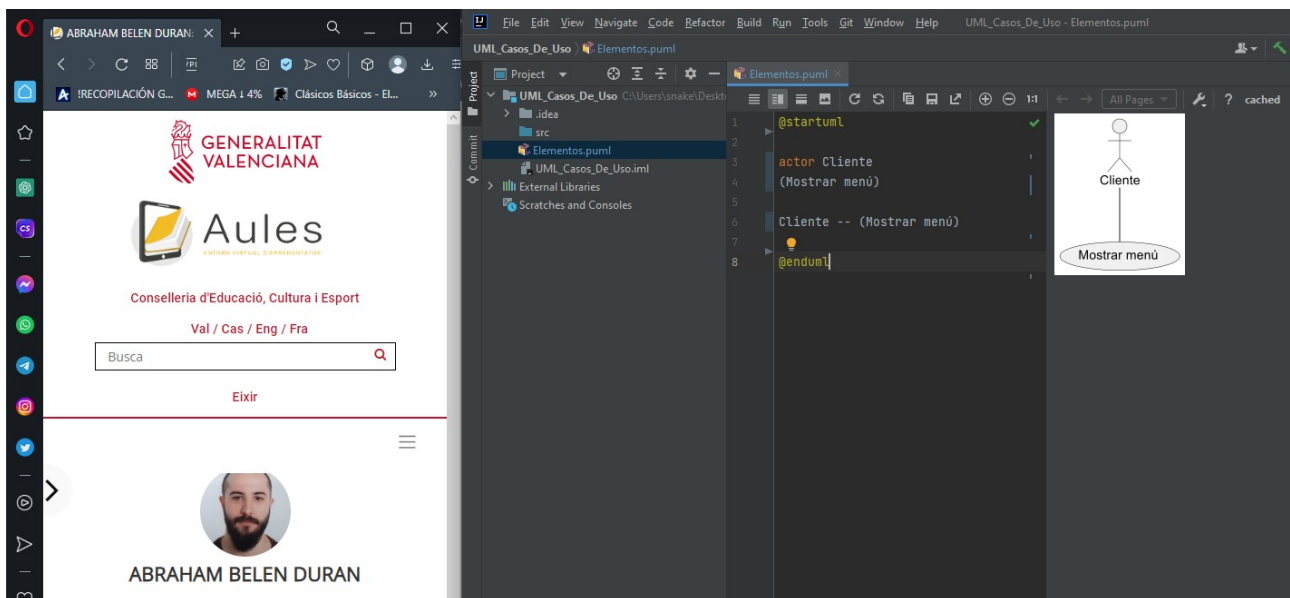
Ahora deberemos representar un caso de uso, es decir, las acciones generales del sistema. Se representa con una cadena de texto entre paréntesis.



Ahora deberemos asociar al actor junto con el caso de uso, no sin antes modificar el nombre del actor tal y como muestro en la siguiente imagen. Le he cambiado el nombre a Cliente sencillamente quitándole los dos puntos que rodeaba a la palabra Actor y poniéndole el actual nombre posteriormente.

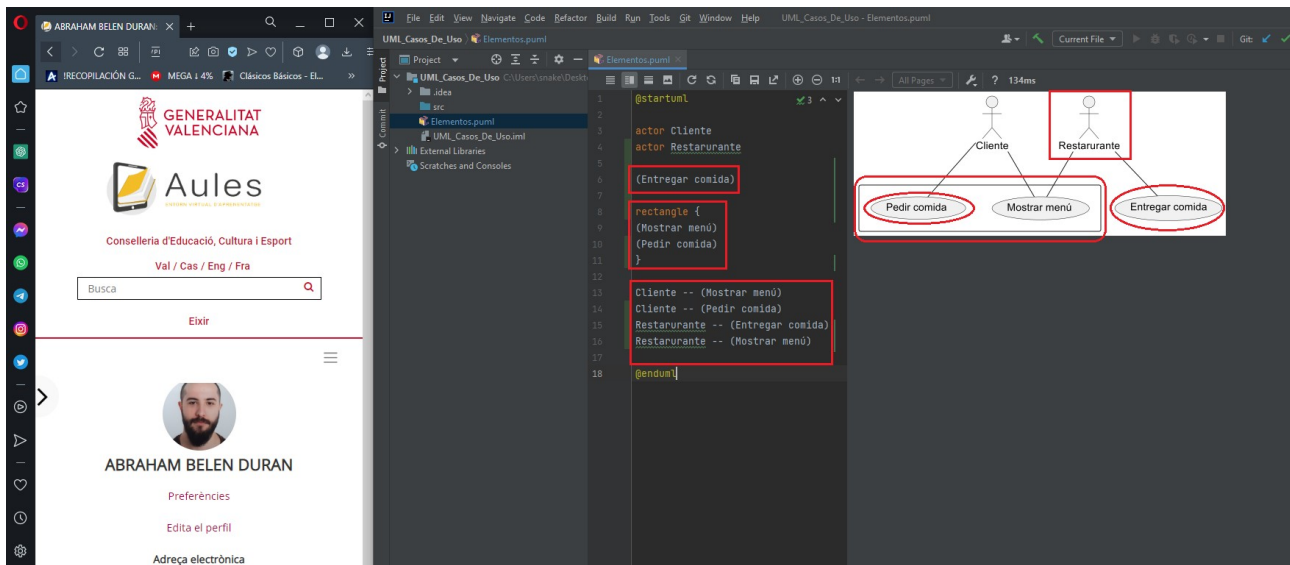


Ahora sí, pasemos a la asociación entre ambas entidades. Esta se realiza uniendo el actor con el caso de uso mediante dos guiones tal y como muestro en la imagen.



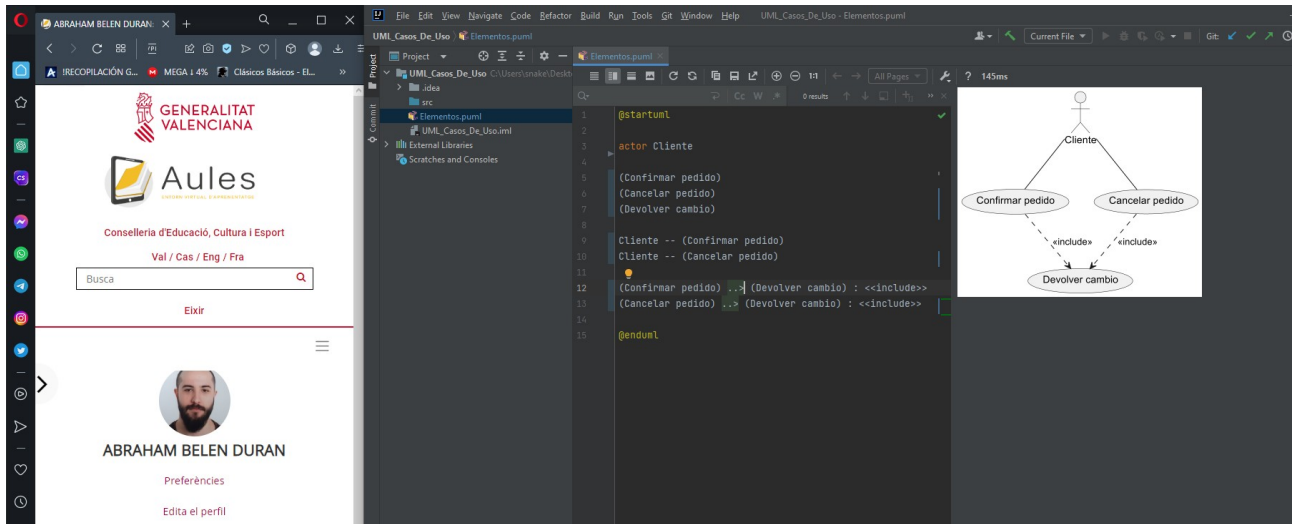
Ahora debemos crear el sistema. En UML se representa como un rectángulo. Además de eso también crearemos otro actor y dos casos de uso, tal y como muestro en la siguiente imagen.

Explicaré un poco todo lo que se ve en la imagen expuesta. Básicamente he creado un nuevo actor llamado 'Restaurante' junto con varios casos de uso, 'Entregar comida' para Restaurante y 'Pedir comida' para Cliente. Una vez he hecho eso los he unido y he creado el 'Sistema' dentro del cual he introducido 'Mostrar menú' y 'Pedir comida'.



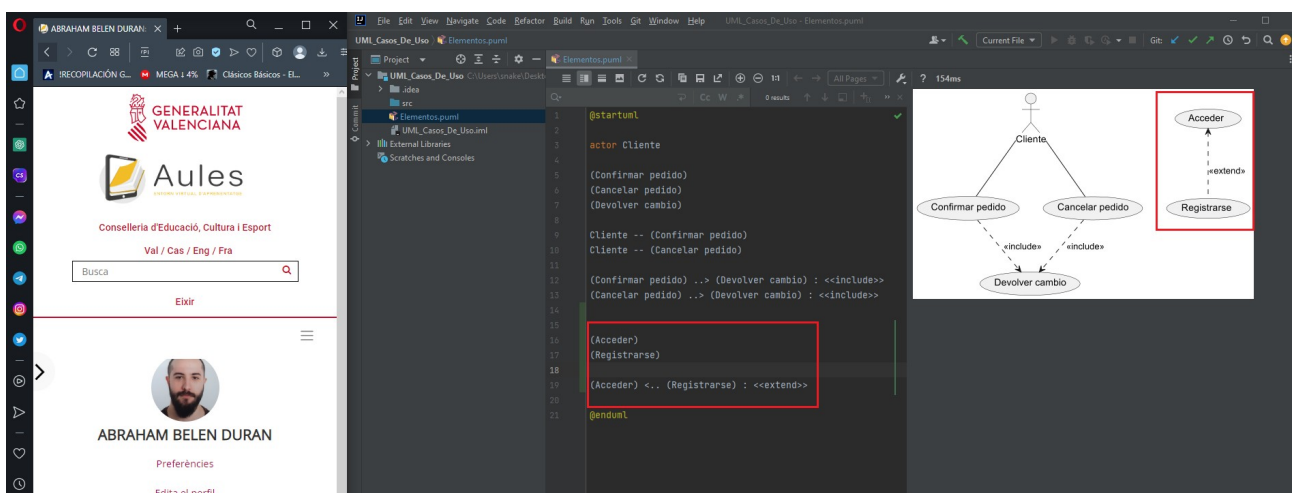
El sistema es el software que vamos a desarrollar. Puede ser un pequeño componente cuyos actores son otros componentes, o puede ser una aplicación completa. Dentro de ella se incluyen los casos de uso soportados por el sistema.

Ahora continuaremos haciendo una inclusión, que consiste en que un caso de uso se incluye dentro de otro caso de uso. Aunque para ello borraremos la mayoría de cosas que habíamos creado en los anteriores ejercicios. No hay de que preocuparse, ya he hecho commit justo en este momento, así que podremos ver lo realizado anteriormente en cualquier momento. De modo que en la siguiente imagen se puede apreciar el nuevo ejercicio realizado.

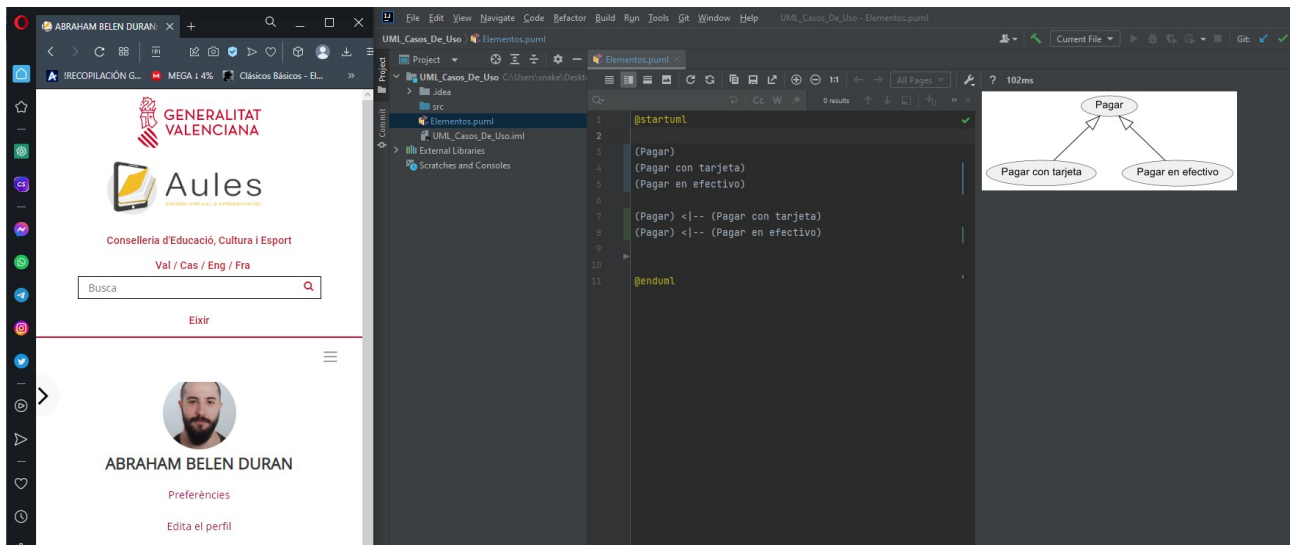


Se utiliza cuando el comportamiento de un caso de uso se incluye dentro del comportamiento de otro. Se representa con una flecha de trazo discontinuo desde el caso que incluye hasta el caso incluido, con el estereotipo «include» o «use».

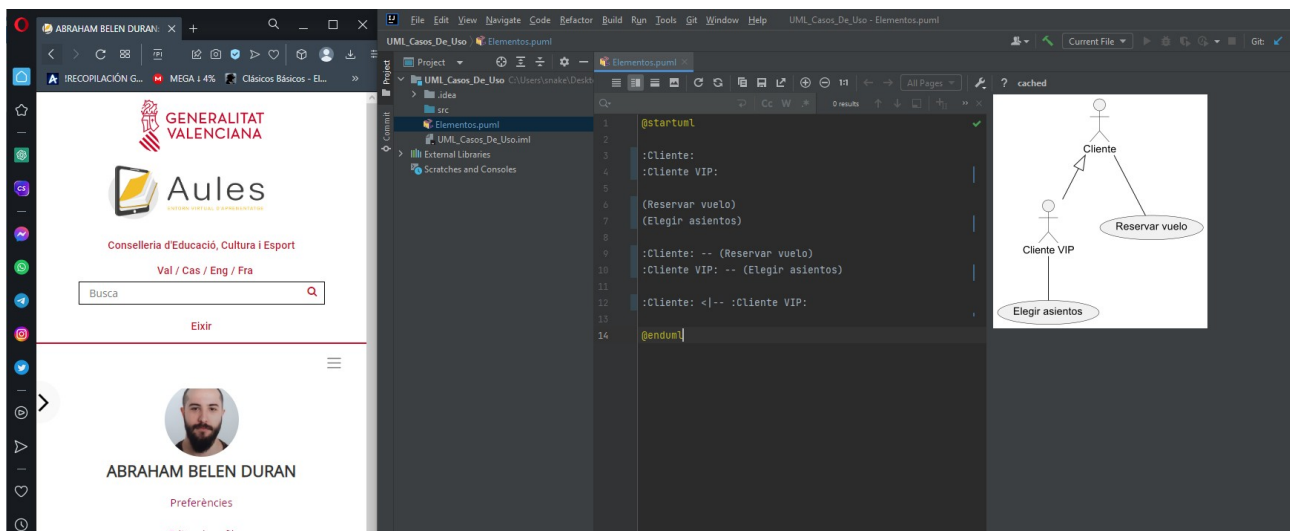
En el siguiente ejercicio haremos una 'Extensión' que se utiliza cuando un caso aporta un comportamiento adicional. No hace falta borrar nada de lo hecho anteriormente, se hace a parte y ya está. Se representa de la siguiente manera.



Para el siguiente ejemplo sí borraré lo hecho anteriormente, ya que necesito el actor Cliente para hacer la generalización.



Aunque también se puede hacer con actores, tal y como muestro a continuación. Además los actores se pueden crear de el modo que muestro en la captura de abajo. Anteriormente los he creado de otra forma, pero ambas son válidas.



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.

Aquí finaliza la primera parte de la práctica, ahora hay que poner en práctica todo lo anterior mediante ejemplos proporcionados por el temario.

Ejemplo: La máquina de café

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

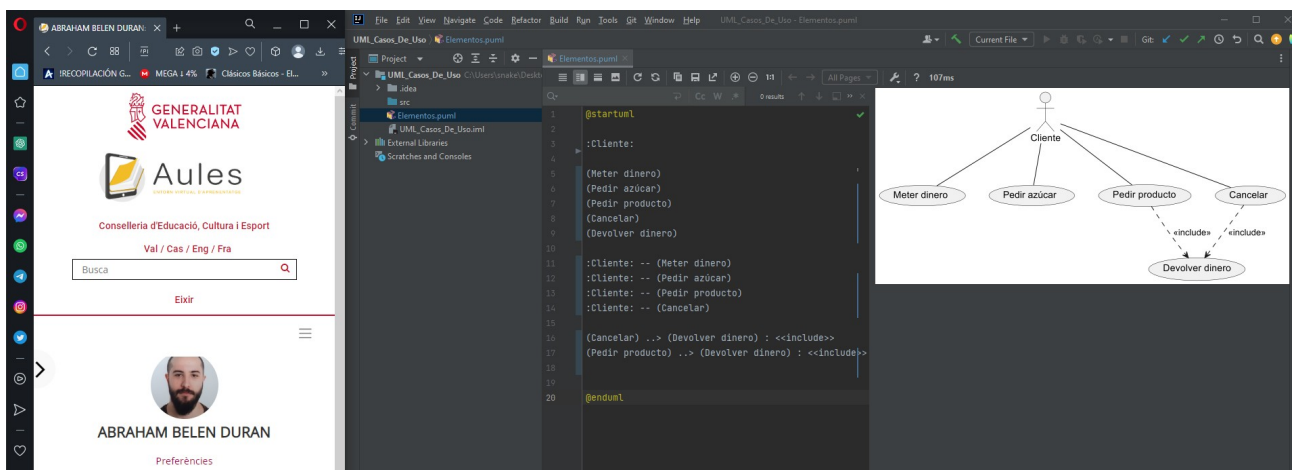
La máquina debe permitir a una persona introducir dinero, escoger uno de los productos de acuerdo a su precio, escoger un nivel de azúcar y entregar el producto y las vueltas.

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

A continuación muestro como queda el ejercicio resuelto.

En este caso 'Cliente' va asociado a 'Meter dinero', 'Pedir azúcar', 'Pedir producto' y 'Cancelar'.

Finalmente 'Pedir producto' y 'Cancelar' se unen a 'Devolver dinero' mediante la Inclusión.



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.

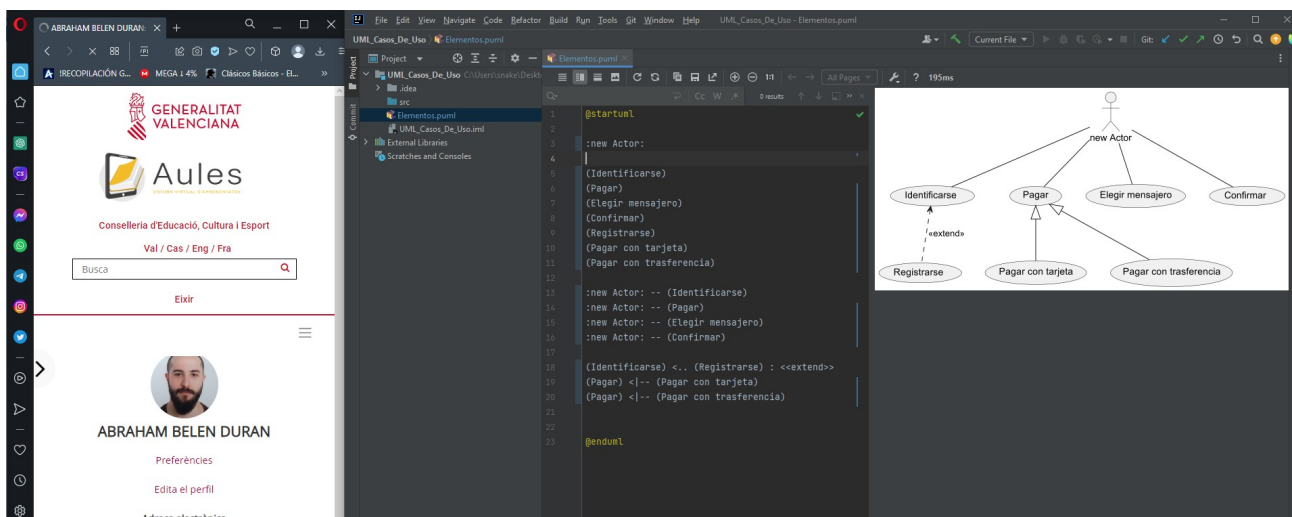
Finalmente se le mostrarán todos los datos del pedido para pedirle que confirme.

A continuación muestro como queda el ejercicio resuelto:

En este caso 'new Actor' va asociado a 'Identificarse', 'Pagar', 'Elegir mensajero' y 'Confirmar'.

'Identificarse' se une a 'Registrarse' mediante una extensión.

Y finalmente 'Pagar con tarjeta' y 'Pagar con transferencia' se unen a 'Pagar' mediante la generalización.



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.

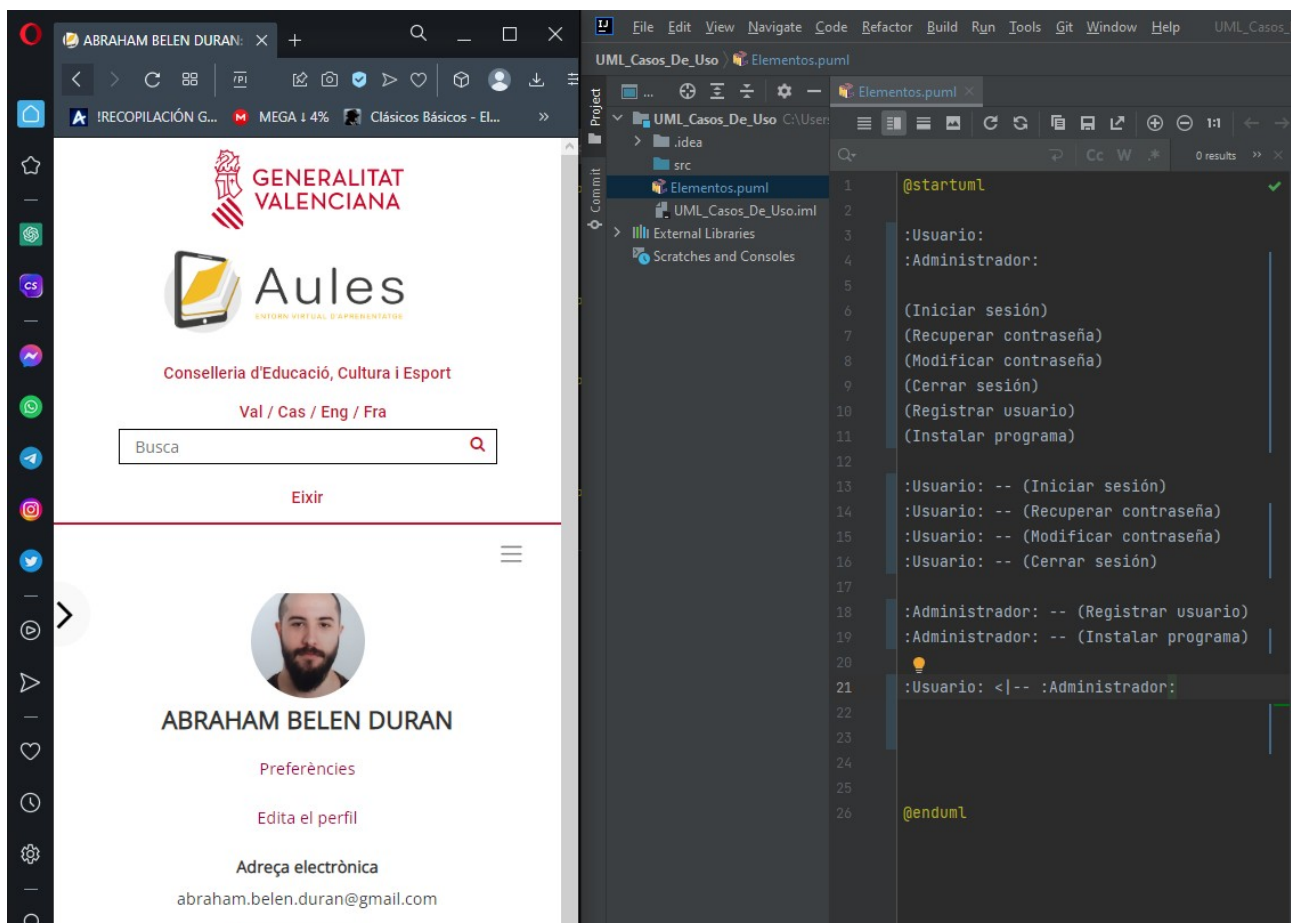
A continuación muestro como queda el ejercicio resuelto:

‘Usuario’ va asociado a ‘Iniciar sesión’, ‘Recuperar contraseña’, ‘Modificar contraseña’ y ‘Cerrar sesión’.

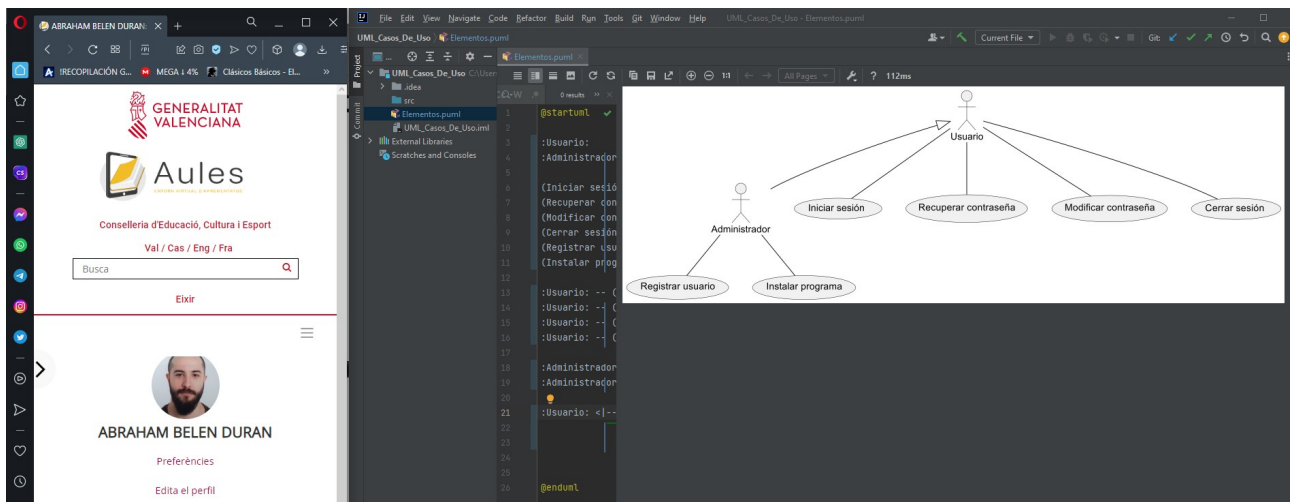
‘Administrador’ va asociado a ‘Registrar usuario’ e ‘Instalar programa’.

Finalmente ‘Usuario’ y ‘Administrador’ se unen mediante una generalización.

Captura de código:



Captura de diagrama UML:



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.

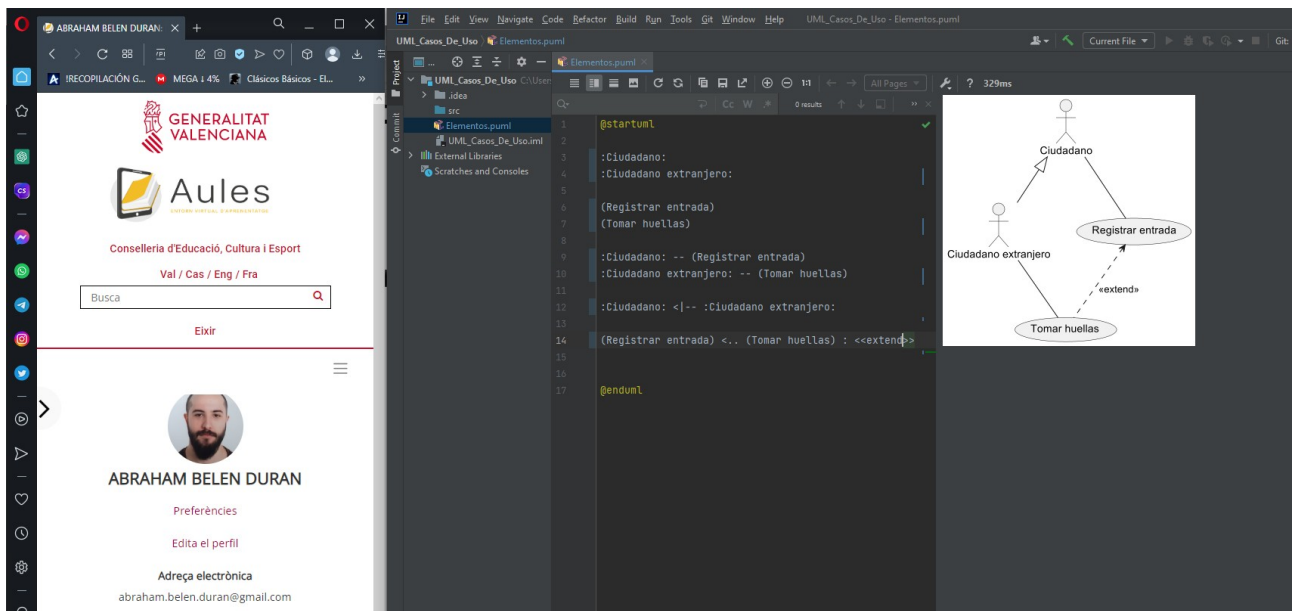
A continuación muestro como queda el ejercicio resuelto:

‘Ciudadano’ va asociado a ‘Registrar entrada’.

‘Ciudadano extranjero’ va asociado a ‘Tomar huellas’.

A continuación ‘Ciudadano extranjero’ se asocia a ‘Ciudadano’ mediante una generalización.

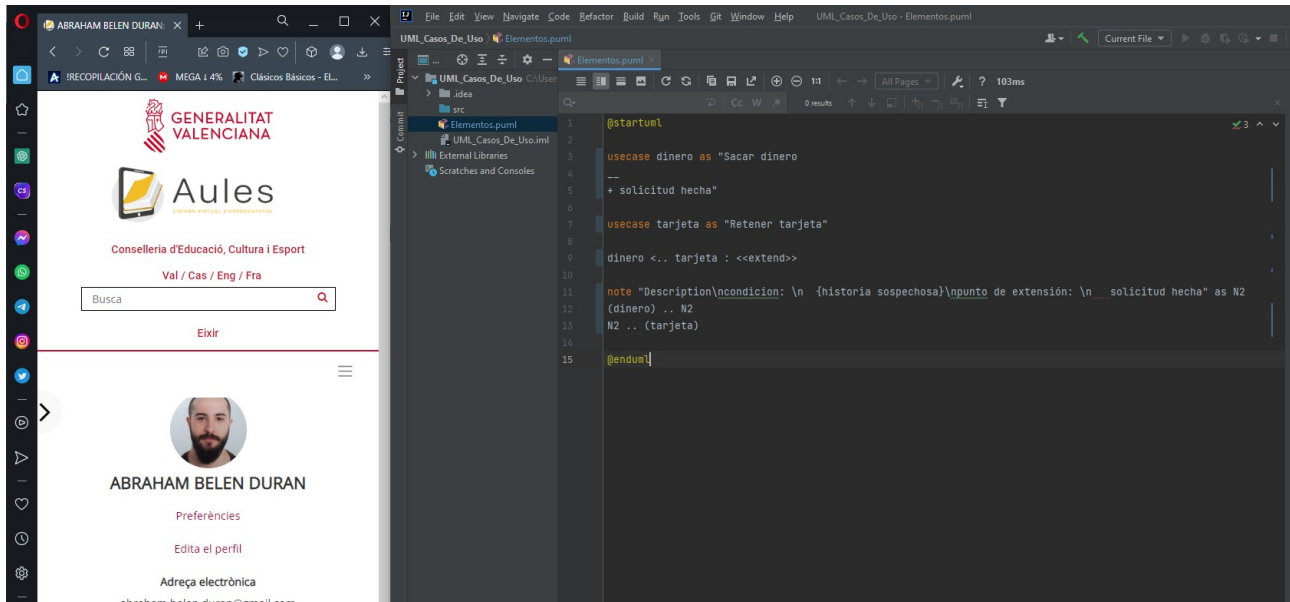
Finalmente ‘Tomar huellas’ y ‘Registrar entrada’ se unen mediante una extensión.



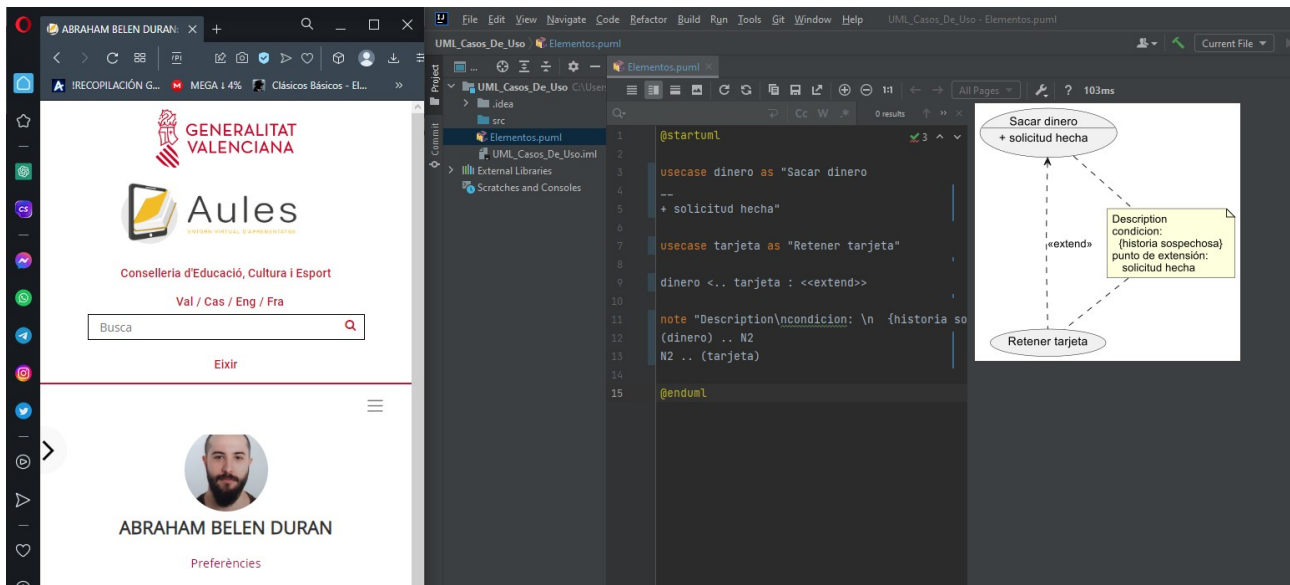
Nuevamente utilizamos la relación «extend» para indicar que la toma de huellas se realiza como parte del registro de entrada cuando se da la condición de que el ciudadano es extranjero, lo que expresamos añadiendo el actor “ciudadano extranjero” que interactúa con este caso de uso.

Y por último vemos los puntos de extensión, estos 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.

Captura de código:



Captura de diagrama UML:



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.