Documento de MODELADO DE REQUISITOS

Sagres

Sistema de Administración y Gestión de RESTaurantes

Samuel Guirado Navarro
Carlos Moreno Muñoz
Gaspar Muñoz Soria

V1.3

ÍNDICE DE CONTENIDO

Apartado de control de versiones

Todas las versiones están especificadas a fondo en el apartado de "Apéndices", al final de este documento, cada apéndice se corresponde en nombre con su número de versión. Por ejemplo, el "Apéndice 0.1" se corresponde con la versión v0.1. Para ver los cambios realizados sobre cada versión, hay que ir deshaciendo los cambios desde el final.

Versión	Fecha	Descripción
V1.0	15/03/10	Documento inicial
V1.1	16/03/10	Revisión de la especificación de requisitos, diagramas de secuencia del sistema y operaciones del sistema.
V1.2	21/03/10	Revisión del diagrama de casos de uso, especificación de los casos de uso y diagramas de secuencia
V1.3	15/04/10	Revisión de las especificaciones de los casos de uso y de los diagramas de secuencia del sistema.

1. Modelo funcional

Identificación de actores

Cocinero Jefe

Es el encargado de gestionar la carta del restaurante. Entre sus tareas están las de añadir, modificar y eliminar elementos de la carta. También es el encargado de gestionar el stock de ingredientes del restaurante. Otra de sus responsabilidades será la de notificar incidencias que ocurran con ingredientes que utilice en la elaboración de los platos del restaurante. También deberá informar de cuando llega un pedido del proveedor.

Metre

Es el encargado de gestionar todo lo referente al stock de bebidas del restaurante. Al igual que el cocinero jefe con los ingredientes, el metre deberá notificar cualquier incidencia relacionada con alguna bebida.

Identificación de los requerimientos funcionales

Estos son los requerimientos funciones obtenidos tras la reunión con el cliente asociados a los subsistemas estudiados en esta primera iteración:

RF1. El sistema contempla a tres tipos de usuario: cliente, camarero jefe y metre.

RF2. El cocinero jefe podrá añadir nuevos elementos a la carta aportando el nombre del nuevo elemento, la descripción y el precio de éste, así como los ingredientes y el gasto de elaboración asociados y una foto del mismo finalizado. Cada elemento deberá tener asignada una y solo una sección de la carta, las cuales serán: entradas, carnes, pescados, bebidas y postres. Además, en el caso de los elementos que no sean bebidas, el cocinero deberá especificar si estos se pueden partir en raciones o no, indicando el número de raciones en caso positivo.

RF3. El cocinero jefe ha de poder realizar pequeñas modificaciones en los datos (salvo en los ingredientes) de cualquier elemento de la carta, siempre y cuando el restaurante permanezca cerrado.

RF4. El cocinero jefe podrá también eliminar cualquier elemento de la carta. Para evitar la posibilidad de que el cocinero elimine un elemento mientras un cliente lo selecciona, esta operación solo podrá llevarse a cabo cuando el restaurante este cerrado.

RF5. El sistema debe ofrecer la posibilidad de que con solo pulsar un botón, se le pueda notificar que ha llegado el pedido del proveedor, actualizándose las cantidades de los productos pedidos automáticamente.

RF6. El cocinero jefe podrá añadir nuevos ingredientes al stock de productos través del sistema. Los datos que deberá proporcionar de cada ingrediente son el nombre, la cantidad actual de la que se dispondrá, el límite mínimo de cantidad que puede haber en stock, el límite máximo y una foto del mismo.

RF7. El cocinero jefe podrá modificar los datos de cualquier ingrediente.

RF8. El cocinero jefe también podrá eliminar ingredientes del sistema.

RF9. El sistema deberá llevar un control de la pérdidas que se produzcan en el restaurante debido a incidencias con productos y bebidas. Para ello, el sistema ofrecerá la posibilidad de poder notificar una incidencia ya sea con un ingrediente o con una bebida. Se deberá indicar tan solo la cantidad de producto afectado.

RF10. El metre será el encargado de añadir, modificar y eliminar las bebidas del stock de productos.

RF11. Los datos necesarios para añadir una bebida serán los mismos que para un ingrediente. Sin embargo, en este caso también será necesario añadir la cantidad de líquido que cabe en el envase en el que se encuentra dicha bebida.

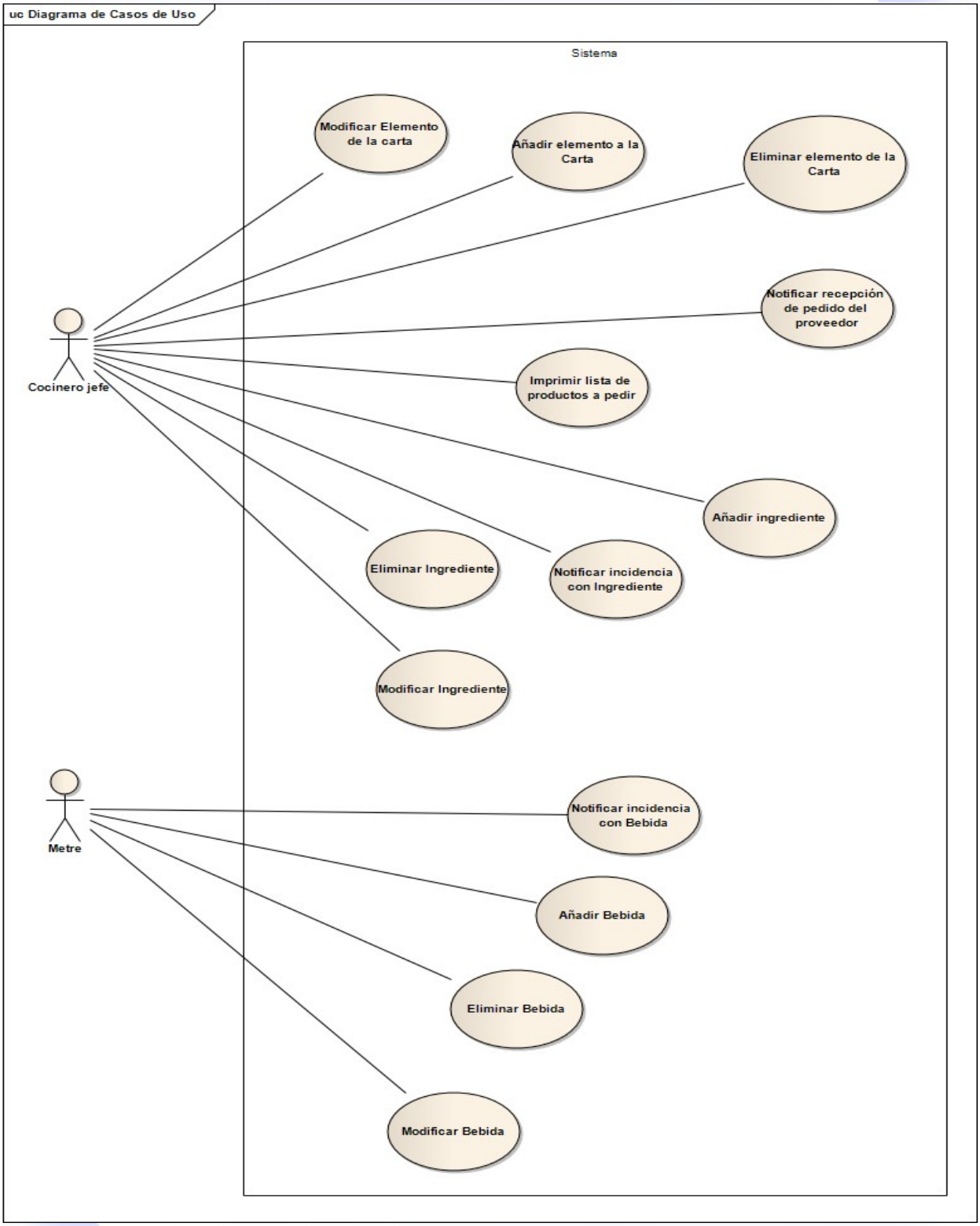
RF12. El sistema deberá elaborar una lista de necesidades con las cantidades requeridas de cada uno de los productos que se deben comprar, a partir de los productos consumidos y del mínimo que debe haber de existencias. Además, el cocinero podrá imprimir esta lista cuando desee para que se pueda realizar el pedido.

RF13. El sistema tendrá la obligación de notificar al usuario cuando se produzca el hecho en el que la cantidad actual en stock de un producto supere su límite mínimo.

RF14. Si un producto llega a cantidad 0 en stock, el sistema invalidará los elementos de la carta que dependan de ese producto. Cuando dicho producto se reponga el sistema deberá rehabilitarlos.

Diagrama de casos de uso

Los casos de uso anterior mente especificados, los podemos representar en el siguiente diagrama global de casos de uso, para concluir la elaboración del modelo funcional de nuestro sistema.



Descripción de los casos de uso

Añadir elemento a la carta

Nombre del caso:	Añadir elemento a la carta
Resumen:	El cocinero jefe desea añadir un a la carta del restaurante. Introduce toda la información correspondiente al nuevo elemento y por último selecciona los ingredientes asociados a este con sus cantidades correspondientes.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha añadido un elemento nuevo a la carta
Curso normal:	1. El caso de uso comienza cuando el usuario desea añadir un elemento a la carta 2. El usuario especifica la sección en la que va a incluir el nuevo elemento. 3. El sistema muestra una lista productos dependiendo de la sección escogida. 4. El usuario rellena los datos correspondientes al elemento: 4.1.1. Introduce el nombre 4.1.2. Introduce la descripción 4.1.3. Introduce la foto 4.1.4. Introduce el precio 4.1.5. Introduce el número de divisiones del elemento 4.1.6. Selecciona la sección 4.1.6.1. Si la sección asignada no es la de bebidas 4.1.6.1.1 El usuario especifica el tiempo de elaboración del plato 4.2. Mientras el usuario quiera asociar productos al elemento: 4.2.1. El usuario selecciona un producto de la lista e introduce la cantidad deseada 4.3. El usuario confirma los datos 4.4. El sistema comprueba que los datos introducidos sean correctos 4.5. Los datos introducidos son correctos. El sistema genera un código asociado al elemento y registra todos los datos 4.6. El sistema confirma que se ha añadido un nuevo elemento a la carta satisfactoriamente
Cursos alternativos:	4.2.1a. El usuario se ha equivocado al seleccionar un producto 4.2.1.1. El usuario elimina el producto deseado 4.2.1b. El usuario se ha equivocado al introducir la cantidad de

	un ingrediente 4.2.1.1. El usuario selecciona el producto de la lista e introduce la nueva cantidad 4.5. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3 *. El usuario cancela la operación
Observaciones:	
Requisitos no funcionales específicos:	- Las secciones asignables a un elemento de la carta serán: **Entrantes**, **Carnes**, **Pescados**, **Bebidas** o **Postres** - Las cantidades de los ingredientes asignados a los platos se establecerán en gramos en caso de ser sólidos y mililitros en caso de ser líquidos **- La foto deberá estar en formato .jpeg y no sobrepasar los 200 kb**

Eliminar elemento a la carta

Nombre del caso:	Eliminar elemento de la carta
Resumen:	El cocinero jefe desea eliminar un elemento de la carta del restaurante. Le basta tan solo con seleccionar el elemento que desea borrar y confirmarlo
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha eliminado un elemento de la carta
Curso normal:	1. El caso de uso comienza cuando el usuario desea eliminar un elemento de la carta 2. El sistema muestra una lista con los elementos de la carta registrados 3. El usuario selecciona el elemento de la carta que desea eliminar 4. El sistema muestra la información asociada al elemento y solicita la confirmación de la operación 5. El usuario confirma la operación 6. El sistema confirma que el elemento ha sido eliminado de la carta satisfactoriamente
Cursos alternativos:	5. El usuario no confirma la operación. Se vuelve al paso 2 *. El usuario cancela la operación
Observaciones:	
Requisitos no funcionales:	

Modificar elemento de la carta

Nombre del caso:	Modificar elemento de la carta
Resumen:	El cocinero jefe desea modificar un elemento de la carta del restaurante. Selecciona el elemento y modifica el o los datos deseados.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se han modificado uno o más datos de un elemento de la carta
Curso normal:	1. El caso de uso comienza cuando el usuario desea modificar un elemento de la carta 2. El sistema muestra una lista con los elementos de la carta registrados 3. El usuario selecciona un elemento de la lista 4. El sistema muestra la información asociada a ese elemento 5. Mientras el usuario desee modificar datos del elemento: 5.1 El usuario modifica el dato deseado 6. El usuario finaliza la modificación de los datos 7. El sistema solicita la confirmación de la modificación de los datos 8. El usuario confirma la operación 9. El sistema comprueba que los datos introducidos son correctos. 10. Los datos introducidos son correctos. El sistema registra los nuevos cambios y confirma que el elemento ha sido modificado con éxito
Cursos alternativos:	10. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 4 *. El usuario cancela la operación
Observaciones:	
Requisitos no funcionales:	

Imprimir lista de productos a pedir

Nombre del caso:	Imprimir lista de productos a pedir
Resumen:	El cocinero jefe desea imprimir una lista de los productos que están bajo mínimos en stock para poder hacer un pedido al proveedor
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha imprimido la lista de productos a pedir • Se ha guardado una copia de la lista en el sistema
Curso normal:	1. El caso de uso comienza cuando el usuario quiere imprimir la lista de productos y cantidades del pedido 2. El sistema obtiene la lista de productos que están bajo mínimos en stock y las cantidades necesarias de cada uno para alcanzar su máximo. 3. El sistema imprime dicha lista 4. El sistema guarda una copia de la lista 5. El sistema muestra la información asociada al pedido
Cursos alternativos:	
Observaciones:	
Requisitos no funcionales específicos:	La lista de productos estará dividida en dos partes: por un lado los ingredientes y por otro lado las bebidas

Notificar recepción de pedido del proveedor

Nombre del caso:	Notificar recepción de pedido del proveedor
Resumen:	El cocinero jefe informa de que ha llegado el pedido realizado al proveedor. Las cantidades de los productos pedidos se actualizan automáticamente
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha modificado la cantidad actual en stock de uno o más productos
Curso normal:	1. El caso de uso comienza cuando el cocinero jefe informa de que ha llegado el pedido del proveedor. 2. El sistema obtiene los datos del pedido y muestra los elementos de la carta inválidados que quedarán habilitados tras la actualización de sus cantidades en caso de que los haya. 3. El sistema actualiza las cantidades de los productos especificados en el pedido 4. El sistema habilita los elementos de la carta correspondientes. 5. El sistema devuelve la información de los productos actualizados y de los elementos habilitados.
Cursos alternativos:	
Observaciones:	El sistema utiliza la copia que posee del pedido para actualizar las cantidades de los productos correspondientes
Requisitos no funcionales específicos:	

Añadir ingrediente

Nombre del caso:	Añadir ingrediente
Resumen:	El cocinero jefe desea añadir un ingrediente al sistema. Introduce los datos correspondientes a dicho ingrediente, se comprueban si son correctos y se confirma la operación.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha registrado un ingrediente en el sistema
Curso normal:	1. El caso de uso comienza cuando el cocinero jefe desea añadir un ingrediente. 2. El usuario introduce los datos correspondientes al nuevo ingrediente: 2.1.1 Introduce el nombre 2.1.2 Introduce el límite mínimo en stock. 2.1.3 Introduce el máximo en stock. 2.1.4 Introduce la cantidad actual en stock 2.1.5 Introduce una foto. 3. El usuario confirma los datos. 4. El sistema comprueba si los datos introducidos son correctos 5. Los datos introducidos son correctos. El sistema genera un código de producto, registra todos los datos y confirma que ha sido añadido satisfactoriamente.
Cursos alternativos:	5. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 2 *. El usuario cancela la operación.
Observaciones:	
Requisitos no funcionales específicos:	- Tanto los límites como la cantidad actual se deberán dar en gramos. - La foto deberá estar en formato .jpeg y no sobrepasar los 200 kb

Eliminar ingrediente

Nombre del caso:	Eliminar ingrediente
Resumen:	El cocinero jefe desea eliminar un ingrediente de nuestro sistema. Para ello lo selecciona de una lista de ingredientes y confirma su eliminación. Una vez hecho esto el sistema lo elimina.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha eliminado un ingrediente del sistema.
Curso normal:	1. El caso de uso comienza cuando el cocinero jefe desea eliminar un ingrediente. 2. El usuario selecciona el ingrediente que quiere eliminar. 3. El sistema obtiene información detallada del ingrediente que deseamos eliminar y la muestra al usuario. Además muestra una lista con los elementos de la carta que se verán afectados tras la eliminación del ingrediente. 4. El usuario confirma la eliminación. 5. El sistema elimina el ingrediente e invalida los elementos de la carta que cuentan con ese ingrediente. Se informa de ello al usuario.
Cursos alternativos:	4. El usuario no confirma la operación. Se vuelve al paso 2. *. El usuario cancela la operación.
Observaciones:	
Requisitos no funcionales específicos:	

Modificar ingrediente

Nombre del caso:	Modificar ingrediente
Resumen:	El cocinero jefe quiere modificar los datos de un determinado ingrediente. Una vez que se le muestran los datos, cambia los parámetros que quiera y una vez confirmados y comprobados estos se registran en el sistema.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha registrado uno o más cambios en los detalles de un ingrediente.
Curso normal:	1. El caso de uso comienza cuando el cocinero jefe desea modificar un ingrediente. 2. El usuario selecciona el ingrediente que quiere modificar. 3. El sistema muestra los datos asociados a ese ingrediente. 4. Mientras el usuario quiera modificar parámetros: 4.1. El usuario modifica el parámetro deseado. 5. El usuario confirma los cambios. 6. El sistema comprueba si los datos introducidos son correctos 7. Los datos introducidos son correctos. El sistema registra los cambios e informa que se han realizado satisfactoriamente.
Cursos alternativos:	5. El usuario no confirma los cambios. Se vuelve al paso 2 7. El sistema informa de que los datos introducidos son incorrectos. Se vuelve al paso 2. *. El usuario cancela la operación.
Observaciones:	
Requisitos no funcionales específicos:	

Notificar incidencia con Ingrediente

Nombre del caso:	Notificar incidencia con Ingrediente
Resumen:	El cocinero jefe desea notificar una incidencia ocurrida con un ingrediente. Para ello selecciona el ingrediente en cuestión e introduce la cantidad de este que se ha visto afectada. Por último especifica el tipo de incidencia.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha registrado una incidencia con un ingrediente
Curso normal:	1. El caso de uso comienza cuando el usuario desea notificar una incidencia ocurrida con un ingrediente 2. El usuario selecciona el ingrediente sobre el que desea notificar la incidencia 3. El usuario introduce la cantidad de producto afectado y especifica el motivo de la incidencia 4. El sistema comprueba la cantidad que queda del ingrediente en stock 4.1. Si la cantidad actual en stock del ingrediente afectado se encuentra por debajo de su mínimo 4.1.1 El sistema informa de que la cantidad de ese producto en stock se encuentra por debajo de su mínimo 5. El sistema comprueba que los datos introducidos son correctos. 6. Los datos introducidos son correctos. El sistema genera un código asociado a la incidencia y registra la incidencia 7. El sistema resta del stock la cantidad del ingrediente especificado por el usuario e invalida los elementos con el ingrediente asociado en caso de que sea necesario. 8. El sistema informa de que la incidencia se ha registrado correctamente
Cursos alternativos:	6. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 2 4.1. Si el ingrediente se ha agotado completamente 4.1.1. El sistema busca los elementos de la carta que usen ese ingrediente y los invalida. Se informa de ello al usuario *. El usuario cancela la operación
Observaciones:	
Requisitos no funcionales específicos:	

Añadir bebida

Nombre del caso:	Añadir bebida
Resumen:	El metre desea añadir una bebida al sistema. Introduce los datos correspondientes a la bebida, se comprueban si son correctos y se confirma la operación.
Dependencias:	
Actores:	Metre
Precondiciones:	
Postcondiciones:	• Se ha registrado una nueva bebida en el sistema
Curso normal:	1. El caso de uso comienza cuando el cocinero jefe desea añadir una bebida. 2. El usuario introduce los datos correspondientes a la bebida: 2.1. Introduce el nombre 2.2. Introduce el límite mínimo en stock. 2.3. Introduce el máximo en stock. 2.4. Introduce la cantidad actual en stock 2.5. Introduce una foto. 3. El usuario confirma los datos. 4. El sistema comprueba si los datos introducidos son correctos 5. Los datos introducidos son correctos. El sistema genera un código de producto, registra todos los datos y confirma que se ha añadido la bebida satisfactoriamente.
Cursos alternativos:	5. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3.1 *. El usuario cancela la operación.
Observaciones:	
Requisitos no funcionales específicos:	- Los límites mínimo y máximo vendrán en número de unidades de envase - La foto deberá estar en formato .jpeg y no sobrepasar los 200 kb

Eliminar bebida

Nombre del caso:	Eliminar bebida
Resumen:	El metre desea eliminar una bebida de nuestro sistema. Para ello la selecciona de una lista de bebidas, y confirma su eliminación. Una vez hecho esto el sistema la elimina.
Dependencias:	
Actores:	Metre.
Precondiciones:	
Postcondiciones:	• Se ha eliminado una bebida del sistema.
Curso normal:	1. El caso de uso comienza cuando el metre desea eliminar una bebida. 2. El usuario selecciona la bebida que quiere eliminar. 3. El sistema obtiene información detallada de la bebida que deseamos eliminar y la muestra al usuario. Además muestra una lista con los elementos de la carta que se verán afectados tras la eliminación de la bebida. 4. El usuario confirma la eliminación. 5. El sistema invalida los elementos de la carta afectados. 6. El sistema elimina la bebida e informa de ello al usuario.
Cursos alternativos:	4. El usuario no confirma la operación. Se vuelve al paso 2. *. El usuario cancela la operación.
Observaciones:	
Requisitos no funcionales específicos:	

Modificar bebida

Nombre del caso:	Modificar bebida
Resumen:	El metre quiere modificar los datos de una determinada bebida. Selecciona la bebida que desea modificar, cambia los parámetros que quiera y una vez confirmados y comprobados estos se registran en el sistema.
Dependencias:	
Actores:	Metre
Precondiciones:	
Postcondiciones:	• Se ha registrado uno o más cambios en los detalles de una bebida.
Curso normal:	1. El caso de uso comienza cuando el metre desea modificar una bebida. 2. El usuario selecciona la bebida que quiere modificar. 3. El sistema muestra los datos asociados a dicha bebida. 4. Mientras el usuario quiera modificar parámetros: 4.1. El usuario modifica el parámetro deseado. 5. El usuario confirma los cambios. 6. El sistema comprueba si los datos introducidos son correctos 7. Los datos introducidos son correctos. El sistema registra los cambios e informa que se han realizado satisfactoriamente.
Cursos alternativos:	5. El usuario no confirma los cambios. Se vuelve al paso 2 6. El sistema informa de que los datos introducidos son incorrectos. Se vuelve al paso 2. *. El usuario cancela la operación.
Observaciones:	

Notificar incidencia con bebida

Nombre del caso:	Notificar incidencia con Bebida
Resumen:	El cocinero jefe desea notificar una incidencia ocurrida con una bebida. Para ello selecciona la bebida en cuestión e introduce la cantidad de esta que se ha visto afectada. Por último especifica el tipo de incidencia.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	• Se ha registrado una incidencia con una bebida
Curso normal:	1. El caso de uso comienza cuando el usuario desea notificar una incidencia ocurrida con una bebida 2. El usuario selecciona la bebida sobre el que desea notificar la incidencia 3. El usuario introduce la cantidad de producto afectado y especifica el motivo de la incidencia 4. El sistema comprueba la cantidad que queda de la bebida en stock 4.1. Si la cantidad actual en stock de la bebida afectada se encuentra por debajo de su mínimo 4.1.1 El sistema informa de que la cantidad de ese producto en stock se encuentra por debajo de su mínimo 5. El sistema comprueba que los datos introducidos son correctos. 6. Los datos introducidos son correctos. El sistema genera un código asociado a la incidencia y registra la incidencia 7. El sistema resta del stock la cantidad de bebida especificada por el usuario e invalida los elementos con la bebida asociado en caso de que sea necesario. 8. El sistema informa de que la incidencia se ha registrado correctamente
Cursos alternativos:	6. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 2 4.1. Si la bebida se ha agotado completamente 4.1.1. El sistema busca los elementos de la carta que usan esta bebida y los invalida. Se informa de ello al usuario *. El usuario cancela la operación
Observaciones:	
Requisitos no funcionales:	

Identificación de los subsistemas funcionales

Para gestionar y optimizar todas las funcionalidades del sistema expuestas en el apartado de "Requerimientos Funcionales", se han divido éstas en dos subsistemas.

Si analizamos el diagrama de casos de usos global, podemos ver estas dos grandes agrupaciones o subsistemas con claridad. La primera de ellas se corresponde a los casos de uso referentes a la carta de nuestro sistema, y la segunda concierne a la gestión de los productos que se usarán en el día a día del restaurante.

Este último, a su vez, podemos fraccionarlo en dos nuevos subsistemas hijos, a los que llamaremos "subsistema de gestión de bebidas" y "subsistema de gestión de ingredientes". El primero alude al trabajo propio del metre, que será el encargado de la administración de las bebidas del stock. El último atañe a la tarea del jefe de cocina, que además de gestionar la carta también deberá encargarse de la administración de los ingredientes.

Con esta división de subsistemas y el especifico reparto de tareas entre cocinero jefe y metre nos ahorramos tener que implementar un sistema autentificación, ya que debido al escaso número de tipos de usuarios (tan solo 3) sería una pérdida de tiempo.

Diagrama de paquetes para subsistemas funcionales

Una vez identificados nuestros subsistemas, los plasmaremos gráficamente en los diagramas de paquetes.

Diagrama de casos de uso global con división en subsistemas

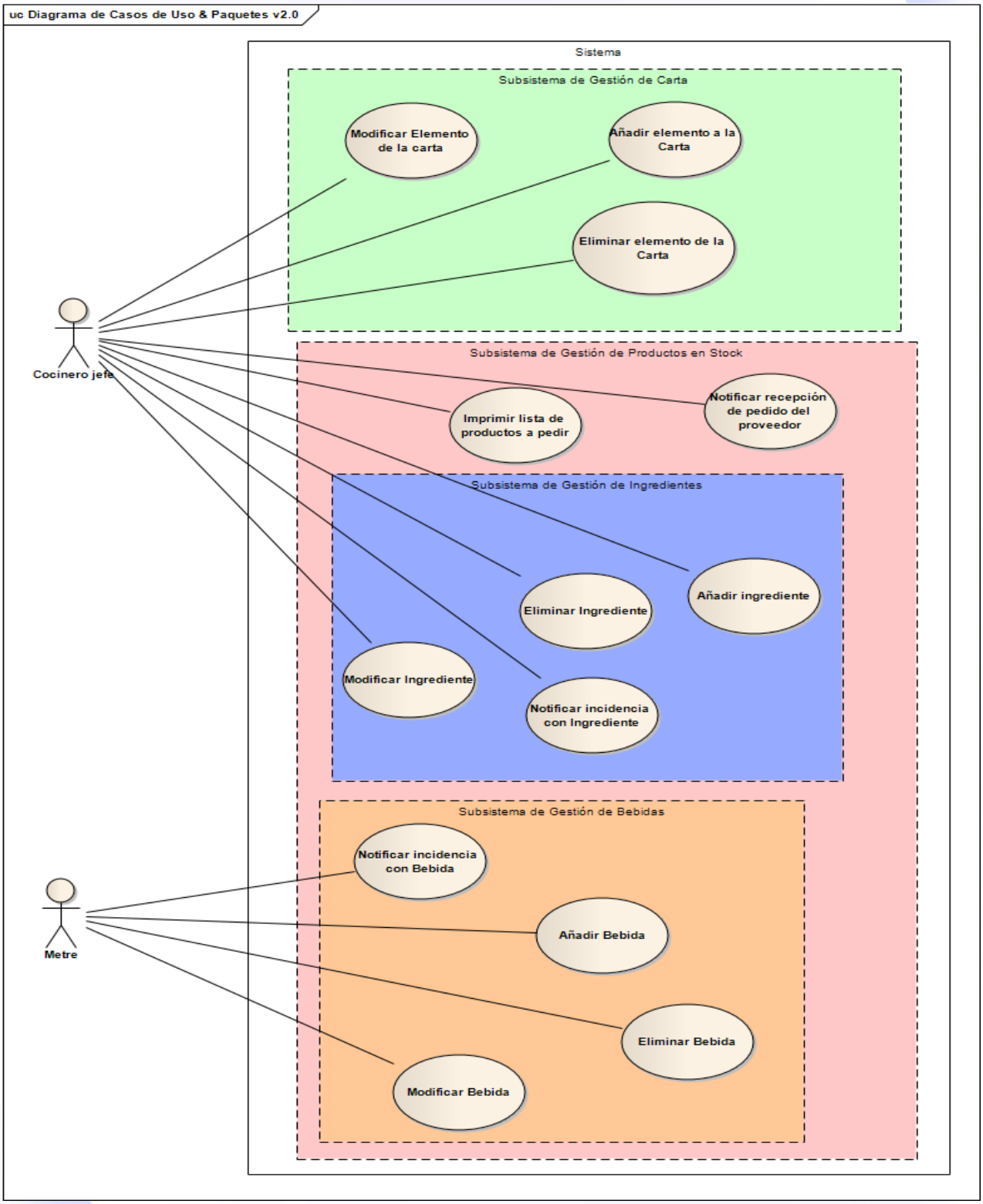
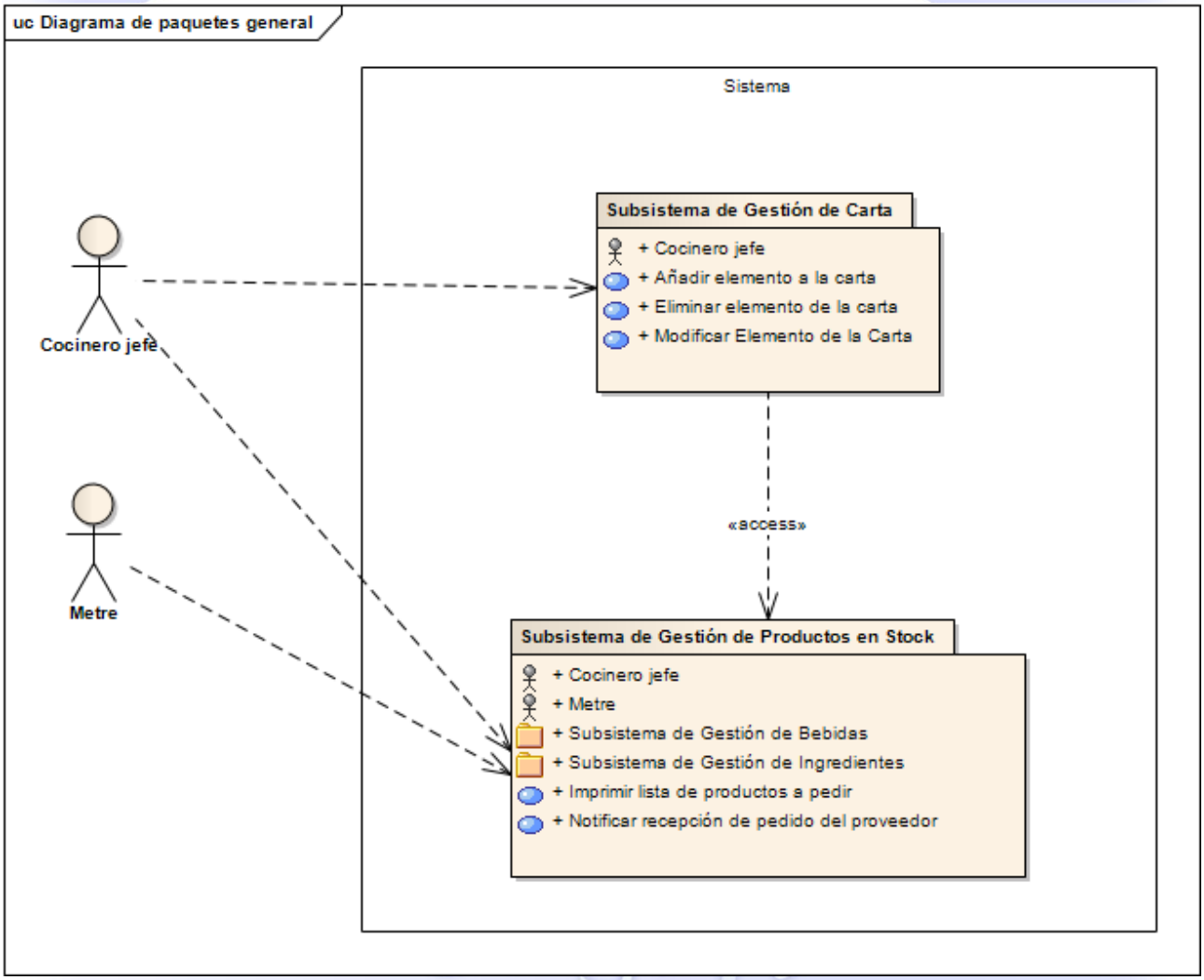
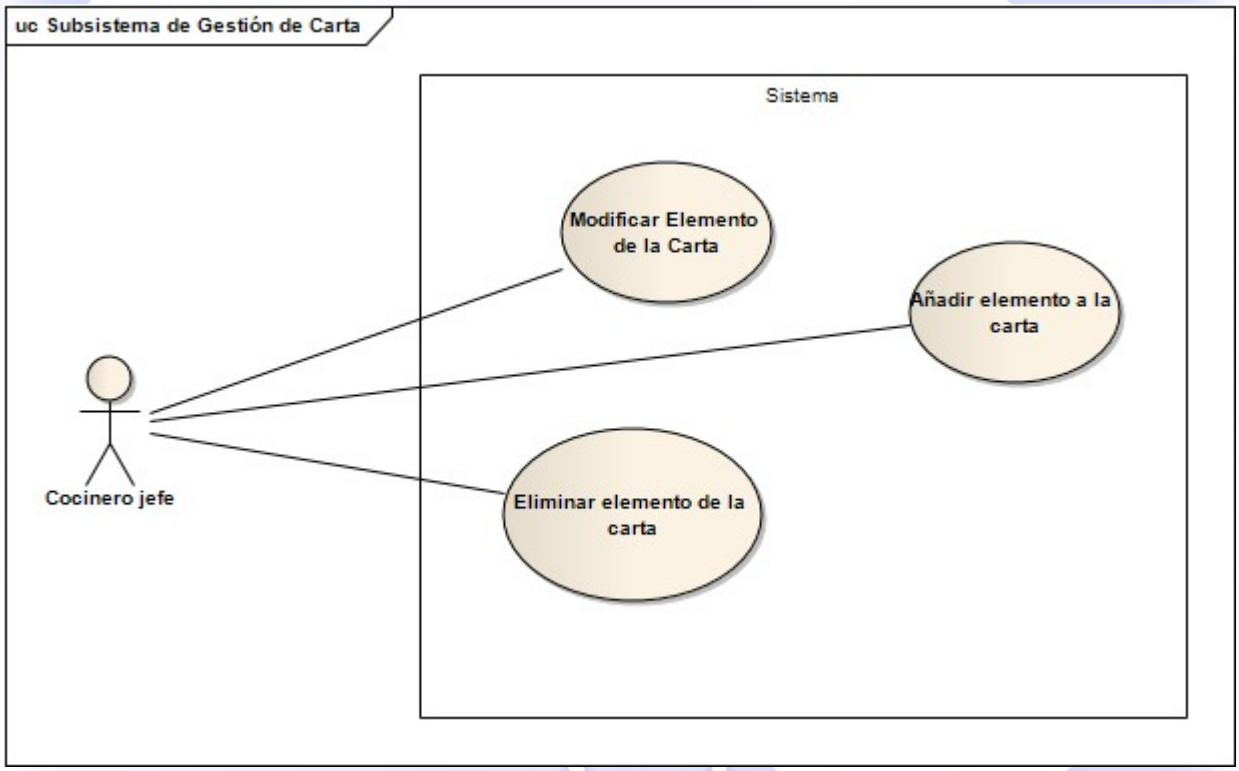


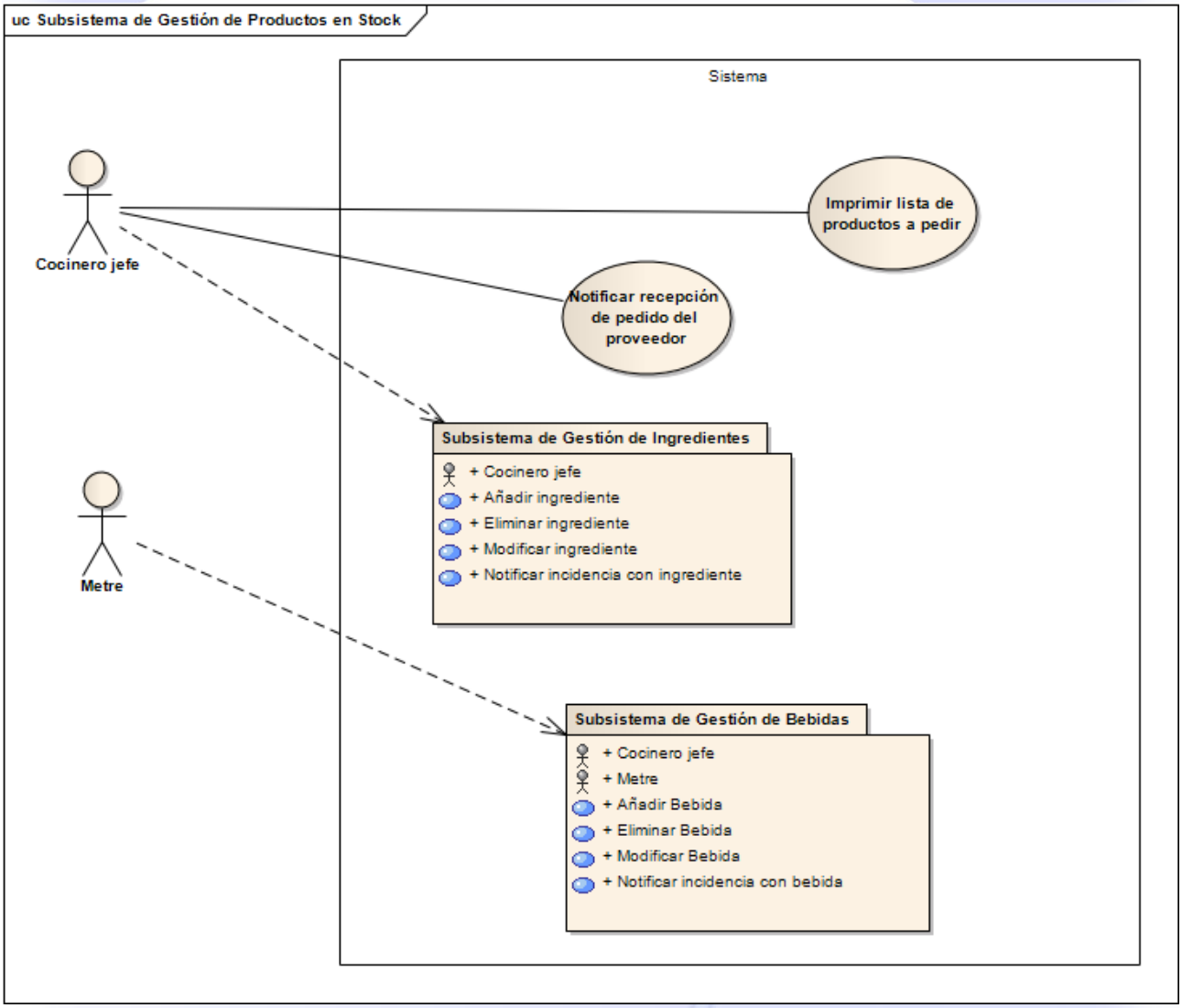
Diagrama de Paquetes



Subsistema de gestión de carta

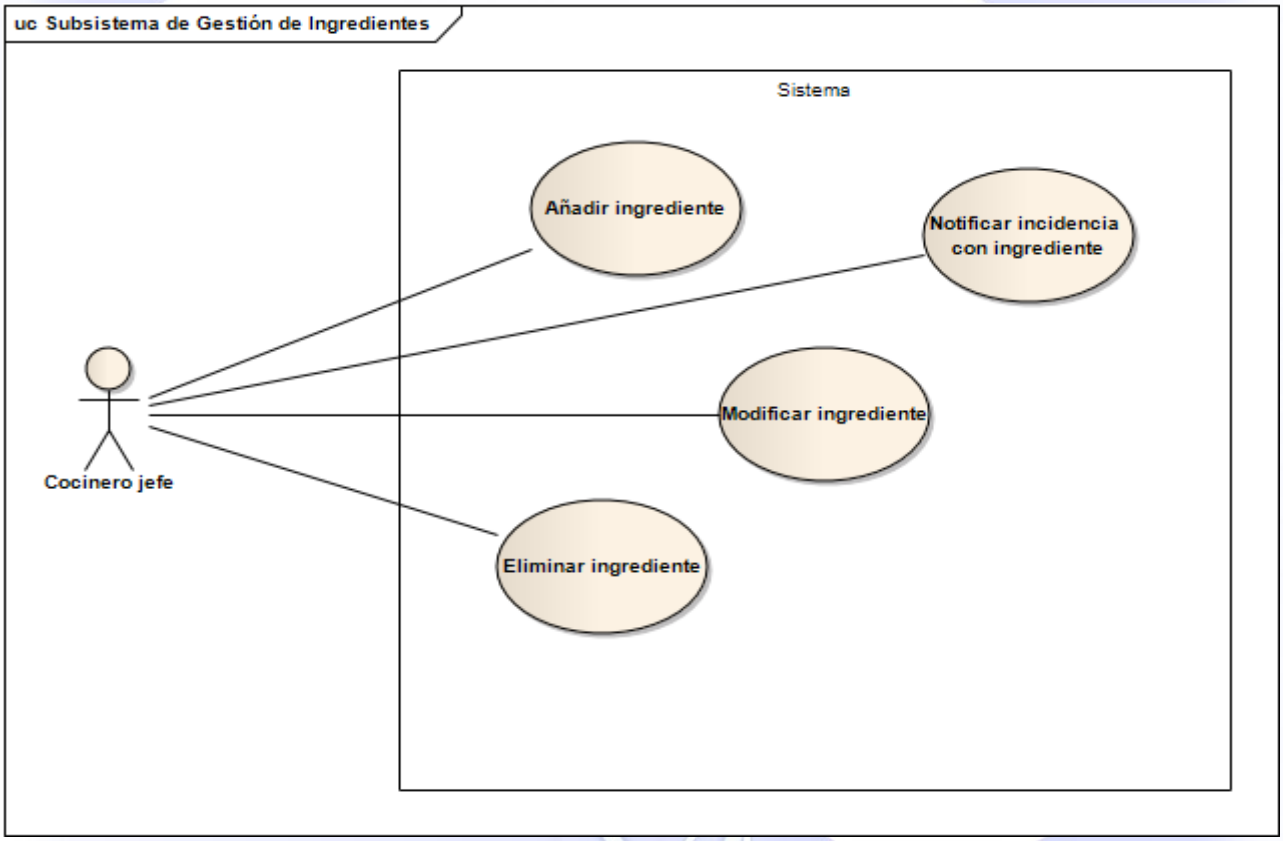


Subsistema de gestión de productos

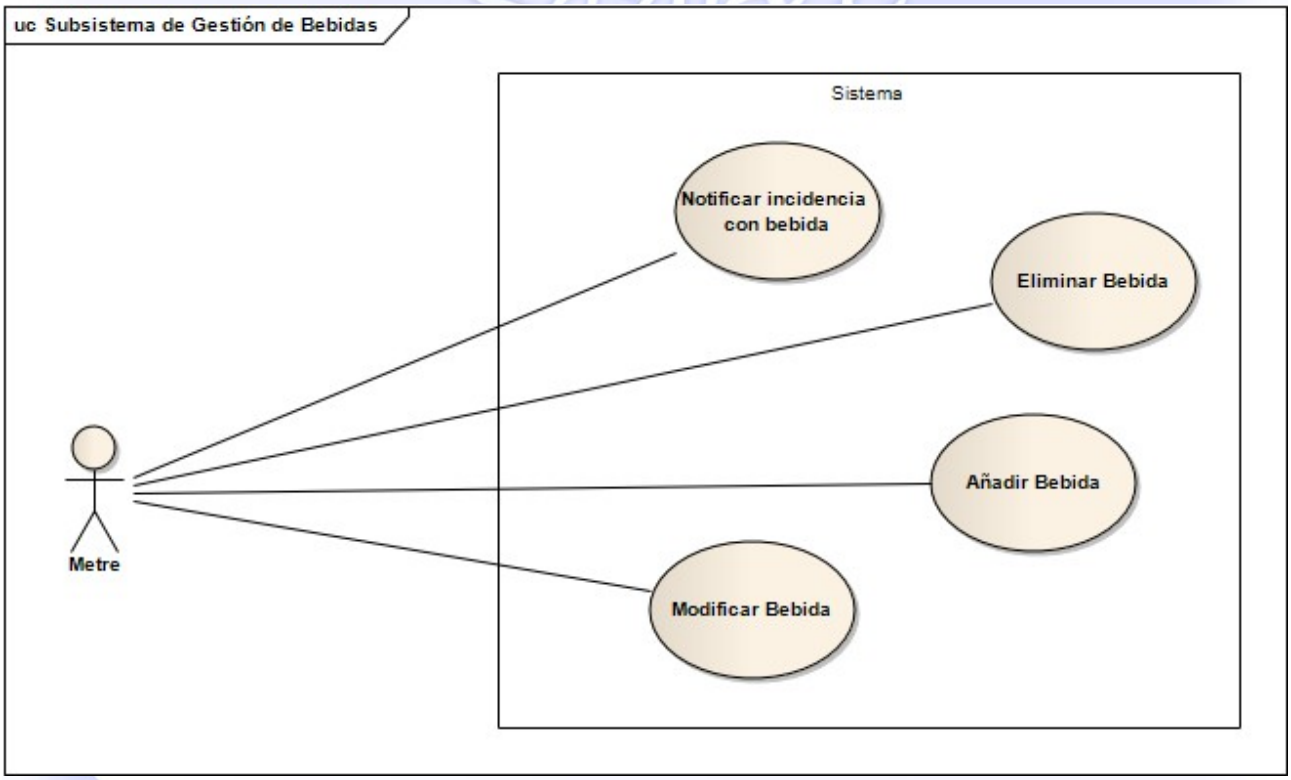


Este sería el diagrama de todo el subsistema de productos, pero como anteriormente comentamos se fracciona en dos nuevos, que son los que veremos a continuación.

Subsistema de gestión de ingredientes



Subsistema de gestión de bebidas



Identificación y descripción de los requisitos no funcionales del sistema

Facilidad de uso

- El sistema debe tener una interfaz de usuario amigable e intuitiva. Un usuario sin conocimientos informáticos deberá poder manejar la aplicación correctamente.
- Al ejecutarse sobre un terminal táctil, la interacción con el usuario se hará mediante un diseño fácil de utilizar y visible.
- El entorno deberá mostrar una breve información sobre cada una de las tareas (tips) que podemos realizar, preferiblemente al estar sobre una opción un determinado tiempo.
- El usuario dispondrá de un sencillo manual de uso sobre toda la funcionalidad del sistema.

Fiabilidad

- El sistema debe tener un grado alto de fiabilidad y robustez.
- Se debe prevenir y tratar cualquier error, mostrando un mensaje de información acerca de lo ocurrido, es decir, garantizamos la correcta captura de excepciones.
- El sistema deberá advertir ante posibles operaciones o acciones inválidas o erróneas que puedan provocar errores.
- Para prevenir de una caída del sistema y/o pérdidas de información, haremos copias de seguridad cada cierto tiempo de nuestros datos.
- Por ello, el número mayor de datos que podemos perder es el de los guardados desde la última copia de seguridad de nuestra base de datos.

Rendimiento

- No hay un tiempo de respuesta determinado hacia tareas en concreto pero al no requerir de cálculos u operaciones complejas deberá ser un tiempo eficiente y rápido.
- El tamaño de espacio ocupado en memoria masiva en el servidor de la base de datos irá en función de la cantidad de información almacenada en el sistema y no variará mucho respecto a otros sistemas que incluyan una base de datos.
- Se permitirá la ejecución de varios clientes concurrentemente y se garantizará la atención correcta y precisa de las peticiones de estos por parte de nuestro servidor.
- La peor situación aceptable para un usuario será la de que sus peticiones tarden un tiempo más de lo normal en ser atendidas.

Soporte

- El producto final será soportado en cualquier equipo con la máquina virtual de java instalada y donde pueda correr una versión compatible del gestor de la base de datos.
- Deberá ser fácilmente actualizable. Las tareas de mantenimiento, tales como actualizaciones a nuevos entornos hardware, serán resueltas por los programadores.

Implementación

- La plataforma hardware consistirá en un terminal táctil con conexión permanente a la red del servidor en el caso de los terminales de las mesas,cocina y bar; y a Internet en el caso de los terminales de las habitaciones.
- Cada cliente web deberá contar con conexión de red al servidor (generalmente Internet u lan que soporte conectividad tcp/ip).
- El lenguaje de programación empleado será java y en la parte del cliente web utilizaremos php + mysql.
- Para la implementación del código de la aplicación se utilizará el IDE gratuito NetBeans, ya que dispone de una opción que nos permitirá utilizar subversión, quedando los archivos de código del proyecto distribuidos en una misma localización en Internet.
- Para dicho subversión, se utilizará la aplicación web de uso gratuita Google Code para almacenar los archivos y SVN Tortoise para gestionar estos últimos (también de uso gratuito).

Interfaz

- El sistema no interactuará con otro sistema externo. Los datos importados serán introducidos por un usuario mediante los menús gráficos aportados por el sistema y de forma táctil o bien provendrán de los equipos clientes vía red local o Internet. Los datos se exportarán desde la aplicación al cliente también web por red local o Internet.

Operaciones

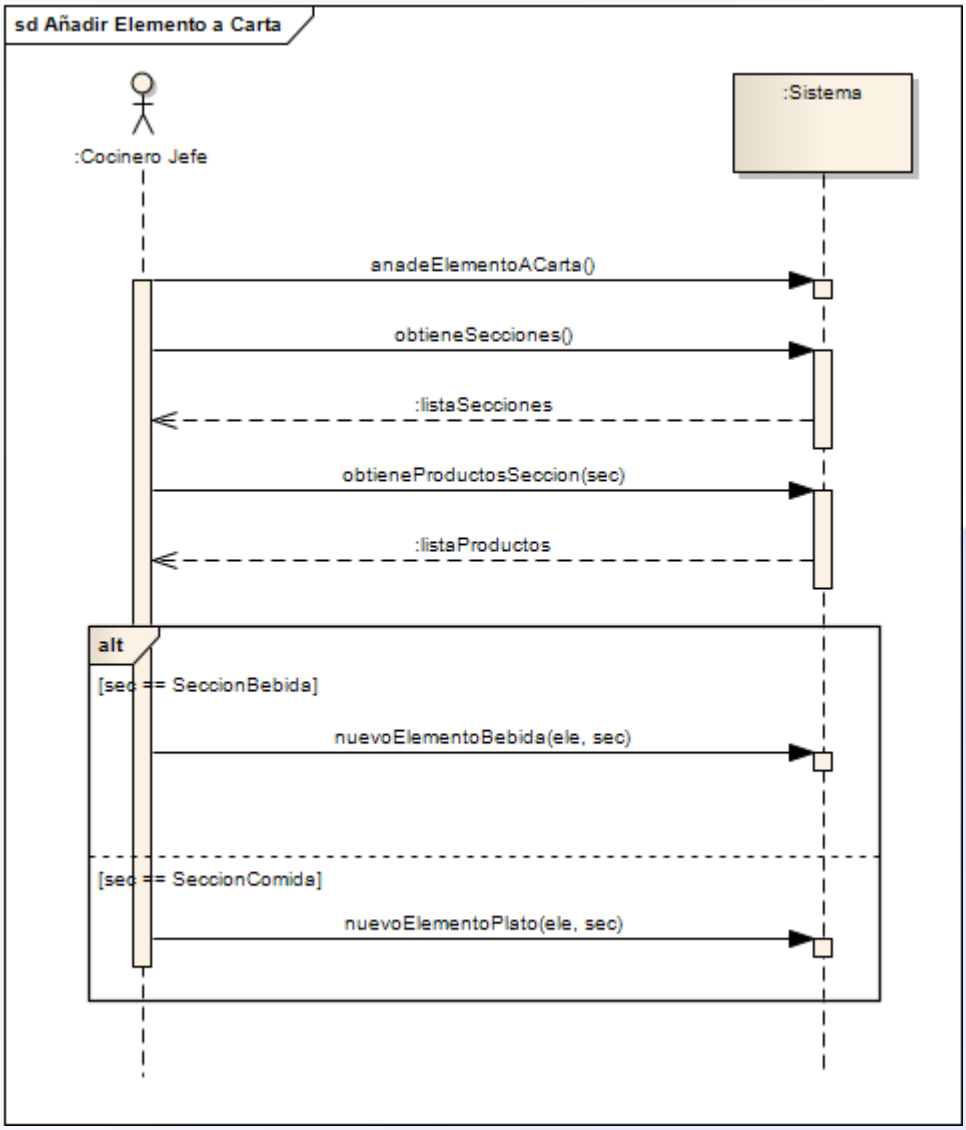
- El sistema al iniciarse dispondrá de la información contenida en su base de datos.
- Los administradores interaccionarán con el sistema cuando surja algún cambio imprevisto que el sistema no sea capaz de detectar.

Legales

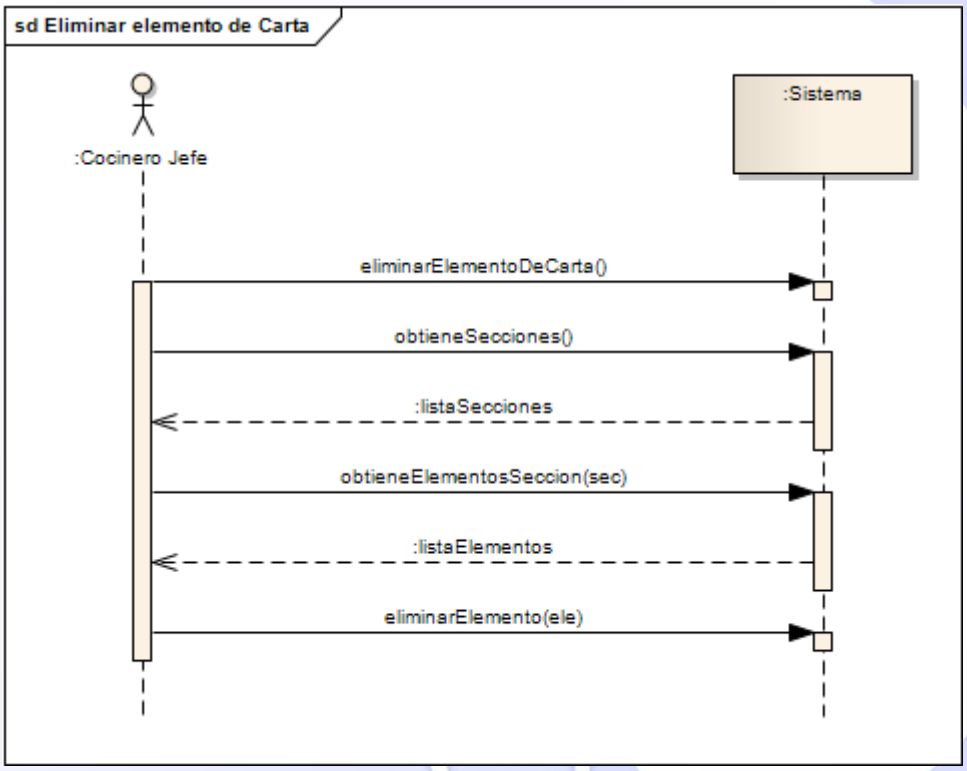
- El sistema debe cumplir las disposiciones recogidas en la Ley Orgánica de Datos Personales y en el Reglamento de medidas de seguridad.
- Al usar software gratuito para el desarrollo del proyecto no será necesaria la compra de licencias de ningún tipo.

Diagramas de secuencia del sistema

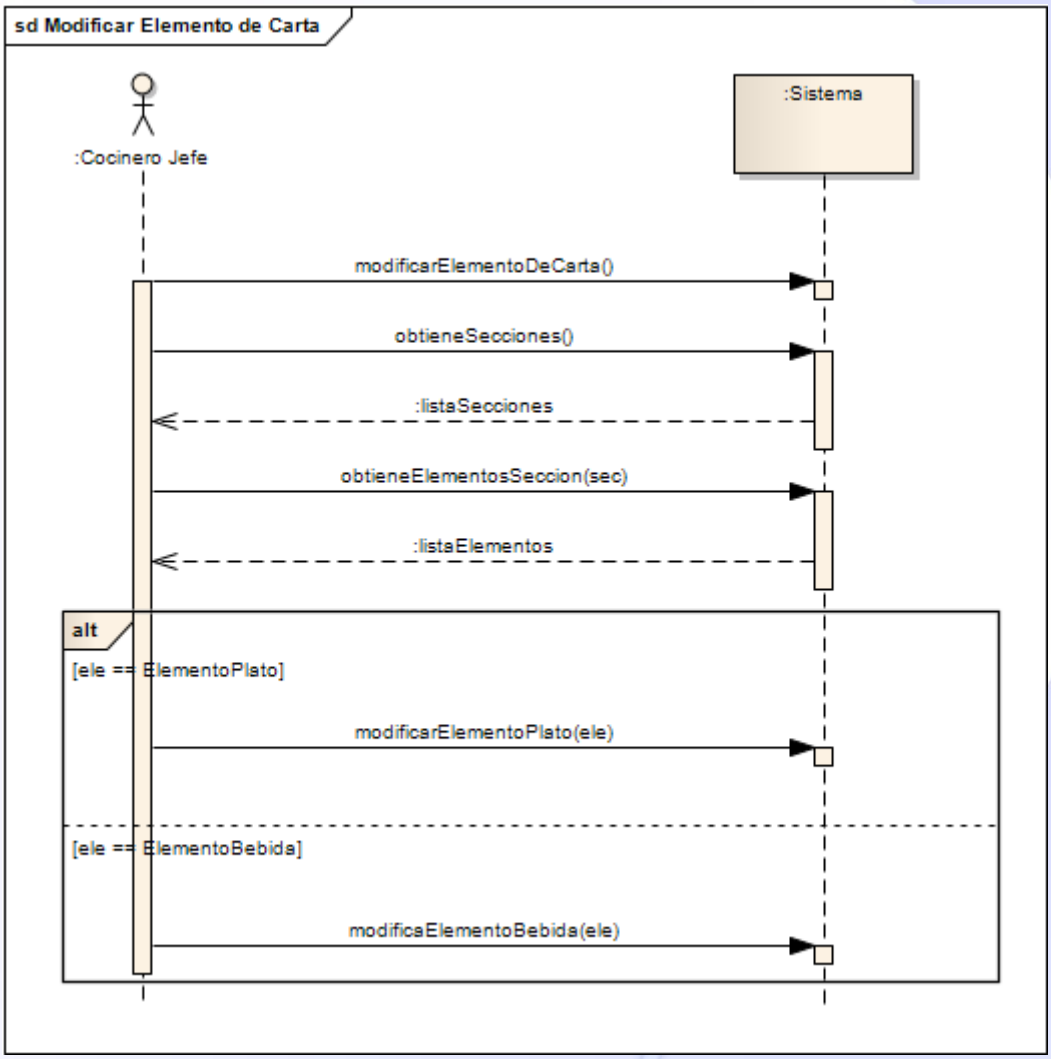
Añadir elemento a la carta



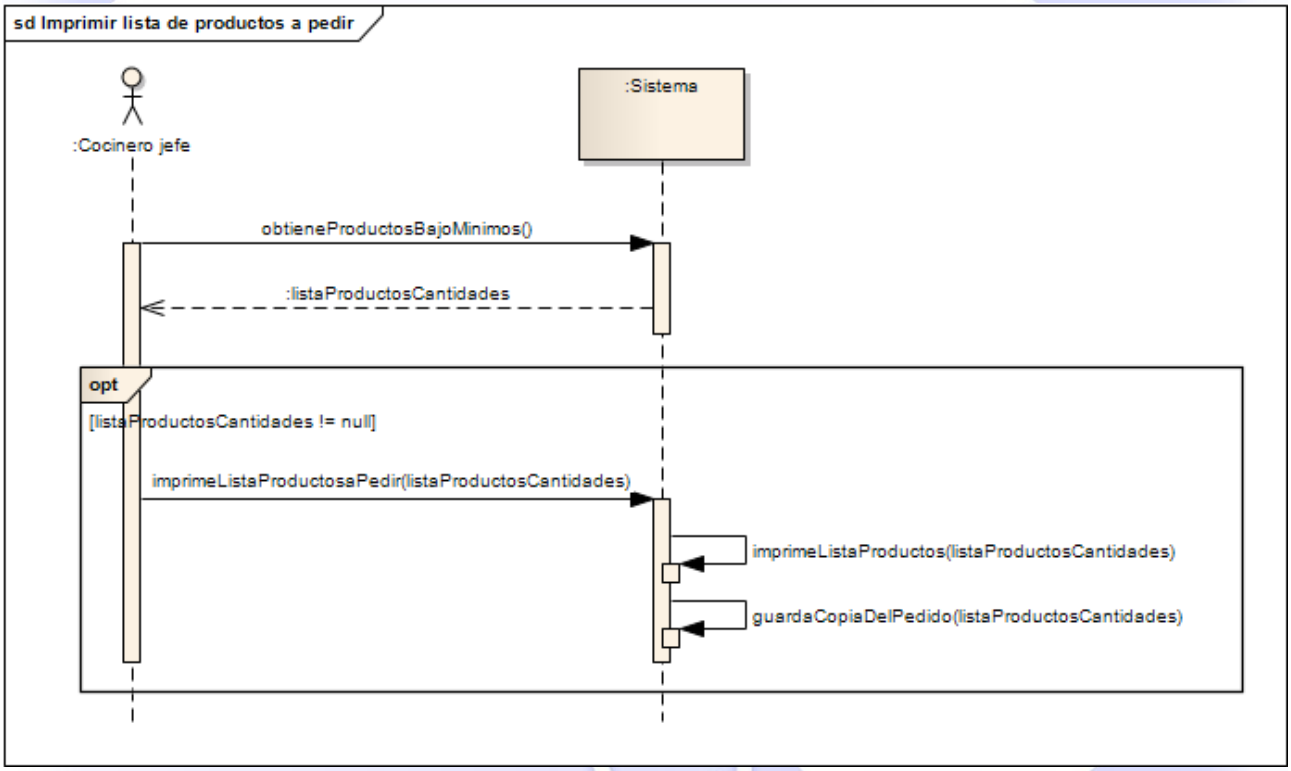
Eliminar elemento de la carta



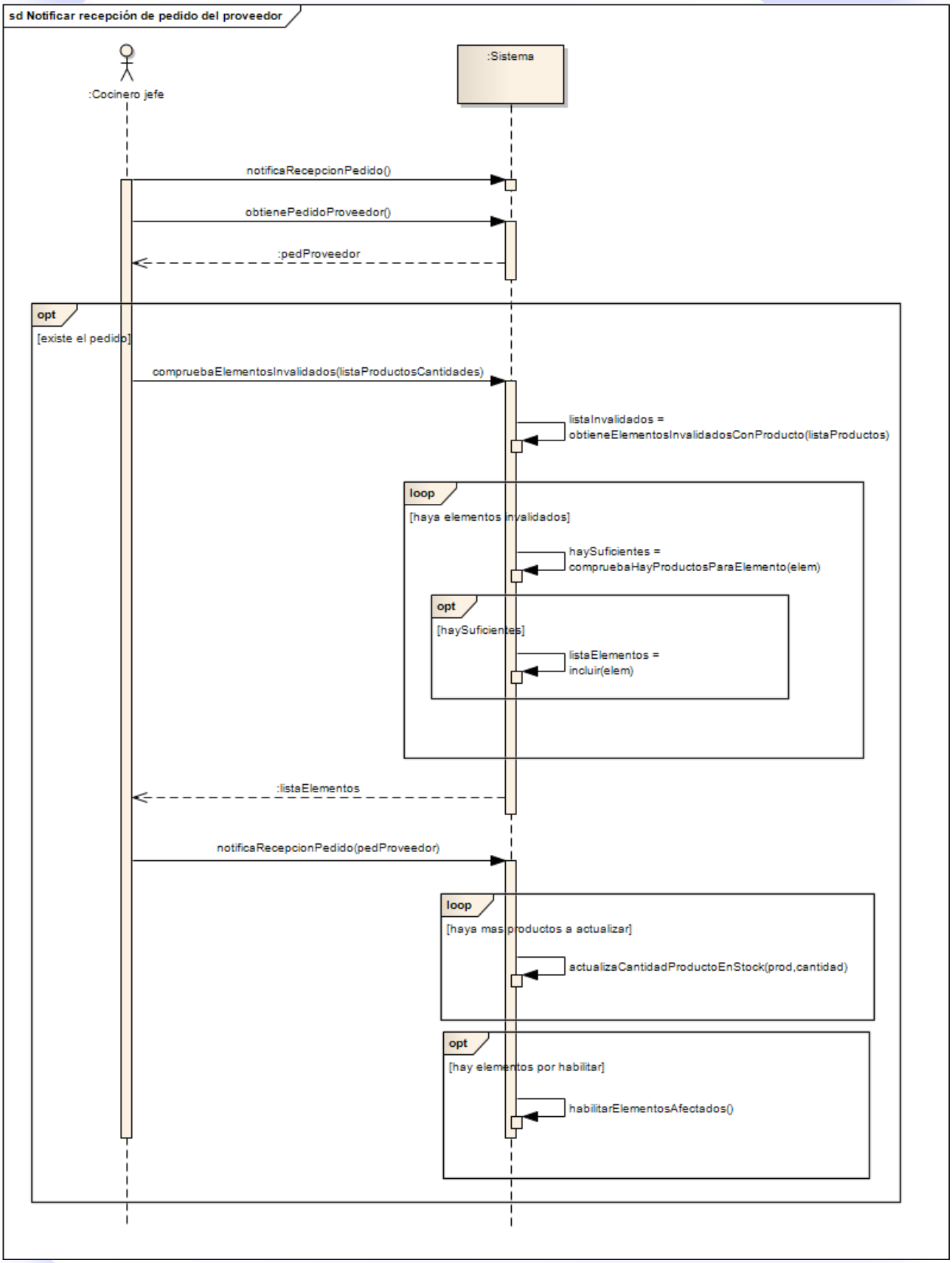
Modificar elemento de la carta



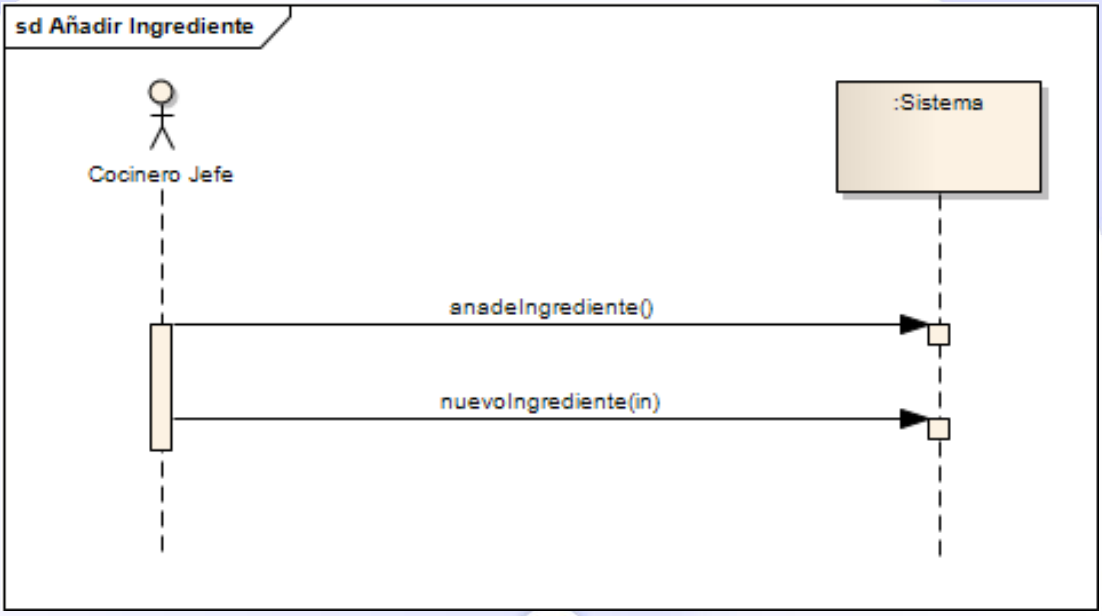
Imprimir lista de productos a pedir



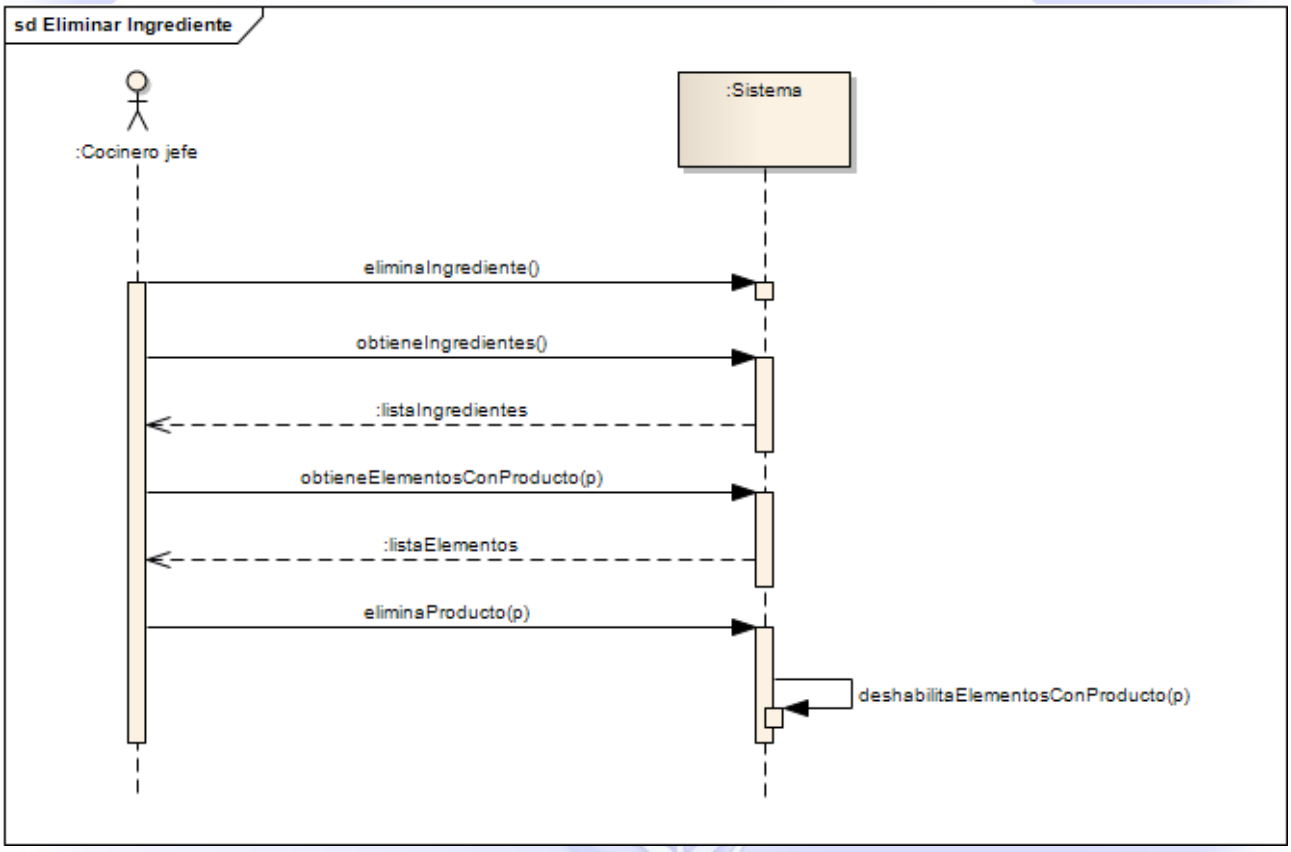
Notificar recepción de pedido del proveedor



Añadir ingrediente

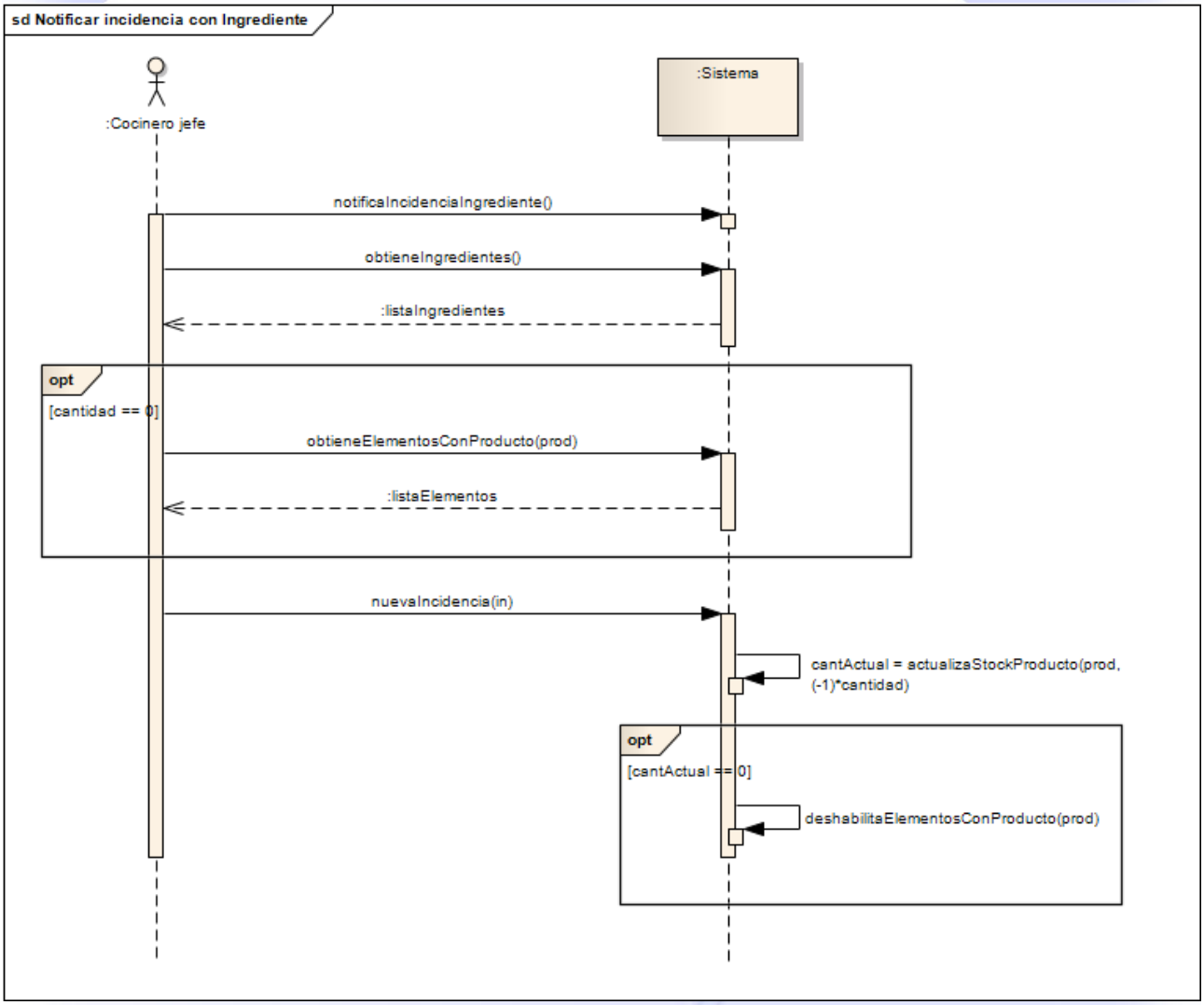


Eliminar ingrediente

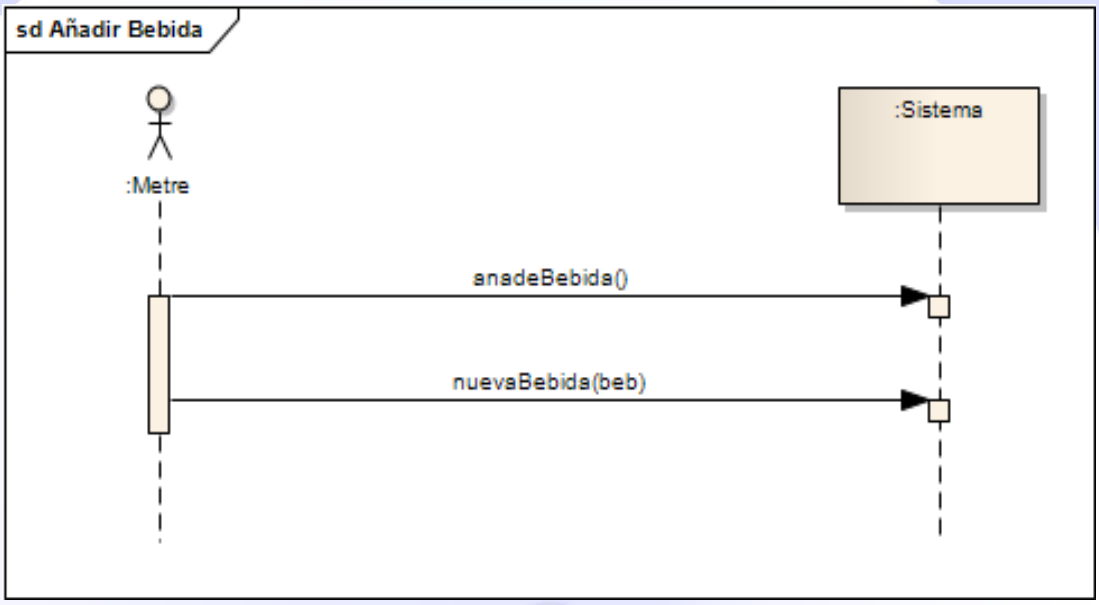


Modificar ingrediente

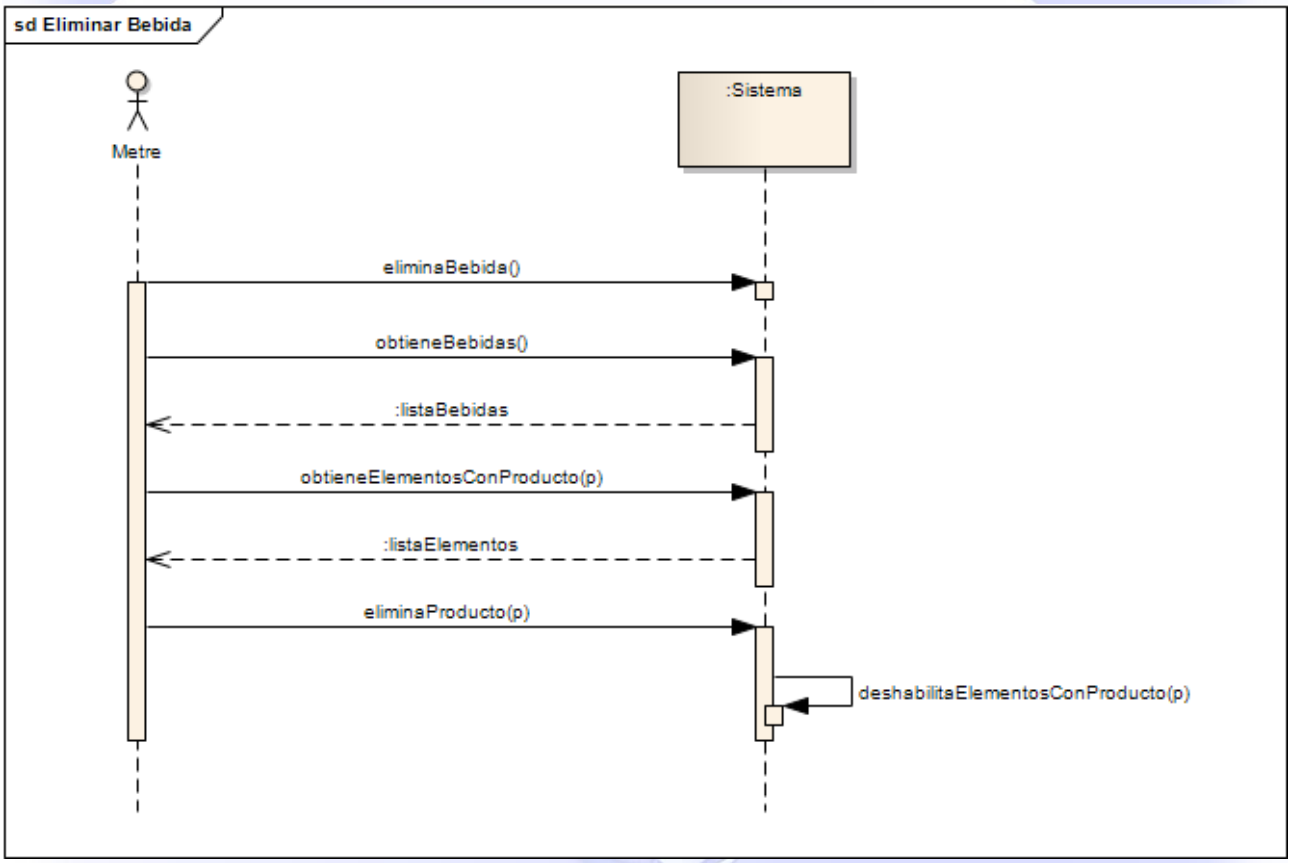
Notificar incidencia con ingrediente



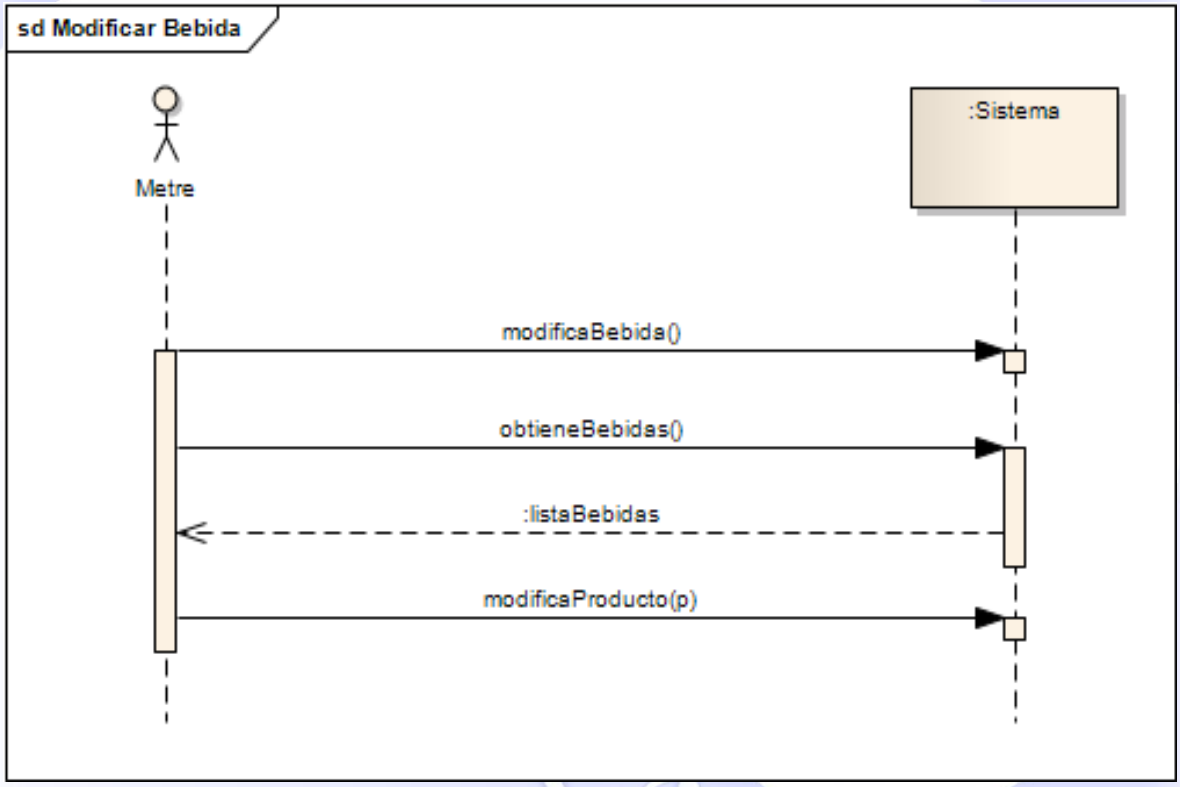
Añadir bebida



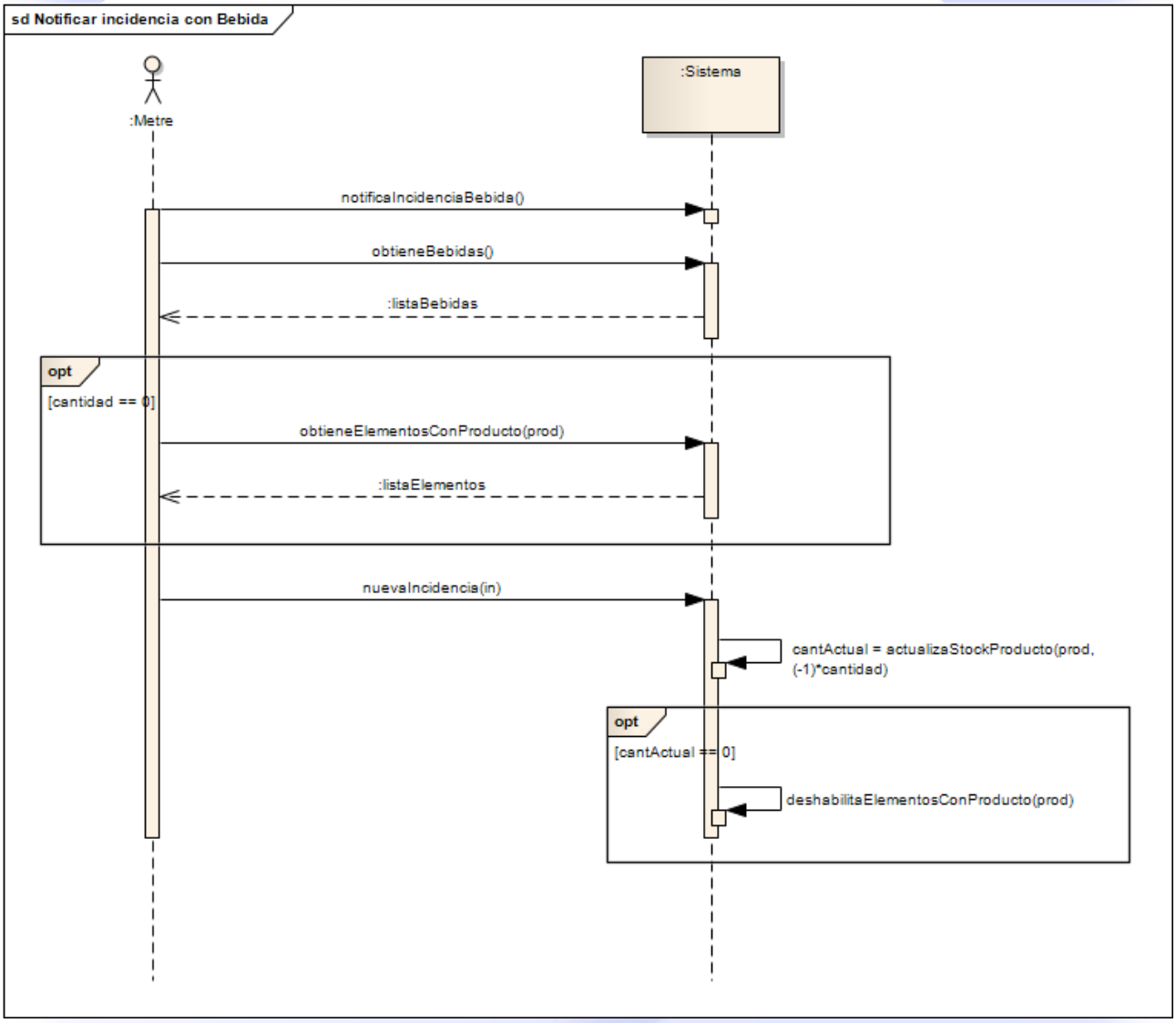
Eliminar bebida



Modificar bebida



Notificar incidencia con bebida



Identificación de las operaciones del sistema

Añadir elemento a la carta

- anadeElementoACarta(): operación que inicia el caso de uso para añadir un nuevo elemento a la carta.
- obtieneSecciones(): Devuelve una lista con las secciones registradas en el sistema.
- obtieneProductosSeccion(sec): Devuelve una lista de productos acordes a esa sección.
- nuevoElementoBebida(ele,sec): Registra un nuevo elemento bebida en el sistema y lo asocia con su sección
- nuevoElementoPlato(ele,sec): Registra un nuevo elemento plato en el sistema y lo asocia con su sección

Eliminar elemento de la carta

- eliminaElementoDeCarta(): operación que inicia el caso de uso para eliminar un elemento de la carta.
- obtieneSecciones(): Devuelve una lista con las secciones registradas en el sistema.
- obtieneElementosSeccion(sec): Devuelve una lista de elementos asociados a esa sección
- eliminaElemento(ele): Elimina el elemento del sistema

Modificar elemento de la carta

- modificaElementoDeCarta(): operación que inicia el caso de uso para modificar un elemento de la carta. Devuelve la lista de elementos registrados en el sistema.
- obtieneSecciones(): Devuelve una lista con las secciones registradas en el sistema.
- obtieneElementosSeccion(sec): Devuelve una lista de elementos asociados a esa sección
- modificaElementoPlato(ele): Registra en el sistema los cambios realizados en el elemento plato.
- modificaElementoBebida(ele): Registra en el sistema los cambios realizados en el elemento bebida.

Imprimir lista de productos a pedir

- obtieneProductosBajoMinimos(): Devuelve una lista de productos cuya cantidad en stock esta por debajo de su mínimo y las cantidades necesarias para que alcancen su máximo.
- imprimeListaProductosaPedir(): Ordena a la impresora que imprima el pedido y guarda una copia en el sistema.

Notificar recepción de pedido del proveedor

- notificaRecepcionPedido(): operación que inicia el caso de uso para notificar al sistema que se ha recibido el pedido del proveedor.
- obtienePedidoProveedor(): Devuelve un objeto con la informacion del último pedido

pendiente de recibir.

- compruebaElementosInvalidados(listaProductosCantidades): Devuelve una lista con los elementos de carta que se habilitarían tras actualizar las cantidades de los productos recibidos.
- notificaRecepcionPedido(pedProveedor): Actualiza las cantidades de los productos en el sistema y habilita los elementos de carta correspondientes.

Añadir ingrediente

- anadeIngrediente(): Operación que inicia el caso de uso para añadir un nuevo ingrediente.
- nuevoIngrediente(in): Registra un nuevo ingrediente en el sistema.

Eliminar ingrediente

- eliminaIngrediente(): Operación que inicia el caso de uso para eliminar un ingrediente.
- obtieneIngredientes(): Devuelve una lista con los ingredientes registrados en el sistema
- obtieneElementosConProducto(p): Devuelve una lista con los elementos de carta que tiene el producto “p” asociado.
- eliminaProducto(p): Elimina el producto p del sistema e invalida los elementos de carta que tienen asociado dicho producto.

Modificar ingrediente

- modificaIngrediente(): Operación que inicia el caso de uso para modificar un ingrediente.
- obtieneIngredientes(): Devuelve una lista con los ingredientes registrados en el sistema
- modificaProducto(p): Registra en el sistema los cambios efectuados en el producto

Notificar incidencia con ingrediente

- notificaIncidenciaIngrediente(): operación que inicial el caso de uso para notificar una incidencia con un ingrediente.
- modificaProducto(p): Registra en el sistema los cambios efectuados en el producto
- obtieneElementosConProducto(prod): Devuelve una lista con los elementos que tienen ese producto asociado.
- nuevaIncidencia(in): Registra una nueva incidencia en el sistema

Añadir Bebida

- anadeBebida(): Operación que inicia el caso de uso para añadir una nueva bebida.
- nuevaBebida(beb): Registra una nueva bebida en el sistema

Eliminar Bebida

- eliminaBebida(): Operación que inicia el caso de uso para eliminar una bebida.
- obtieneBebidas(): Devuelve una lista con las bebidas registrados en el sistema
- obtieneElementosConProducto(p): Devuelve una lista con los elementos de carta que tiene el producto “p” asociado.
- eliminaProducto(p): Elimina el producto p del sistema e invalida los elementos de carta que tienen asociado dicho producto.

Modificar bebida

- modificaBebida(): Operación que inicia el caso de uso para modificar una bebida.
- obtieneBebidas(): Devuelve una lista con las bebidas registrados en el sistema
- modificaProducto(p): Registra en el sistema los cambios efectuados en el producto

Notificar incidencia con bebida

- notificaIncidenciaBebida(): operación que inicial el caso de uso para notificar una incidencia con una bebida.
- obtieneElementosConProducto(prod): Devuelve una lista con los elementos que tienen ese producto asociado.
- nuevaIncidencia(in): Registra una nueva incidencia en el sistema

Apéndice 1.0

Fecha	15/03/10
Descripción del problema	-
Impacto del problema	-
Soluciones adoptadas	• Se generado el documento de modelado de requisitos inicial
Anexos a la versión	

Apéndice 1.1

Fecha	16/03/10
Descripción del problema	Se han encontrado unos pequeños problemas en las especificaciones de algunos casos de uso que nos han obligado a modificar tanto estos como sus respectivos diagramas de secuencia. Además, el hecho de que los diagramas de secuencia se hayan repartido ha provocado que aparezcan distintas denominaciones para operaciones del sistema que realmente tienen la misma funcionalidad. También nos hemos percatado de que no habíamos contemplado el requisito funcional RF14, el cual especifica que si un producto se agota se deben deshabilitar los elementos de la carta que dependan de él.
Impacto del problema	Los problemas mencionados en la descripción del problema son concretamente que en algunos casos de uso realizábamos demasiadas llamadas a operaciones de consulta. Esto podría resultar perjudicial para la eficiencia de nuestra aplicación. El hecho de que haya operaciones con distinto nombre pero igual funcionalidad iba a suponer un coste tremendo para nosotros en primer lugar porque hubiéramos hecho los contratos de operaciones repetidas, y en segundo lugar para el equipo de implementación puesto que habrían invertido un tiempo valioso en repetir operaciones del sistema absurdamente. No tener en cuenta el requisito funcional RF14 hubiera provocado que el cliente hubiese podido seleccionar un plato para el cual no hubiera habido ingredientes para prepararlo.
Soluciones adoptadas	• Se han corregido los resúmenes de los casos de uso para que fuesen más claros. • Se han eliminado consultas innecesarias en los cursos normales de algunos casos de uso. • Se han unificado los nombres de las operaciones del sistema con funcionalidad equivalente. • Se han modificado los diagramas de secuencia para que se contemplen las soluciones descritas en los puntos anteriores. • Se ha incluido el requisito funcional RF14 tanto es la especificaciones de los casos de uso involucrados como en los diagramas de secuencia de esos casos de uso.
Anexos a la versión	

Apéndice 1.2

Fecha	21/03/10
Descripción del problema	Se han encontrado unos pequeños problemas en los diagramas de secuencia del sistema de algunos casos de uso. Estos problemas tenían que ver con algunas operaciones de consultas y con los nombres de algunas operaciones del sistema que no explicaban bien su cometido. No queda del todo claro la función de los casos de uso "Consultar..."
Impacto del problema	Algunos diagramas de secuencia tenían la operación salir() cuando en realidad no hacía falta. Esto podría suponer un error de interpretación para el equipo de implementación. Algunos nombres de las operaciones no se correspondían con su cometido, a entendimientos ambiguos por parte de los integrantes de nuestro equipo. El carácter cíclico que hemos querido darles a los casos de uso nos ha supuesto muchos problemas en el diseño de los diagramas de secuencia.
Soluciones adoptadas	• Se han eliminado tanto de los diagramas de secuencia como de las especificaciones de los casos de uso el último paso correspondiente siempre a "el usuario finaliza la operación". • Se han modificado algunos nombres de operaciones y la correspondiente lista final de éstas. • Se ha optado por borrar los casos de uso que hacían referencia a todo tipo de "consultas", puesto que consideramos que es mejor tratarlas como operaciones privadas del sistema. • También se ha corregido el formato de los cursos alternativos de algunas especificaciones de casos de uso ya que no quedaban del todo claros. • Se ha eliminado de los casos de uso la posibilidad de realizarlos varias veces. Ahora cada caso de uso solo se ejecutará una vez, quedando su especificación mucho más clara.
Anexos a la versión	

Apéndice 1.3

Fecha	15/04/10
Descripción del problema	Tras la entrega de toda documentación de la 1ª iteración y la defensa de ésta, el equipo de planificación de la 2ª iteración ha propuesto una revisón completa de la 1ª iteración para solventar los fallos encontrados por el profesor.
Impacto del problema	Cuando se diseñaron los diagramas de secuencia del sistema, se llegó al acuerdo de que las operaciones del sistema recibirían los parámetros necesarios para la creación de los objetos. Esto presenta un problema a la hora de añadir nuevo parámetros a los objetos, lo cual obligaría a cambiar las cabeceras de las funciones. Otro problema detectado ha sido que el usuario no podía tener conocimiento previo de qué consecuencias tendría la realización de una operación concreta antes de realizarla.
Soluciones adoptadas	• Se han modificado las cabeceras de las funciones para trabajar directamente con objetos. • Se han añadido operaciones de comprobación para que el usuario pueda ver el estado futuro del sistema tras la realización de una operación.
Anexos a la versión	