

# Documento de ANÁLISIS

Sagres

Sistema de Administración y  
Gestión de RESTaurantes



Gaspar Muñoz Soria  
Samuel Guirado Navarro  
Carlos Moreno Muñoz

V1.0

## ÍNDICE DE CONTENIDO

Apartado de control de versiones.....	3
1. Modelo estático.....	4
<i>Descripción de clases conceptuales.....</i>	<i>4</i>
<i>Relaciones entre clases conceptuales.....</i>	<i>5</i>
<i>Atributos asociados a clases y/o relaciones.....</i>	<i>6</i>
<i>Diagrama de clases.....</i>	<i>8</i>
2. Modelo del comportamiento externo.....	9
<i>Descripción de contratos del sistema.....</i>	<i>9</i>
<i>Diagramas de colaboración para cada contrato.....</i>	<i>18</i>
Apéndice 1.0.....	27



## **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>
1.0	22-03-2010	Documento de análisis inicial.

  
Sagres

## 1. MODELO ESTÁTICO

---

Dentro del proceso unificado, el siguiente paso es el análisis. Para ello lo primero es realizar un modelo estático de nuestro sistema. El modelo estático o diagrama de clases contiene todos los elementos que juegan un papel importante y merecen ser representados como clases en el sistema. Se compone de la definición de las clases, sus atributos, métodos y las relaciones entre ellas.

### Descripción de clases conceptuales

En esta primera iteración nos encontramos con las siguientes clases.

- **Carta:** Esta clase es la encargada de almacenar, controlar y gestionar toda la información referente a la carta del restaurante.
- **Sección:** La carta del restaurante se encuentra dividida en secciones. Esta clase se encarga de dividir los distintos elementos que componen la carta según el tipo de cada uno.
- **SecciónComida:** Es un tipo de sección. Su función es organizar los elementos de la carta que son platos de comida.
- **SecciónBebida:** Es un tipo de sección. Su función es organizar los elementos de la carta que son bebidas.
- **Elemento:** Representa a cada elemento de la carta del restaurante. Almacena la información asociada al plato o bebida que representa.
- **ElementoPlato:** Elemento de tipo plato. Representa a los platos de comida que se forman con la combinación de distintos ingredientes.
- **ElementoBebida:** Elemento de tipo bebida. Representa a las bebidas que se ofrecen en la carta.
- **Producto:** Se trata de una clase que representa a cualquiera de los productos, tanto bebidas como ingredientes, que se usan en el restaurante.
- **Ingrediente:** Son los ingredientes usados para confeccionar los platos que se presentan en la carta del restaurante.
- **Bebida:** Son las bebidas que se ofrecen en el restaurante.
- **PedidoProveedor:** Es una clase que controla y gestiona los pedidos que se han realizado al proveedor o que están pendientes de recibir.
- **Incidencia:** Refleja una incidencia que ha sucedido con un producto. Controla las pérdidas que se producen en el restaurante por estos problemas.

## Relaciones entre clases conceptuales

- **Carta – Sección:** Como se ha definido anteriormente una carta está compuesta por secciones. Se trata de una relación de composición puesto que la carta se trata como un “todo” del cual las secciones forman parte y no tendrían sentido por sí solas si este “todo” no existiese.
- **SeccionBebida – Seccion – SeccionComida:** Se establece una generalización entre estas tres clases en la que Sección es la clase padre y las otras dos son las que heredan de ella.
- **ElementoPlato – Elemento - ElementoBebida:** Se establece una generalización entre estas tres clases en la que Elemento es la clase padre y las otras dos son las que heredan de ella.
- **SeccionComida - ElementoPlato:** Está claro que una sección de la carta organiza a una serie de elementos de un tipo concreto. En este caso, una seccion de comida organiza elementos de tipo plato, es decir, elementos de la carta que están realizados con ingredientes. La relación que se establece es de agregación, puesto que aunque los elementos se vinculan con un “todo” como es la sección, estos pueden existir de manera independiente.
- **SeccionBebida - ElementoBebida:** Se da la misma situación que para la relación anterior.
- **Ingrediente – Producto – Bebida:** Se establece una generalización entre estas tres clases en la que Producto es la clase padre y las otras dos son las que heredan de ella.
- **Producto – Incidencia:** Una incidencia lleva asociada un producto, de ahí la relación entre estas dos clases.
- **PedidoProveedor – Producto:** Cuando se realiza un pedido al proveedor este lleva asociada una lista de productos con las cantidades de cada uno que se desean pedir.

Sagres



## Atributos asociados a clases y/o relaciones

- **Carta**
  - fechaUltimaModificación: Indica la fecha más reciente en la que la carta fue modificada. Este atributo se utilizará para controlar que la carta no sufra cambios importantes hasta pasados los 6 meses desde la última modificación.
- **Sección**
  - nombre: Nombre de la sección.
- **Elemento**
  - codElem: número que identifica a un elemento.
  - descripción: breve descripción del elemento.
  - disponible: nos dice si un elemento está disponible para su pedido o está invalidado por cualquier motivo.
  - foto: imagen para visualizar el elemento.
  - nombre: nombre del elemento que se visualizará en la carta.
- **ElementoPlato**
  - racionesMax: número de raciones en las que se puede dividir el plato.
  - racionesActuales: número de raciones que quedan del plato. Cuando se acaben se volverá a hacer el plato
  - tiempoElaboracion: tiempo en minutos que se tarda en elaborar el plato. Solo se tiene en cuenta en el caso de que no haya raciones.
- **ElementoBebida**
  - tipo: Indica si se vende por copas, chupitos, etc... o se tiene que vender por unidad.
- **Producto**
  - codPro: Identifica inequívocamente a un producto en nuestro sistema..
  - nombre: El nombre que se le da a un producto. En las consultas y listas será el que aparecerá identificando al producto.
  - mínimo: indica el límite mínimo de cantidad mínima en stock a partir del cual el sistema considerará que éste producto es escaso y se añadirá a una lista para realizar un próximo pedido.
  - máximo: indica la máxima cantidad del que se puede disponer de un producto en el almacén.
  - cantidad: es la cantidad actual en stock del producto.
  - foto: imagen del producto.
- **Bebida**
  - cantidadporEnvase: número de ml que contiene un envase de una bebida.

- **PedidoProveedor**

- fechaPedido: fecha en la que se realizó el pedido.
- recibido: indica si el pedido en cuestión ha sido recibido o no.

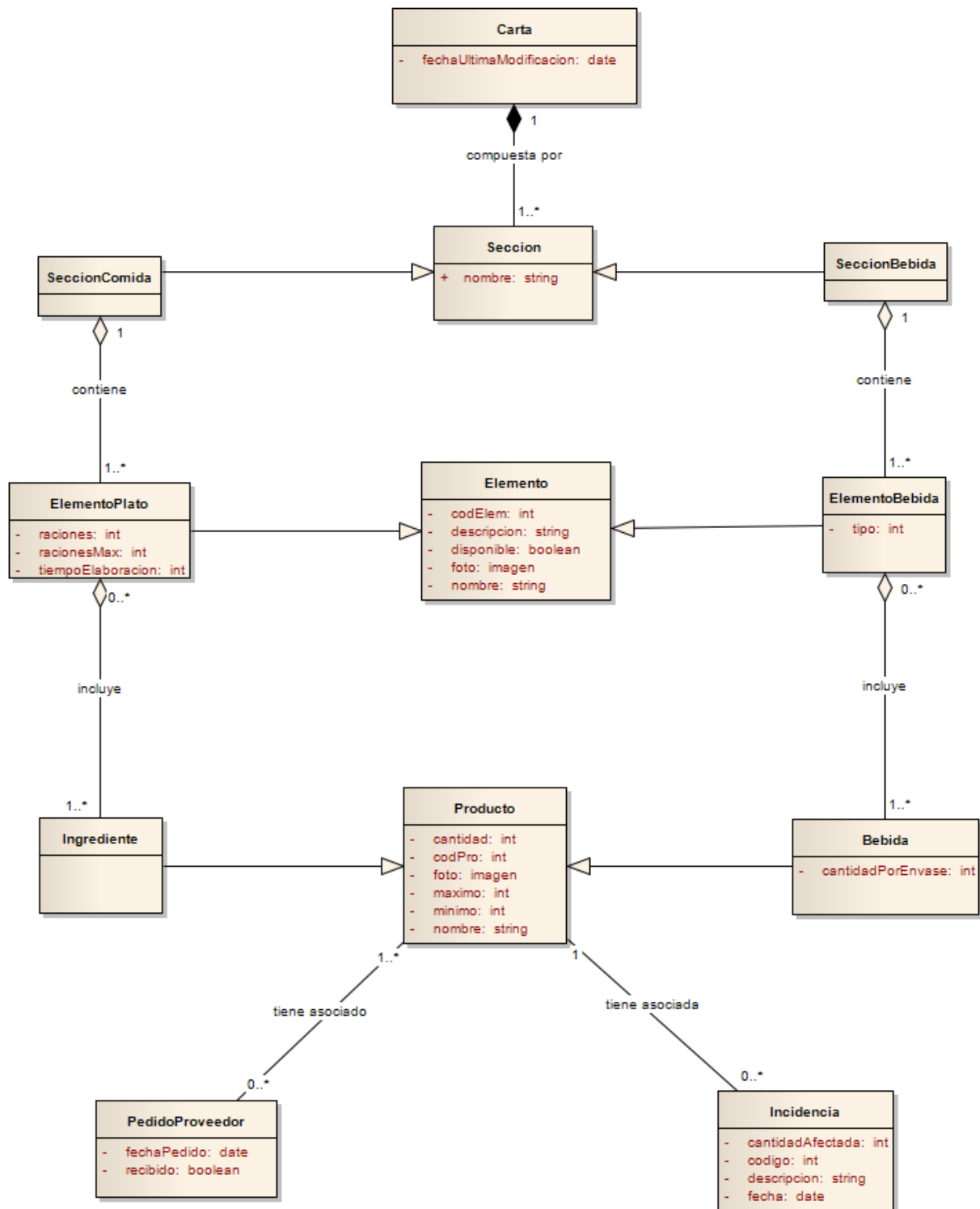
- **Incidencia**

- cantidadAfectada: cantidad de producto que se ha visto afectada en la incidencia.
- codigo: numero que identifica a una incidencia.
- descripción: breve descripción del suceso
- fecha: fecha en la que sucedió el incidente.



# Diagrama de clases

class Diagrama de clases





## 2.MODELO DEL COMPORTAMIENTO EXTERNO

### Descripción de contratos del sistema

<b>Nombre:</b>	listaProductos = imprimirListaProductosaPedir()
<b>Responsabilidad:</b>	Imprimir la lista de productos a pedir y guardar una copia de esa lista
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Imprimir lista de productos a pedir”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"><li>• Si no hay ningún producto cuya cantidad actual en stock esté bajo mínimos</li></ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"><li>• Se creó un PedidoProveedor</li><li>• Se asoció una lista de Productos al PedidoProveedor.</li></ul>
<b>Salida:</b>	<ul style="list-style-type: none"><li>• La lista de productos junto con las cantidades necesitadas de cada uno</li></ul>

Sagres

<b>Nombre:</b>	introducirDatosIncidencia(codProducto, cantidadInvolucrada, tipoIncidencia)
<b>Responsabilidad:</b>	Comprobar que los datos introducidos son correctos, generar un código de incidencia, registrar la incidencia, actualizar la cantidad en stock del producto afectado y, generar una alerta si la cantidad en stock del producto afectado ha superado su límite mínimo o invalidar los elementos de la carta que incluyan ese producto si la cantidad de éste es 0.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Notificar incidencia con bebida” y “Notificar incidencia con Ingrediente”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si los datos son incorrectos</li> <li>• Si no existe el producto identificado por codProducto</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó una Incidencia</li> <li>• Se ha modificado el valor del parámetro Producto.cantidad del Producto identificado por codProducto</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	listaProductos, listaHabilitados = notificarRecepcionPedido()
<b>Responsabilidad:</b>	Actualizar las cantidades en stock de los productos que han sido pedidos y habilitar los elementos de la carta inválidos si todos sus productos asociados están disponibles en stock
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Notificar recepción de pedido del proveedor”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no hay ningún pedido pendiente de recibir</li> </ul>
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>• Se ha imprimido la lista de productos a pedir</li> </ul>
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se ha modificado el valor del parámetro Producto.cantidad de varios productos.</li> <li>• Se ha modificado el valor del parámetro PedidoProveedor.recibido a verdadero</li> </ul>
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• La lista de productos que se han visto actualizados</li> <li>• La lista de elementos de la carta que han sido habilitados</li> </ul>

<b>Nombre:</b>	introducirDatosIngrediente(nombre, cantidad, minimo, maximo, foto)
<b>Responsabilidad:</b>	Comprobar que los datos introducidos son correctos, generar un código de producto y registrar el ingrediente en el sistema.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso: Añadir ingrediente
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si alguno de los datos introducidos no se corresponden a la realidad (por ejemplo, son negativos o la imagen no es del tipo requerido...)</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó un ingrediente.</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	introducirDatosBebida(nombre, cantidad, minimo, maximo, foto, cantidadPorEnvase)
<b>Responsabilidad:</b>	Comprobar que los datos introducidos son correctos, generar un código de producto y registrar la bebida en el sistema.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso: Añadir bebida
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si alguno de los datos introducidos no se corresponden a la realidad (por ejemplo, son negativos o la imagen no es del tipo requerido...)</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó una bebida</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	obtenerDetallesProducto(codProducto)
<b>Responsabilidad:</b>	Operación que muestra los detalles de un producto a eliminar. De una lista selecciona para eliminar el producto con código codProducto como identificador y muestra los detalles de éste.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso: 'Eliminar bebida', 'Modificar Bebida', 'Eliminar Ingrediente' y 'Modificar Ingrediente'
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si el producto identificado por codProducto no existe</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	Información detallada sobre el producto seleccionado

<b>Nombre:</b>	eliminarProducto(codProducto)
<b>Responsabilidad:</b>	Eliminar el elemento identificado por codProducto del sistema e invalidar los elementos de la carta que tengan asociado dicho producto
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de Uso 'Eliminar bebida' y 'Eliminar Ingrediente'
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>◦ Si no existe el producto identificado por codProducto</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se eliminó el producto indetificado por codProducto.</li> <li>• Se modificó el valor del parametro Elemento.disponible a FALSO en los elementos de la carta que tenían asociado el producto</li> </ul>
<b>Salida:</b>	Una lista con los elementos de la carta que han sido invalidados

<b>Nombre:</b>	modificarBebida(codProducto,nombre, cantidad, minimo, maximo, foto, cantidadPorEnvase)
<b>Responsabilidad:</b>	Comprobar que los datos introducidos son correctos y registrar los cambios en los datos de la bebida.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso: 'Modificar Bebida'
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si alguno de los datos introducidos no se corresponden a la realidad (por ejemplo, son negativos o la imagen no es del tipo requerido...) se lanza excepción y se vuelve a la pantalla de introducción de datos.</li> <li>• Si no existe la bebida identificada por codProducto</li> <li>• Si ha ocurrido un error al registrar los cambios.</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se modificó una bebida.</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	modificarIngrediente(codProducto,nombre, cantidad, minimo, maximo, foto)
<b>Responsabilidad:</b>	Comprobar que los datos introducidos son correctos y registrar los cambios en los datos del ingrediente.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso: 'Modificar Ingrediente'
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si alguno de los datos introducidos no se corresponden a la realidad (por ejemplo, son negativos o la imagen no es del tipo requerido...) se lanza excepción y se vuelve a la pantalla de introducción de datos.</li> <li>• Si no existe el ingrediente identificado por codProducto</li> <li>• Si ha ocurrido un error al registrar los cambios.</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se modificó un ingrediente.</li> </ul>
<b>Salida:</b>	



<b>Nombre:</b>	ListaProductos = asignarSeccion(sec)
<b>Responsabilidad:</b>	Asignar una sección al nuevo elemento y obtener una lista de productos vinculados a esa sección
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir Elemento a Carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no existe la sección identificada por sec</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	Una lista de productos acordes con la sección asignada

<b>Nombre:</b>	ListaProductosDeElemento = añadirProductoaLista(codProducto, cantidad)
<b>Responsabilidad:</b>	Añadir producto, junto con su cantidad, a la lista de productos del elemento que se quiere añadir a la carta
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir Elemento a Carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no existe el producto identificado por codProducto</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se añadió un producto a la lista de productos asociados al nuevo elemento</li> </ul>
<b>Salida:</b>	La lista de productos asociados al nuevo elemento actualizada



<b>Nombre:</b>	eliminarProductoDeLista(codProducto)
<b>Responsabilidad:</b>	Eliminar el producto identificado por codProducto de la lista de productos asociados al elemento que se quiere añadir a la carta
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir Elemento a Carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si el producto identificado por codProducto no se encuentra en la lista de productos asociados al elemento</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se elimino un producto de la lista de productos asociados al nuevo elemento</li> </ul>
<b>Salida:</b>	La lista de productos asociados al nuevo elemento actualizada

<b>Nombre:</b>	modificarProductoLista(codProducto, cantidad)
<b>Responsabilidad:</b>	Modificar la cantidad de un producto de la lista de productos asociados al elemento que se quiere añadir a la carta
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir Elemento a Carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si el producto identificado por codProducto no se encuentra en la lista de productos asociados al elemento</li> <li>• Si la cantidad introducida es errónea</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se ha modificado la cantidad de un producto</li> </ul>
<b>Salida:</b>	La lista de productos asociados al nuevo elemento actualizada

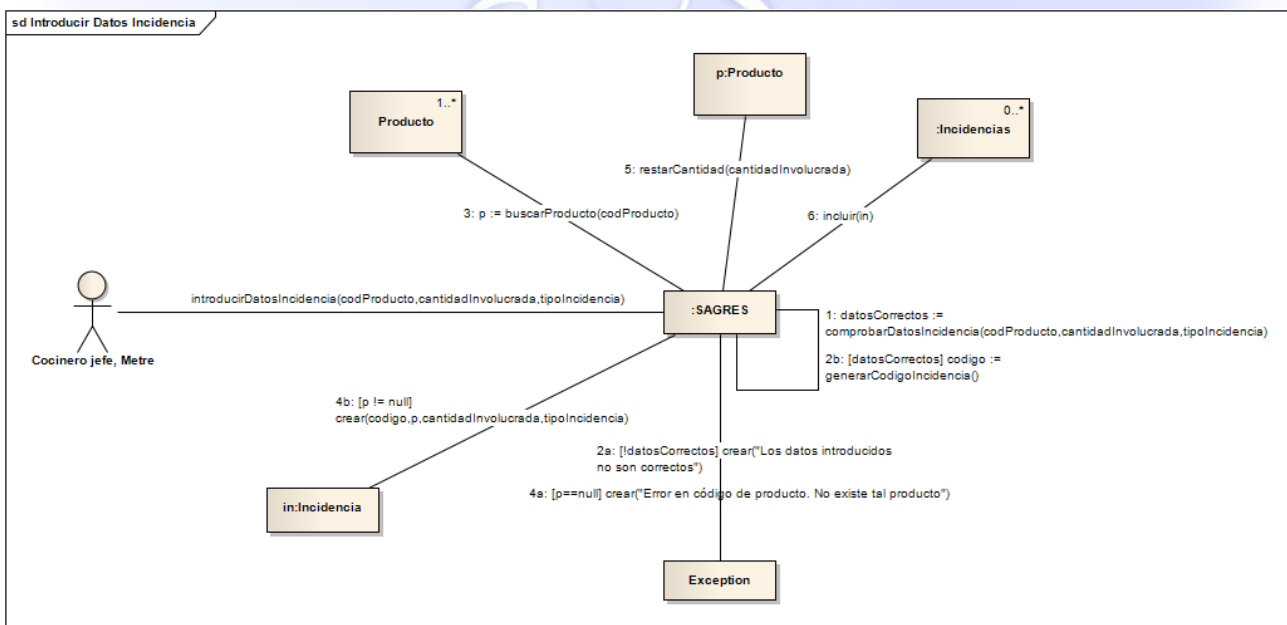
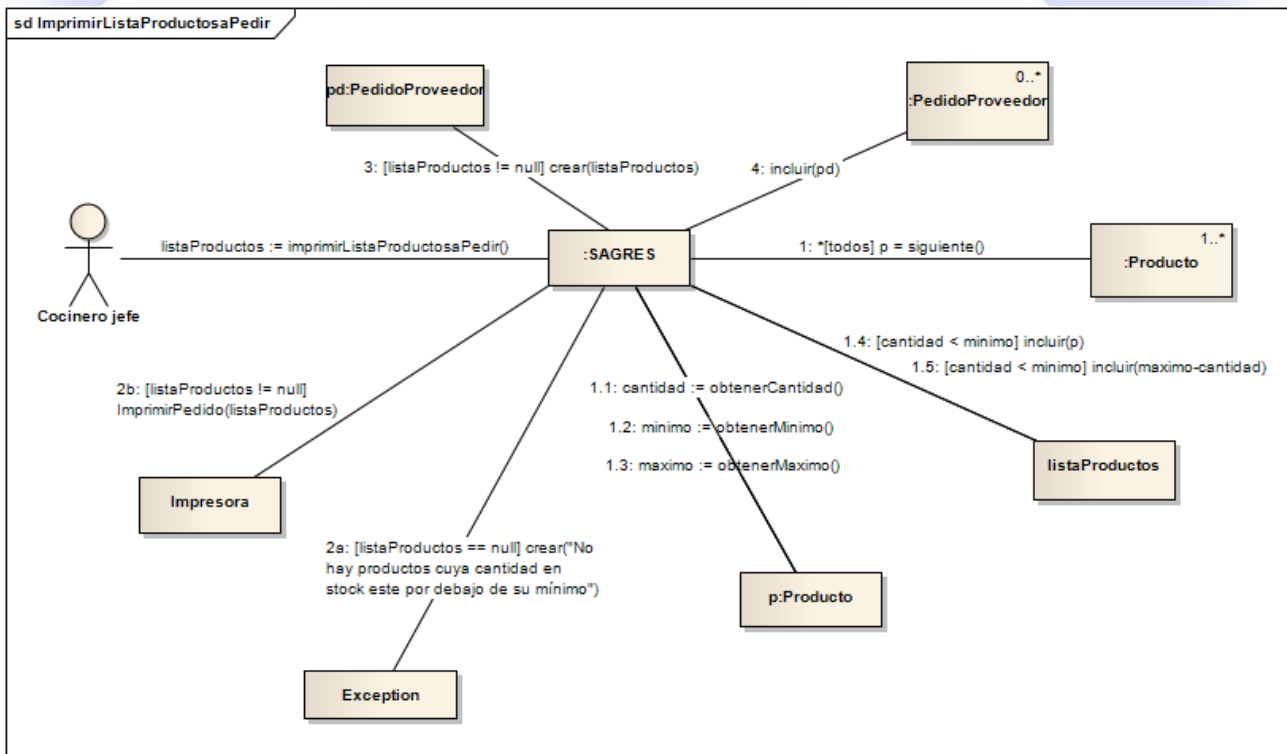
<b>Nombre:</b>	introducirDatosElemento(nombre, sec, descripcion, precio, foto, tiempoPreparacion, diviMax)
<b>Responsabilidad:</b>	Comprobar que los datos introducidos son correctos, generar un codigo de elemento y registrar el nuevo elemento en el sistema.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir Elemento a Carta”
<b>Nota:</b>	“diviMax” hace referencia al número de porciones o divisiones máximas que se pueden hacer del elemento. En el caso de platos de comida serán raciones y en el de las botellas serán divisiones por cada unidad de envase.
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si los datos introducidos no son correctos</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó un Elemento</li> <li>• Se creó un vínculo entre Elemento y Sección</li> <li>• Se crearon varios vínculos entre Elemento y Productos</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	obtenerDetallesElemento(codElemento)
<b>Responsabilidad:</b>	Obtener la información detallada del elemento
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Eliminar Elemento de Carta” Caso de uso “Modificar Elemento de Carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no existe el elemento identificado por codElemento</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	Información detallado del elemento

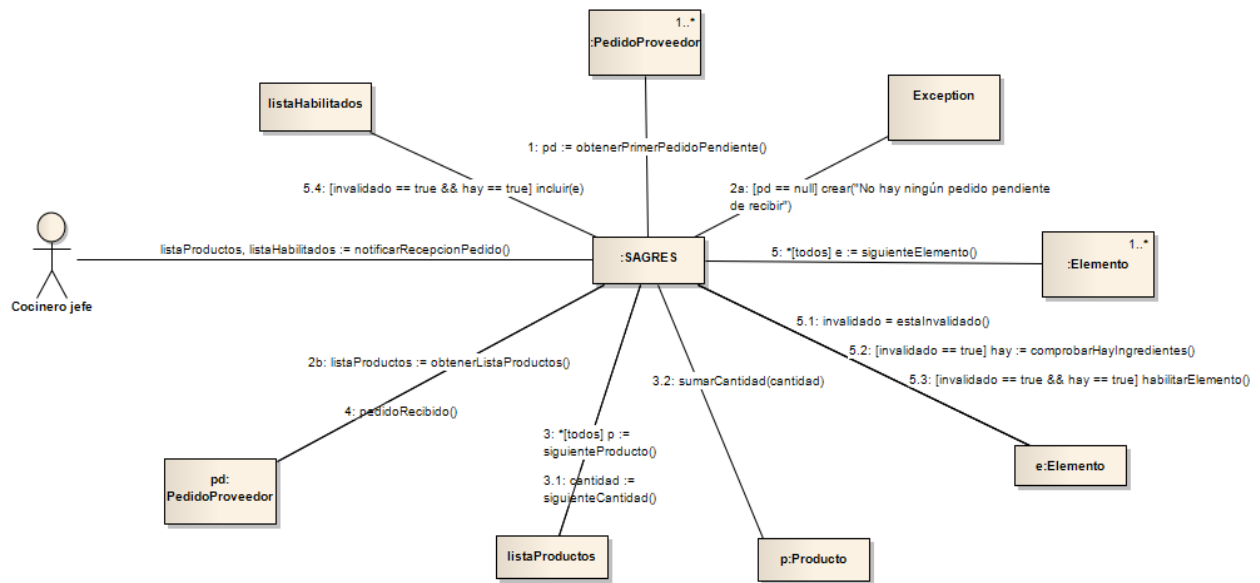
<b>Nombre:</b>	eliminarElemento(codElemento)
<b>Responsabilidad:</b>	Eliminar un elemento de la lista de elementos de la carta
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Eliminar Elemento de Carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no existe el elemento identificado por codElemento</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se eliminó un elemento de la lista de elementos de carta</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	modificarDatosElemento(codElemento, nombre, descripcion, precio, foto, tiempoPreparacion, diviMax)
<b>Responsabilidad:</b>	Comprobar que los datos introducidos son correctos y registrar los cambios realizados en el elemento
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Modificar Elemento de Carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si los datos introducidos son incorrectos</li> <li>• Si no existe el elemento identificado por codElemento</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se ha modificado un elemento</li> </ul>
<b>Salida:</b>	

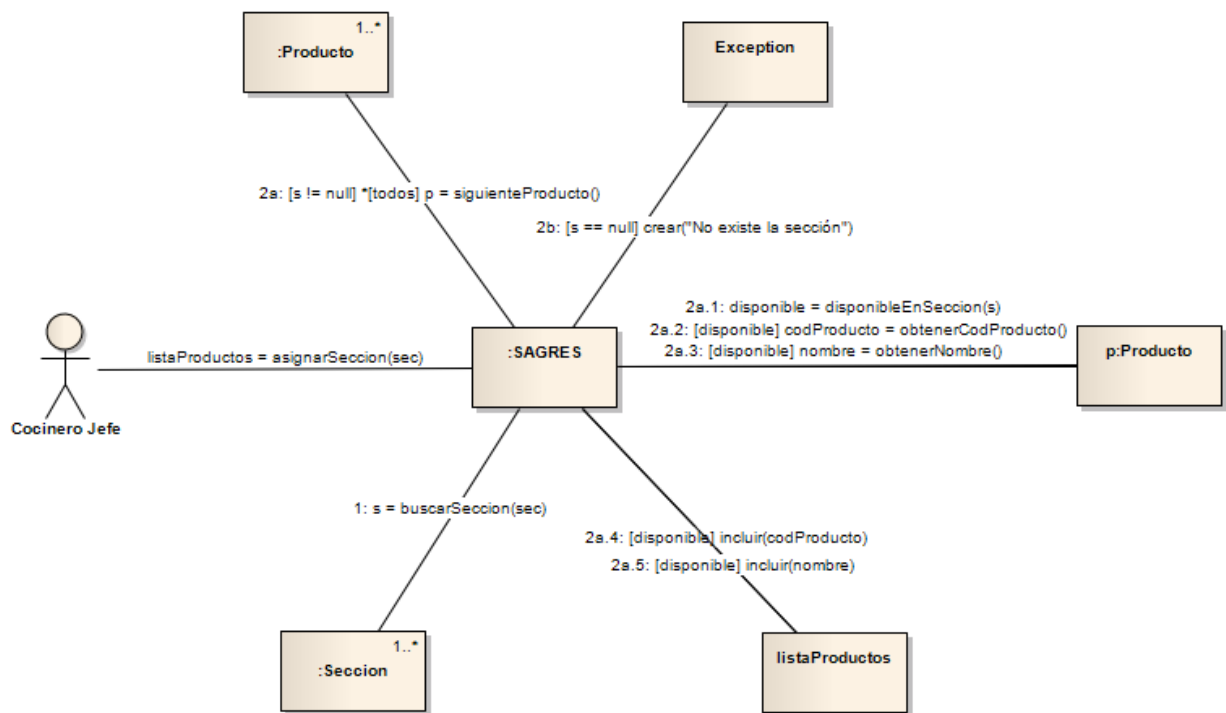
## Diagramas de colaboración para cada contrato



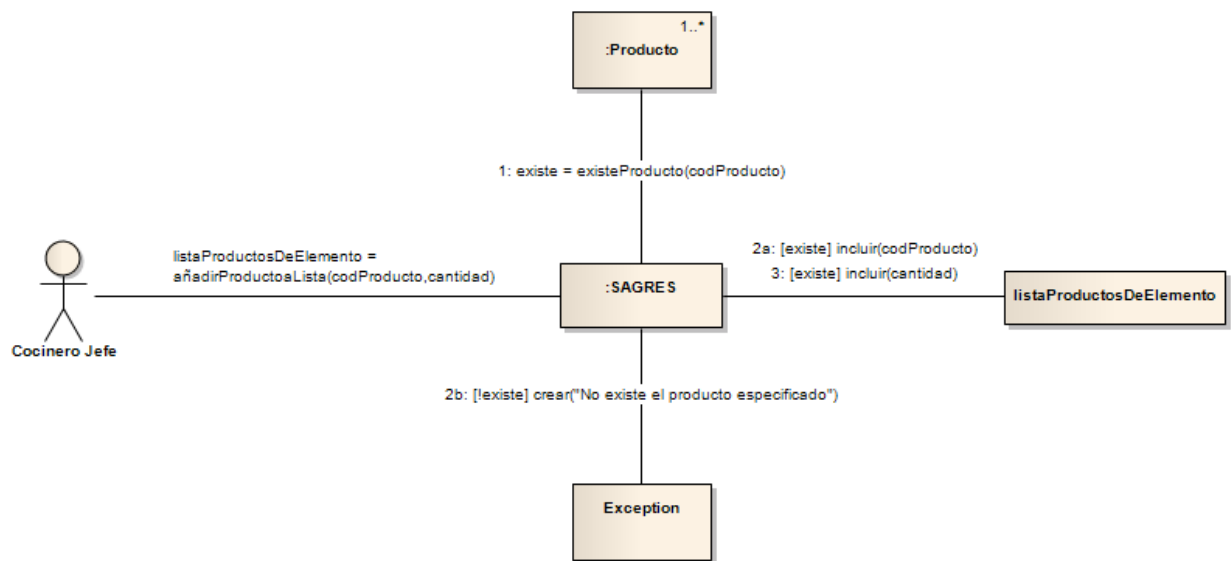
# sd Notificar Recepcion de Pedido



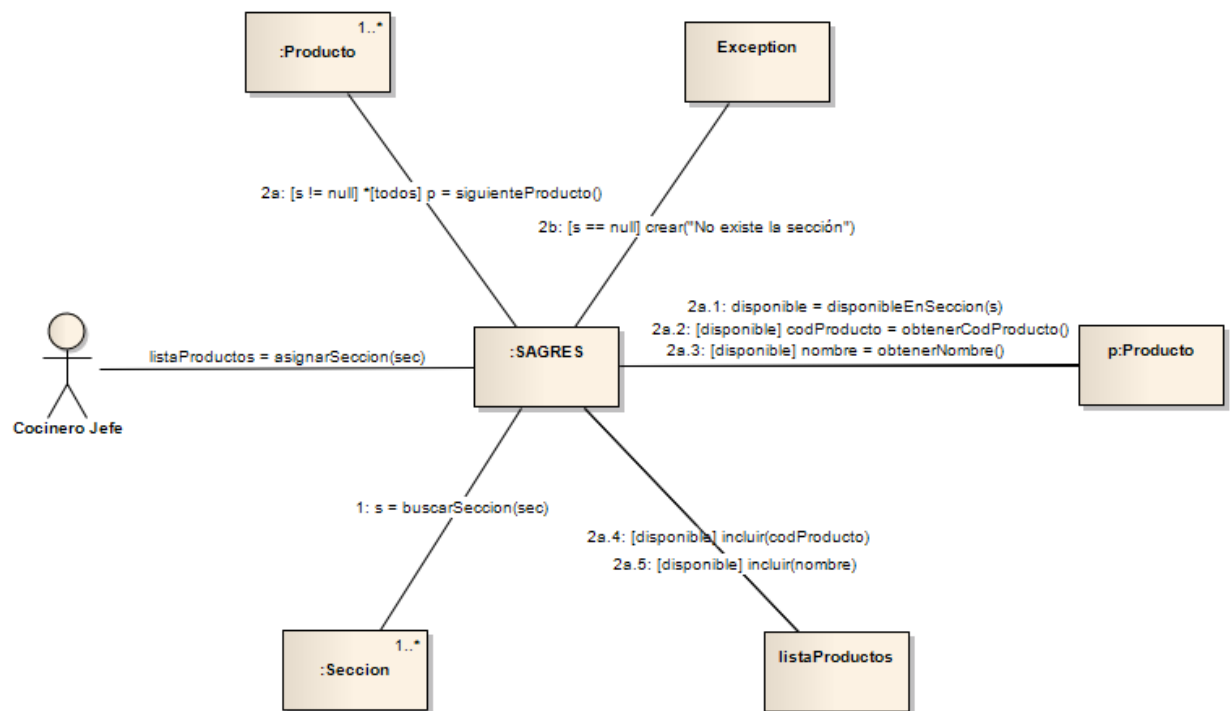
# sd Asignar Seccion



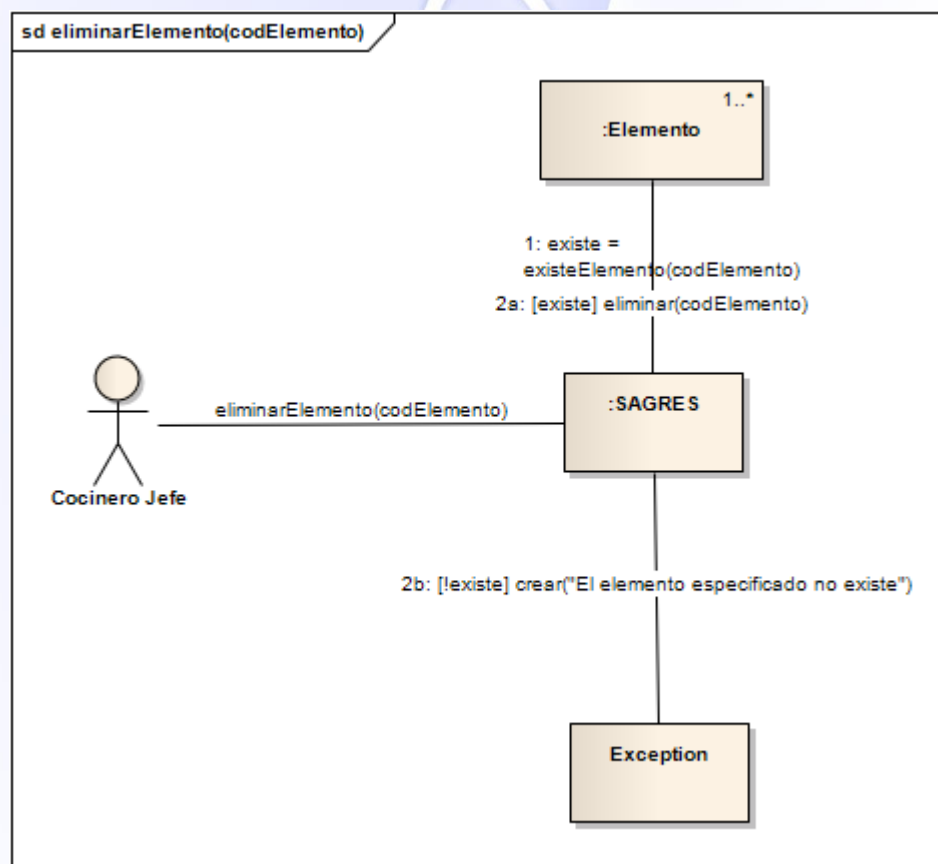
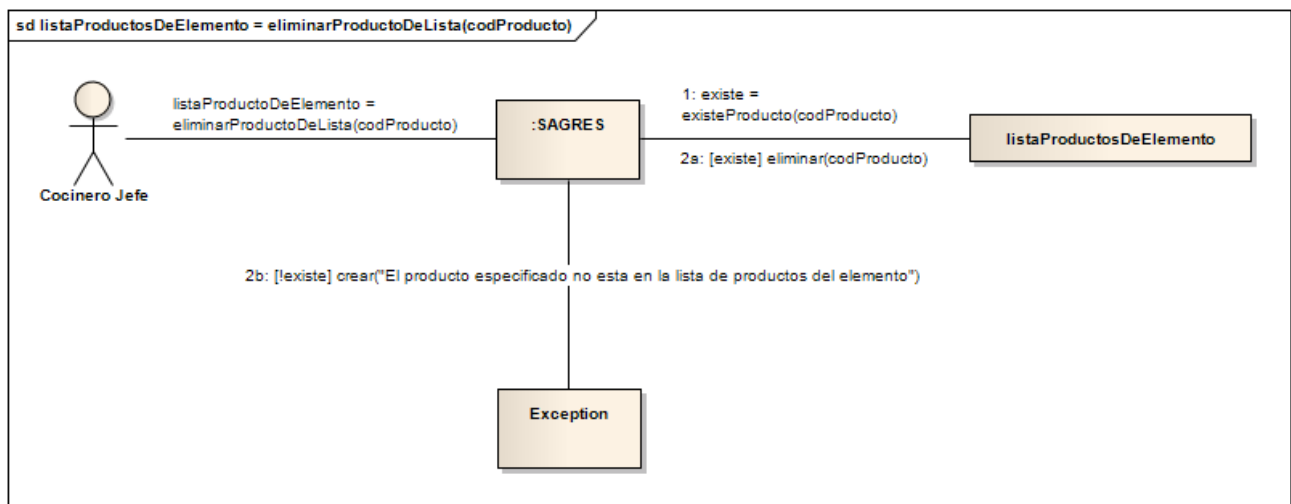
sd listaProductosDeElemento = añadirProductoaLista(codProducto,cantidad)



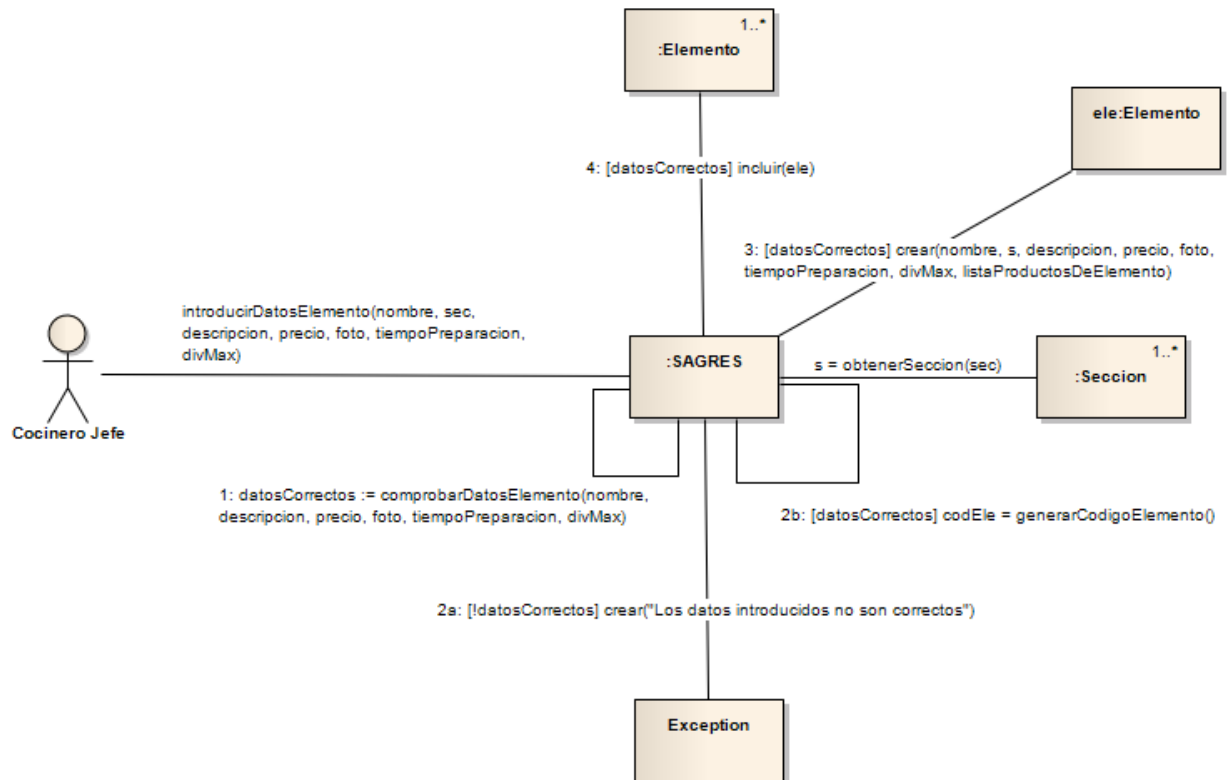
sd listaProductos = asignarSeccion(sec)





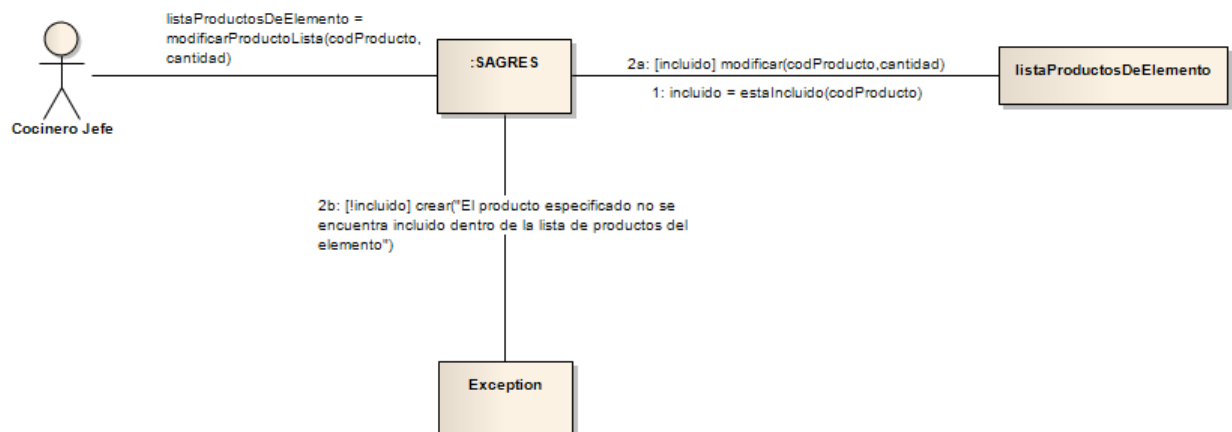


sd introducirDatosElemento(nombre, descripcion, precio, foto, tiempoPreparacion, divMax) (añadir elemento)

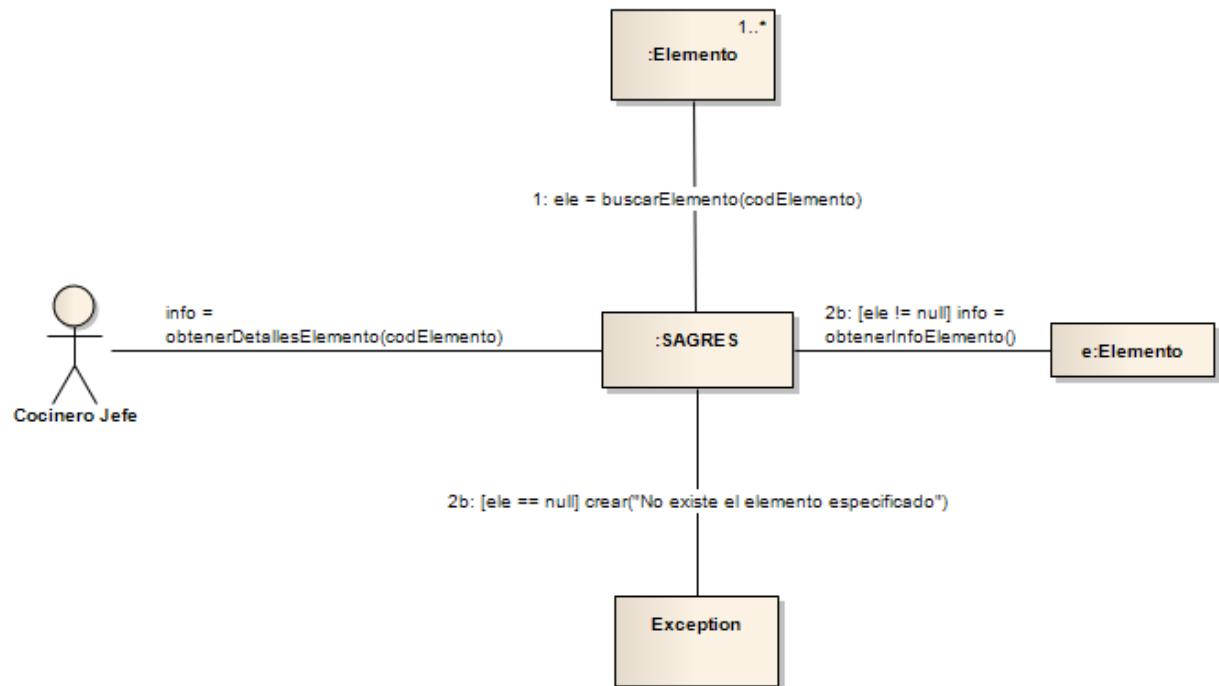


Sagres

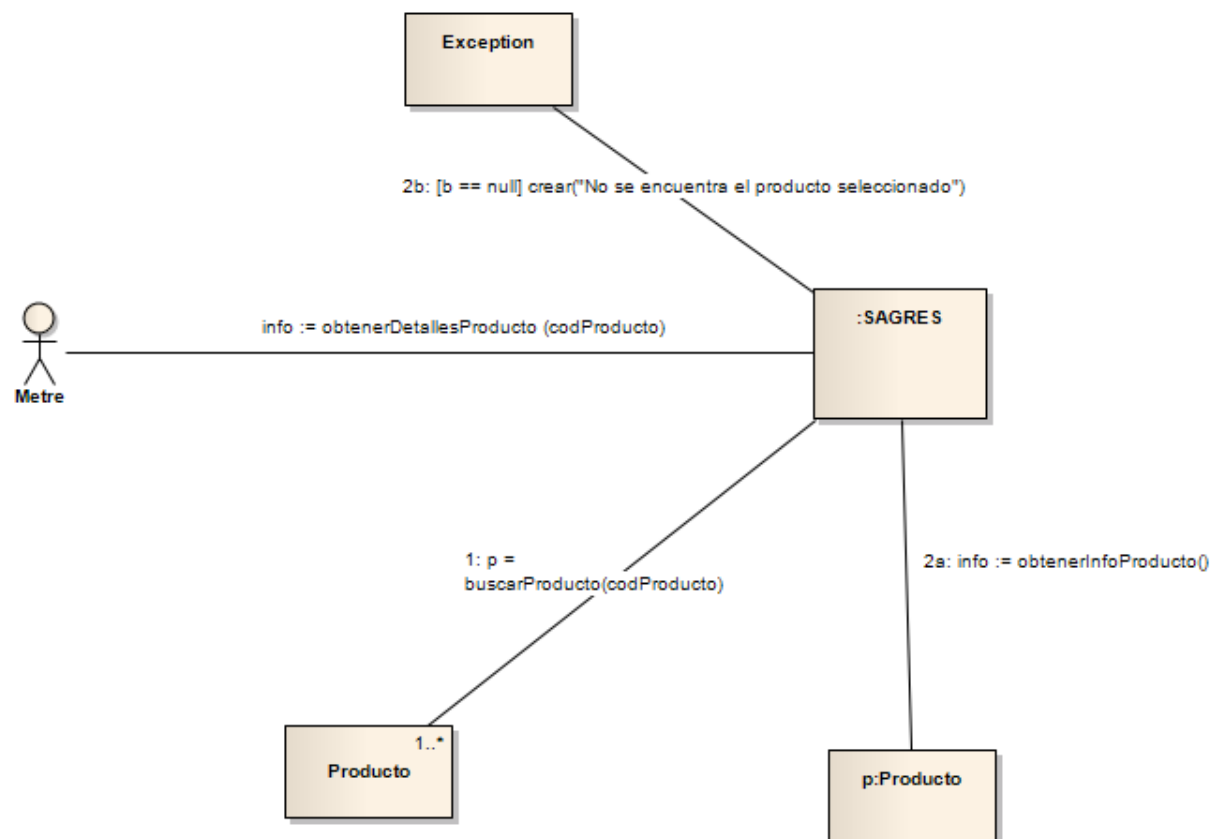
sd listaProductoDeElemento = modificarProductoLista(codProducto,cantidad)



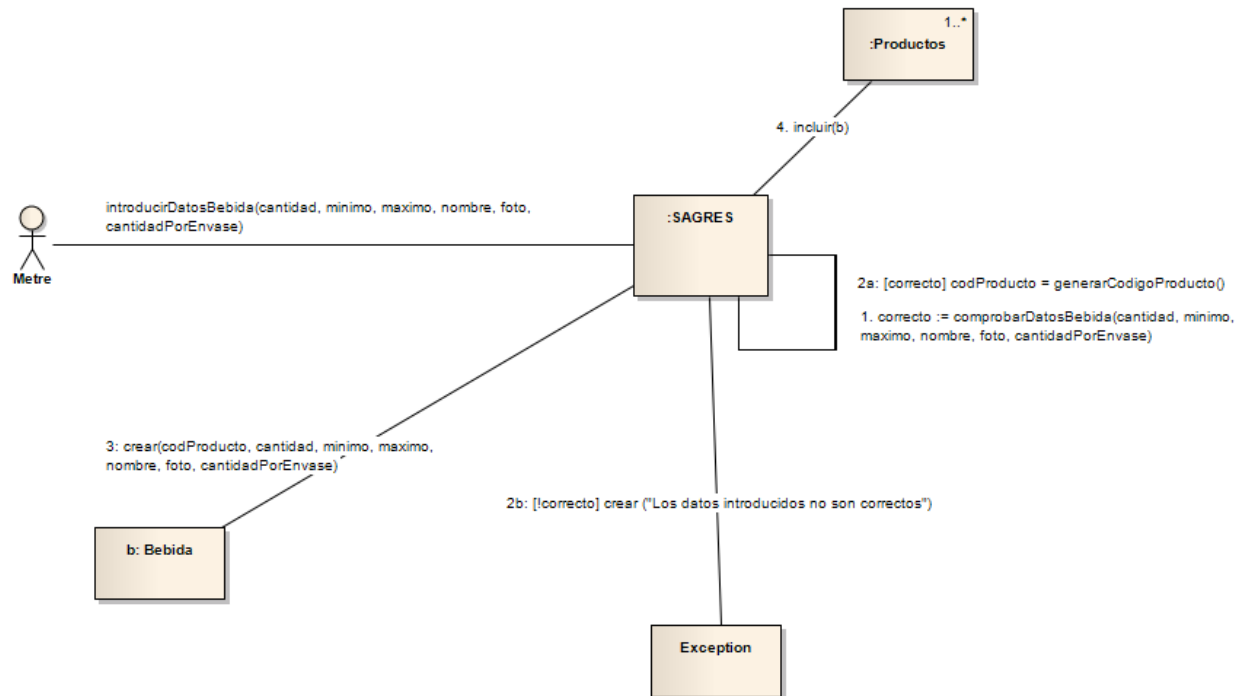
sd info = obtenerDetallesElemento(codElemento)



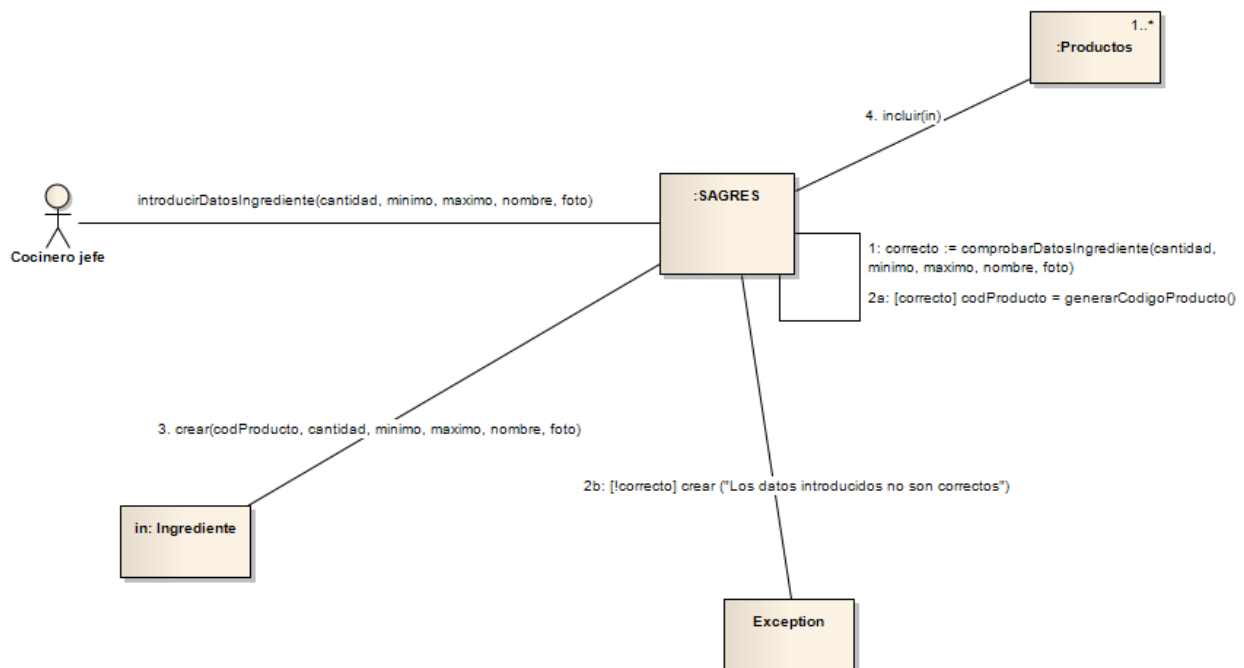
sd obtenerDetallesProducto



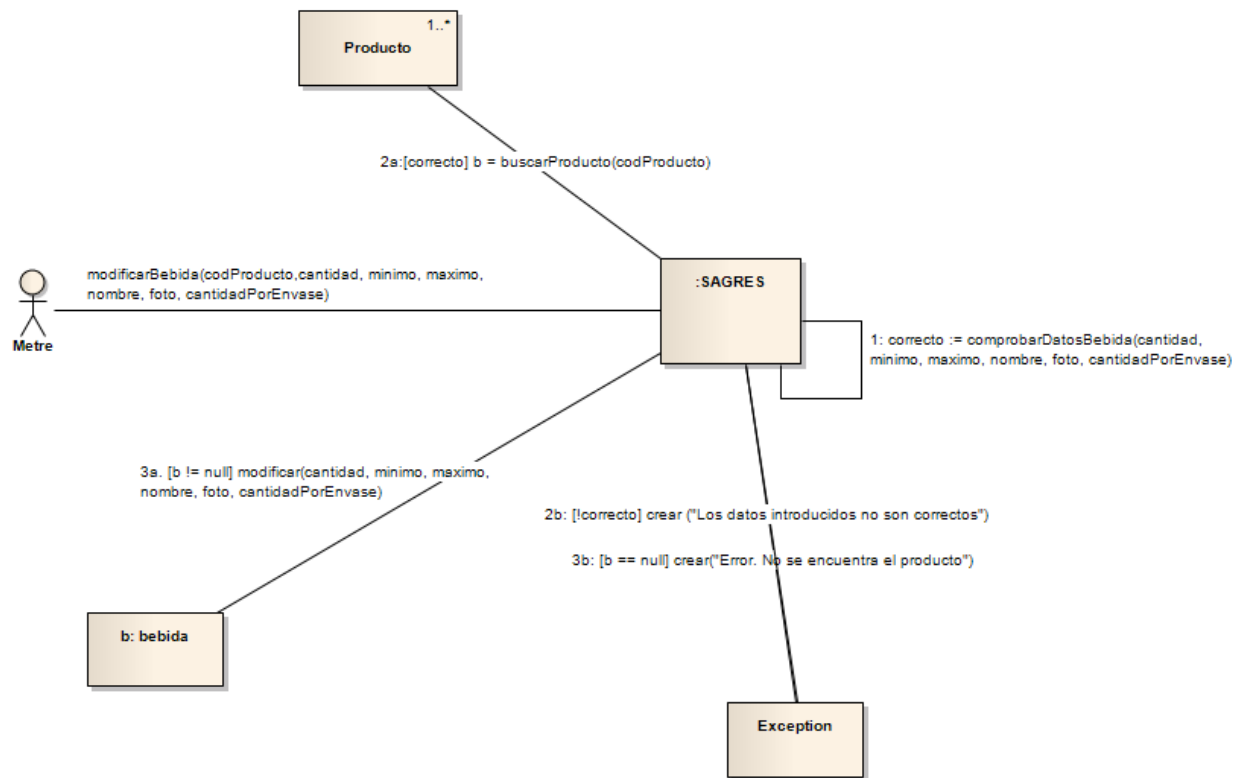
# sd introducirDatosBebida



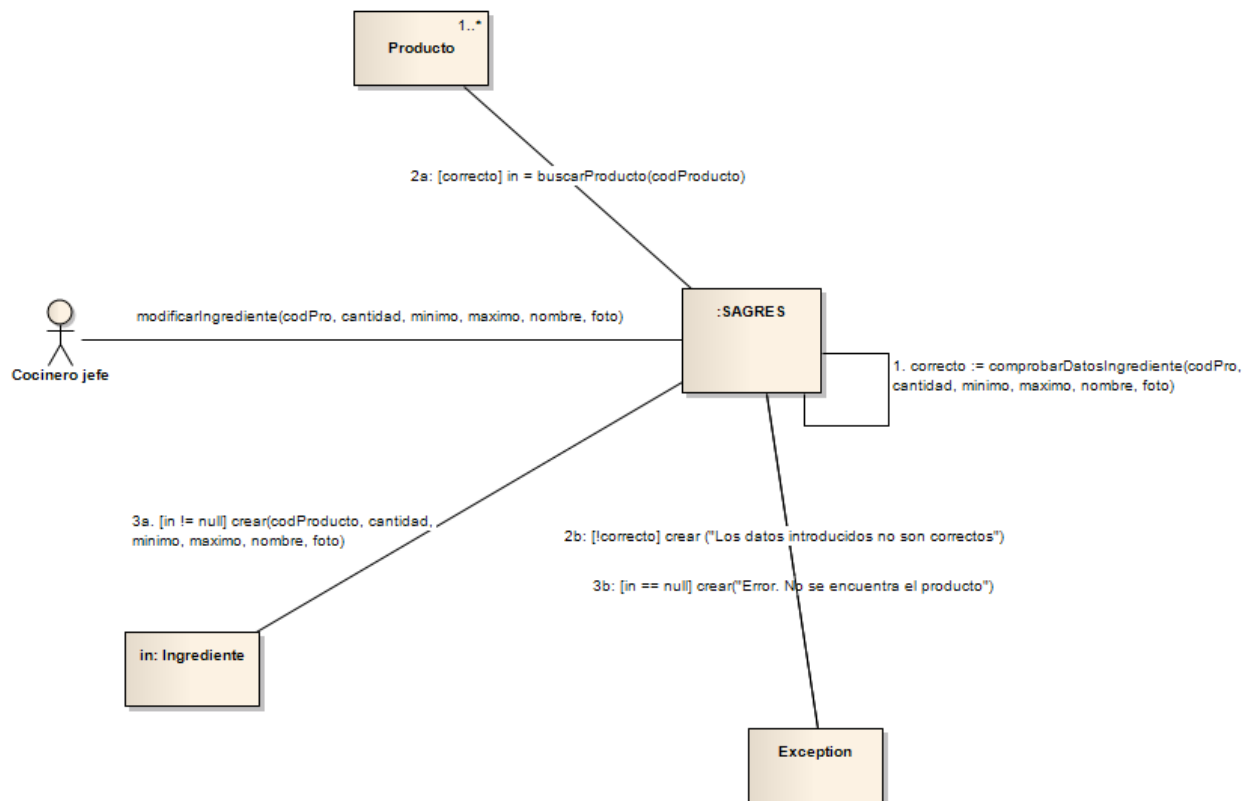
# sd introducirDatosIngrediente

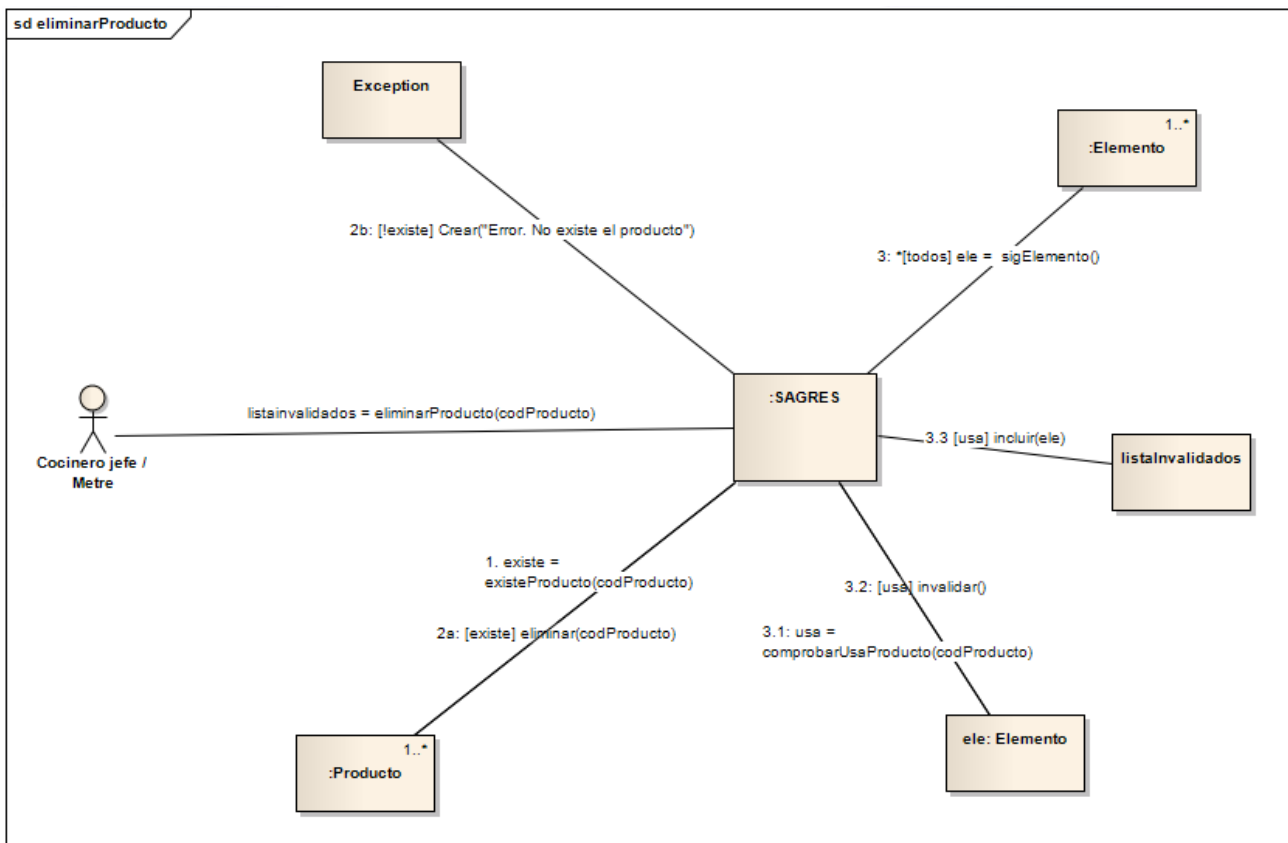


# sd modificarBebida(datos)



# sd modificarIngrediente(datos)





Sagres



## APÉNDICE 1.0

Fecha	22/03/2010
Descripción del problema	
Impacto del problema	
Soluciones adoptadas	<ul style="list-style-type: none"><li>• Se ha generado el documento de análisis inicial.</li></ul>
Anexos a la versión	

Sagres