



SPRINT 04: DIAGRAMES DE  
SEQÜÈNCIA

● 14/05- 01/06

# SPRINT 4 QUE MENGES

GRUP 511 05

## MIEMBRES DEL GRUP:

Daniel Alberto Vaca

Nour Kobbi Laghmich

Yang Chen

Liying Sun

# Composició de L'equip



**MIEMBRO 1**

DANIEL ALBERTO  
VACA



**MIEMBRO 2**

NOUR KOBBI  
LAGHMICH



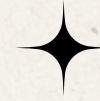
**SCRUM MASTER**

YANG CHEN



**MIEMBRO 4**

LIYING SUN



# Organizació en TRELLO

VISTA PREVIA DEL NOSTRE TAULER

The screenshot shows a Trello board with the following structure:

- Backlog:**
  - Crear diagrama de secuencia
  - Actualizar el documento de especificaciones
- Sprint Backlog:**
  - + Añade una tarjeta
- To do:**
  - Corregir Diagramas de Clase 3 (YC)
  - Actualizar Product Backlog (YC)
  - Corregir Diagramas de Clase 4 (DV)
  - Crear y actualizar BurnDown Chart (YC)
  - Reporte diario de los Daily Meetings al Github (YC)
- En proceso:**
  - Corregir Diagramas de Clase 5 y 6 (NK) - 1 card in progress
  - Creación Presentación Sprint 4 (NK)
- Hecho:**
  - Completar Diagrama de Clase 3 (DV)
  - Subir Códigos UML Diagramas de Clase 3 (DV)
  - Completar Diagramas de Clase 5 y 6 (LS)
  - Subir Códigos UML Diagramas de Clase 4 (NK)
  - Completar Diagrama de Clase (NK)

At the top right, there are filter buttons for NK (Not Yet Started), DV (Doing), LS (In Progress), VM (Waiting for Me), YC (In Progress), and +1 (Completed). There is also a 'Compartir' (Share) button.

ORGANITZACIÓ:  
DANIEL  
NOUR  
YANG  
LIYING  
TODOS

# Tasques realitzades: TRELLO

## TASQUES RELITZADES PER CADA MEMBRE:

- \* Completar Diagramas de Clase 1 y 2.
- \* Completar Diagrama de Clase 3.
- \* Completar Diagrama de Clase 4.
- \* Completar Diagramas de Clase 5 y 6.
- \* Subir Códigos UML Diagramas de Clase 1 y 2.
- \* Subir Códigos UML Diagramas de Clase 3.
- \* Subir Códigos UML Diagramas de Clase 4.
- \* Subir Códigos UML Diagramas de Clase 5 y 6.
- \* Actualizar Product Backlog.
- \* Crear y actualizar BurnDown Chart.
- \* Reporte diario de los Daily Meetings al Github
- \* Corregir Diagramas de Clase 1 y 2.
- \* Corregir Diagramas de Clase 3.
- \* Corregir Diagramas de Clase 4.
- \* Corregir Diagramas de Clase 5 y 6.
- \* Creación carpetas GitHub para Diagramas Secuencia.
- \* Creación Diagramas de Secuencia Bloque 1 y 2.
- \* Creación Diagrama de Secuencia Bloque 3.
- \* Creación Diagramas de Secuencia Bloque 5 y 6.
- \* Creación Diagrama de Secuencia Bloque 4.
- \* Subir código UML Diagramas de Secuencia Bloque 1 y 2.
- \* Subir código UML Diagrama de Secuencia Bloque 5 y 6.
- \* Subir código UML Diagrama de Secuencia Bloque 4.
- \* Subir código UML Diagrama de Secuencia Bloque 3.
- \* Crear diagrama de secuencia.
- \* Unificar Diagramas de Clase.
- \* Preparar presentaci'on Sprint.
- \* Corregir Diagrama de Secuencia Bloq1-2.
- \* Corregir Diagrama Secuencia Bloq3.
- \* Corregir Diagrama Secuencia Bloq4.
- \* Corregir Diagrama Secuencia Bloq5-6.

# GitHub

<https://github.com/dabien3/ES25UAB51105>

The screenshot shows a GitHub repository interface. On the left, the file structure is displayed under the 'Files' tab, showing various UML models and sequence diagrams. On the right, the 'Code' tab displays the statechart code for a payment process.

**Code Tab Content:**

```
1 @startuml
2 actor Cliente
3 participant Comanda
4 control Pago
5 boundary PlataformaPago
6 boundary Tarjeta
7 boundary PayPal
8 entity Cuiner
9 entity Repartidor
10
11 activate Cliente
12 Cliente --> Comanda : iniciarPagoComanda()
13 activate Comanda
14
15 Comanda --> Pago : procesarPagoComanda()
16 activate Pago
17
18 Pago --> PlataformaPago : elegirPlataforma()
19 activate PlataformaPago
20
21 alt Metodo == Tarjeta
22     PlataformaPago --> Tarjeta : procesarPagoT()
23     activate Tarjeta
24     return procesado()
25     deactivate Tarjeta
26 else Metodo == PayPal
27     PlataformaPago --> PayPal : procesarPagoPP()
28     activate PayPal
29     return procesado()
30     deactivate PayPal
31 end
32 deactivate PlataformaPago
33
34 Pago --> Cuiner : procesarPagoCuiner()
35 activate Cuiner
36 return ok
37 deactivate Cuiner
```

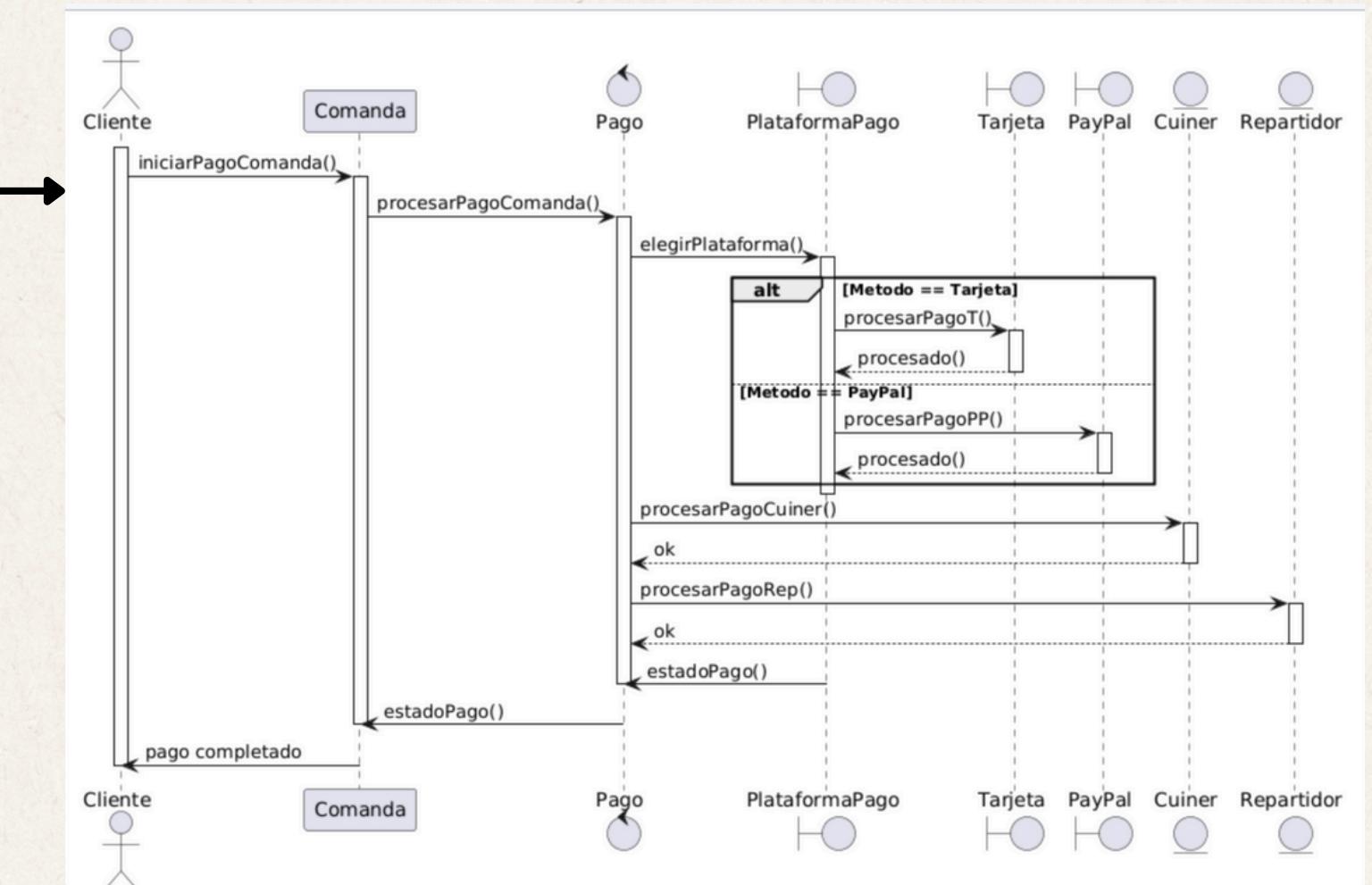
## EXEMPLE MOSTRA DEL DIAGRAMA DE SEQÜÈNCIA DEL BLOC 4

```

1  @startuml
2  actor Cliente
3  participant Comanda
4  control Pago
5  boundary PlataformaPago
6  boundary Tarjeta
7  boundary PayPal
8  entity Cuiner
9  entity Repartidor
10
11 activate Cliente
12 Cliente -> Comanda : iniciarPagoComanda()
13 activate Comanda
14
15 Comanda -> Pago : procesarPagoComanda()
16 activate Pago
17
18 Pago -> PlataformaPago : elegirPlataforma()
19 activate PlataformaPago
20
21 alt Metodo == Tarjeta
22   PlataformaPago -> Tarjeta : procesarPagoT()
23   activate Tarjeta
24   return procesado()
25   deactivate Tarjeta
26 else Metodo == PayPal
27   PlataformaPago -> PayPal : procesarPagoPP()
28   activate PayPal
29   return procesado()
30   deactivate PayPal
31 end
32 deactivate PlataformaPago
33
34 Pago -> Cuiner : procesarPagoCuiner()
35 activate Cuiner
36 return ok
37 deactivate Cuiner
  
```

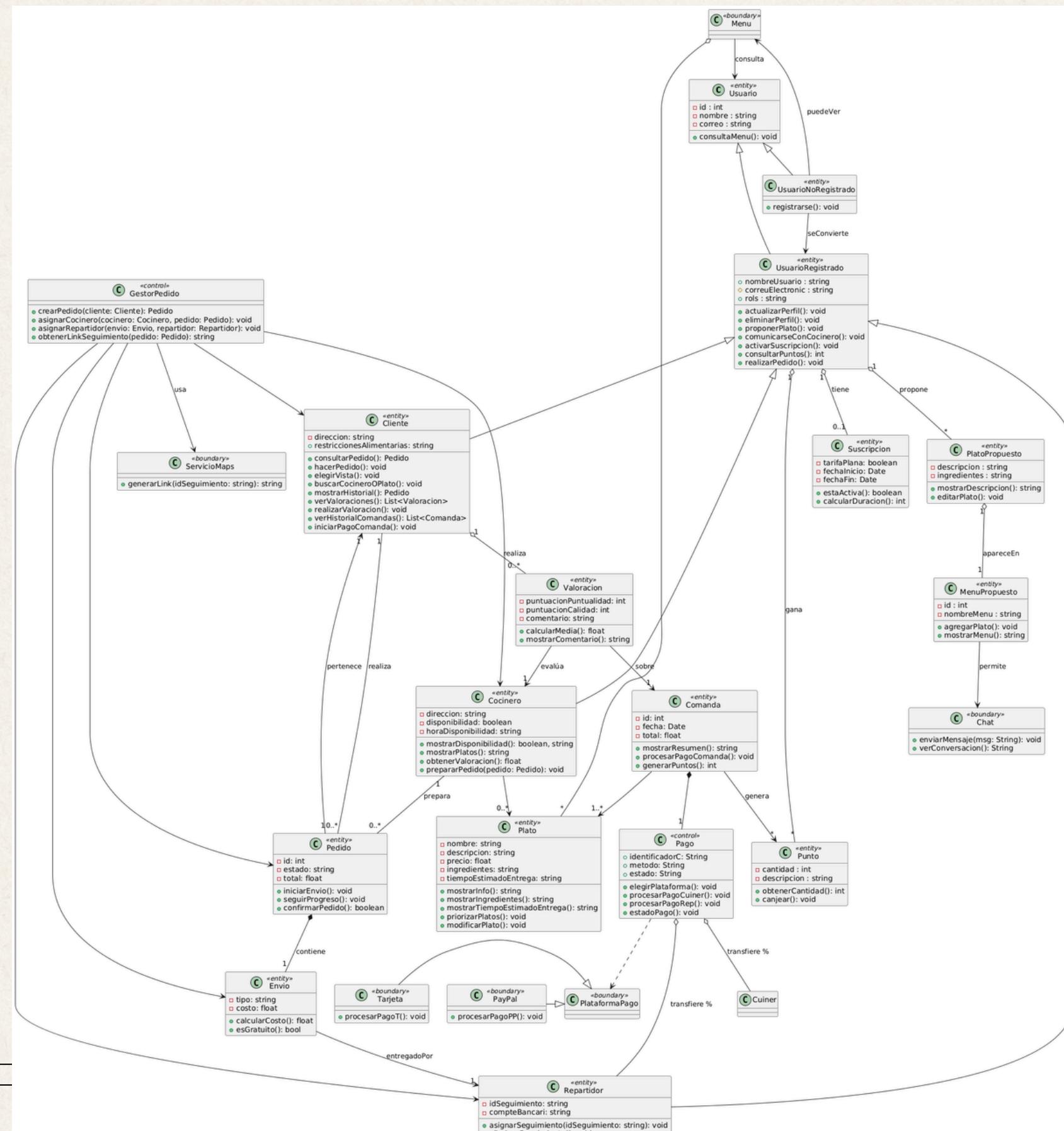
DIAGRAMA SEQÜÈNCIA PNG

DIAGRAMA SEQÜÈNCIA P.UML



# DIAGRAMA FINAL DE CLASSES

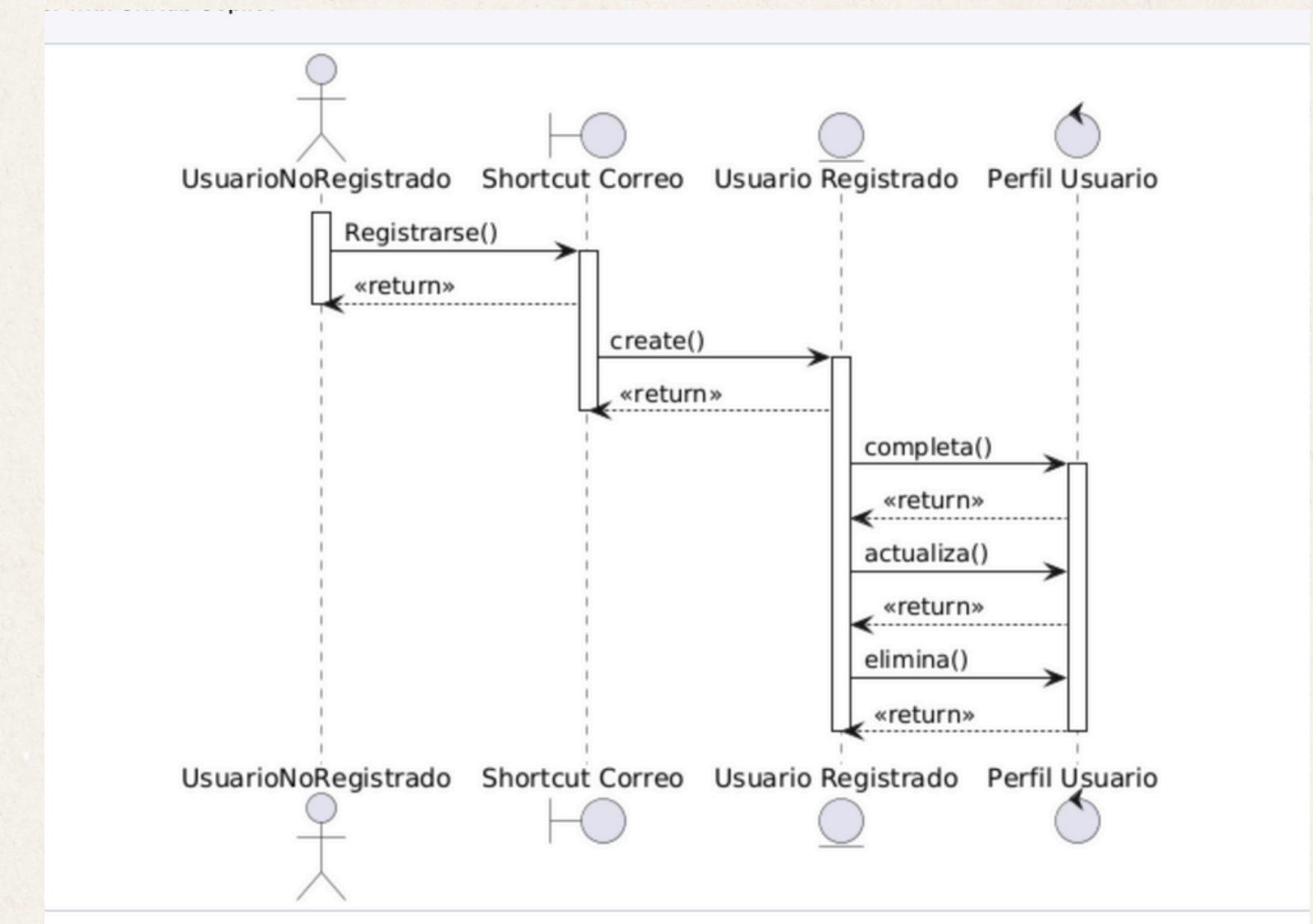
SPRINT 4



# DIAGRAMA DE SEQUÈNCIA BLOC 1

SPRINT 4

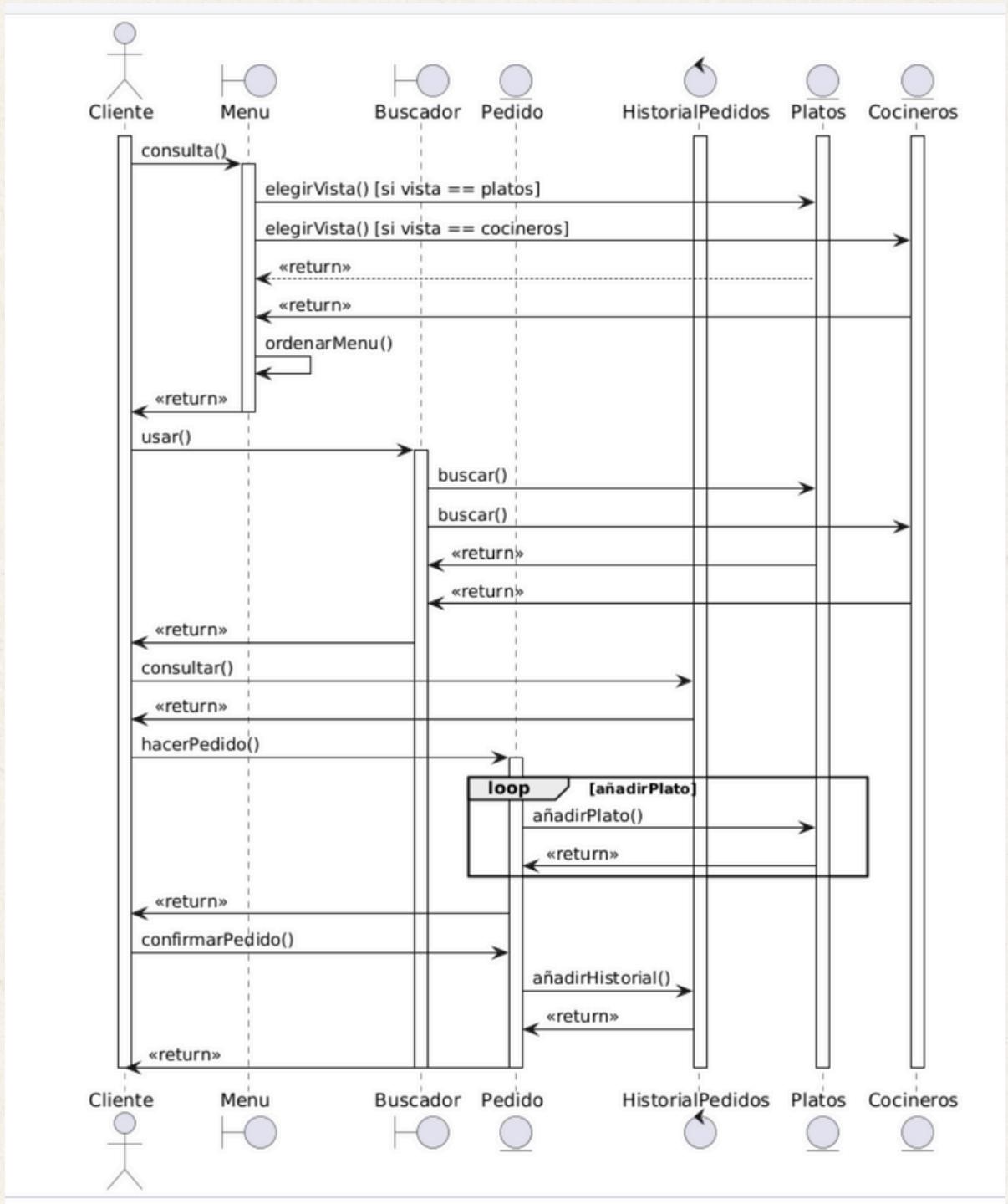
1	<b>CASOS de USO</b>	Bloque 1: Gestión de cuentas y creación de perfiles de usuario.
2	Num Versión	1.0
3	Data	09/04/2025
3	Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Dávalos, Nour Kobbi Laghmich.
4	Descripción	En este caso de uso vemos como se relaciona los actores con relación a la gestión de cuentas y creación de perfiles de usuario.
5	Actores	Cliente, Cocinero, Repartidos, Usuario no Registrado.
6	Precondición	N/A
7	Flujo principal	<ul style="list-style-type: none"> <li>1. Los usuarios podrán actualizar su perfil como también eliminarlo a cualquier momento.</li> <li>2. Los usuarios no registrados podrán tener la opción de registrarse en cualquier momento haciendo uso de Google o Outlook.</li> <li>3. Todos los usuarios registrados serán capaces de pedir comandas.</li> <li>4. Todos los usuarios podrán consultar el menú.</li> </ul>
8	Subflujos	<ul style="list-style-type: none"> <li>1. Los datos del perfil deberán ser llenados de la manera correcta.</li> <li>2. Tanto el nombre de usuario como contraseñas deberán tener su formato correcto, si no saltará un error.</li> </ul>
9	Flujos alternativos	N/A
10	Postcondición	N/A
11	Requisitos no Funcionales	<p>RNF-1-01: El nombre de usuario tendrá 5 caracteres como mínimo y deberá de ser único.</p> <p>RNF-1-02: La contraseña de la cuenta tendrá como mínimo 8 caracteres, de los cuales mínimo 2 serán mayúsculas y 2 caracteres especiales.</p>
12	Prioridad	Normal.
13	Comentarios	N/A



# DIAGRAMA DE SEQUÈNCIA BLOC 2

SPRINT 4

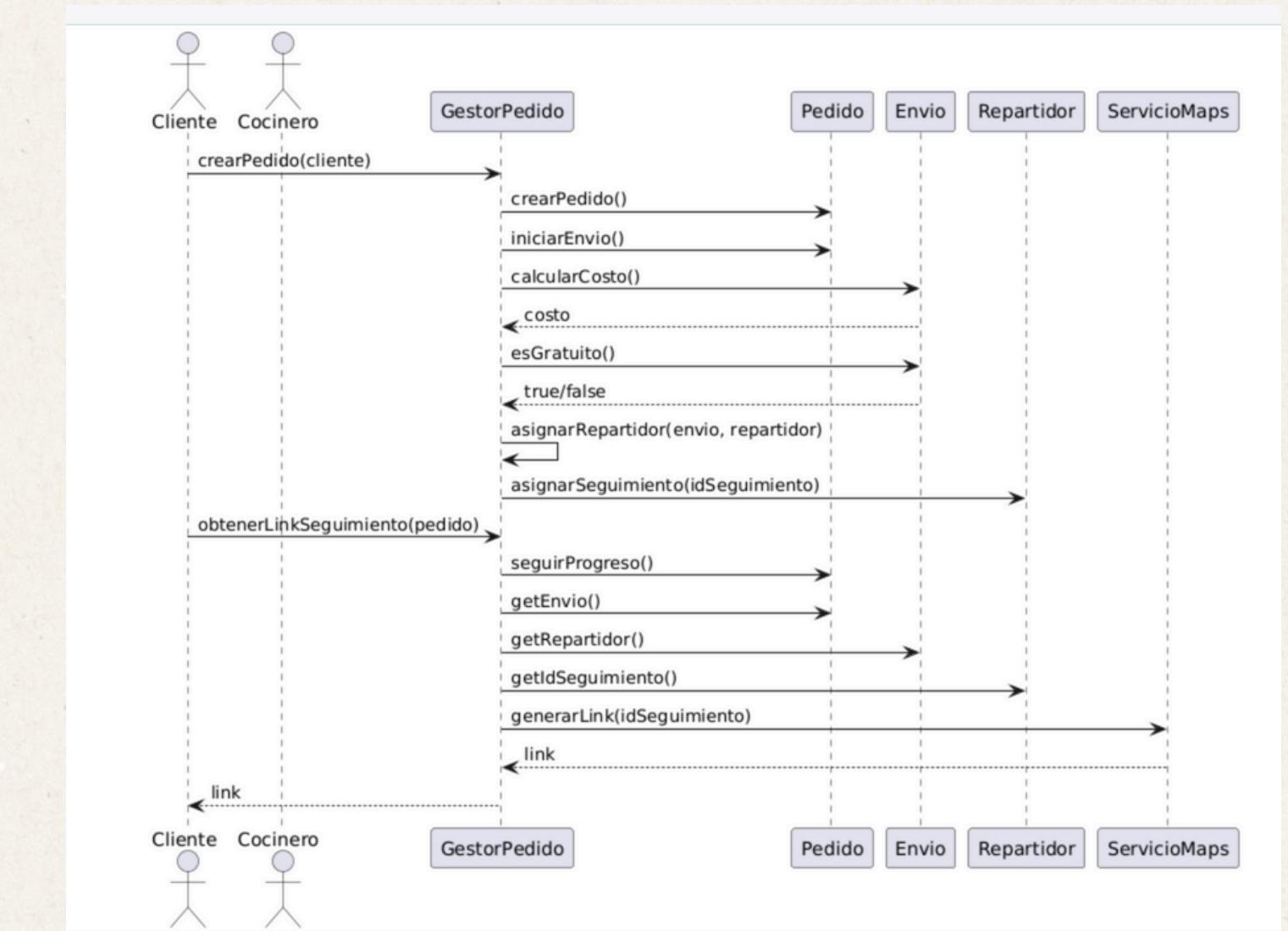
CASOS de USO		Bloque 2: Realización de comandas
1	Num Versión	1.0 Data 09/04/2025
2	Autores	
3	Yang Chen, Liying Sun, Daniel Alberto Vaca Davalos, Nour Kobbi.	
4	Descripción	
5	En este caso de uso se representa cómo los clientes realizan comandas, desde la visualización de platos hasta la confirmación del pedido.	
6	Actores	
7	Cliente	
8	Precondición	
9	Cliente debe estar registrado para realizar pedidos.	
10	Flujo principal	
11	1. El cliente puede ver platos según sus restricciones alimentarias. 2. El cliente puede ver el tiempo estimado de entrega. 3. El cliente puede cambiar entre la vista de platos y cocineros. 4. El cliente puede consultar los platos ofrecidos por un cocinero. 5. El cliente puede buscar platos o cocineros específicos. 6. El cliente puede ver la disponibilidad de un cocinero. 7. El cliente puede consultar los ingredientes y alérgenos de los platos, y en caso de necesario, eliminar ingredientes opcionales de los platos. 8. El cliente puede confirmar y cerrar el pedido. 9. El cliente puede consultar su historial de pedidos. 10. El cliente puede repetir un pedido anterior.	
12	Subflujos	
13	NA	
14	Flujos alternativos	
15	N/A	
16	Postcondición	
17	El pedido queda confirmado o reservado.	
18	Requisitos no Funcionales	
19	RNF-2-01: El sistema debe mostrar resultados priorizados según distancia y restricciones alimentarias de manera eficiente. RNF-2-02: El sistema debe limitar a tres los pedidos activos por cliente.	
20	Prioridad	
21	Normal	
22	Comentarios	
23	Este módulo depende de la correcta configuración del perfil del cliente y de la disponibilidad de los cocineros.	



# DIAGRAMA DE SEQÜÈNCIA BLOC 3

SPRINT 4

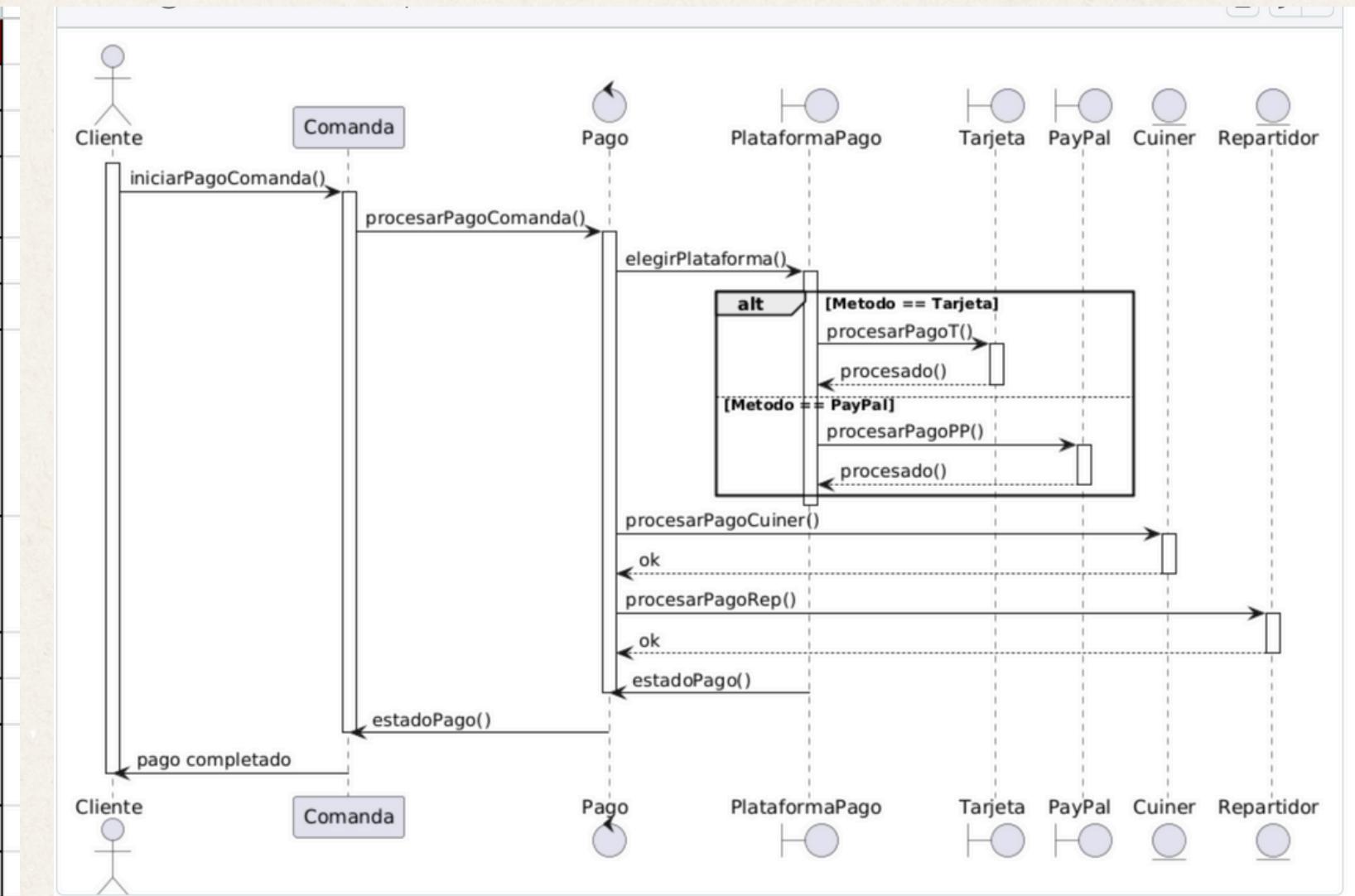
CASOS de USO   Bloque 3: Envío de comandas				
1	Num Versión	1.0	Data 09/04/2025	
2	Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Davalos, Nour Kobbi.		
3	Descripción	En este caso de uso se muestra cómo el cliente interactúa con las distintas opciones de envío del pedido.		
4	Actores	Cliente, Cocinero, Repartidos.		
5	Precondición	El cliente debe haber confirmado un pedido.		
6	Flujo principal	1. El cliente y el cocinero puede seleccionar el tipo de entrega (repartidor, cocinero, o recogida en casa del cocinero). 2. El cliente puede visualizar el costo de envío en función de la distancia entre él y el cocinero. Si escoge recogida, se indica que el envío es gratuito. 3. Una vez iniciado el envío, el cliente puede seguir el estado del pedido en tiempo real. 4. El cliente puede ver la ubicación del repartidor en un mapa.		
7	Subflujos	NA		
8	Flujos alternativos	N/A		
9	Postcondición	El pedido es entregado o recogido por el cliente según la opción seleccionada.		
10	Requisitos no Funcionales	RNF-3-01: Integración con Google Maps para visualizar la posición del repartidor en tiempo real.		
11	Prioridad	Normal		
12	Comentarios	N/A		



# DIAGRAMA DE SEQÜÈNCIA BLOC 4

SPRINT 4

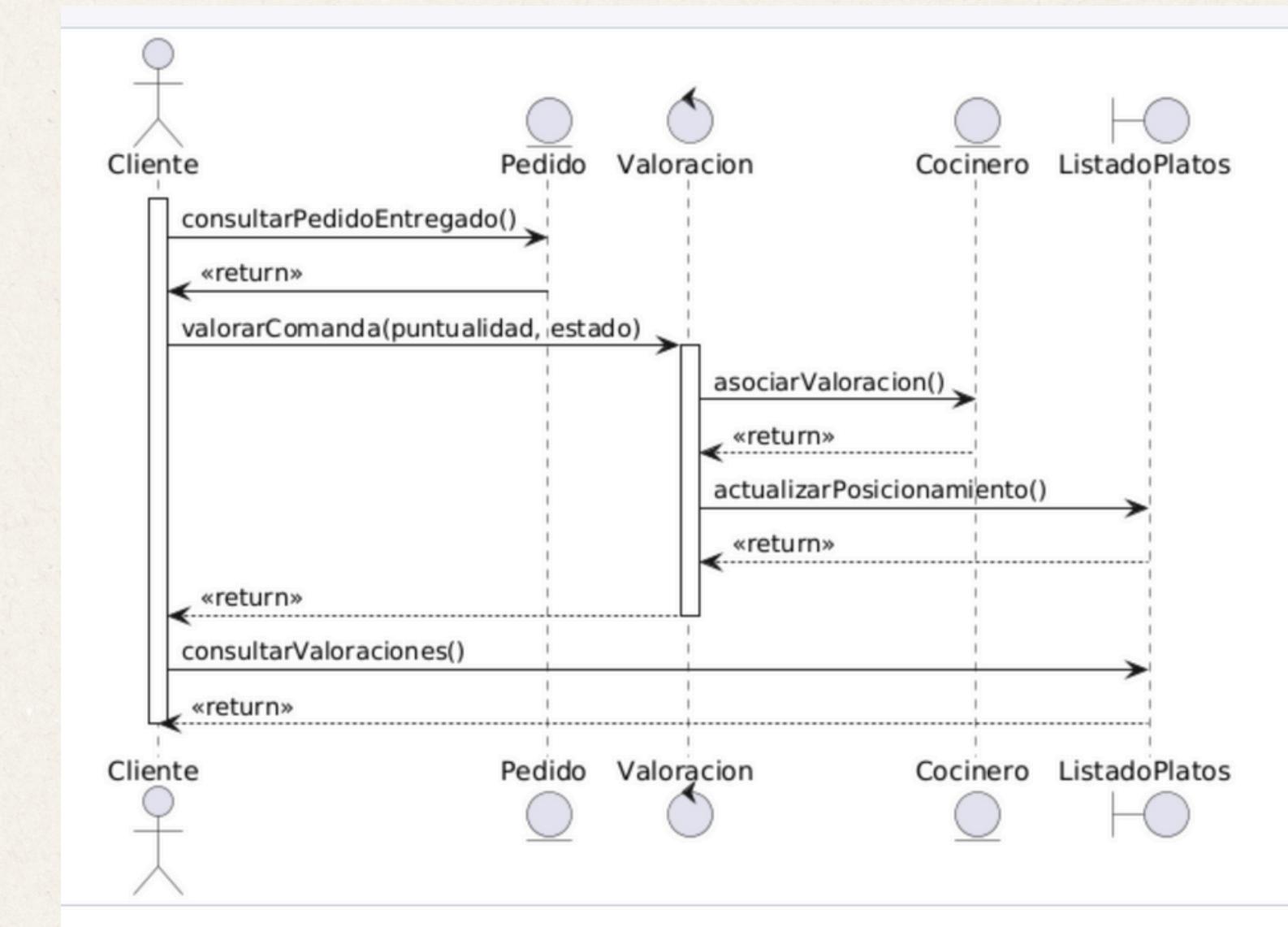
CASOS DE USO Bloque 4: Pago de la comanda			
Num Versión	1.0	Data	09/04/2025
Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Dávalos, Nour Kobbi Laghmich.		
Descripción	Este caso de uso describe cómo se realiza el proceso de pago de una comanda desde la aplicación, así como el reparto del ingreso correspondiente al cocinero y al repartidor.		
Actores	Cliente, Cocinero, Repartidor.		
Precondición	N/A		
Flujo principal	1. El cliente accede al apartado de pago desde la aplicación. 2. Selecciona un método de pago: tarjeta bancaria o PayPal. 3. La aplicación conecta con la plataforma correspondiente para procesar el pago. 4. Una vez realizado el pago, se efectúa el ingreso al cocinero y al repartidor usando sus cuentas bancarias asociadas.		
Subflujos	1. El sistema redirige al cliente al entorno seguro de la plataforma de pago seleccionada. 2. El sistema usa los datos bancarios de los perfiles para repartir el dinero automáticamente.		
Flujos alternativos	N/A		
Postcondición	El pago de comanda queda realizada correctamente.		
Requisitos no Funcionales	RNF-4-01: El sistema debe conectarse con la plataforma de pago correspondiente para autenticar la transacción.		
Prioridad	Alta		
Comentarios	N/A		



# DIAGRAMA DE SEQUÈNCIA BLOC 5

SPRINT 4

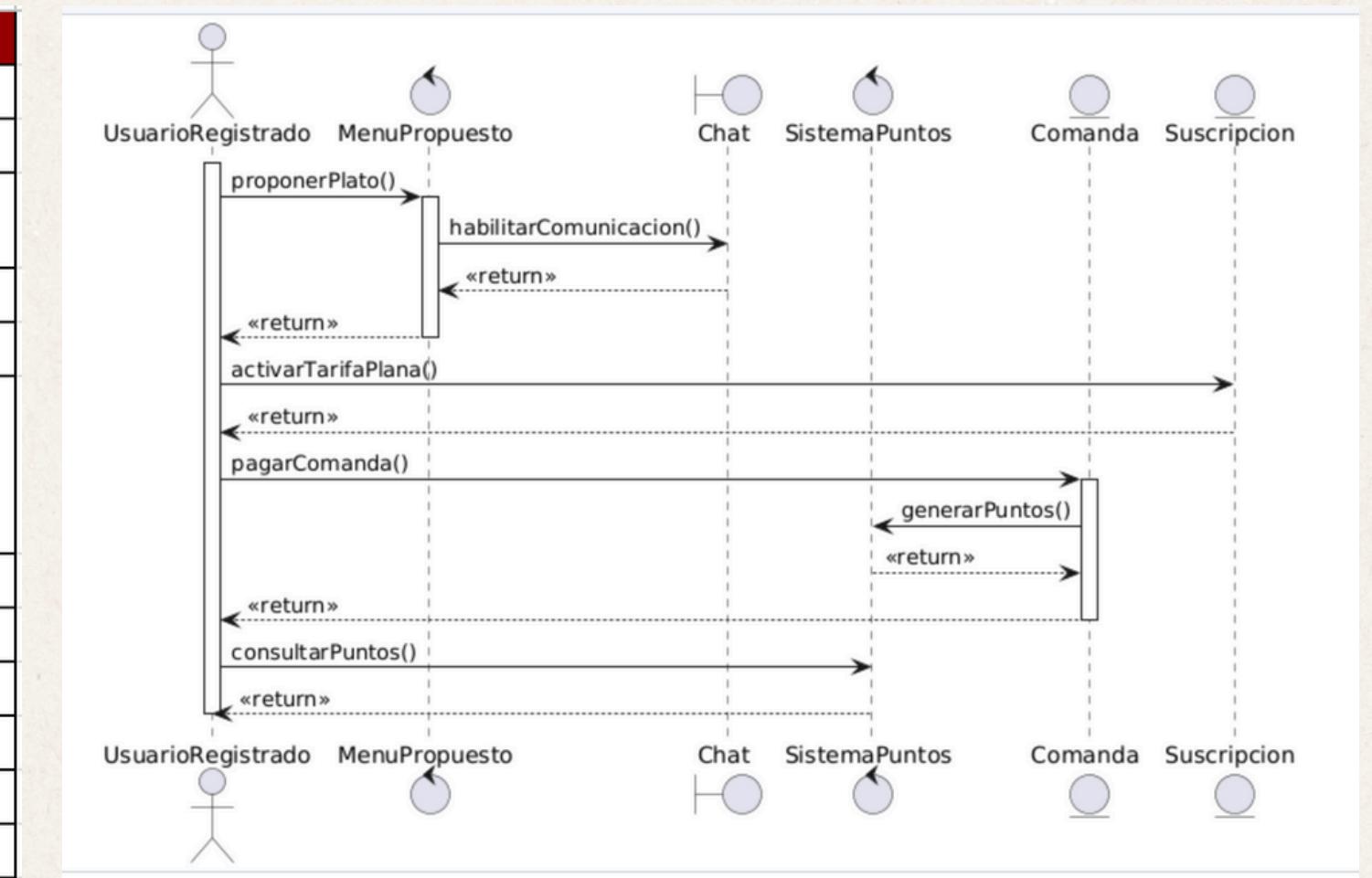
CASOS DE USO Bloque 5: Valoración de la comanda			
Num Versión	1.0	Data	09/04/2025
Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Dávalos, Nour Kobbi Laghmich.		
Descripción	Este caso de uso describe cómo el cliente puede valorar una comanda recibida, incluyendo aspectos como la puntualidad y el estado del plato, y cómo estas valoraciones afectan al posicionamiento del cocinero.		
Actores	Cliente y Cocinero.		
Precondición	La comanda ha sido entregada al cliente.		
Flujo principal	1. El cliente accede a la aplicación para valorar una comanda recibida. 2. Introduce la valoración, incluyendo puntuaciones sobre la puntualidad y el estado del plato. 3. El sistema registra la valoración y la asocia al cocinero correspondiente. 4. La posición del cocinero en el listado de platos disponibles se actualiza en función de sus valoraciones acumuladas.		
Subflujos	1. El cliente puede consultar las valoraciones de otros usuarios antes de realizar un pedido. 2. El cocinero puede acceder a su historial de valoraciones en cualquier momento. 3. El sistema mantiene un registro detallado de cada valoración, vinculada a la comanda y al cocinero.		
Flujos alternativos	N/A		
Postcondición	El cocinero recibe la valoración de una comanda.		
Requisitos no Funcionales	N/A		
Prioridad	Normal.		
Comentarios	N/A		



# DIAGRAMA DE SEQUÈNCIA BLOC 6

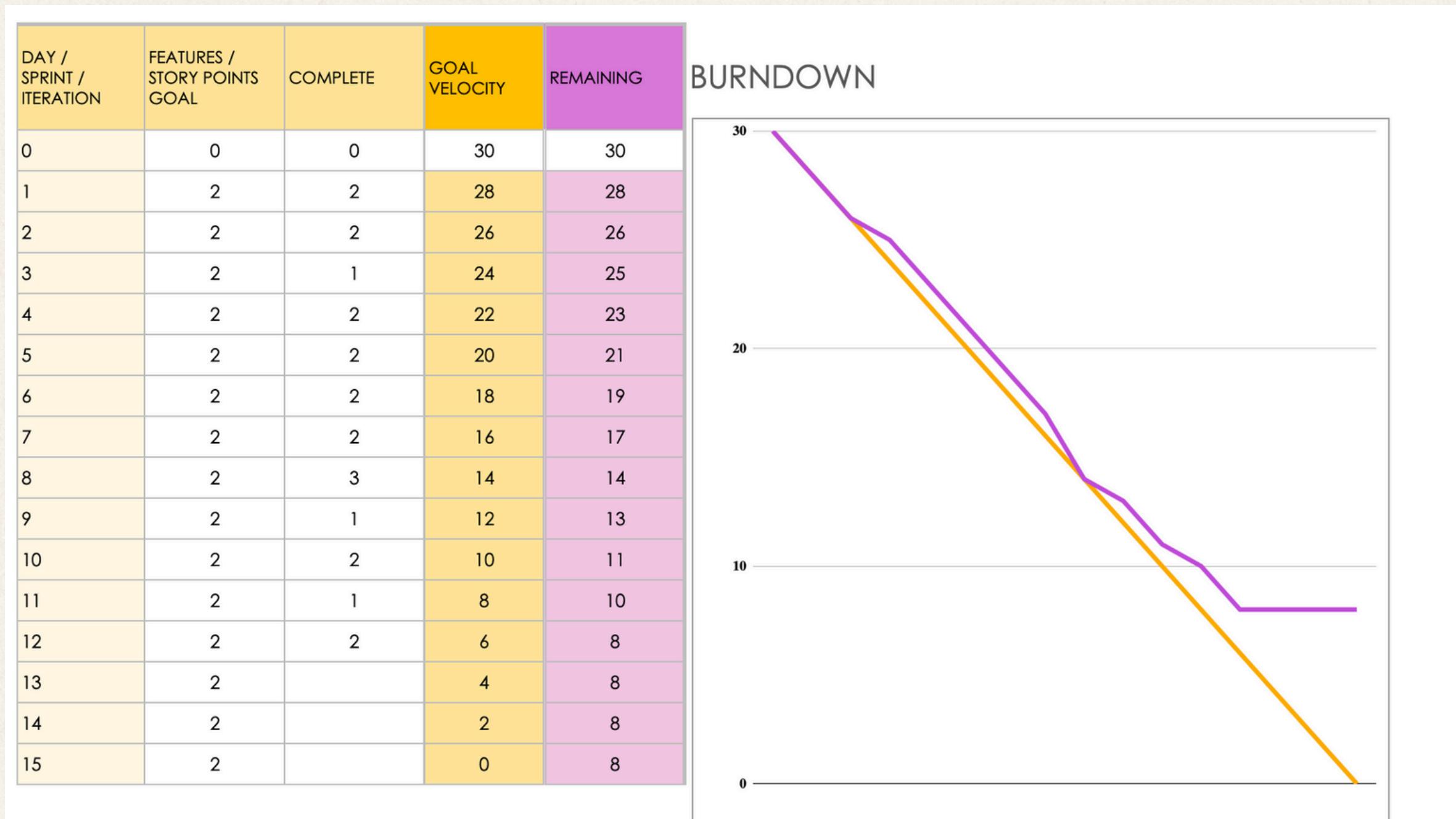
SPRINT 4

CASOS de USO Bloque 6: Funciones adicionales			
Num. Versión	1.0	Data	09/04/2025
Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Dávalos, Nour Kobbi Laghmich		
Descripción	En este escenario veremos como interactuan algunas funcionalidades adicionales con el cliente y el cocinero		
Actores	Cliente y cocinero		
Precondición	N/A		
Flujo principal	1. El cliente propone un plato. 2. El sistema preparará un chat entre el cocinero y el cliente. 3. El cocinero se comprometerá a cocinar el plato personalizado. 4. Si el cliente esta suscrito, obtendrá puntos para canjearlo por descuentos.		
Subflujos	N/A		
Flujos alternativos	N/A		
Postcondición	N/A		
Requisitos no Funcionales	N/A		
Prioridad	Normal		
Comentarios	Comentarios adicionales		



# BURNDOWN CHART

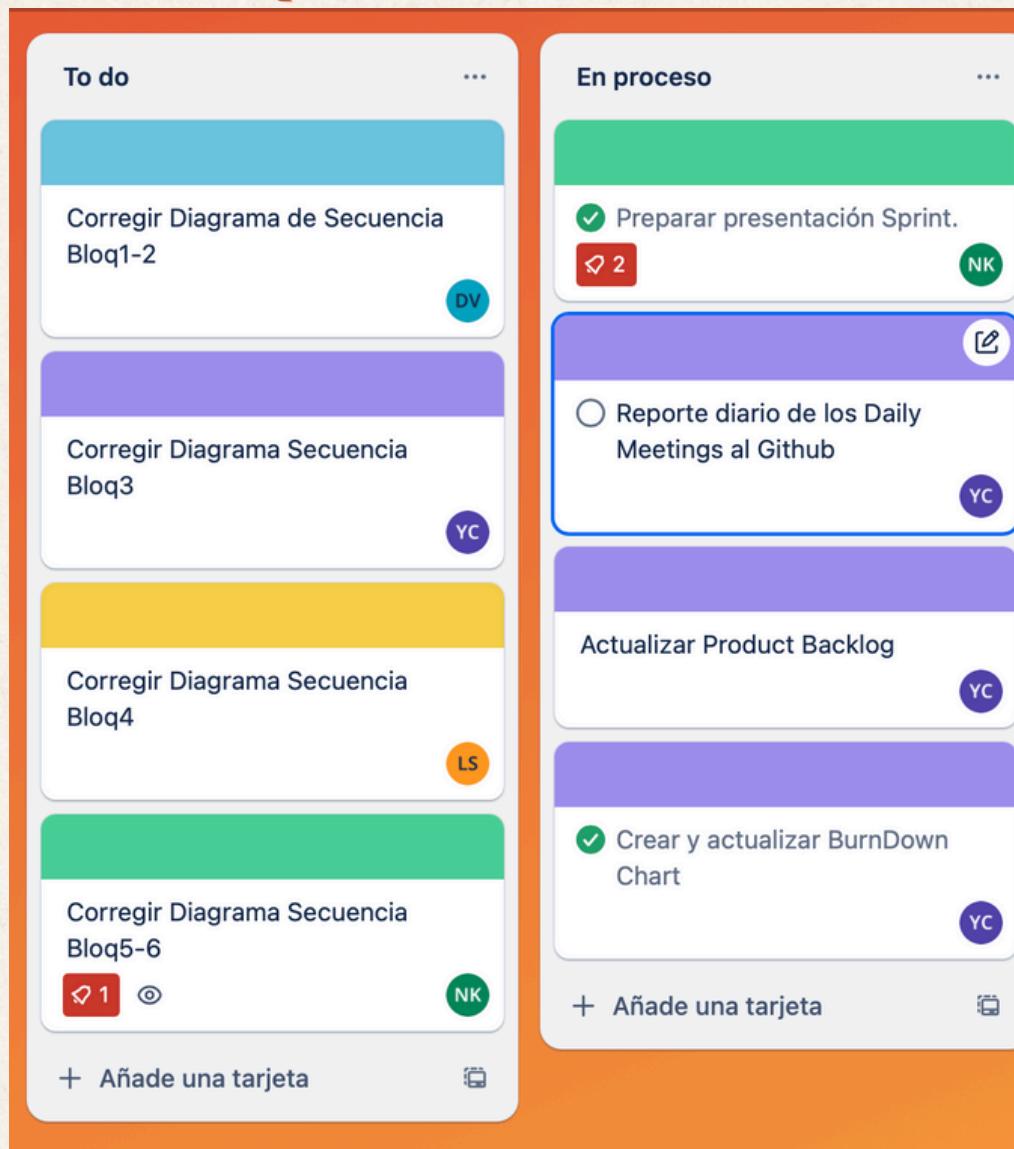
SPRINT 4



# Tasques pendents

ORGANIZACIÓ I PLANIFICACIÓ

## TASQUES PENDENTS



## TREBALL FINAL SPRINT 1 - SPRINT 4

- **Sprint 1:** Definició de Requisits
- **Sprint 2 :** Document S.R.S. i Diagrames de Casos d'ús
- **Sprint 3 :** Especificacions, Diagrama de Classes
- **Sprint 4 :** Diagrames de Seqüència

**DATA: 01/06**



SPRINT 04: DIAGRAMES DE  
SEQÜÈNCIA

**i 14/05- 01/06**

# SPRINT 4 QUE MENGES

**GRUP 511 05**

## **MIEMBRES DEL GRUP:**

Daniel Alberto Vaca

Nour Kobbi Laghmich

Yang Chen

Liying Sun