

Documento de MODELADO DE REQUISITOS



**Sistema de Administración y
Gestión de RESTaurantes**



Samuel Guirado Navarro
Carlos Moreno Muñoz
Gaspar Muñoz Soria

V1.1

ÍNDICE DE CONTENIDO

Apartado de control de versiones.....	4
1. Modelo funcional.....	5
Identificación de actores.....	5
Cocinero Jefe.....	5
Metre.....	5
Identificación de los requerimientos funcionales.....	6
Descripción de los casos de uso.....	7
Añadir elemento a la carta.....	7
Eliminar elemento a la carta.....	9
Modificar elemento de la carta.....	10
Consultar elementos de la carta.....	11
Imprimir lista de productos a pedir.....	12
Notificar incidencia con Ingrediente.....	13
Notificar incidencia con bebida.....	14
Notificar recepción de pedido del proveedor.....	15
Añadir ingrediente.....	16
Eliminar ingrediente.....	17
Modificar ingrediente.....	18
Añadir bebida.....	19
Eliminar bebida.....	20
Modificar bebida.....	21
Consultar lista de productos.....	22
Consultar lista de ingredientes.....	23
Consultar lista de bebidas.....	24
Elaboración del Diagrama de casos de uso.....	25
2. Subsistemas funcionales.....	26
Identificación de los subsistemas funcionales.....	26
Elaboración del diagrama de paquetes para subsistemas funcionales.....	27
Diagrama de casos de uso global con división en subsistemas.....	27
Subsistema de gestión de carta.....	28
Subsistema de gestión de productos.....	28
Subsistema de gestión de ingredientes.....	29
Subsistema de gestión de bebidas.....	29

3. Requisitos no funcionales.....	30
<i>Identificación y descripción de los requisitos no funcionales del sistema.....</i>	<i>30</i>
Facilidad de uso.....	30
Fiabilidad.....	30
Rendimiento.....	30
Soporte.....	31
Implementación.....	31
Interfaz.....	31
Operaciones.....	31
Legales.....	31
4. Operaciones del sistema.....	32
<i>Elaboración de diagramas de secuencia del sistema</i>	<i>32</i>
Añadir elemento a la carta.....	32
Eliminar elemento de la carta.....	33
Modificar elemento de la carta.....	34
Consultar elementos de la carta.....	35
Imprimir lista de productos a pedir.....	35
Notificar recepción de pedido del proveedor.....	36
Consultar lista de productos.....	36
Añadir ingrediente.....	37
Eliminar ingrediente.....	38
Modificar ingrediente.....	39
Consultar lista de ingredientes.....	40
Notificar incidencia con ingrediente.....	41
Añadir bebida.....	42
Eliminar bebida.....	43
Modificar bebida.....	44
Consultar lista de bebidas.....	45
Notificar incidencia con bebida.....	46
<i>Identificación de las operaciones del sistema</i>	<i>47</i>
Apéndice 1.0.....	54

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.

<i>Versión</i>	<i>Fecha</i>	<i>Descripción</i>
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.



Sagres

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 o bebidas 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, éste también deberá notificar de cualquier incidencia relacionada con alguna bebida.

Sagres

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.

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 uno o más elementos 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:	Incluye al caso de uso “Consultar lista de Productos”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none"> Se han añadido uno o más elementos nuevos a la carta
Curso normal:	<ol style="list-style-type: none"> El caso de uso comienza cuando el usuario desea añadir uno o más elementos a la carta Se inicia el caso de uso “Consultar lista de Productos” Mientras el usuario desee añadir elementos a la carta: <ol style="list-style-type: none"> El usuario rellena los datos correspondientes al elemento: <ol style="list-style-type: none"> Introduce el nombre Introduce la descripción Introduce la foto Introduce el precio Introduce el tiempo de elaboración Selecciona la sección <ol style="list-style-type: none"> Si la sección asignada no es la de bebidas <ol style="list-style-type: none"> El usuario especifica si el plato se puede dividir en raciones Mientras el usuario quiera asociar productos al elemento: <ol style="list-style-type: none"> El usuario selecciona un producto de la lista El usuario introduce la cantidad deseada de dicho producto El usuario confirma los datos El sistema comprueba que los datos introducidos sean correctos Los datos introducidos son correctos. El sistema genera un código asociado al elemento y registra todos los datos El sistema confirma que se ha añadido un nuevo elemento a la carta satisfactoriamente El sistema pregunta al usuario si desea seguir añadiendo más elementos a la carta El usuario finaliza la operación
Cursos alternativos:	<ol style="list-style-type: none"> El usuario se ha equivocado al seleccionar un producto <ol style="list-style-type: none"> El usuario elimina el producto deseado

	<p>3.2.1. El usuario se ha equivocado al introducir la cantidad de un ingrediente</p> <p>3.2.1.1. El usuario selecciona el producto de la lista</p> <p>3.2.1.2. El usuario introduce la nueva cantidad</p> <p>3.5. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3.1</p> <p>*. El usuario cancela la operación</p>
<i>Observaciones:</i>	
<i>Requisitos no funcionales específicos:</i>	<p>- Las secciones asignables a un elemento de la carta serán: Entrantes, Carnes, Pescados, Bebidas o Postres</p> <p>- 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</p> <p>- La foto deberá estar en formato .jpeg y no sobrepasar los 200 kb</p>



Sagres

Eliminar elemento a la carta

Nombre del caso:	Eliminar elemento de la carta
Resumen:	El cocinero jefe desea eliminar uno o más elementos de la carta del restaurante. Le basta con tan solo con seleccionar el elemento que desea borrar y confirmarlo
Dependencias:	Incluye al caso de uso “Consultar elementos de la carta”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se han eliminado uno o más elementos de la carta
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario desea eliminar uno o más elementos de la carta2. Se inicia el caso de uso “Consultar elementos de la carta”3. Mientras el usuario quiera eliminar elementos de la carta:<ol style="list-style-type: none">3.1 El usuario selecciona el elemento de la carta que desea eliminar3.2 El sistema muestra la información asociada al elemento y solicita la confirmación de la operación3.3 El usuario confirma la operación3.4 El sistema confirma que el elemento ha sido eliminado de la carta satisfactoriamente3.5 El sistema pregunta al usuario si desea eliminar más elementos de la carta4. El usuario finaliza la operación
Cursos alternativos:	<ol style="list-style-type: none">3.3. El usuario no confirma la operación. Se vuelve al paso 3 <p>*. El usuario cancela la operación</p>
Observaciones:	
Requisitos no funcionales:	

Modificar elemento de la carta

Nombre del caso:	Modificar elemento de la carta
Resumen:	El cocinero jefe desea modificar uno o más elementos de la carta del restaurante. Selecciona el elemento y modifica el o los datos deseados.
Dependencias:	Incluye al caso de uso “Consultar elementos de la carta”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se han modificado uno o más elementos de la carta
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario desea modificar uno o más elementos de la carta2. Se inicia el caso de uso “Consultar elementos de la carta”3. Mientras el usuario necesite modificar elementos:<ol style="list-style-type: none">3.1 El usuario selecciona un elemento de la lista3.2 El sistema muestra la información asociada a ese elemento3.3 Mientras el usuario desee modificar datos del elemento:<ol style="list-style-type: none">3.3.1 El usuario modifica el dato deseado3.4 El usuario finaliza la modificación de los datos3.5 El sistema solicita la confirmación de la modificación de los datos3.6 El usuario confirma la operación3.7 El sistema comprueba que los datos introducidos son correctos.3.8 Los datos introducidos son correctos. El sistema registra los nuevos cambios y confirma que el elemento ha sido modificado con éxito3.9 El sistema pregunta al usuario si desea modificar otro elemento4. El usuario finaliza la operación
Cursos alternativos:	<ol style="list-style-type: none">3.8. El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3.2 <p>*. El usuario cancela la operación</p>
Observaciones:	
Requisitos no funcionales:	

Consultar elementos de la carta

Nombre del caso:	Consultar elementos de la carta
Resumen:	El cocinero jefe desea obtener una lista de los elementos que actualmente figuran en la carta del restaurante
Dependencias:	Es incluido por los casos de uso “Modificar elemento de la carta” y “Eliminar elemento de la carta”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se ha generado una lista con los elementos que forman la carta del restaurante
Curso normal:	<ol style="list-style-type: none">1. El caso de uso se inicia cuando el usuario desea conocer cuales son los elementos que componen la carta2. El sistema proporciona una lista detallada de todos los elementos incluidos en la carta
Cursos alternativos:	
Observaciones:	
Requisitos no funcionales específicos:	

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:	<ul style="list-style-type: none">• Se ha imprimido la lista de productos a pedir• Se ha guardado una copia de la lista en el sistema
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario quiere imprimir una lista con los productos que están bajos mínimos en stock2. El sistema obtiene la lista de productos que están bajo mínimos en stock3. El sistema imprime dicha lista4. El sistema guarda una copia de la lista5. 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

Añadir ingrediente

Nombre del caso:	Añadir ingrediente
Resumen:	El cocinero jefe desea añadir uno o más ingredientes al sistema. Introduce los datos correspondientes a cada ingrediente se comprueban si son correctos y se confirma cada operación.
Dependencias:	
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se han registrado uno o más productos en el sistema
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el cocinero jefe desea añadir uno o más ingredientes.2. Mientras que el usuario quiera añadir ingredientes:<ol style="list-style-type: none">2.1. El usuario introduce los datos correspondientes al nuevo ingrediente:<ol style="list-style-type: none">2.1.1 Introduce el nombre2.1.2 Introduce el límite mínimo en stock.2.1.3 Introduce el máximo en stock.2.1.4 Introduce la cantidad actual en stock2.1.5 Introduce una foto.2.2. El usuario confirma los datos.2.3. El sistema comprueba si los datos introducidos son correctos2.4. Los datos introducidos son correctos. El sistema genera un código de producto, registra todos los datos y confirma que ha sido añadido satisfactoriamente.2.5. El sistema pregunta al usuario si desea seguir añadiendo productos3. El usuario finaliza la operación
Cursos alternativos:	<p>2.4b El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3.1</p> <p>*. El usuario cancela la operación.</p>
Observaciones:	
Requisitos no funcionales específicos:	<ul style="list-style-type: none">- 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 y pregunta por si quiere seguir eliminando.
Dependencias:	Incluye al caso de uso “Consultar lista de ingredientes”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se han eliminado uno o más productos del sistema.
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el cocinero jefe desea eliminar uno o más ingredientes.2. Se inicia el caso de uso “Consultar lista de ingredientes”3. Mientras que el usuario quiera eliminar ingredientes:<ol style="list-style-type: none">3.1. El usuario selecciona el ingrediente que quiere eliminar.3.2. El sistema solicita la confirmación de la operación.3.3. El usuario confirma la operación.3.4. El sistema comprueba los platos que contienen este ingrediente y los invalida.3.5. El sistema elimina el ingrediente e informa de ello al usuario.3.6. El sistema pregunta al usuario si desea eliminar otro elemento4. El usuario finaliza la operación
Cursos alternativos:	<p>3.3b El usuario no confirma la operación. Se vuelve al paso 2.</p> <p>*. El usuario cancela la operación.</p>
Observaciones:	

Modificar ingrediente

Nombre del caso:	Modificar ingrediente
Resumen:	El cocinero jefe quiere modificar los datos de unos determinados ingredientes. Una vez que se le muestran los datos, cambia los parámetros que quiera y una vez confirmados y comprobados estos se registra la operación.
Dependencias:	Incluye al caso de uso “Consultar lista de ingredientes”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se ha registrado uno o más cambios en los detalles de un ingrediente.
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el cocinero jefe desea modificar uno o mas ingredientes.2. Se inicia el caso de uso “Consultar lista de ingredientes”3. Mientras el usuario quiera modificar ingredientes:<ol style="list-style-type: none">3.1. El usuario selecciona el ingrediente que quiere modificar.3.2. El sistema muestra los datos asociados a ese ingrediente.3.3. Mientras el usuario quiera modificar parámetros:<ol style="list-style-type: none">3.3.1. El usuario modifica el parámetro deseado.3.4. El sistema solicita al usuario que confirme los cambios3.5. El usuario confirma los cambios.3.6. El sistema comprueba si los datos introducidos son correctos3.7. Los datos introducidos son correctos. El sistema registra los cambios e informa que se han realizado satisfactoriamente.3.8. El sistema pregunta al usuario si desea modificar otro ingrediente4. El usuario finaliza la operación
Cursos alternativos:	<p>3.5b El usuario no confirma los cambios. Se vuelve al paso 2</p> <p>3.6b El sistema informa de que los datos introducidos son incorrectos. Se vuelve al paso 2.</p> <p>*. El usuario cancela la operación.</p>
Observaciones:	

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:	Incluye el caso de uso “Consultar lista de Ingredientes”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none"> Se ha registrado una incidencia con un ingrediente
Curso normal:	<ol style="list-style-type: none"> El caso de uso comienza cuando el usuario desea notificar una incidencia ocurrida con un ingrediente Se inicia el caso de uso “Consultar lista de Ingredientes” Mientras el usuario desee notificar incidencias: <ol style="list-style-type: none"> El usuario selecciona el ingrediente sobre el que desea notificar la incidencia El usuario introduce la cantidad de producto afectado y especifica el motivo de la incidencia El sistema comprueba que los datos introducidos son correctos. Los datos introducidos son correctos. El sistema genera un código asociado a la incidencia y registra la incidencia El sistema resta del stock la cantidad del ingrediente especificado por el usuario El sistema comprueba la cantidad que queda del ingrediente en stock <ol style="list-style-type: none"> Si la cantidad actual en stock del ingrediente afectado se encuentra por debajo de su mínimo <ol style="list-style-type: none"> El sistema informa de que la cantidad ese producto en stock se encuentra por debajo de su mínimo El sistema informa de que la incidencia se ha registrado correctamente El sistema pregunta al usuario si desea notificar alguna incidencia más El usuario finaliza las notificaciones
Cursos alternativos:	<ol style="list-style-type: none"> El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3.1 Si el ingrediente se ha agotado completamente <ol style="list-style-type: none"> El sistema busca los elementos de la carta que usen ese ingrediente y los invalida e informa de ello al usuario <p>*. El usuario cancela la operación</p>

Observaciones:	
Requisitos no funcionales:	

Sagres

Añadir bebida

Nombre del caso:	Añadir bebida
Resumen:	El cocinero jefe desea añadir una o más bebidas al sistema. Introduce los datos correspondientes a cada ingrediente se comprueban si son correctos y se confirma cada operación.
Dependencias:	
Actores:	Metre
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se ha registrado una nueva bebida en el sistema
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el cocinero jefe desea añadir una o más bebidas.2. Mientras que el usuario quiera añadir bebidas:<ol style="list-style-type: none">2.1. El usuario introduce los datos correspondientes a la bebida:<ol style="list-style-type: none">2.1.1 Introduce el nombre2.1.2 Introduce el límite mínimo en stock.2.1.3 Introduce el máximo en stock.2.1.4 Introduce la cantidad actual en stock2.1.5 Introduce una foto.2.1.6 El usuario deberá introducir la cantidad por envase.2.2. El usuario confirma los datos.2.3. El sistema comprueba si los datos introducidos son correctos2.4. Los datos introducidos son correctos. El sistema genera un código de producto y registra todos los datos.2.5. El sistema confirma que se ha añadido la bebida satisfactoriamente.2.6. El sistema pregunta al usuario si desea seguir añadiendo bebidas.3. El usuario finaliza la operación
Cursos alternativos:	<p>2.4b El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3.1</p> <p>*. El usuario cancela la operación.</p>
Observaciones:	
Requisitos no funcionales específicos:	<ul style="list-style-type: none">- Los límites mínimo y máximo vendrán en número de unidades de envase, y la cantidad por envase en mililitros.- La foto deberá estar en formato .jpeg y no sobrepasar los 200 kb

Eliminar bebida

Nombre del caso:	Eliminar bebida
Resumen:	El cocinero jefe desea eliminar una bebida de nuestro sistema. Para ello lo selecciona de una lista de bebidas, y confirma su eliminación. Una vez hecho esto el sistema la elimina y pregunta por si quiere seguir eliminando.
Dependencias:	Incluye al caso de uso “Consultar lista de bebidas”
Actores:	Metre.
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se han eliminado uno o más bebidas del sistema.
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el cocinero jefe desea eliminar una o más bebidas.2. Se inicia el caso de uso “Consultar lista de bebidas”3. Mientras que el usuario quiera eliminar bebidas:<ol style="list-style-type: none">3.1. El usuario selecciona la bebida que quiere eliminar.3.2. El sistema solicita la confirmación de la operación.3.3. El usuario confirma la operación.3.4. El sistema comprueba los elementos de la carta que contienen esta bebida y los invalida.3.5. El sistema elimina la bebida informa de ello al usuario.3.6. El sistema pregunta al usuario si desea eliminar otro elemento4. El usuario finaliza la operación
Cursos alternativos:	<p>3.3b El usuario no confirma la operación. Se vuelve al paso 2.</p> <p>*. El usuario cancela la operación.</p>
Observaciones:	

Modificar bebida

Nombre del caso:	Modificar bebida
Resumen:	El cocinero jefe quiere modificar los datos de unas determinadas bebidas. Una vez que se le muestran los datos, cambia los parámetros que quiera y una vez confirmados y comprobados estos se registra la operación.
Dependencias:	Incluye al caso de uso “Consultar lista de bebidas”
Actores:	Metre
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none">• Se ha registrado uno o más cambios en los detalles de una bebida.
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el cocinero jefe desea modificar uno o mas ingredientes.2. Se inicia el caso de uso “Consultar lista de bebidas”3. Mientras el usuario quiera modificar bebidas:<ol style="list-style-type: none">3.1. El usuario selecciona la bebida que quiere modificar.3.2. El sistema muestra los datos asociados a esa bebida.3.3. Mientras el usuario quiera modificar parámetros:<ol style="list-style-type: none">3.3.1. El usuario modifica el parámetro deseado.3.4. El sistema solicita al usuario que confirme los cambios3.5. El usuario confirma los cambios.3.6. El sistema comprueba si los datos introducidos son correctos3.7. Los datos introducidos son correctos. El sistema registra los cambios e informa que se han realizado satisfactoriamente.3.8. El sistema pregunta al usuario si desea modificar otra bebida4. El usuario finaliza la operación
Cursos alternativos:	<p>3.5b El usuario no confirma los cambios. Se vuelve al paso 3.2</p> <p>3.7b El sistema informa de que los datos introducidos son incorrectos. Se vuelve al paso 3.2.</p> <p>*. El usuario cancela la operación.</p>
Observaciones:	

Notificar incidencia con bebida

Nombre del caso:	Notificar incidencia con Bebida
Resumen:	El usuario 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:	Incluye el caso de uso “Consultar lista de Bebidas”
Actores:	Metre, Cocinero jefe
Precondiciones:	
Postcondiciones:	<ul style="list-style-type: none"> Se ha registrado una incidencia con una bebida
Curso normal:	<ol style="list-style-type: none"> El caso de uso comienza cuando el usuario desea notificar una incidencia ocurrida con una bebida Se inicia el caso de uso “Consultar lista de Bebidas” Mientras el usuario desee notificar incidencias: <ol style="list-style-type: none"> El usuario selecciona la bebida sobre el que desea notificar la incidencia El usuario introduce la cantidad de producto afectado y especifica el motivo de la incidencia El sistema comprueba que los datos introducidos son correctos. Los datos introducidos son correctos. El sistema genera un código asociado a la incidencia y registra la incidencia El sistema resta del stock la cantidad del bebida especificad por el usuario El sistema comprueba la cantidad que queda de bebida en stock <ol style="list-style-type: none"> Si la cantidad actual en stock de la bebida afectada se encuentra por debajo de su mínimo <ol style="list-style-type: none"> El sistema informa de que la cantidad ese producto en stock se encuentra por debajo de su mínimo El sistema informa de que la incidencia se ha registrado correctamente El sistema pregunta al usuario si desea notificar alguna incidencia más El usuario finaliza las notificaciones
Cursos alternativos:	<ol style="list-style-type: none"> El sistema informa de que los datos introducidos no son correctos. Se vuelve al paso 3.1 Si la bebida se ha agotado completamente <ol style="list-style-type: none"> El sistema busca los elementos de la carta que usen esa bebida, los invalida e informa de ello al usuario <p>*. El usuario cancela la operación</p>

<i>Observaciones:</i>	El cocinero jefe podrá notificar incidencias con las bebidas puesto que puede utilizarlas como ingredientes
<i>Requisitos no funcionales:</i>	

Sagres

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:	<ul style="list-style-type: none">• Se ha modificado la cantidad actual en stock de uno o más productos
Curso normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción "Notificar recepción de pedido del proveedor" en el terminal táctil.2. El sistema obtiene los datos del pedido3. El sistema actualiza las cantidades de los productos especificados en el pedido4. El sistema obtiene una lista con los elementos de la carta que están invalidados5. Mientras haya elementos en la lista:<ol style="list-style-type: none">5.1 El sistema comprueba que haya de cada ingrediente asociado al elemento en stock.5.2 Si hay de todos los ingredientes en stock, el sistema habilita el elemento de la carta6. El sistema devuelve la información de los productos actualizados y de los elementos habilitados en caso de que los haya.
Cursos alternativos:	
Observaciones:	El sistema utiliza la copia que posee del pedido para actualizar las cantidades de los productos correspondientes
Requisitos no funcionales:	

Consultar lista de productos

Nombre del caso:	Consultar lista de Productos
Resumen:	Se obtiene una lista con los productos registrados en el sistema, dividida en dos secciones, una la de bebidas y otra la de ingredientes.
Dependencias:	Es incluido por los casos de uso “Añadir Elemento a la Carta”. Incluye a los casos de uso “Consultar lista de bebidas” y “Consultar lista de ingredientes”
Actores:	Cocinero Jefe
Precondiciones:	
Postcondiciones:	
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario desea ver la lista de productos registrados en el sistema2. Se inicia el caso de uso 'Consultar lista de bebidas'3. Se inicia el caso de uso 'Consultar lista de ingredientes'4. El sistema combina las dos listas y muestra una lista completa de todos los productos.
Cursos alternativos:	
Observaciones:	
Requisitos no funcionales específicos:	

Consultar lista de ingredientes

Nombre del caso:	Consultar lista de ingredientes
Resumen:	Se obtiene una lista con los productos registrados en el sistema dependiendo del tipo de usuario que la solicite
Dependencias:	Es incluido por los casos de uso “Eliminar ingrediente”, “Modificar ingrediente”, “Notificar incidencia con ingrediente”, “Añadir Elemento a la Carta” y “Consultar lista de productos”
Actores:	Cocinero jefe
Precondiciones:	
Postcondiciones:	
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario desea ver la lista de productos registrados en el sistema2. El sistema muestra una lista completa de todos los ingredientes registrados
Cursos alternativos:	
Observaciones:	
Requisitos no funcionales específicos:	

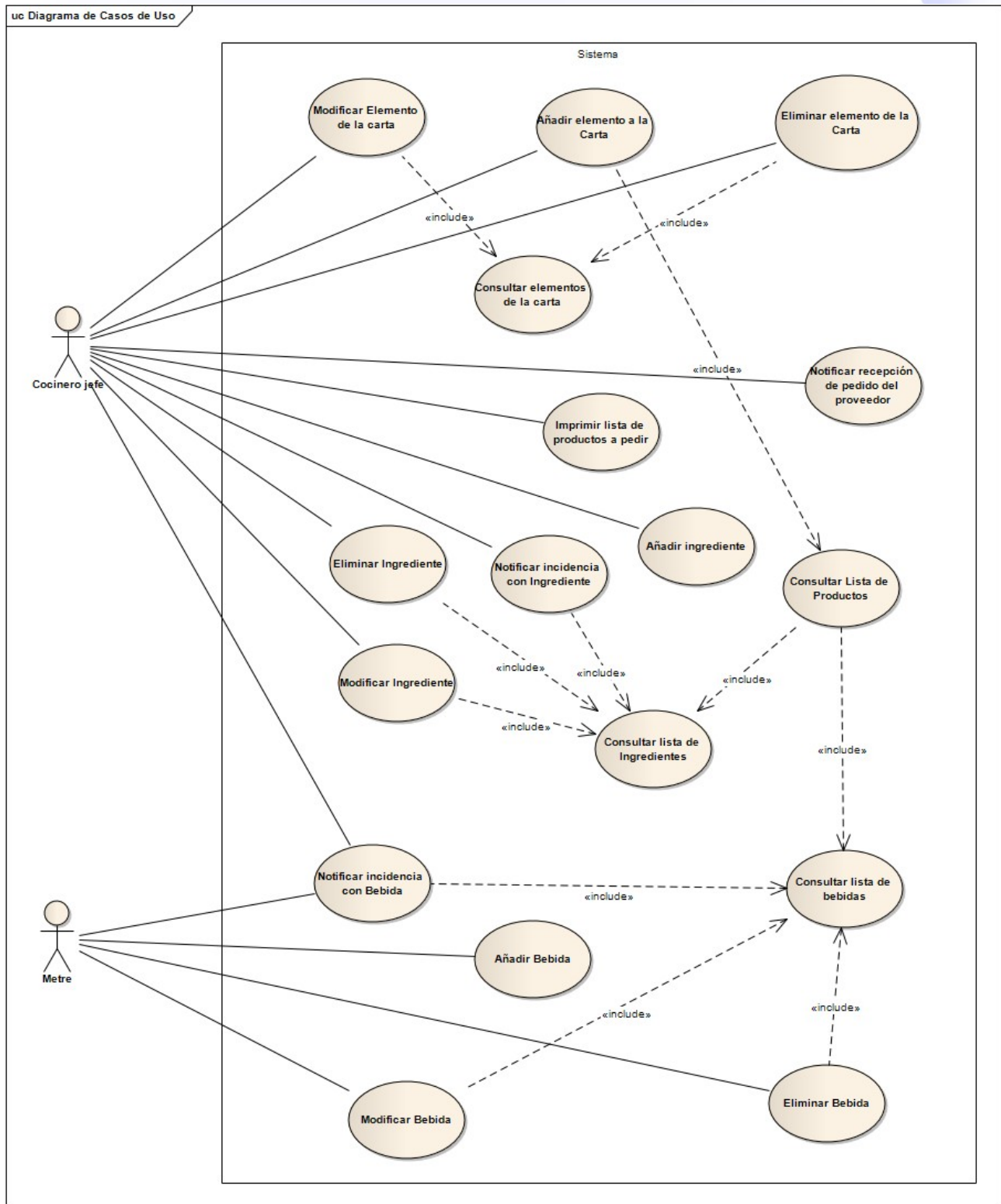
Consultar lista de bebidas

Nombre del caso:	Consultar lista de bebidas
Resumen:	Se obtiene una lista con las bebidas registradas
Dependencias:	Es incluido por los casos de uso “Eliminar bebidas”, “Modificar bebidas”, “Notificar incidencia con bebida”, “Añadir Elemento a la Carta” y “Consultar lista de productos”
Actores:	Metre, Cocinero jefe
Precondiciones:	
Postcondiciones:	
Curso normal:	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario desea ver la lista de bebidas registradas en el sistema2. El sistema muestra una lista completa de todas las bebidas registradas
Cursos alternativos:	
Observaciones:	

Sagres

Elaboración del 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.



2. SUBSISTEMAS FUNCIONALES

Identificación de los subsistemas funcionales

Para gestionar y optimizar todas las funcionalidades del sistema expuestas en el apartado de “Requerimientos Funcionales”, se han dividido 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.

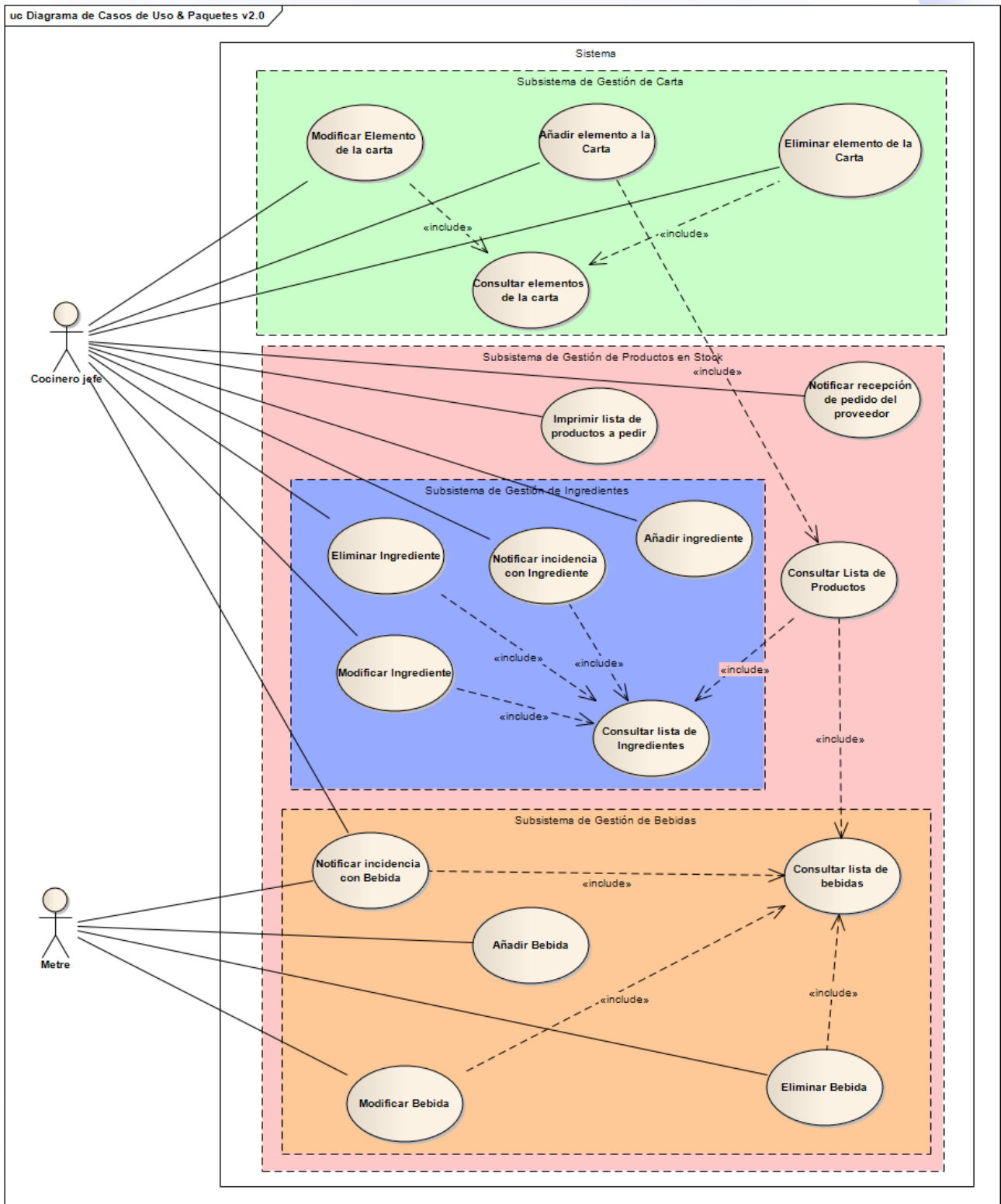
Con esta división de subsistemas y el específico reparto de tareas entre cocinero jefe y metre nos ahorramos tener que implementar un sistema autenticación, ya que debido al escaso número de tipos de usuarios (tan solo 3) sería una pérdida de tiempo.

Sagres

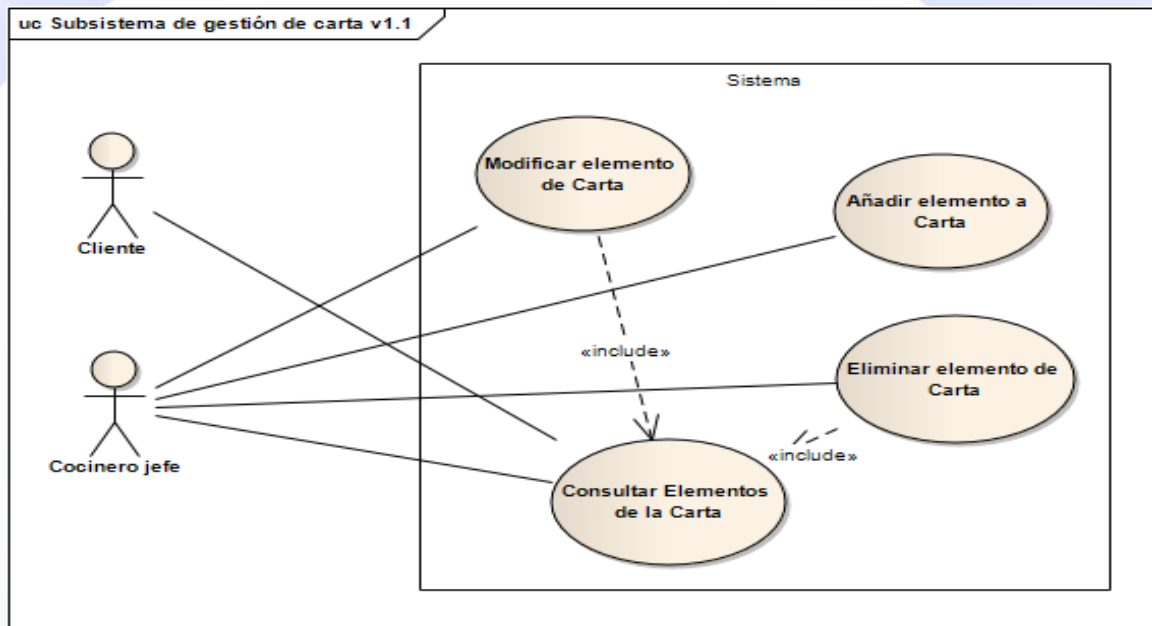
Elaboración del diagrama de paquetes para subsistemas funcionales

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

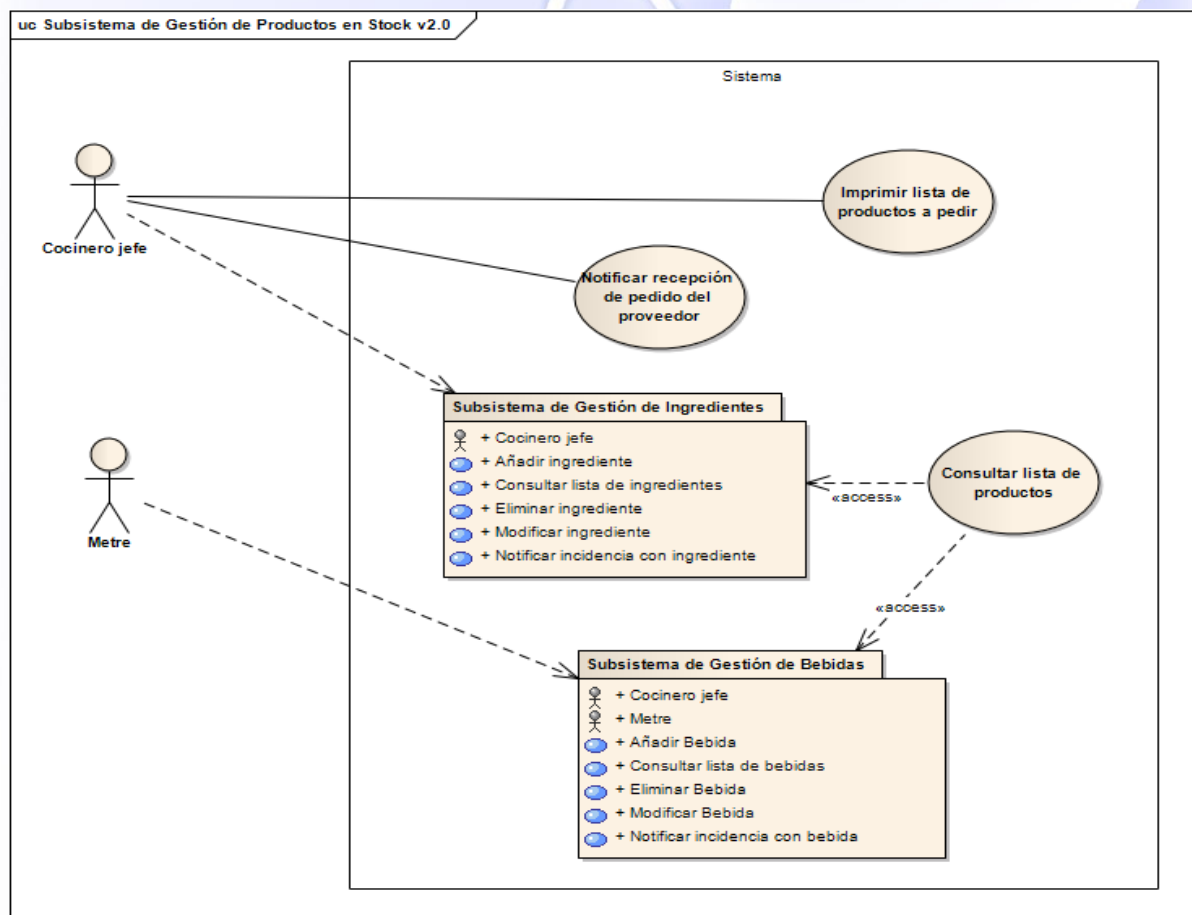
Diagrama de casos de uso global con división en subsistemas



Subsistema de gestión de carta

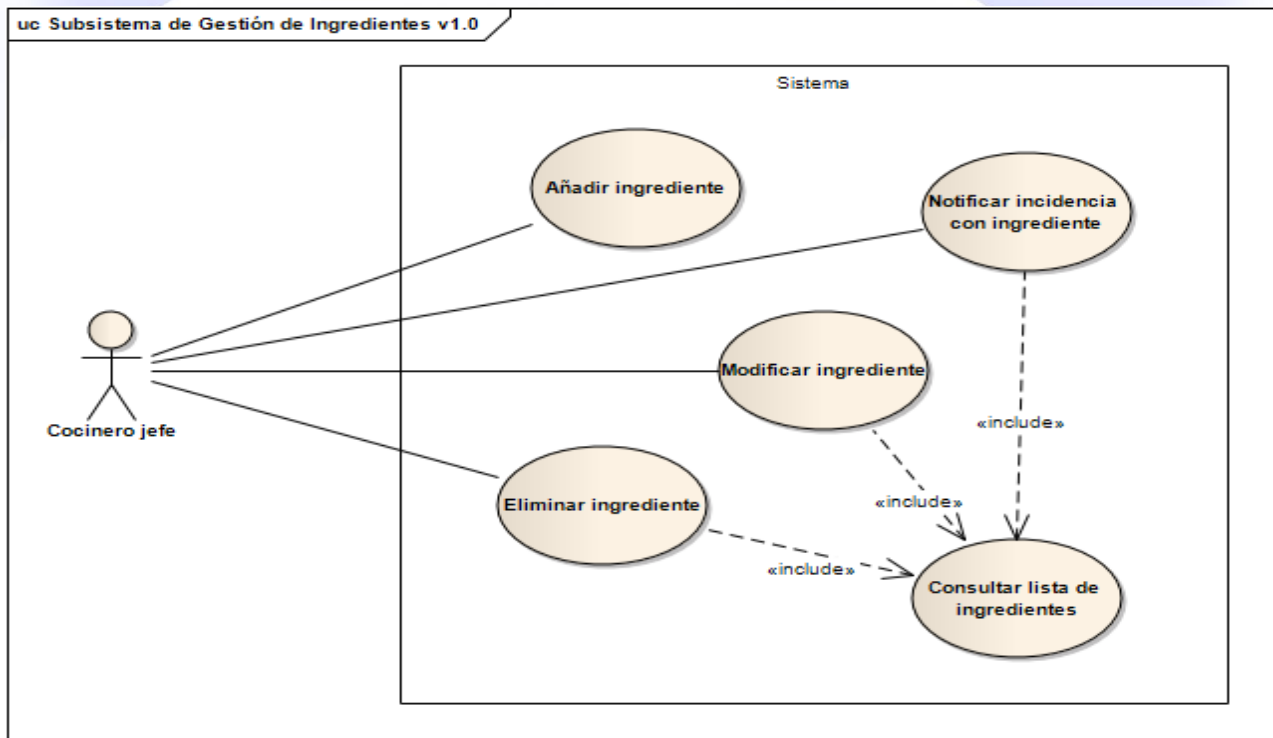


Subsistema de gestión de productos

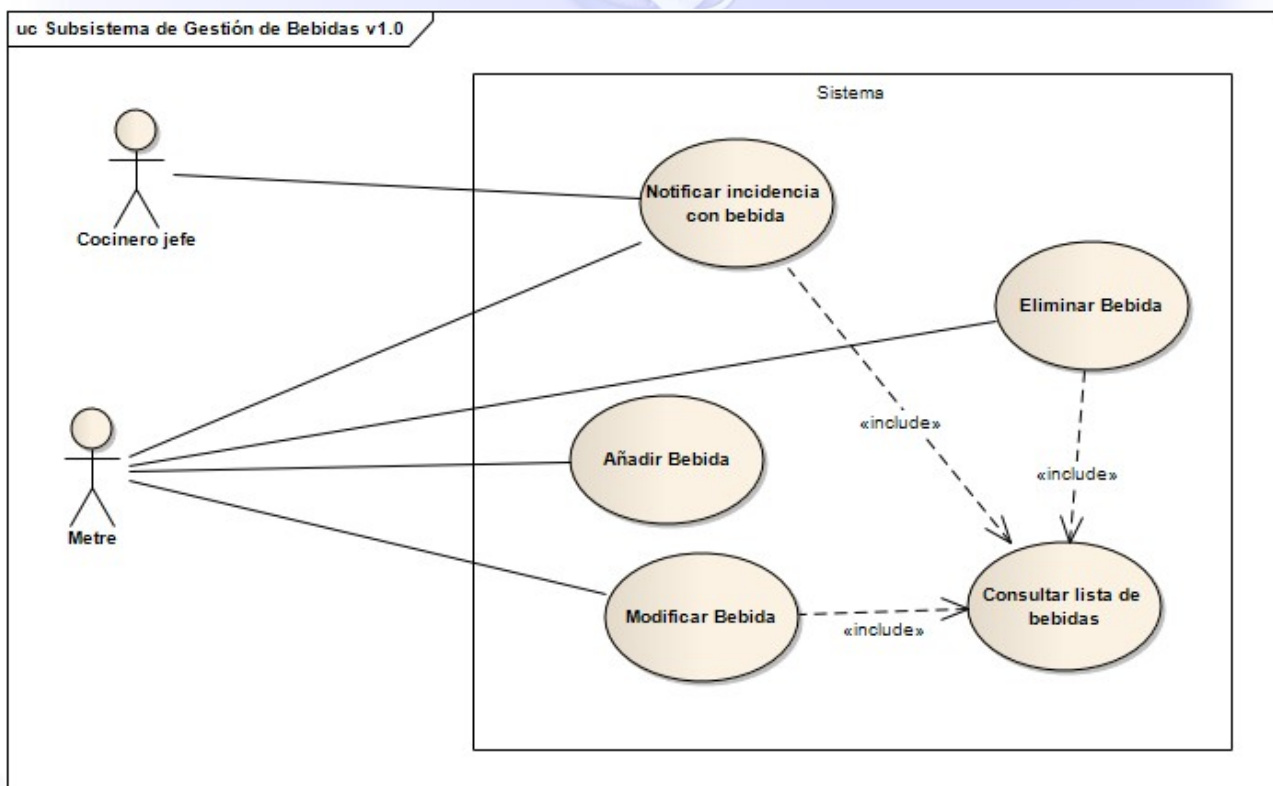


Este sería el diagrama de todo el subsistema de productos, pero como anteriormente comentamos se fracciona en dos nuevos, que son los que veremos a continuación.

Subsistema de gestión de ingredientes



Subsistema de gestión de bebidas



3. REQUISITOS NO FUNCIONALES

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.

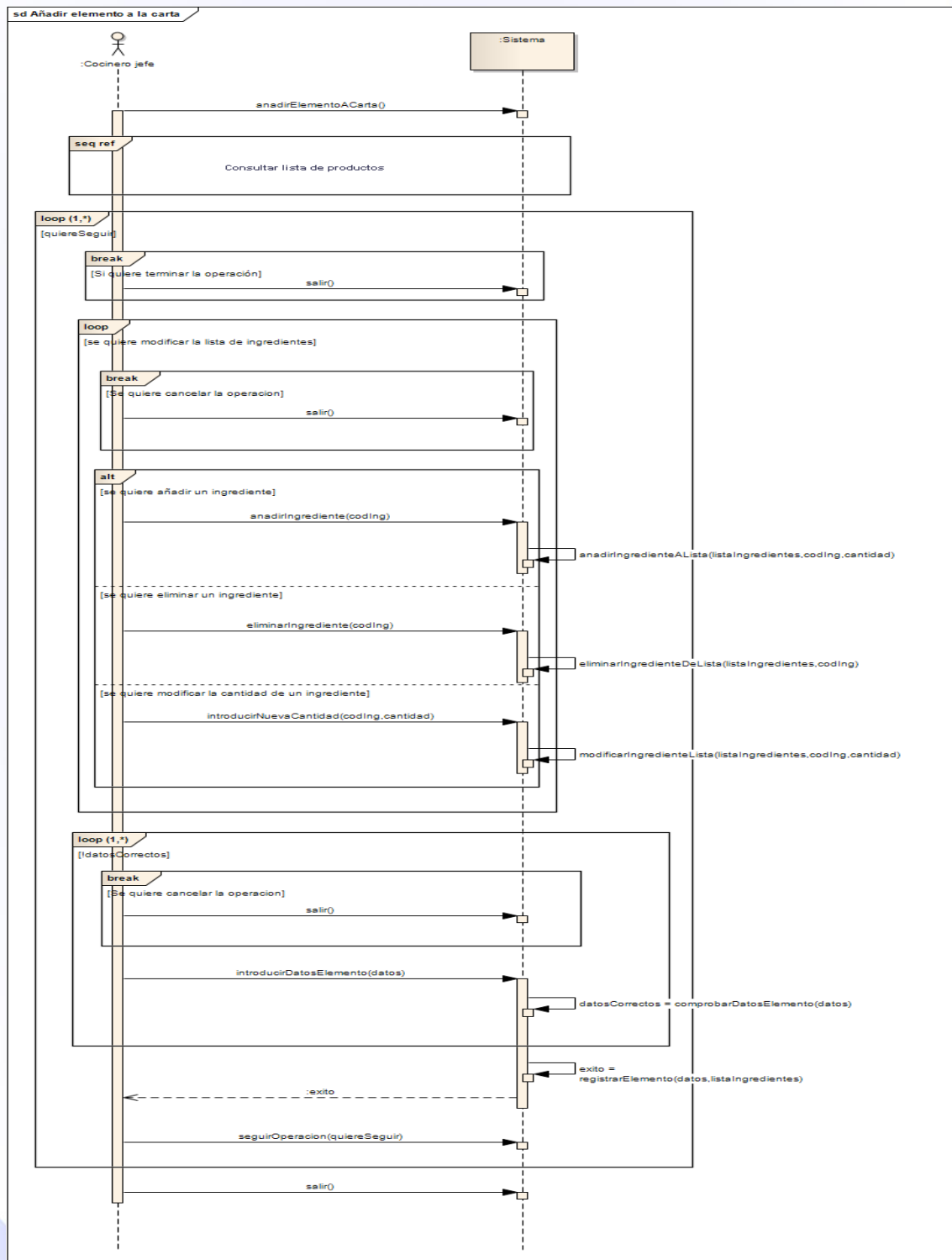
Legales

- 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.

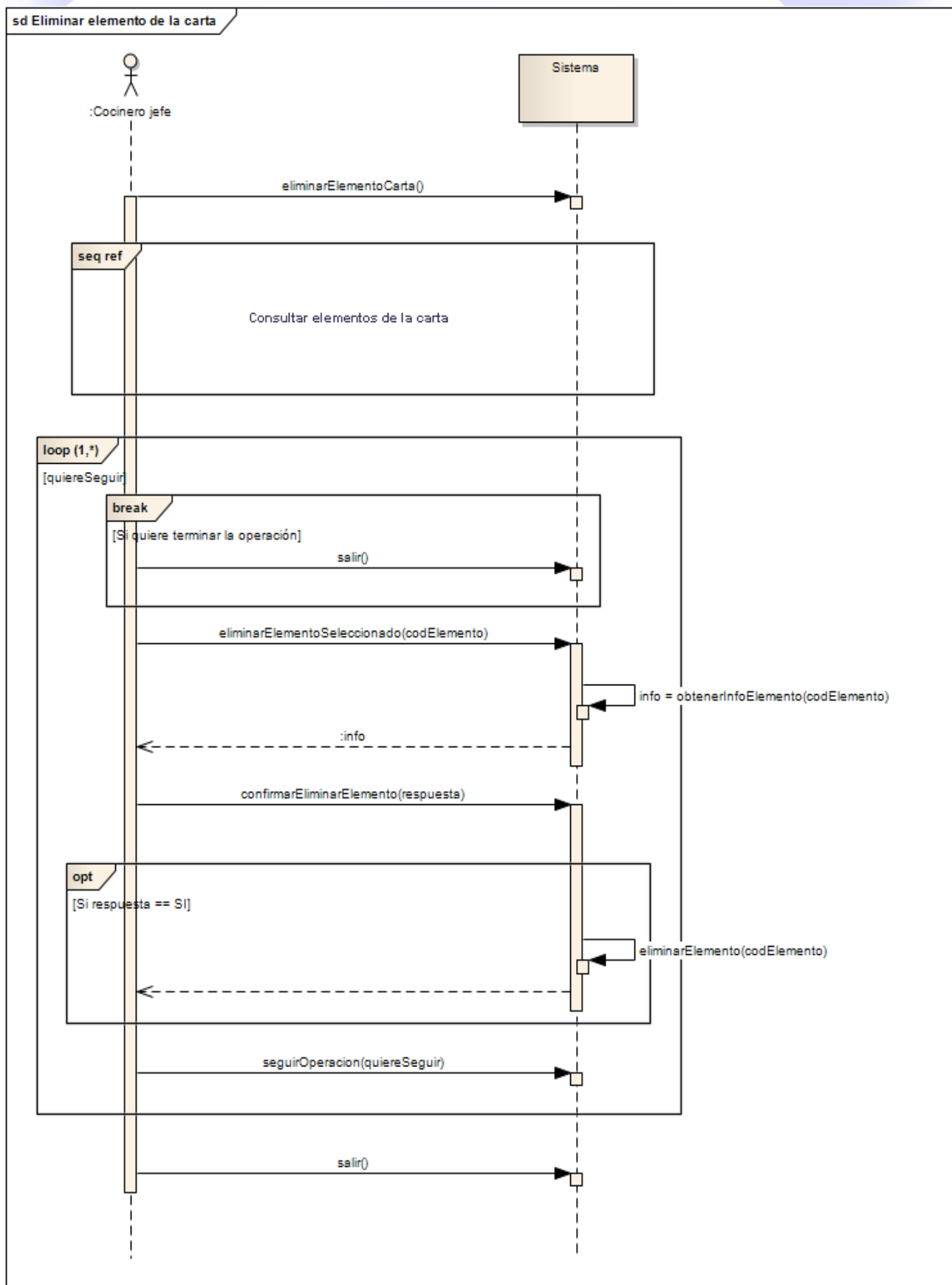
4. OPERACIONES DEL SISTEMA

Elaboración de diagramas de secuencia del sistema

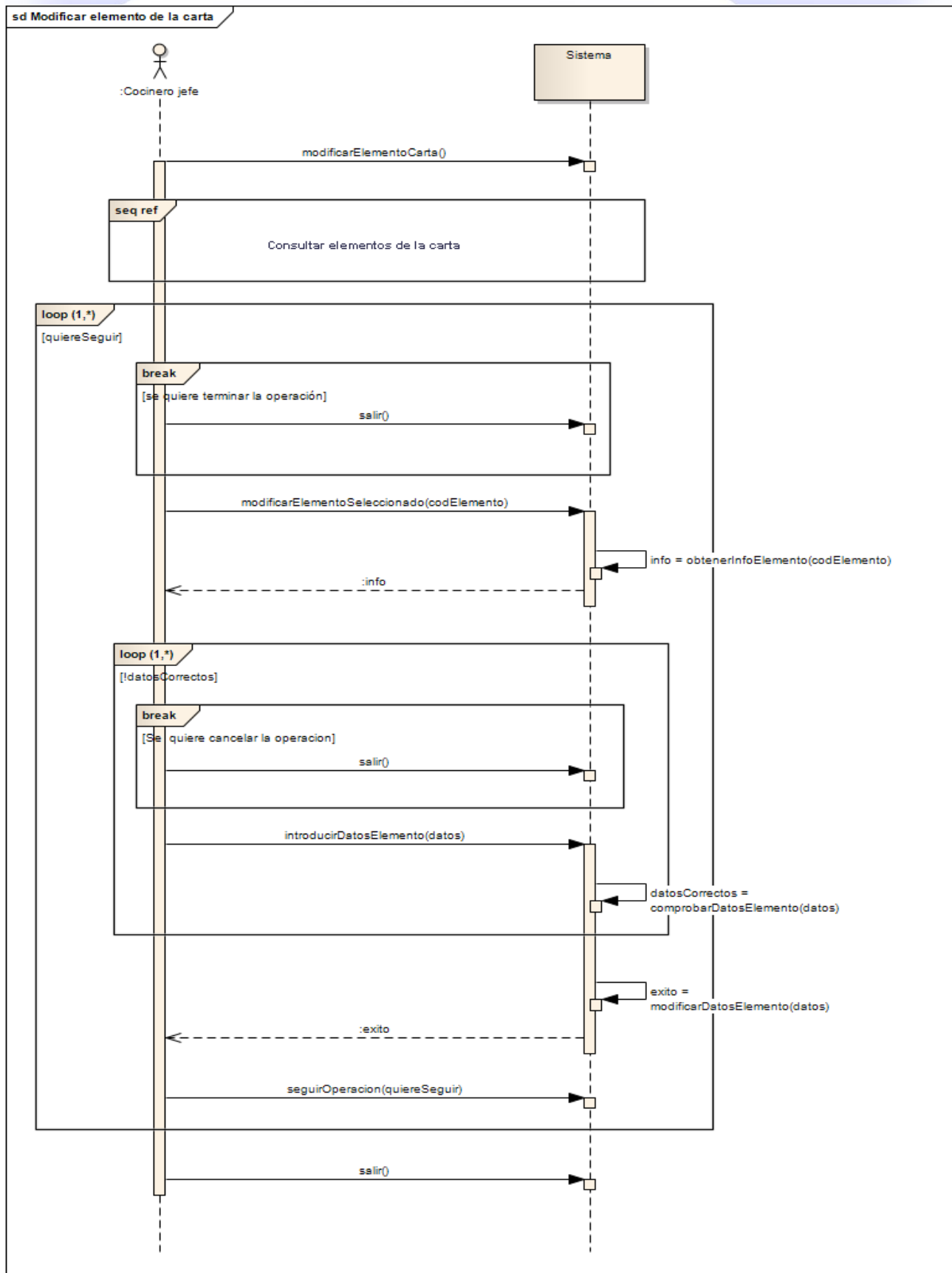
Añadir elemento a la carta



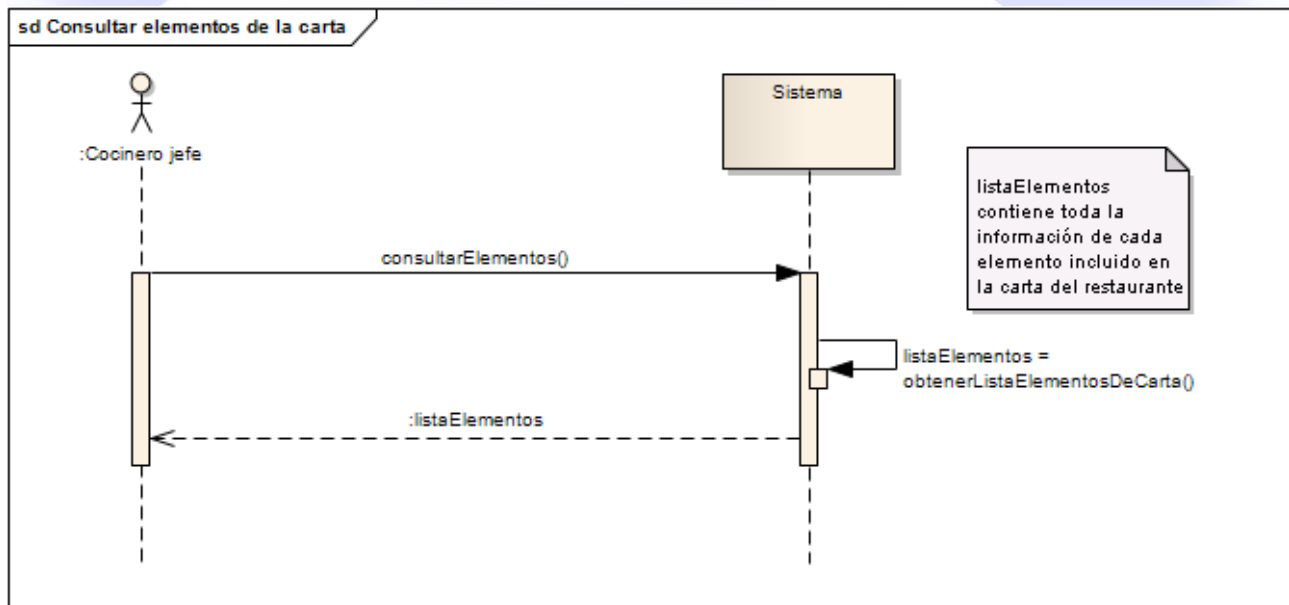
Eliminar elemento de la carta



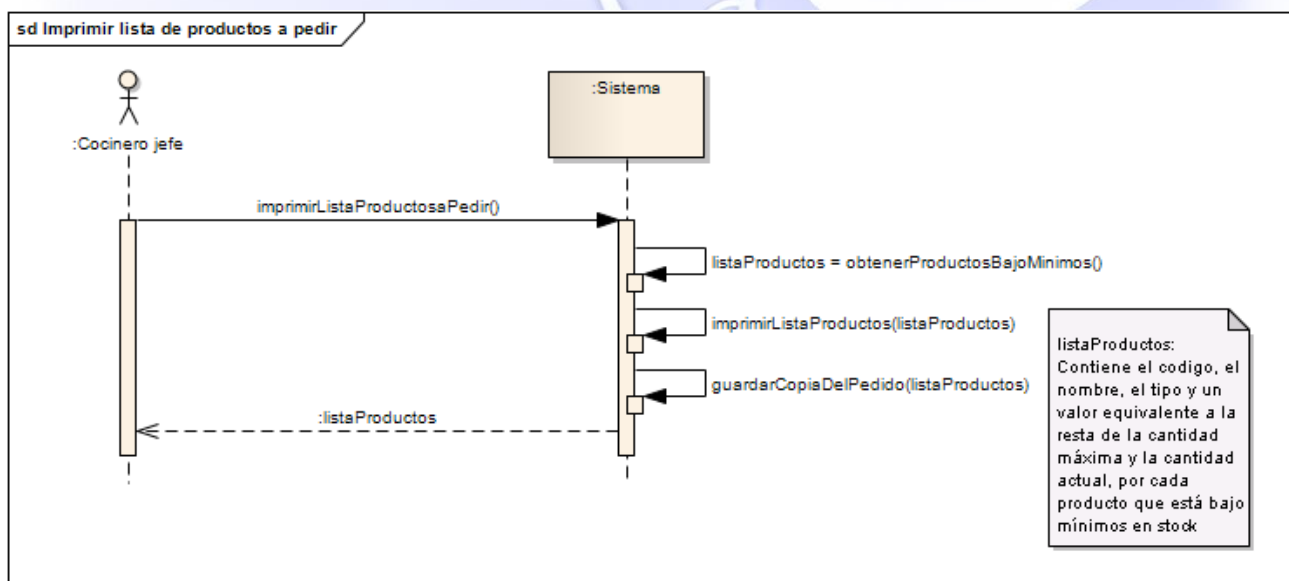
Modificar elemento de la carta



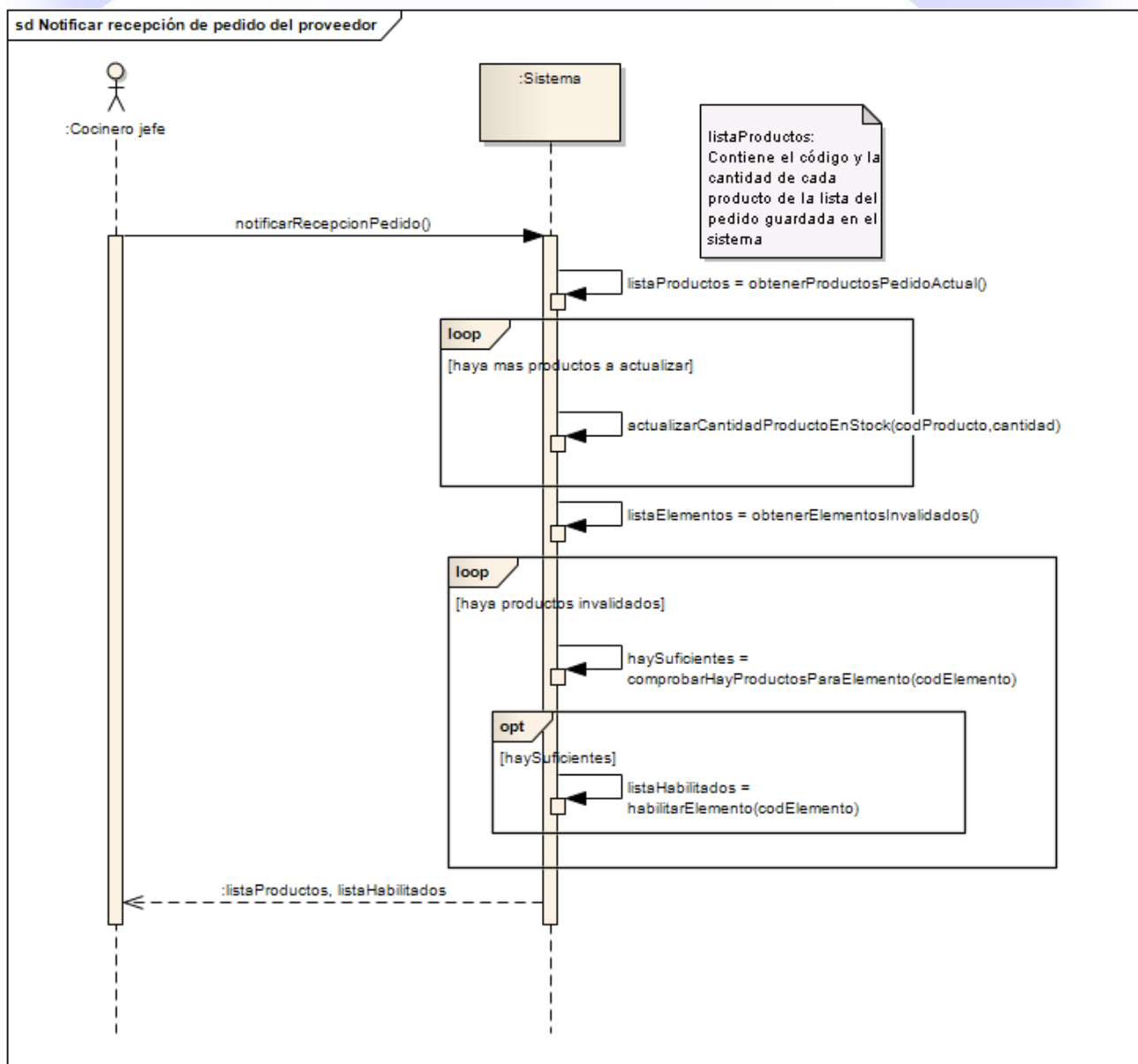
Consultar elementos de la carta



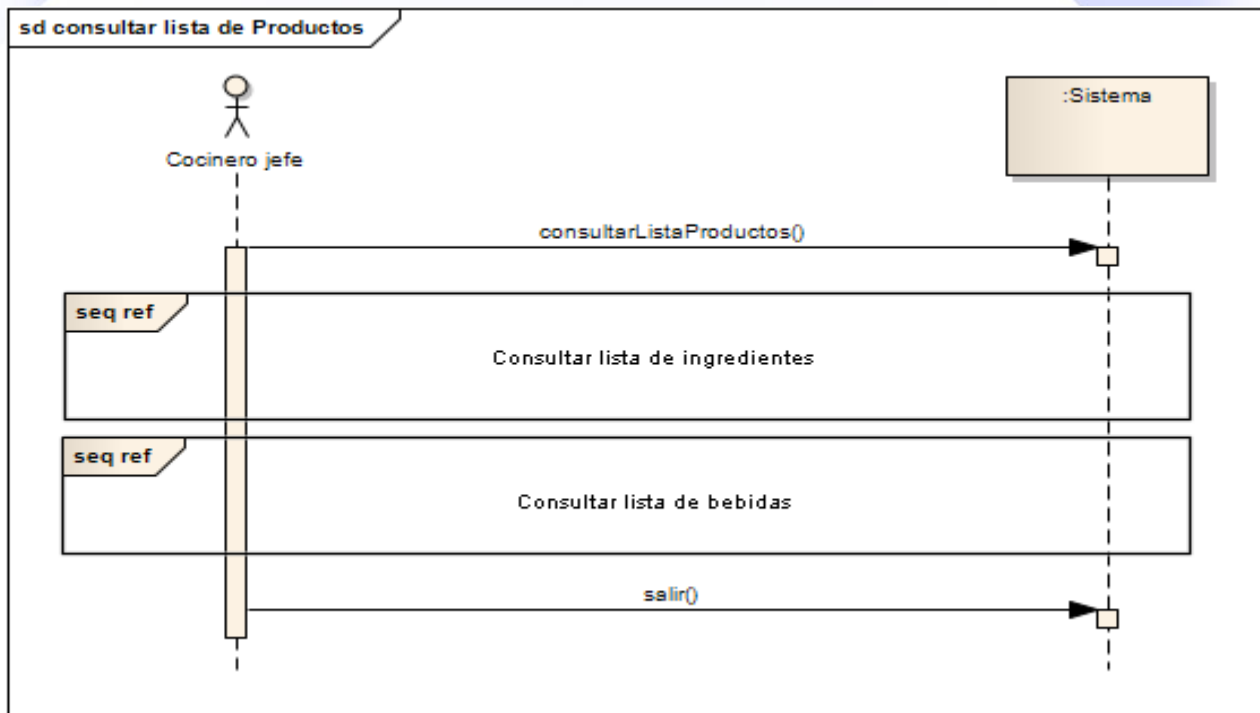
Imprimir lista de productos a pedir



Notificar recepción de pedido del proveedor

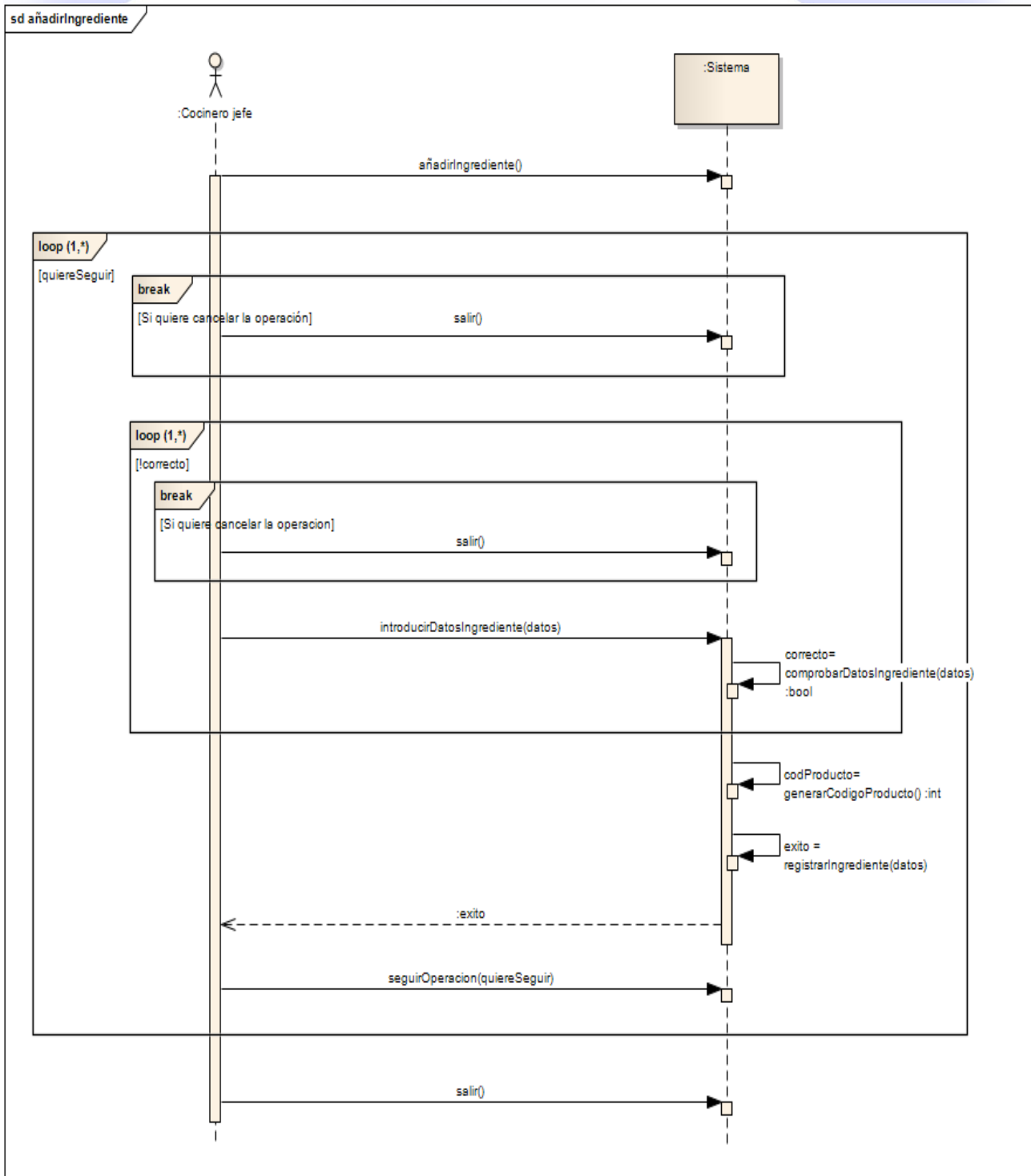


Consultar lista de productos

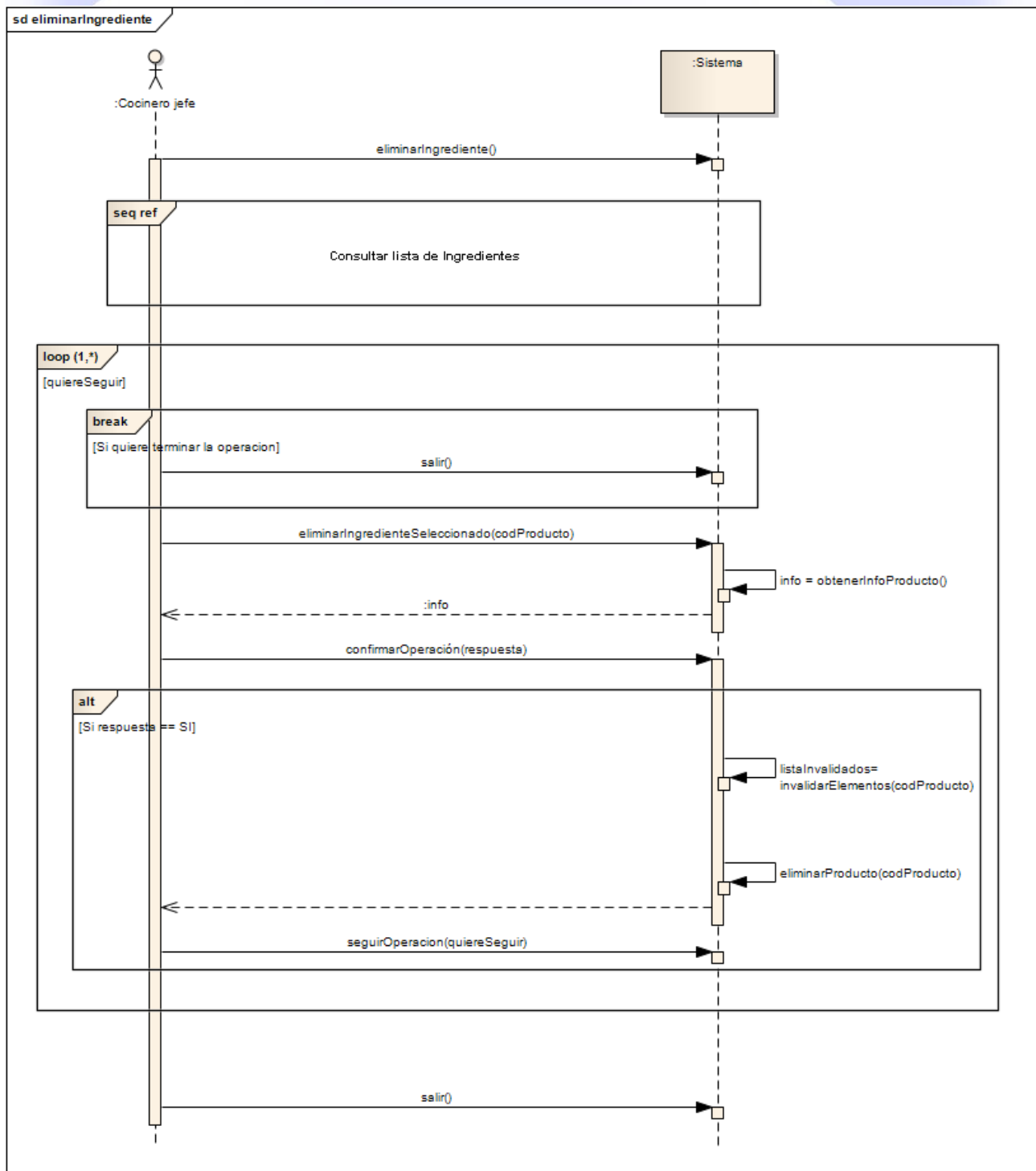


Sagres

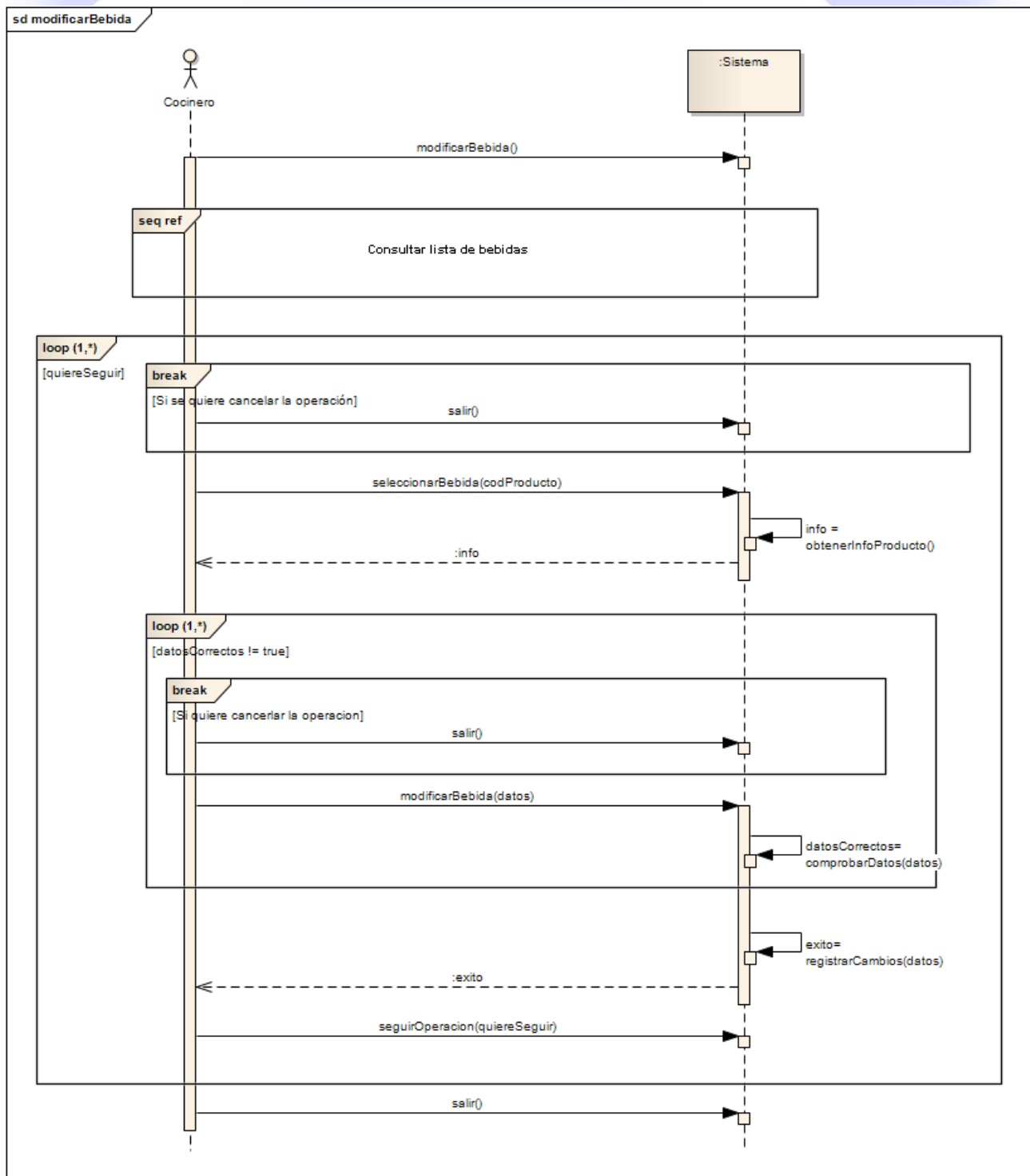
Añadir ingrediente



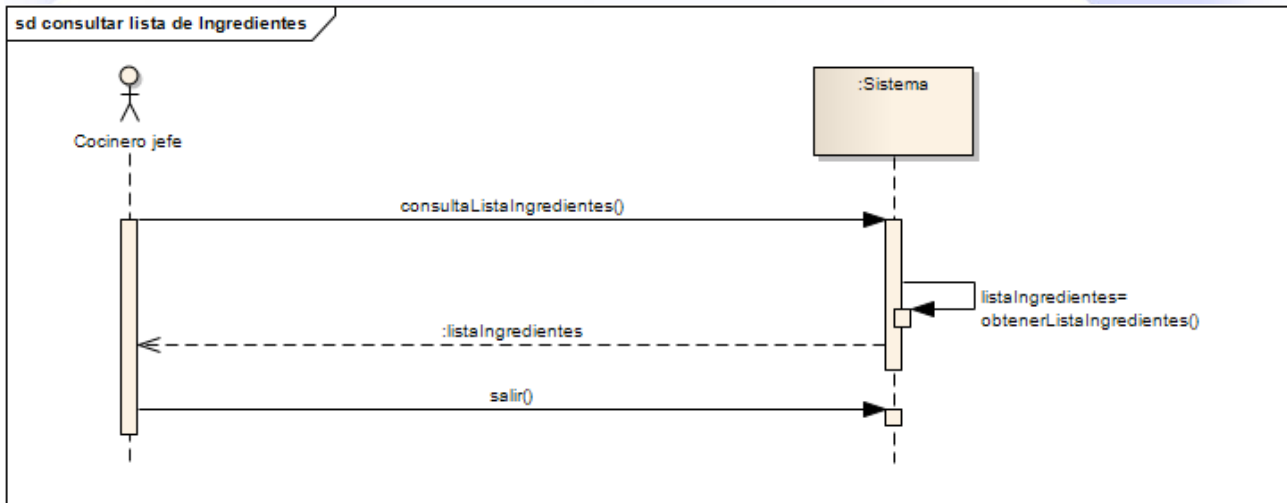
Eliminar ingrediente



Modificar ingrediente

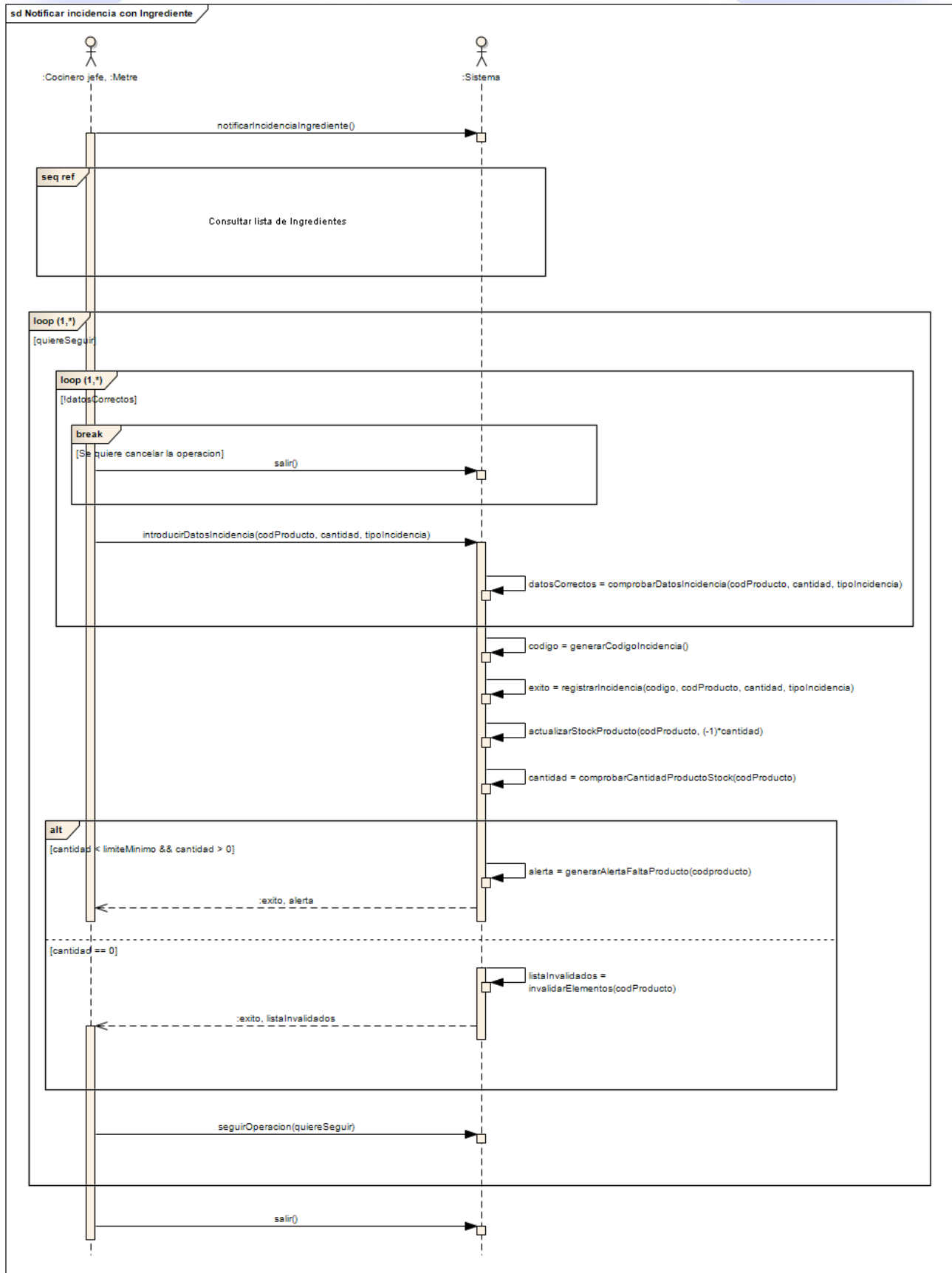


Consultar lista de ingredientes

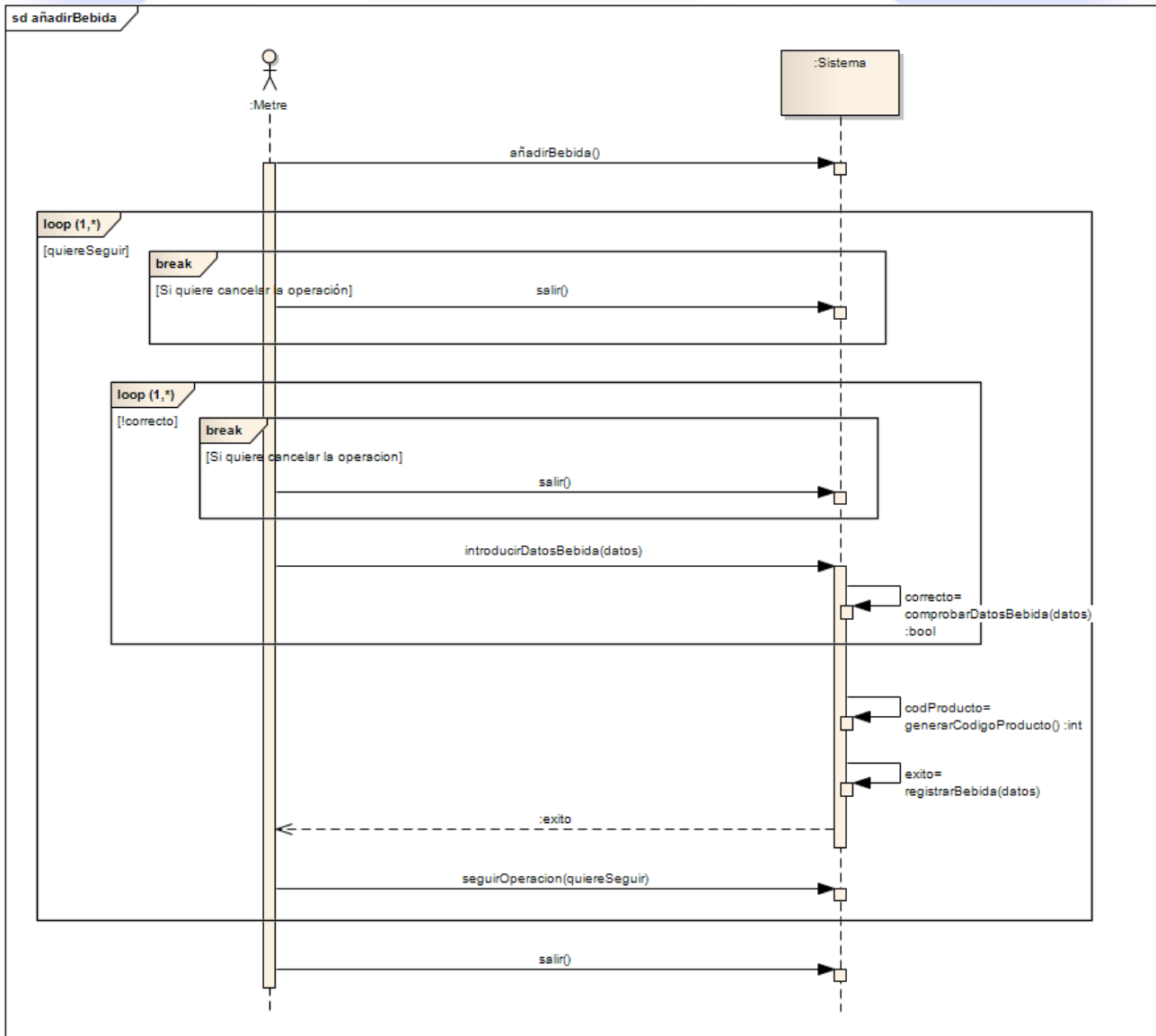


Sagres

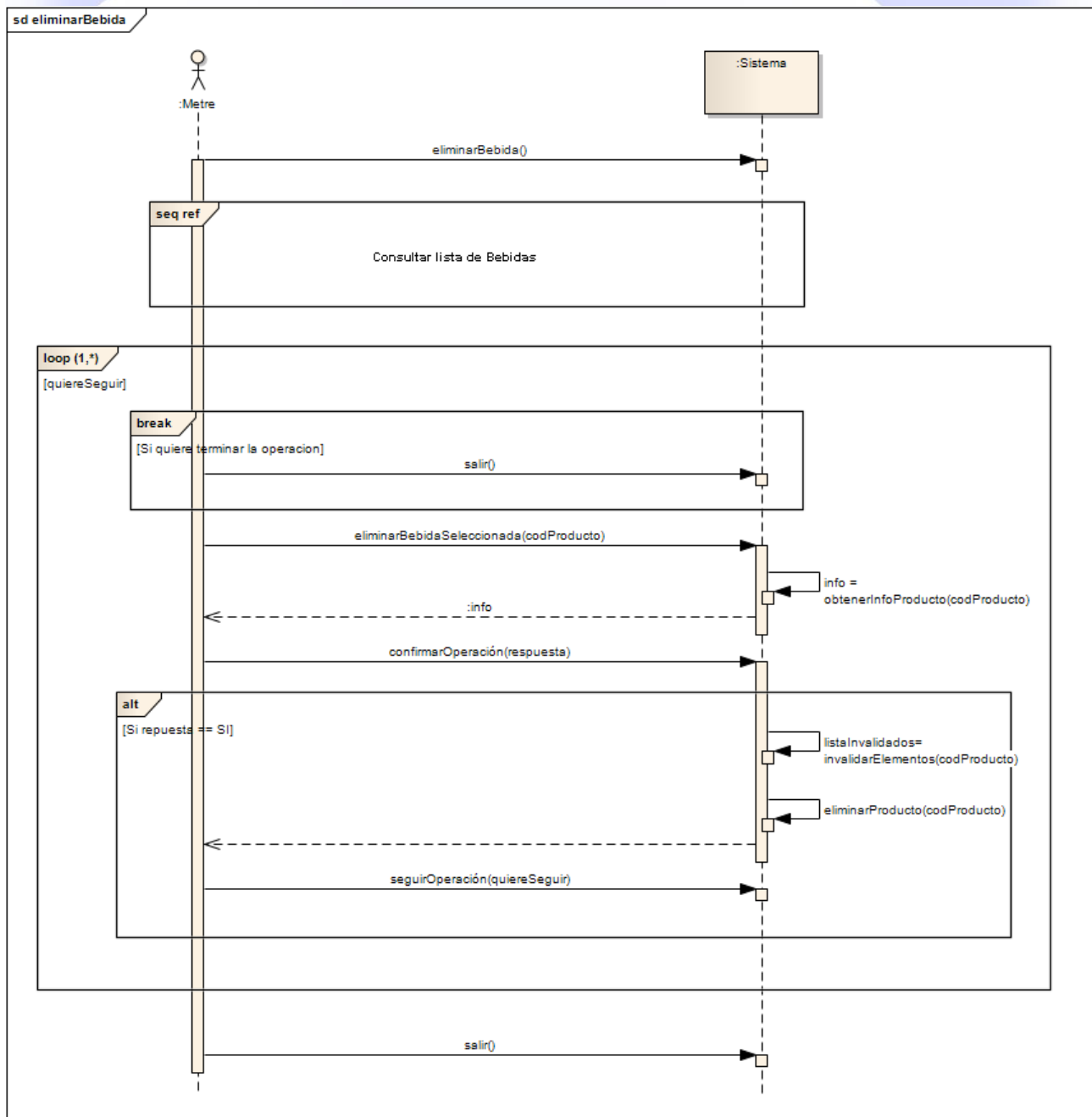
Notificar incidencia con ingrediente



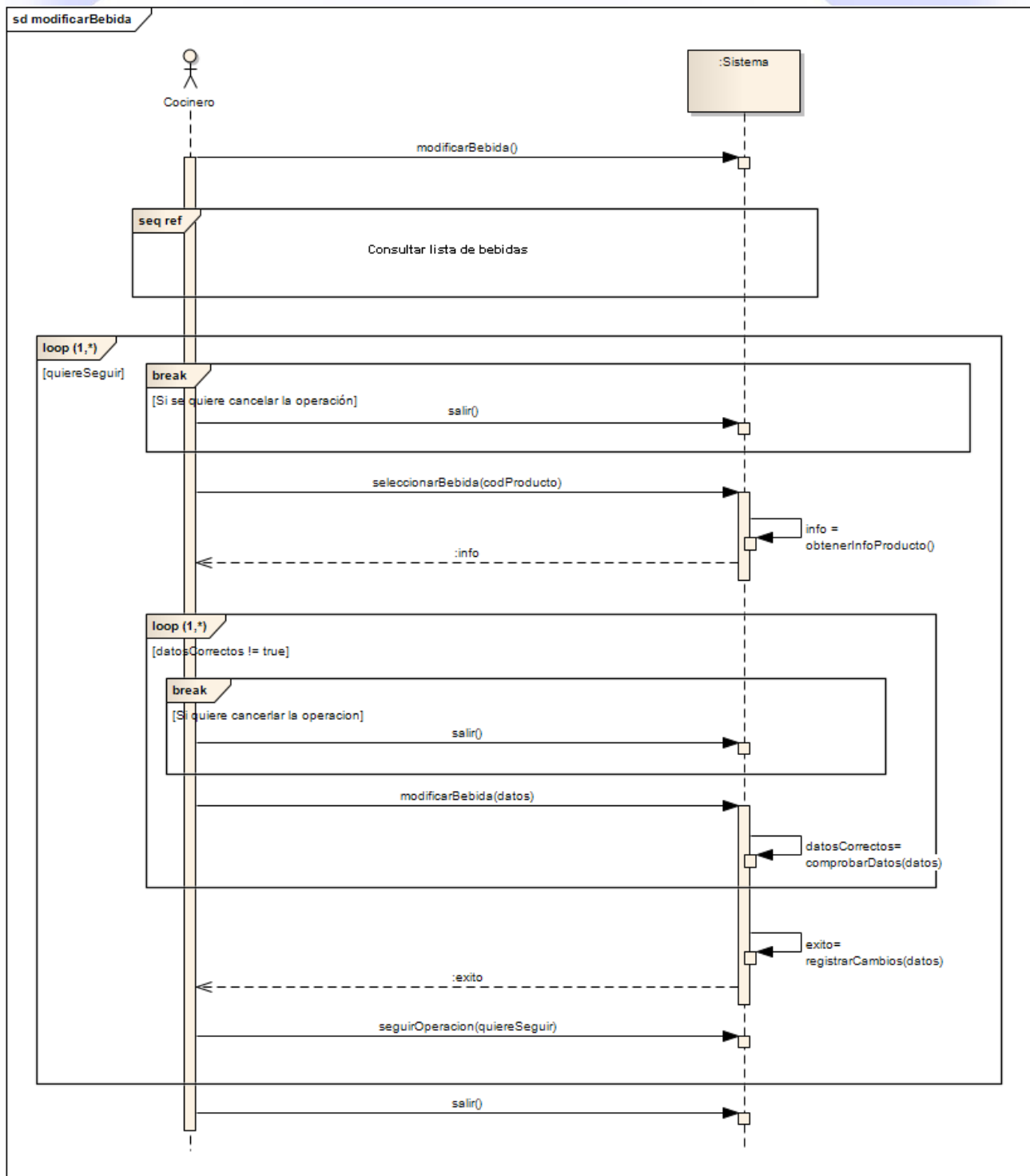
Añadir bebida



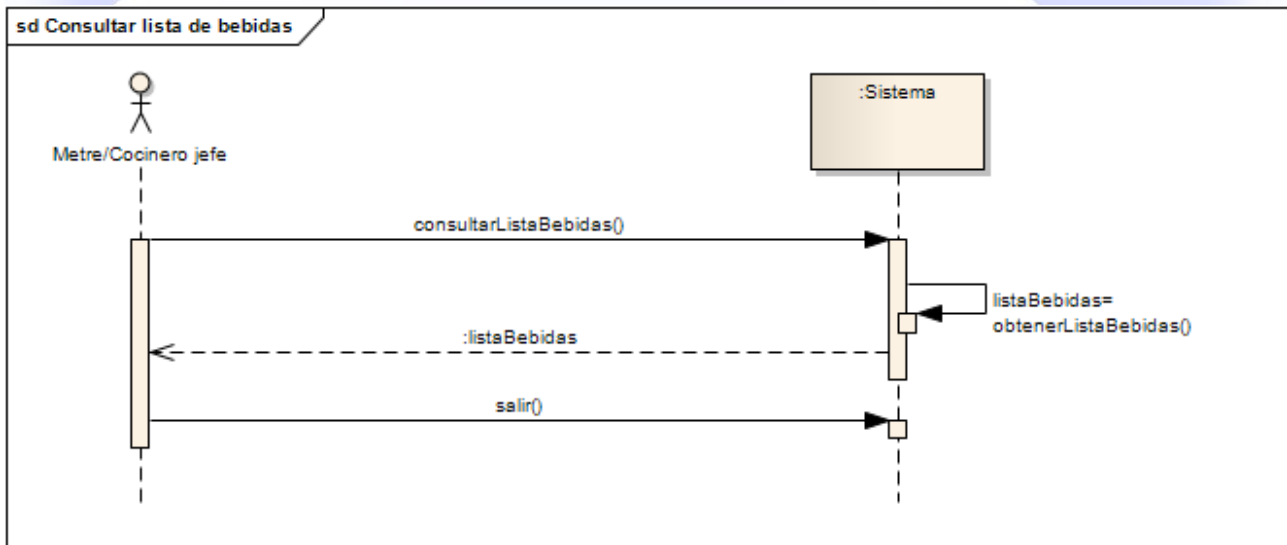
Eliminar bebida



Modificar bebida

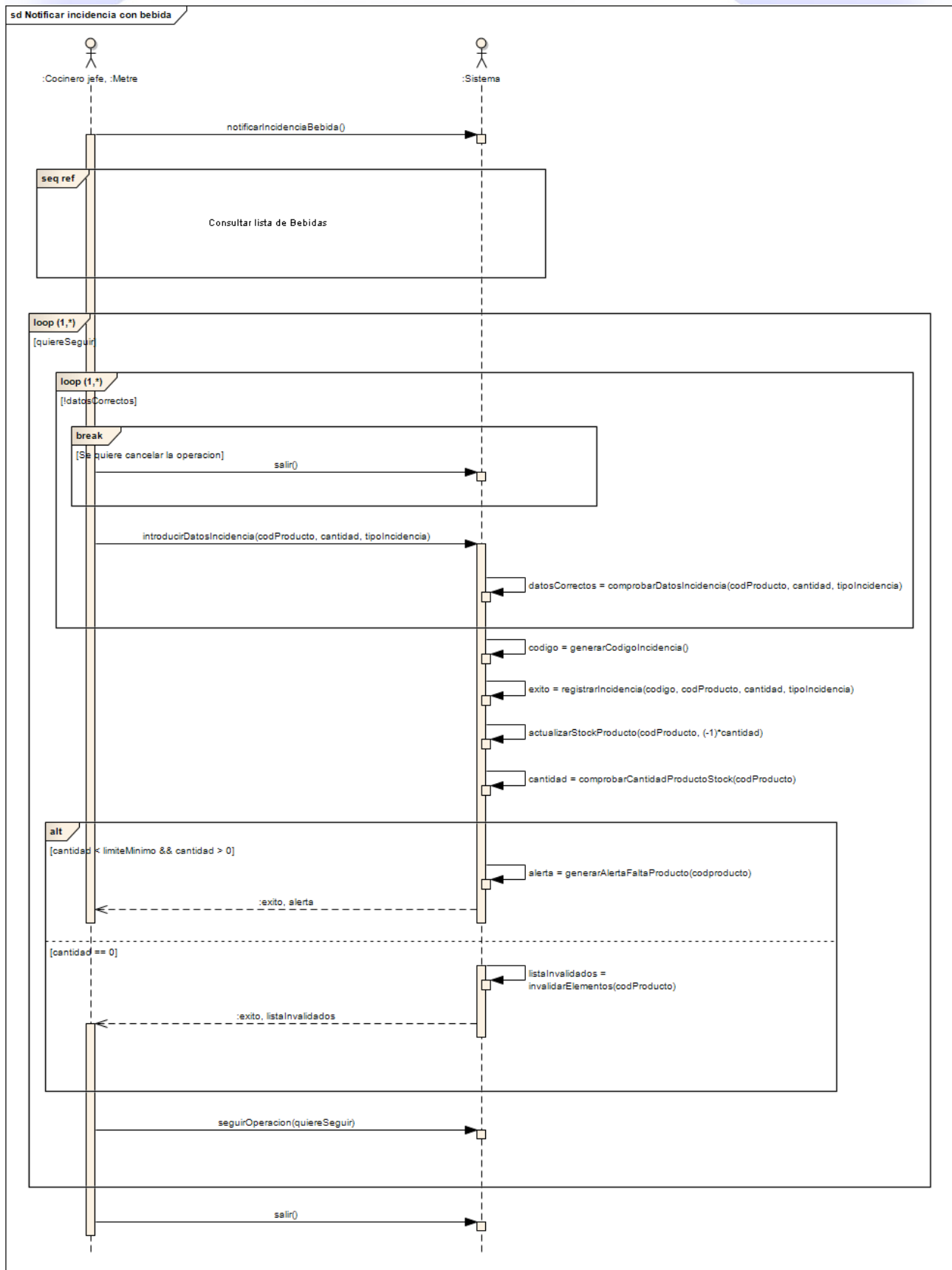


Consultar lista de bebidas



Sagres

Notificar incidencia con bebida



Identificación de las operaciones del sistema

Añadir elemento a la carta

- `añadirElementoACarta()`: operación que inicia el caso de uso para añadir un nuevo elemento a la carta.
- `salir()`: operación con la que el usuario decide finalizar la operación
- `seguirOperacion(quiereSeguir)`: operación con la que el usuario hace saber al sistema si quiere continuar con la operación o no.
- `añadirIngrediente(codIng,cantidad)`: se introduce el código y la cantidad del ingrediente deseado en el sistema
- `añadirIngredienteAlista(listaIngredientes,codIng,cantidad)`: el sistema añade el código y la cantidad del ingrediente a la lista de ingredientes asociada al elemento de la carta que se va a añadir.
- `eliminarIngrediente(codIng)`: se introduce el código del ingrediente que se desea eliminar de la lista de ingredientes
- `eliminarIngredienteDeLista(listaIngredientes,codIng)`: el sistema elimina el ingrediente con código `codIng` de la lista de ingredientes asociados al elemento que se va a añadir
- `introducirNuevaCantidad(codIng,cantidad)`: se introduce el código del ingrediente del cual se desea modificar la cantidad por el valor nueva cantidad.
- `modificarIngredienteDeLista(listaIngredientes,codIng,cantidad)`: el sistema modifica la cantidad del ingrediente con código `codIng` de la lista de ingredientes por el valor especificado en el parámetro `cantidad`.
- `introducirDatosElemento(datos)`: se introducen los datos correspondientes al nuevo elemento. Estos datos son: nombre, descripción, precio, foto, tiempo de preparación y número de raciones en que se divide el elemento.
- `comprobarDatosElemento(datos)`: se comprueba que los datos introducidos sean correctos. Estos datos son: nombre, descripción, precio, foto, tiempo de preparación y número de raciones en que se divide el elemento.
- `registrarElemento(datos,listaIngredientes)`: el sistema crea un nuevo elemento en el sistema y le asocia los datos y la lista de ingredientes.

Eliminar elemento de la carta

- `eliminarElementoCarta()`: operación que inicia el caso de uso para eliminar un elemento de la carta
- `salir()`: operación con la que el usuario decide finalizar la operación
- `seguirOperacion(quiereSeguir)`: operación con la que el usuario hace saber al sistema si quiere continuar con la operación o no.
- `eliminarElementoSeleccionado(codElemento)`: se introduce el código del elemento que se desea eliminar
- `obtenerInfoElemento(codElemento)`: el sistema obtiene toda la información asociada al

elemento con código codElemento

- confirmarEliminarElemento(respuesta): se introduce la confirmación o negación para eliminar el producto finalmente.
- eliminarElemento(codElemento): el sistema elimina el elemento cuyo código es codElemento

Modificar elemento de la carta

- modificarElementoCarta(): operación que inicia el caso de uso para modificar un elemento de la carta
- salir(): operación con la que el usuario decide finalizar la operación
- seguirOperacion(quiereSeguir): operación con la que el usuario hace saber al sistema si quiere continuar con la operación o no.
- modificarElementoSeleccionado(codElemento): se introduce el código del elemento que se desea modificar.
- modificarDatosElemento(datos): se introducen los nuevos datos del elemento. Estos datos son nombre, descripción, precio, foto, tiempo de preparación y el número de raciones en el que se puede dividir el plato.
- obtenerInfoElemento(codElemento): se obtiene toda la información asociada al elemento con código codElemento.
- comprobarDatosElemento(datos): el sistema comprueba si los datos introducidos son correctos.
- modificarDatosElemento(datos): el sistema registra los nuevos cambios en los datos del elemento.

Consultar elementos de la carta

- consultarElementos(): operación que inicia el caso de uso para consultar los elementos de la carta registrados.
- ObtenerListaElementosDeCarta(): el sistema obtiene una lista con los elementos de la carta registrados

Imprimir lista de productos a pedir

- imprimirListaProductosaPedir(): operación que inicia el caso de uso para imprimir la lista de productos a pedir
- obtenerProductosBajoMinimos(): el sistema obtiene una lista de los productos cuya cantidad actual en stock está por debajo de su límite
- imprimirListaProductos(listaProductos): el sistema manda un mensaje a la impresora con la información que debe imprimir.
- guardarCopiaDelPedido(listaProductos): el sistema almacena una copia de la información del pedido

Notificar recepción de pedido del proveedor

- `notificarRecepcionPedido()`: operación que inicia el caso de uso para notificar al sistema que se ha recibido el pedido del proveedor
- `obtenerProductosPedidoActual()`: el sistema recupera la información asociada al pedido realizado
- `actualizarCantidadProductoEnStock(codProducto,cantidad)`: el sistema actualiza la cantidad del producto identificado por `codProducto`. La manera de actualizar la cantidad sera realizando una suma entre la cantidad actual del producto y la cantidad pasada por parámetro Si se pasara un valor de cantidad negativo por parámetro, esto operación de suma se transformaría en una resta.
- `ObtenerElementosInvalidos()`: se obtiene una lista con los elementos de la carta que están invalidados por falta de productos.
- `comprobarHayProductosParaElemento(codElemento)`: se comprueba si hay de todos los ingredientes que necesita el elemento identificado por `codElemento`.
- `habilitarElemento(codElemento)`: se habilita el elemento de la carta identificado por el `codElemento`

Consultar lista de productos

- `iniciarconsultaListaProductos()`: Operación que inicia el caso de uso para consultar la lista de productos.
- `FinalizarconsultarListaProductos()`: Operación que finaliza el caso de uso para consultar la lista de productos.

Añadir ingrediente

- `añadirIngrediente()`: Operación que inicia el caso de uso para añadir un nuevo ingrediente.
- `introducirDatosIngrediente(datos)`: se introduce los datos necesarios para añadir un nuevo ingrediente. Estos son: nombre, mínimo, máximo, nivel actual y foto.
- `comprobarDatosIngrediente(datos)`: Se comprueba que los datos que el usuario ha introducido son correctos, es decir, entran dentro de las limitaciones, son números 'lógicos'...
- `generarCódigoProducto()`: El sistema asigna un código de producto al nuevo ingrediente. Será consecutivo al último que hay.
- `registrarIngrediente(datos)`: Se guarda toda la información correspondiente al nuevo ingrediente en el sistema.
- `seguirOperacion(quiereSeguir)`: el usuario responde al sistema acerca de si desea realizar una nueva operación del mismo tipo.
- `salir()`: Operación que finaliza el caso de uso.

Eliminar ingrediente

- `eliminarIngrediente()`: Operación que inicia el caso de uso para eliminar un ingrediente.
- `eliminarIngredienteSeleccionado(codProducto)`: El usuario selecciona el ingrediente que quiere eliminar con código `codProducto`.

- obtenerInfoProducto(): obtiene información del producto requerido y se lo devuelve al usuario.
- confirmarOperacion(respuesta): el usuario confirma la eliminación del ingrediente seleccionado.
- invalidarElementos(codProducto): invalida los elementos de la carta en los que el ingrediente que queremos eliminar esté siendo usado. Devuelve esta lista de elementos invalidados.
- eliminarProducto(codProducto): elimina definitivamente del sistema el producto con el código codProducto.
- seguirOperacion(quiereSeguir): el usuario responde al sistema acerca de si desea realizar una nueva operación del mismo tipo.
- salir(): Operación que finaliza el caso de uso.

Modificar ingrediente

- modificarIngrediente(): Operación que inicia el caso de uso para modificar un ingrediente.
- seleccionarIngrediente(codProducto): El usuario selecciona el ingrediente con el código codProducto.
- obtenerInfoProducto(): obtiene información del producto requerido y se lo devuelve al usuario.
- modificarIngrediente(datos): el usuario modifica los datos del ingrediente. Datos son los datos una vez modificados, es decir, los datos nuevos.
- comprobarDatos(datos): Se comprueba que los datos que el usuario ha introducido son correctos, es decir, entran dentro de las limitaciones, son números 'lógicos'...
- registrarCambios(datos): El sistema guarda definitivamente los cambios en los datos del ingrediente.
- seguirOperacion(quiereSeguir): el usuario responde al sistema acerca de si desea realizar una nueva operación del mismo tipo.
- salir(): Operación que finaliza el caso de uso.

Notificar incidencia con ingrediente

- notificarIncidenciaIngrediente(): operación que inicial el caso de uso para notificar una incidencia con un ingrediente
- salir(): operación con la que el usuario decide finalizar la operación
- seguirOperacion(quiereSeguir): operación con la que el usuario hace saber al sistema si quiere continuar con la operación o no.
- introducirDatosIncidencia(codProducto,cantidad,tipoIncidencia): se introducen los datos de la incidencia. En este caso los datos son el código de producto, la cantidad de producto implicado y el tipo de incidencia.

- `comprobarDatosIncidencia()`: el sistema comprueba que los datos introducidos sean correctos.
- `generarCodigoIncidencia()`: el sistema genera un código para la incidencia
- `registrarIncidencia(codigo,codProducto,cantidad,tipoIncidencia)`: el sistema registra la incidencia
- `actualizarStockProducto(codProducto,cantidad)`: el sistema actualiza la cantidad del producto identificado por `codProducto`. La manera de actualizar la cantidad sera realizando una suma entre la cantidad actual del producto y la cantidad pasada por parámetro Si se pasara un valor de cantidad negativo por parámetro, esto operación de suma se transformaría en una resta.
- `comprobarCantidadProductoStock()`: el sistema comprueba si la cantidad del producto actual esta por debajo o por encima del limite, o si ya no queda nada en stock.
- `generarAlertaFaltaProducto(codProducto)`: el sistema genera un mensaje de alerta informando al usuario de que la cantidad actual en stock del producto identificado por `codProducto` ha superado su límite mínimo.
- `invalidarElementos(codProducto)`: se invalidan los elementos de la carta que tengan asociados el producto identificado por `codProducto`

Consultar lista de ingredientes

- `consultarListaIngredientes()`: Operación que inicia el caso de uso para consultar la lista de ingredientes.
- `obtenerListaIngredientes()`: obtiene una lista de ingredientes registrados en el sistema
- `salir()`: Operación que finaliza el caso de uso.

Añadir bebida

- `añadirBebida()`: Operación que inicia el caso de uso para añadir una nueva bebida.
- `introducirDatosBebida(datos)`: se introduce los datos necesarios para añadir una nueva bebida: nombre, mínimo, máximo, nivel actual, foto y cantidad por Envase.
- `comprobarDatosBebida(datos)`: Se comprueba que los datos que el usuario ha introducido son correctos, es decir, entran dentro de las limitaciones, son números 'lógicos'...
- `generarCodigoProducto()`: El sistema asigna un código de producto a la nueva bebida. Será consecutivo al último que hay.
- `registrarBebida(datos)`: Se guarda toda la información correspondiente a la nueva bebida en el sistema.
- `seguirOperacion(quiereSeguir)`: el usuario responde al sistema acerca de si desea realizar una nueva operación del mismo tipo.
- `salir()`: Operación que finaliza el caso de uso.

Eliminar bebida

- `eliminarBebida()`: Operación que inicia el caso de uso para eliminar una bebida.
- `eliminarBebidaSeleccionada(codProducto)`: El usuario selecciona la bebida con el código `codProducto`.
- `ObtenerInfoProducto()`: obtiene información del producto requerido y se lo devuelve al usuario.
- `confirmarOperación(respuesta)`: el usuario confirma la operación.
- `invalidarElementos()`: invalida los elementos de la carta en los que la bebida que queremos eliminar esté siendo usada. Devuelve una lista de elementos invalidados.
- `eliminarProducto(codProducto)`: elimina definitivamente del sistema el producto con el código `codProducto`.
- `seguirOperacion(quiereSeguir)`: el usuario responde al sistema acerca de si desea realizar una nueva operación del mismo tipo.
- `salir()`: Operación que finaliza el caso de uso.

Modificar bebida

- `modificarBebida()`: Operación que inicia el caso de uso para modificar una bebida.
- `seleccionarBebida(codProducto)`: El usuario selecciona la bebida con el código `codProducto`.
- `obtenerInfoProducto()`: obtiene información del producto requerido y se lo devuelve al usuario.
- `modificarBebida(datos)`: el usuario modifica los datos de la bebida. Datos son los datos una vez modificados, es decir, los datos nuevos.
- `comprobarDatos(datos)`: Se comprueba que los datos que el usuario ha introducido son correctos, es decir, entran dentro de las limitaciones, son números 'lógicos'...
- `registrarCambios(datos)`: El sistema guarda definitivamente los cambios en los datos del ingrediente.
- `seguirOperacion(quiereSeguir)`: el usuario responde al sistema acerca de si desea realizar una nueva operación del mismo tipo.
- `salir()`: Operación que finaliza el caso de uso.

Consultar lista de bebidas

- `consultarListaBebidas()`: Operación que inicia el caso de uso para consultar la lista de bebidas.
- `obtenerListaBebidas()`: obtiene una lista de bebidas registradas en el sistema
- `salir()`: Operación que finaliza el caso de uso.

Notificar incidencia con bebida

- `iniciarNotificarIncidenciaBebida()`: Operación que inicia el caso de uso para notificar una incidencia con una bebida.
- `introducirDatosIncidencia(codProducto, cantidad, tipoIncidencia)`: se introducen los datos de la incidencia. En este caso los datos son el código de producto, la cantidad de producto implicado y el tipo de incidencia.
- `ComprobarDatosIncidencia()`: el sistema comprueba que los datos introducidos sean correctos.
- `GeneraCodigoIncidencia()`: el sistema genera un código para la incidencia.
- `registrarIncidencia(codigo,codProducto,cantidad,tipoIncidencia)`: el sistema registra la incidencia
- `actualizarStockProducto(codProducto,cantidad)`: el sistema actualiza la cantidad del producto identificado por `codProducto`. La manera de actualizar la cantidad sera realizando una suma entre la cantidad actual del producto y la cantidad pasada por parámetro Si se pasara un valor de cantidad negativo por parámetro, esto operación de suma se transformaría en una resta.
- `comprobarCantidadProductoEnStock()`: el sistema comprueba si la cantidad del producto actual esta por debajo o por encima del limite.
- `generarAlertaFaltaDeProducto(codProducto)`: el sistema genera un mensaje de alerta informando al usuario de que la cantidad actual en stock del producto identificado por `codProducto` ha superado su límite mínimo.
- `invalidarElementos(codProducto)`: se invalidan los elementos de la carta que tengan asociados el producto identificado por `codProducto`
- `seguirOperacion(quiereSeguir)`: el usuario responde al sistema acerca de si desea realizar una nueva operación del mismo tipo.
- `salir()`: Operación que finaliza el caso de uso.

APÉNDICE 1.1

<i>Fecha</i>	16/03/10
<i>Descripción del problema</i>	<p>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 echo de que los diagramas de secuencia se hayan repartido a provocado que aparezcan distintas denominaciones para operaciones del sistema que realmente tienen la misma funcionalidad.</p> <p>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 dependen de él.</p>
<i>Impacto del problema</i>	<p>Los problemas mencionados en la descripción del problema son concretamente que en algunos casos de uso realizabamos demasiadas llamadas a operaciones de consulta. Esto podría resultar perjudicial para la eficiencia de nuestra aplicación.</p> <p>El echo de que haya operaciones con distinto nombre pero igual funcionalidad iba a suponer un coste tremendo para nosotros en primer lugar porque hubieramos 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.</p> <p>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.</p>
<i>Soluciones adoptadas</i>	<ul style="list-style-type: none">• 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 en las especificaciones de los casos de uso involucrados como en los diagramas de secuencia de esos casos de uso.
<i>Anexos a la versión</i>	