

Documento De Especificaciones

Autors:

Yang Chen
Nour Kobbi Laghmich
Liyang Sun
Daniel Alberto Vaca Davalos

Versión 1	16/03	
Versión 2	17/03	<ul style="list-style-type: none">- Añadir el índice.- Añadir la tabla de versiones.- Actualizar historias de usuario.- Actualizar la lista de los requisitos(no formal).- Crear la tabla de requerimientos.
Versión 3	23/03	<ul style="list-style-type: none">- Actualizar la lista formal de los requisitos.- Actualizar el índice.- Añadir la versión 3 a la tabla de versión.

Versió 4	06/05	- Añadir apartado 6 de Diagramas de casos de uso
Versió 5	10/05	<ul style="list-style-type: none"> - Actualizar enlaces apartado 6 - Añadir tablas descripció caso de uso.
Versió 6	12/06	- Enlaces presentaciones Sprints

Índice

1. Lista de los integrados del equipo de desarrollo.....	2
2. Enlaces al proyecto a Trello i GitHub.....	2
3. Historias de usuario.....	2
4. Lista de los perfiles de usuario identificados.....	6
5. Lista formal de requisitos organizada por categorías.....	6
5.1. Grupo 1: Gestión de cuentas y creación de perfiles de usuario.....	6
5.2. Grupo 2: Realización de comandas.....	9
5.3. Grupo 3: Envío de comandas.....	12
5.4. Grupo 4: Pago de la comanda.....	14
5.5. Grupo 5: Valoración de la comanda	15
5.6. Grupo 6: Funcionalidades adicionales.....	16
5.7. Otras consideraciones.....	17
6. Diagramas de casos de uso.....	18

1. LISTA INTEGRANTES DEL PORYECTO

- 1) Daniel Alberto Vaca Davalos
- 2) Nour Kobbi Laghmich
- 3) Liying Sun
- 4) Yang Chen

2. ENLACES DEL TABLERO TRELLO Y REPOSITORIO GITHUB

- *Trello*

<https://trello.com/invite/b/67cf23fdc295a12aba038187/ATTI38696ccc363fb4b49b705111effe9d598DE3F93E/es25uab-511-05>

- *Github*

<https://github.com/dabien3/ES25UAB-511-05.git>

3. HISTORIAS DE USUARIO

3.1. Bloque 1: GESTIÓ DE COMPTES I CREACIÓ DE PERFILS D' USUARI

- Como usuario quiero poder mirar los menús diferentes a cada momento para así saber las diferentes opciones de comida a elegir.
- Como usuario quiero poder identificarme como usuario o por plataformas altamente usadas como Google o Outlook, para poder facilitar el uso y registramiento dentro de la aplicación.
- Como usuario registrado quiero poder actualizar en cualquier momento mi perfil, para mantener mis datos actualizados.
- Como usuario registrado quiero dar de baja mi perfil en cualquier momento.
- Como usuario quiero elegir el tipo de cuenta que crearé (cliente, repartidor o cocinero), para determinar mi rol en la aplicación
- Como usuario quiero poder ser los 3 tipos de roles de la aplicación, para adaptarme a mis necesidades.

3.2. Bloque 2: REALITZACIÓ DE COMANDES

- Como usuario quiero poder tener una preferencia de platos, a través de datos proporcionados en mi perfil, como pueden ser alérgenos,

de esta manera los primeros platos que me salgan serán platos de mi agrado.

- Como usuario quiero poder ver el tiempo de espera de la entrega de los platos para así poder tener un rasgo más a elegir entre las diferentes opciones.
- Como usuario quiero poder tener una lista de platos o menús individuales como también de cocineros que ofrecen sus servicios culinarios, en este último caso además, querría poder ver los platos que ofrece dicho cocinero, todo esto para poder tener diferentes vistas a elegir, no solo platos como tal, sino también su cocinero.
- Como usuario quiero que la lista que se me muestre, sea de cocineros o platos, este ordenado según la distancia física entre usuario y cocinero, así habrá un rasgo más que me hará decantar por un plato u otro.
- Como usuario quiero poder tener un buscador tanto de platos como de cocineros, esto ayudará a poder tener una búsqueda más específica en el caso de que sepa de antemano que o de quien querré pedir.
- Como cocinero quiero poder aparecer en una lista de no disponibles, que indique mi horario de disponibilidad, de esta manera los usuarios sabrán cuando no y cuando si enviarme comandas.
- Como usuario quiero poder reservar un plato a un cocinero para una hora a la que esté disponible, así no hay que estar atento a que llegue la hora para poder pedir, ya se deja pedido de antemano.
- Como usuario quiero poder revisar los ingredientes que tiene un plato, como también modificar alguno que sea indicado por el cocinero como opcional. Así puedo quitar aquellos ingredientes que no sean de mi agrado.
- Como usuario quiero poder consultar los alergenicos, para así poder saber si ese plato puede ser consumido o no.
- Como cocinero y usuario quiero que una vez el plato haya sido seleccionado se cierre la comanda y se proceda el envío, así evitamos que haya errores.
- Como usuario, no podré tener más de tres comandas activas, así se garantiza el correcto funcionamiento de todo.

- Como usuario quiero poder mirar todas las comandas activas como también las hechas en el pasado, con la opción de poder repetir una misma comanda si es necesario, de esta manera puedo controlar mi comanda a la vez que agilizo el proceso de pedir reiterativamente una misma comanda.

3.3. Bloque 3: ENVIAMENT DE COMANDES

- Como cliente quiero elegir entre los 3 métodos de entrega, para elegir la opción que más me convenga.
- Como cliente quiero que el precio de entrega se calcule en la aplicación directamente, para así saber qué opción de entrega me conviene más.
- Como cliente que ha pagado un pedido quiero seguir el progreso de mi envío en todo momento, para saber qué mi comida ya está en camino.
- Como cliente que ha pagado un pedido quiero seguir en todo momento por donde va el repartidor con mi pedido para saber cuánto le falta para llegar.
- Como cliente que ha pagado un pedido quiero seguir el pedido en Google Maps para saber visualmente por donde va.

3.4. Bloque 4: PAGAMENT DE LA COMANDA

- Como cliente, quiero poder elegir entre pagar con tarjeta bancaria o Paypal, para poder pagar con el método que me convenga.
- Como cliente, quiero que se me redirija a la plataforma correspondiente a autenticar mis datos, para poder completar la transacción de manera segura.
- Como repartidor o cocinero, quiero que la aplicación ingrese directamente el pago del pedido a mi cuenta bancaria para recibir mi pago correctamente.
- Como repartidor o cocinero, quiero vincular mi cuenta bancaria a mi perfil en cualquier momento para recibir todos los pagos a la cuenta especificada.

3.5. Bloque 5: VALORACIÓ DE LA COMANDA

- Como cliente quiero poder valorar la comanda recibida, pudiendo indicar diferentes aspectos como puntualidad o estado de la comida, de esta manera puedo contribuir a otros usuarios a la hora de decidir qué o de quien comer.
- Como cocinero quiero que esas valoraciones se puedan asociar a mí y que me dé una pequeña ventaja a la hora de salir en las listas de cocineros o platos disponibles, de esta manera se recompensará mi esfuerzo.

3.6. Bloque 6: FUNCIONALITATS ADDICIONALS

- Como cliente quiero poder proponer platos de mi agrado para consumir, de esta manera podré consumir aquello que quiera aunque no esté disponible.
- Como cliente quiero que esos platos elegidos por mi aparezcan en un menú específico por el cual el cocinero se comprometerá a cocinarlo, como también poder tener un chat privado con él, de esta manera me aseguro de que el plato sea cocinado a mi gusto.
- Como cliente quiero poder tener la opción a pagar un plan mensual que ofrezca una tarifa plana de envíos y un programa de puntos, que se consiga pagando por comidas en la aplicación y que pueden canjearse por un descuento en la comida en un futuro, de esta manera puedo tener un beneficio extra de otra manera.

3.7. Bloque7:ALTRES CONSIDERACIONS

- Como usuario quiero que la aplicación esté disponible en catalán, castellano e inglés, para poder usar la interfaz en el idioma con el que me sienta más cómoda.
- Como usuario quiero poder acceder a la aplicación tanto desde la versión web como desde la versión móvil, para adaptarse al dispositivo que esté utilizando en cada momento.
- Como usuario quiero que la aplicación móvil funcione tanto en Android como en IOS para asegurarme de que mi sistema operativo está soportado.

- Como usuario quiero que la aplicación utilice el tipo de letra Calibri en todos los apartados, para mantener la coherencia visual.
- Como usuario quiero que el color principal de la aplicación sea naranja y que el logo esté visible en todas las pantallas, para identificar fácilmente la marca de QueMenges.
- Como usuario quiero que la aplicación tenga un tiempo de respuesta inferior a 5 ms cuando realizo una petición, para que la experiencia sea rápida y eficiente.
- Como administrador quiero que la aplicación pueda soportar hasta 1000 usuarios conectados simultáneamente, para garantizar la estabilidad en horas punta.
- Como usuario quiero que la aplicación utilice el protocolo OpenVPN para la encriptación de mis datos personales y de pago, para garantizar la seguridad y privacidad de mi información.

4. LISTA DE PERFILES DE USUARIO IDENTIFICADOS

- Cliente-Realiza pedidos de platos caseros.
- Cocinero-Vende sus platos caseros.
- Repartidor-Se encarga de las entregas de los pedidos.
- Usuario no registrado-Aquel usuario que no se registró con Google o Outlook.

5. LISTA FORMAL DE REQUISITOS POR BLOQUES

Enlace al repositorio a la carpeta de requisitos

<https://github.com/dabien3/ES25UAB-511-05/tree/main/requirements>

5.1. Grupo 1: Gestión de cuentas y creación de perfiles de usuario

Requisitos Funcionales

RF-1-01 : Un usuario podrá consultar los menús, esté o no registrado.

- Entrada: Usuario(registrado o no) accede a la app o web.
- Proceso: El sistema permite el acceso a la visualización de menús sin necesidad de autenticación.
- Salida: Lista de menús visible para cualquier usuario.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-1-01.md

RF-1-02: Solo los usuarios registrados podrán hacer pedidos.

- Entrada: Usuario intenta realizar un pedido.
- Proceso: El sistema analiza si el usuario que intenta realizar el pedido está registrado o no.
- Salida: Si el usuario es un cliente puede realizar el pedido, en caso contrario no podrá hacerlo.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-1-02.md

RF-1-03: Los usuarios podrán registrarse con una cuenta de correo.

- Entrada: Un usuario intenta registrarse.
- Proceso: El sistema recibirá la petición de registro y la procesa.
- Salida: El usuario ya está registrado en la aplicación.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-1-03.MD

RF-1-04: Los usuarios podrán registrarse a través de shortcuts, en este caso de Gmail y Outlook .

- Entrada: Un usuario usa un shortcut
- Proceso: La aplicación lo redirige a la plataforma correspondiente para que realice la autenticación.
- Salida: El usuario ya está registrado a través de un shortcut.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-01-04.md

RF-1-05: Los clientes que vayan a realizar un pedido deberán completar un perfil donde completarán sus datos, entre ellos un nombre de usuario.

- Entrada: Un cliente registrado intenta realizar un pedido
- Proceso: El sistema solicita al cliente que complete su perfil, incluyendo datos como nombre de usuario, dirección, teléfono, etc.
- Salida: El perfil del cliente queda completo y puede proceder a realizar el pedido.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-01-05.md

RF-1-06: Un usuario puede actualizar en todo momento su perfil, incluso cambiar sus roles dentro de la aplicación.

- Entrada: Un usuario registrado accede a la sección de edición de perfil y realiza cambios.
- Proceso: El sistema valida los cambios solicitados y actualiza la información del perfil.
- Salida: El perfil del usuario se actualiza con los nuevos datos y/o roles seleccionados.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-01-06.md

RF-1-07: Un cliente puede eliminar en cualquier momento su perfil.

- Entrada: Un usuario registrado solicita la eliminación de su perfil.
- Proceso: El sistema elimina el perfil del usuario
- Salida: El perfil del usuario se elimina

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-01-07.md

RF-1-08:A la hora de registrarse, el usuario tiene que indicar que tipo de cuenta se puede crear(cliente, cocinero o repartidor).

- Entrada: Un nuevo usuario al registrarse selecciona el tipo de cuenta que desea crear.
- Proceso: El sistema registra al usuario con el tipo de cuenta seleccionado.
- Salida: El usuario se crea la cuenta con los roles seleccionados.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-01-08.md

RF-1-09:Un usuario puede tener todos los tipos de roles disponibles si lo desea.

- Entrada: Un usuario asocia múltiples roles a su perfil.
- Proceso: El sistema asocia los roles seleccionados por el usuario a su perfil.
- Salida: El usuario ya tiene asignados los roles seleccionados a su perfil.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_FUNCIONALES/RF-01-09.md

Requisitos no Funcionales

-Decisión de diseño:

RNF-1-01:El nombre de usuario tendrá 5 caracteres como mínimo y deberá de ser único.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_NO_FUNCIONAL/ES/RNF-1-01.md

- Decisión de diseño:

RNF-1-02:La contraseña de la cuenta tendrá como mínimo 8 caracteres, de los cuales mínimo 2 serán mayúsculas y 2 caracteres especiales.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE1/REQUISITOS_NO_FUNCIONAL/ES/RNF-01-02.md

5.2. Grupo 2: Realización de comandas

Requisitos Funcionales

RF-2-01 : Los platos y menús se muestran priorizando aquellos que cumplen con las restricciones alimentarias del perfil del cliente.

- Entrada: Perfil del cliente con restricciones alimentarias.
- Proceso: El sistema filtra y prioriza los platos/menús que se ajustan a dichas restricciones.
- Salida: Lista personalizada de platos y menús priorizados.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-01.md

RF-2-02 : Se muestra el tiempo estimado de entrega de cada plato o menú.

- Entrada: Información del plato/menú y ubicación del cliente.
- Proceso: El sistema calcula el tiempo estimado de entrega basado en la distancia y disponibilidad.
- Salida: Tiempo estimado visible en la interfaz del cliente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-02.md

RF-2-03 : Los clientes pueden elegir entre dos vistas: la lista de platos y menús o la lista de cocineros disponibles.

- Entrada: Selección del tipo de vista por el cliente.
- Proceso: El sistema cambia la visualización según la preferencia del usuario.
- Salida: Visualización de platos/menús o cocineros disponibles.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-03.md

RF-2-04 : Si se consulta un cocinero, se muestra la lista de todos los platos que ofrece dicho cocinero.

- Entrada: Selección de un cocinero específico.
- Proceso: El sistema recupera y muestra los platos vinculados al cocinero.
- Salida: Lista de platos ofrecidos por el cocinero.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-04.md

RF-2-05 : Los clientes pueden utilizar un buscador para buscar cocineros o platos específicos.

- Entrada: Palabras clave ingresadas en el buscador.
- Proceso: El sistema realiza una búsqueda en la base de datos.
- Salida: Resultados relevantes mostrados al cliente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-05.md

RF-2-06 : Los cocineros pueden aparecer como "no disponibles", indicando la hora a partir de la cual aceptarán pedidos, a la cual los clientes pueden reservar pedidos para entonces.

- Entrada: Disponibilidad del cocinero.
- Proceso: El sistema muestra la hora futura en la que el cocinero aceptará pedidos y permite realizar reservas.
- Salida: Estado de disponibilidad visible con opción de reserva futura.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-06.md

RF-2-07 : Los clientes pueden consultar la lista de ingredientes de cada plato, así como sus alérgenos, y si es necesario, eliminar algún ingrediente opcional.

- Entrada: Selección de un plato.
- Proceso: El sistema muestra los ingredientes y permite modificar los opcionales.
- Salida: Visualización personalizada del plato con modificaciones.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-07.md

RF-2-08 : El cliente puede confirmar el plato y cerrar el pedido para iniciar el envío.

- Entrada: Confirmación del plato del cliente.
- Proceso: El sistema registra el pedido y lo prepara para envío.
- Salida: Pedido confirmado y en proceso de envío.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-08.md

RF-2-09 : Los clientes pueden consultar el historial de pedidos realizados y, si es necesario y posible, repetir pedidos anteriores.

- Entrada: Solicitud del historial del cliente.
- Proceso: El sistema recupera el historial y permite repetir pedidos si están disponibles.
- Salida: Visualización del historial y opción de repetir pedido.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_FUNCIONALES/RF-2-09.md

Requisitos no Funcionales

Requisitos de rendimiento:

RNF-2-01 :El sistema debe mostrar los resultados priorizados según la distancia y las restricciones alimentarias de manera eficiente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_NO_FUNCIONAL/ES/RNF-2-01.md

Restricciones de rendimiento

RNF-2-02: El sistema debe limitar a tres los pedidos activos por cliente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE2/REQUISITOS_NO_FUNCIONAL/ES/RNF-2-02.md

5.3. Grupo 3: Envío de comandas

Requisitos funcionales (RF):

RF-3-01: El sistema ofrece tres opciones de envío: repartidor contratado por la aplicación, el cocinero entrega el pedido, o el cliente recoge el pedido en el domicilio del cocinero.

- Entrada: Selección del método de envío.
- Proceso: El sistema gestiona la logística según la opción elegida.
- Salida: Confirmación de método de envío seleccionado.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE3/REQUISITOS_FUNCIONALES/RF-3-01.md

RF-3-02: El costo de envío se calcula según la distancia entre el cocinero y el cliente.

- Entrada: Dirección del cliente y del cocinero.
- Proceso: Cálculo de distancia y tarifa de envío.
- Salida: Monto del costo de envío mostrado al cliente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE3/REQUISITOS_FUNCIONALES/RF-3-02.md

RF-3-03: El envío es gratuito si el cliente recoge el pedido personalmente.

- Entrada: Selección de recogida personal.
- Proceso: Se omite el cálculo del coste de envío.
- Salida: Costo de envío indicado como gratuito.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE3/REQUISITOS_FUNCIONALES/RF-3-03.md

RF-3-04: Una vez pagado el pedido e iniciado el envío, el cliente puede seguir su progreso en tiempo real.

- Entrada: Pedido pagado y confirmado.
- Proceso: El sistema actualiza y muestra el estado del envío en tiempo real.
- Salida: Visualización del estado del pedido para el cliente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE3/REQUISITOS_FUNCIONALES/RF-3-04.md

RF-3-05: Se muestra la posición actual del repartidor en un mapa en la web.

- Entrada: Localización del repartidor mediante GPS.
- Proceso: Integración con mapa y actualización en tiempo real.
- Salida: Posición del repartidor mostrada al cliente en el mapa.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE3/REQUISITOS_FUNCIONALES/RF-3-05.md

Requisitos no funcionales (RNF):

- Requisitos sobre las interfaces externas:

RNF-3-01: Integración con Google Maps para la visualización de la posición del repartidor en tiempo real.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/4927d5f6ff81270901ec4a0e26cf2c6666dcf4e3/requirements/BLOQUE3/REQUISITOS_NO_FUNCIONAL/ES/RNF-3-01.md

5.4. Grupo 4: Pago de la comanda

Requisitos funcionales

RF-4-01: El sistema debe permitir el pago de la comanda desde la aplicación.

- Entrada: Datos de pago ingresados por el cliente.
- Proceso: La aplicación procesa el pago a través de la plataforma correspondiente.
- Salida: Confirmación de pago exitoso o fallo en la transacción.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE4/REQUISITOS_FUNCIONALES/RF-4-01.md

RF-4-02: Debe ofrecer métodos de pago mediante tarjeta bancaria y PayPal.

- Entrada: Selección del método de pago por parte del cliente.
- Proceso: El sistema redirige a la plataforma de pago correspondiente.
- Salida: Opción de pagar con PayPal y opción de pago con tarjeta bancaria. La connexió amb la plataforma escollida queda establerta.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE4/REQUISITOS_FUNCIONALES/RF-4-02.md

RF-4-03: El sistema debe realizar el ingreso del dinero al cocinero y al repartidor según los datos bancarios asociados a su perfil.

- Entrada: Datos bancarios del cocinero y del repartidor.
- Proceso: El sistema divide el pago según la proporción establecida y transfiere el dinero a las cuentas correspondientes.
- Salida: Ingreso de la parte proporcional a las cuentas del cocinero y repartidor.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE4/REQUISITOS_FUNCIONALES/RF-4-03.md

Requisitos no funcionales

- Requisitos sobre las interfaces externas.

RNF-4-01: El sistema debe conectarse con la plataforma de pago correspondiente para autenticar la transacción.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE4/REQUISITOS_NO_FUNCIONALES/RNF-4-01.md

5.5. Grupo 5: Valoración de la comanda

Requisitos funcionales

RF-5-01: El cliente debe poder valorar la comanda después de recibirla.

- Entrada: Evaluación ingresada por el cliente.
- Proceso: El sistema registra la valoración una vez entregado el pedido.
- Salida: La valoración queda almacenada y asociada a la comanda.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE5/REQUISITOS_FUNCIONALES/RF-5-01.md

RF-5-02: La valoración incluirá criterios como puntualidad y estado del plato.

- Entrada: Puntuación y comentarios sobre puntualidad y calidad del plato.
- Proceso: El sistema almacena la valoración junto con los criterios especificados.
- Salida: La valoración detallada queda registrada en la base de datos.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE5/REQUISITOS_FUNCIONALES/RF-5-02.md

RF-5-03: El sistema asociará la valoración con el cocinero que ha preparado el plato.

- Entrada: Valoración ingresada por el cliente.
- Proceso: El sistema identifica al cocinero que preparó el plato y vincula la valoración a su perfil.

- Salida: La valoración queda registrada y asociada al cocinero correspondiente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE5/REQUISITOS_FUNCIONALES/RF-5-03.md

RF-5-04: Las valoraciones influirán en el posicionamiento de los cocineros en el listado de platos disponibles.

- Entrada: Conjunto de valoraciones recibidas por el cocinero.
- Proceso: El sistema calcula las valoraciones y ajusta la posición del cocinero en el listado.
- Salida: El cocinero tendrá acceso a sus valoraciones acumuladas.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE5/REQUISITOS_FUNCIONALES/RF-5-04.md

RF-5-05: Los clientes deben poder ver las valoraciones de otros usuarios antes de hacer un pedido.

- Entrada: Conjunto de valoraciones recibidas al cocinero.
- Proceso: El sistema ajusta la posición del cocinero en el listado en función de sus valoraciones.
- Salida: El cocinero aparece en una posición acorde a sus valoraciones en la lista de platos.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE5/REQUISITOS_FUNCIONALES/RF-5-05.md

5.6. Grupo 6: Funcionalidades adicionales

Requisitos funcionales:

RF-6-01: El usuario debe poder proponer platos a consumir aparte de los mostrados en la lista de platos.

- Entrada: Platos propuestos por el cliente.
- Proceso: La aplicación propone estos platos al cocinero y este los acepta o no.
- Salida: El cocinero envía su respuesta al cliente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE6/REQUISITOS_FUNCIONALES/RF-6-01.md

RF-6-02: Estos platos propuestos por el usuario deben aparecer en un menú distinto, que permitirá la comunicación usuario-cocinero a través de un chat.

- Entrada: Platos propuestos por el cliente.
- Proceso: Se enseñará los platos al cocinero en un menú diferente y se permitirá la comunicación cliente-cocinero.
- Salida: Un chat usuario-cocinero y un menú diferente.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE6/REQUISITOS_FUNCIONALES/RF-6-02.md

RF-6-03: La aplicación debe poder tener la opción de pagar una mensualidad que ofrezca una tarifa plana de envíos, como también un sistema de puntos, los cuales se consiguen pagando comandas dentro de la aplicación, y gracias a estos poder obtener descuentos en futuras comandas.

- Entrada: Paga mensualidad y sistema de puntos.
- Proceso: La mensualidad ofrecerá una tarifa plana de envíos y un sistema de puntos que se ganaran al hacer comandas y se podrá canjear estos.
- Salida: Una tarifa plana de envíos y descuentos gracias a los puntos.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE6/REQUISITOS_FUNCIONALES/RF-6-03.md

5.7. Otras consideraciones

Requisitos no funcionales:

Decisión de diseño:

RNF-7-01: El software debe estar disponible tanto en catalán, castellano e inglés.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-01.md

RNF-7-02:El software debe usar el protocolo OpenVPN para encriptación de datos y pagos.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-02.md

RNF-7-03:El software debe poder ser accedida tanto por web como desde una aplicación móvil.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-03.md

RNF-7-04:El color principal de la aplicación debe ser el naranja.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-04.md

RNF-7-05:El logo debe poder ser visible en todas las pantallas.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-05.md

RNF-7-06:El software debe usar la tipografía estilo Calibri.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-06.md

- Restricción de diseño:

RNF-7-07:El software debe poder funcionar tanto en Android como en IOS.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-07.md

- Requisitos de rendimiento:

RNF-7-08:Las peticiones hechas por el usuario deben ser respondidas en un tiempo inferior a 5ms.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-08.md

RNF-7-09:El software debe poder soportar hasta 1000 usuarios simultáneos.

Enlace:

https://github.com/dabien3/ES25UAB-511-05/blob/main/requirements/BLOQUE7/REQUISITOS_NO_FUNCIONALES/RNF-7-09.md

6. DIAGRAMAS CASOS DE USO

Link del repositorio GitHub y general de los diagramas de casos de uso individuales

Enlace a github carpeta general Diagramas casos de uso:

<https://github.com/dabien3/ES25UAB-511-05/tree/1cec8573aad3d526bd52bd6c73423c97845f7ea7/documents/Diagrames%20Casos%20d'Us>

6.1 Enlaces del repositorio Github de cada bloque:

6.1.1. Diagrama caso de uso bloque 1:

Diagrama:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Diagrames/Diagrama%20Caso%20de%20uso%20B1.png>

Codigo Plantuml:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Code%20Plantuml/code_bloq1.pUML

6.1.2. Diagrama caso de uso bloque 2:

Diagrama :

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Diagrames/Diagrama%20Caso%20de%20uso%20B2.png>

Codigo Plantuml:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Code%20Plantuml/code_bloq2.pUML

6.1.3. Diagrama caso de uso bloque 3:

Diagrama :

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Diagrames/Diagrama%20Caso%20de%20uso%20B3.png>

Codigo Plantuml:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Code%20Plantuml/code_bloq3.pUML

6.1.4. Diagrama caso de uso bloque 4:

Diagrama :

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Diagrames/Diagrama%20Caso%20de%20uso%20B4.png>

Codigo Plantuml:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Code%20Plantuml/code_bloq4.pUML

6.1.5 Diagrama caso de uso bloque 5:

Diagrama :

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Diagrames/Diagrama%20Caso%20de%20uso%20B5.png>

Codigo Plantuml:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Code%20Plantuml/code_bloq5.pUML

6.1.6. Diagrama caso de uso bloque 6:

Diagrama :

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Diagrames/Diagrama%20Caso%20de%20uso%20B6.png>

Codigo Plantuml:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Code%20Plantuml/code_bloq6.pUML

6.1.7. Diagrama caso de uso bloque 7:

Al tratarse de un bloque de requisitos NO FUNCIONALES no requiere de diagramas.

6.1.8. Diagrama caso de uso FINAL:

Diagrama :

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Diagrames/diagrama%20FINAL.png>

Codigo Plantuml:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrames%20Casos%20d'Us/Code%20Plantuml/code_FINAL.pUML

7. DIAGRAMAS DE CLASES

Link del repositorio GitHub y general de los diagramas de clases individuales

<https://github.com/dabien3/ES25UAB-511-05/tree/main/documents/Diagrama%20Clases>

7.1. Enlaces del repositorio Github de cada bloque:

7.1.1. Diagrama de clases bloque 1:

Diagrama :

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/Diagramas/bloq1_diagrama_clase.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/CodeUML/bloq1.pUML>

7.1.2. Diagrama de clases bloque 2:

Diagrama :

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/Diagramas/bloq2_diagrama_clase.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/CodeUML/bloq2.pUML>

7.1.3. Diagrama de clases bloque 3:

Diagrama :

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/Diagramas/bloq3_diagrama_clase.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/CodeUML/bloq3.pUML>

7.1.4. Diagrama de clases bloque 4:

Diagrama :

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/Diagramas/bloq4_diagrama_clase.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/CodeUML/bloq4.pUML>

7.1.5. Diagrama de clases bloque 5:

Diagrama :

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/Diagramas/bloq5_diagrama_clase.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/CodeUML/bloq5.pUML>

7.1.6. Diagrama de clases bloque 6:

Diagrama :

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/Diagramas/bloq6_diagrama_clase.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagrama%20Clases/CodeUML/bloq6.pUML>

7.2. Diagrama de clases final

Diagrama :

Codigo Plantuml:

8. DIAGRAMAS DE SECUENCIA

Link del repositorio GitHub y general de los diagramas de secuencia.

<https://github.com/dabien3/ES25UAB-511-05/tree/main/documents/Diagramas%20Secuencia>

8.1 Enlaces del repositorio Github de cada bloque:

8.1.1. Diagrama de secuencia bloque 1:

Diagrama:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/Diagramas/bloq1_diagrama_secuencia.png

Código Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/CodigoUML/bloq1.pUML>

Plantilla del diagrama de caso de uso del Bloque 1:

<https://docs.google.com/spreadsheets/d/1FcZ3hVIpd60LNm0S3755OLOgf-qIkWcOhUjSgZa-ljA/edit?gid=0#gid=0>

CASOS de USO				Bloque 1: Gestión de cuentas y creación de perfiles de usuario.			
Num Versión	1.0	Data					09/04/2025
Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Davalos, Nour Kobbi Laghmich.						
Descripción	En este caso de uso vemos como se relaciona los actores con relacion a la gestión de cuentas y creación de perfiles de usuario.						
Actores	Cliente, Cocinero, Repartidores, Usuario no Registrado.						
Precondición	N/A						
Flujo principal	1. El usuario no registrado podrá registrarse usando un correo. 2. Los usuarios registrados podran actualizar su perfil en cualquier momento. 3. Todos los usuarios registrados seran capaces de poder eliminar su perfil en cualquier momento.						
Subflujos	1. Tanto el nombre de usuario como contraseñas deberan tener su formato correcto, si no saltará un error.						
Flujos alternativos	N/A						
Postcondición	N/A						
Requisitos no Funcionales	RNF-1-01:El nombre de usuario tendrá 5 caracteres como mínimo y deberá de ser único. RNF-1-02:La contraseña de la cuenta tendrá como mínimo 8 caracteres, de los cuales mínimo 2 serán mayúsculas y 2 caracteres especiales.						
Prioridad	Normal.						
Comentarios	N/A						

8.1.2. Diagrama de secuencia bloque 2:

Diagrama:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/Diagramas/bloq2_diagrama_secuencia.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/CodigoUML/bloq2.pUML>

Plantilla del diagrama de caso de uso del Bloque 2:

<https://docs.google.com/spreadsheets/d/1FcZ3hVIpd60LNm0S3755OLOgf-qIkWcOhUjSgZa-ljA/edit?gid=0#gid=0>

CASOS de USO Bloque 2: Realización de comandas			
Num Versión	1.0	Data	09/04/2025
Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Davalos, Nour Kobbi.		
Descripción	En este caso de uso se representa cómo los clientes realizan comandas, desde la visualización de platos hasta la confirmación del pedido.		
Actores	Cliente		
Precondición	Cliente debe estar registrado para realizar pedidos.		
Flujo principal	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.		
Subflujos	NA		
Flujos alternativos	N/A		
Postcondición	El pedido queda confirmado o reservado.		
Requisitos no Funcionales	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.		
Prioridad	Normal		
Comentarios	Este módulo depende de la correcta configuración del perfil del cliente y de la disponibilidad de los cocineros.		

8.1.3. Diagrama de secuencia bloque 3:

Diagrama:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/Diagramas/bloq3_diagrama_secuencia.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/CodigoUML/bloq3.pUML>

Plantilla del diagrama de caso de uso del Bloque 3:

<https://docs.google.com/spreadsheets/d/1FcZ3hVIpd60LNm0S3755OLOgf-qIkWcOhUjSgZa-ljA/edit?gid=0#gid=0>

CASOS de USO Bloque 3: Envío de comandas			
Num Versión	1.0	Data	09/04/2025
Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Davalos, Nour Kobbi.		
Descripción	En este caso de uso se muestra cómo el cliente interactúa con las distintas opciones de envío del pedido.		
Actores	Cliente y Cocinero.		
Precondición	El cliente debe haber confirmado un pedido, es decir que va comer.		
Flujo principal	<ol style="list-style-type: none"> 1. El cliente accede a la aplicación y selecciona el tipo de entrega del pedido. 2. El sistema muestra las opciones disponibles: <ol style="list-style-type: none"> a. Repartidor externo b. Entrega por parte del cocinero c. Recogida en casa del cocinero 3. El cliente selecciona una opción. 4. El sistema calcula el costo del envío (si aplica) y lo muestra al cliente. 5. El cliente confirma la opción de entrega. 6. El sistema registra la elección y actualiza el pedido. 		
Subflujos	<p><u>Entrega por repartidor:</u></p> <ol style="list-style-type: none"> 7. El sistema inicia el proceso de envío y asigna un repartidor. 8. Se calcula el costo de envío en base a la distancia entre el cocinero y el cliente. 9. El sistema proporciona un enlace de seguimiento generado a través de Google Maps. 10. El cliente puede consultar el estado del envío en tiempo real. <p><u>Entrega por cocinero:</u></p> <ol style="list-style-type: none"> 7. El cocinero accede a la aplicación y acepta realizar la entrega personalmente. 8. El sistema asigna el pedido al cocinero como repartidor. 9. Se calcula el costo del envío en función de la distancia, si aplica. 10. El cliente puede consultar el estado del pedido y su ubicación si el cocinero ha activado el seguimiento. <p><u>Recogida en casa del cocinero:</u></p> <ol style="list-style-type: none"> 7. El cliente elige recoger el pedido en casa del cocinero. 8. El sistema marca el envío como gratuito. 9. Se informa al cocinero de la elección y se notifica cuándo el cliente puede venir a recoger el pedido. 10. No se genera ningún enlace de seguimiento. 		
Flujos alternativos	N/A		

Postcondición	El pedido es entregado o recogido por el cliente según la opción seleccionada.
Requisitos no Funcionales	RNF-3-01: Integración con Google Maps para visualizar la posición del repartidor en tiempo real.
Prioridad	Normal
Comentarios	N/A

8.1.4. Diagrama de secuencia bloque 4:

Diagrama:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/Diagramas/bloq4_diaframes_sequencia.png

Código Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/CodigoUML/bloq4.pUML>

Plantilla del diagrama de caso de uso del Bloque 4:

<https://docs.google.com/spreadsheets/d/1FcZ3hVIpd60LNm0S3755OLOgf-qIkWcOhUjSgZa-ljA/edit?gid=0#gid=0>

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 Davalos, 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		
Precondición	El cliente ha finalizado una comanda y está listo para pagar.		
Flujo principal	1. El cliente accede al apartado de pago desde la aplicación. 2. Selecciona un método de pago: Tarjeta o PayPal. 3. El sistema redirige a la plataforma correspondiente. 4. El cliente introduce sus datos y se validan. 6. El sistema distribuye automáticamente los importes al cocinero y al repartidor. 7. Se informa al cliente de que el pago ha sido completado.		
Subflujos	Si los datos bancarios introducidos son incorrectos: El sistema muestra un mensaje de error al cliente . El cliente debe volver a ingresar los datos.El sistema repite la validación hasta que los datos sean correctos.		
Flujos alternativos	N/A		
Postcondición	El pago de comanda queda realizado correctamente y su estado cambio		

	de la comanda a: PAGADO
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

8.1.5. Diagrama de secuencia bloque 5:

Diagrama:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/Diagramas/bloq5_diagrama_secuencia.png

Código Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/CodigoUML/bloq5.pUML>

Plantilla del diagrama de caso de uso del Bloque 5:

<https://docs.google.com/spreadsheets/d/1FcZ3hVIpd60LNm0S3755OLOgf-qIkWcOhUjSgZa-ljA/edit?gid=0#gid=0>

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 Davalos, 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	N/A		

Requisitos no Funcionales	N/A
Prioridad	Normal.
Comentarios	N/A

8.1.6. Diagrama de secuencia bloque 6:

Diagrama:

https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/Diagramas/bloq6_diagrama_secuencia.png

Codigo Plantuml:

<https://github.com/dabien3/ES25UAB-511-05/blob/main/documents/Diagramas%20Secuencia/CodigoUML/bloq6.pUML>

Plantilla del diagrama de caso de uso del Bloque 6:

<https://docs.google.com/spreadsheets/d/1FcZ3hVIpd60LNm0S3755OLOgf-qIkWcOhUjSgZa-ljA/edit?gid=0#gid=0>

CASOS de USO	Bloque 6: Funciones adicionales			
Num. Versión	1.0		Data	09/04/2025
Autores	Yang Chen, Liying Sun, Daniel Alberto Vaca Davalos, 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	Flujo principal del caso de uso, es decir, decir como irian apareciendo las acciones			
Subflujos	Alternativas al flujo principal			
Flujos alternativos	Variaciones en los flujos principales o casos de excepción			
Postcondición	Postcondición del caso de uso			
Requisitos no Funcionales	Lista de los RNF que tienen que ver con los funcionales			
Prioridad	varia entre urgente, normal o no prioritario, normalmente seran normales			
Comentarios	Comentarios adicionales			

9. INFORMACIÓN SPRINTS

9.1. Fechas, Masters y presentaciones correspondientes

9.1.1. Sprint 1

Fecha inicio:

Fecha fin:

Master: Daniel Alberto Vaca

Enlace: https://github.com/dabien3/ES25UAB-511-05/blob/main/others/Presentaciones%20Sprint/sprint_1.pdf

9.1.2. Sprint 2

Fecha inicio: 24/03/2025

Fecha fin: 13/04/2025

Master: Nour Kobbi Laghmich

Enlace: https://github.com/dabien3/ES25UAB-511-05/blob/main/others/Presentaciones%20Sprint/sprint_2.pdf

9.1.3. Sprint 3

Fecha inicio: 14/04/2025

Fecha fin: 11/05/2025

Master: Liying Sun

Enlace: https://github.com/dabien3/ES25UAB-511-05/blob/main/others/Presentaciones%20Sprint/sprint_3.pdf

9.1.4. Sprint 4

Fecha inicio: 14/05/2025

Fecha fin: 01/06/2025

Master: Yang Chen

Enlace: https://github.com/dabien3/ES25UAB-511-05/blob/main/others/Presentaciones%20Sprint/sprint_4.pdf