

# Documento de ANÁLISIS

Sagres

**Sistema de Administración y  
Gestión de RESTaurantes**



Gaspar Muñoz Soria  
Samuel Guirado Navarro  
Carlos Moreno Muñoz

V1.1

## 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>24</i>
Apéndice 1.0.....	30
Apéndice 1.1.....	31

Sagres

## **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.
1.1	16-04-2010	Revisión del documento de análisis de la 1ª iteración



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

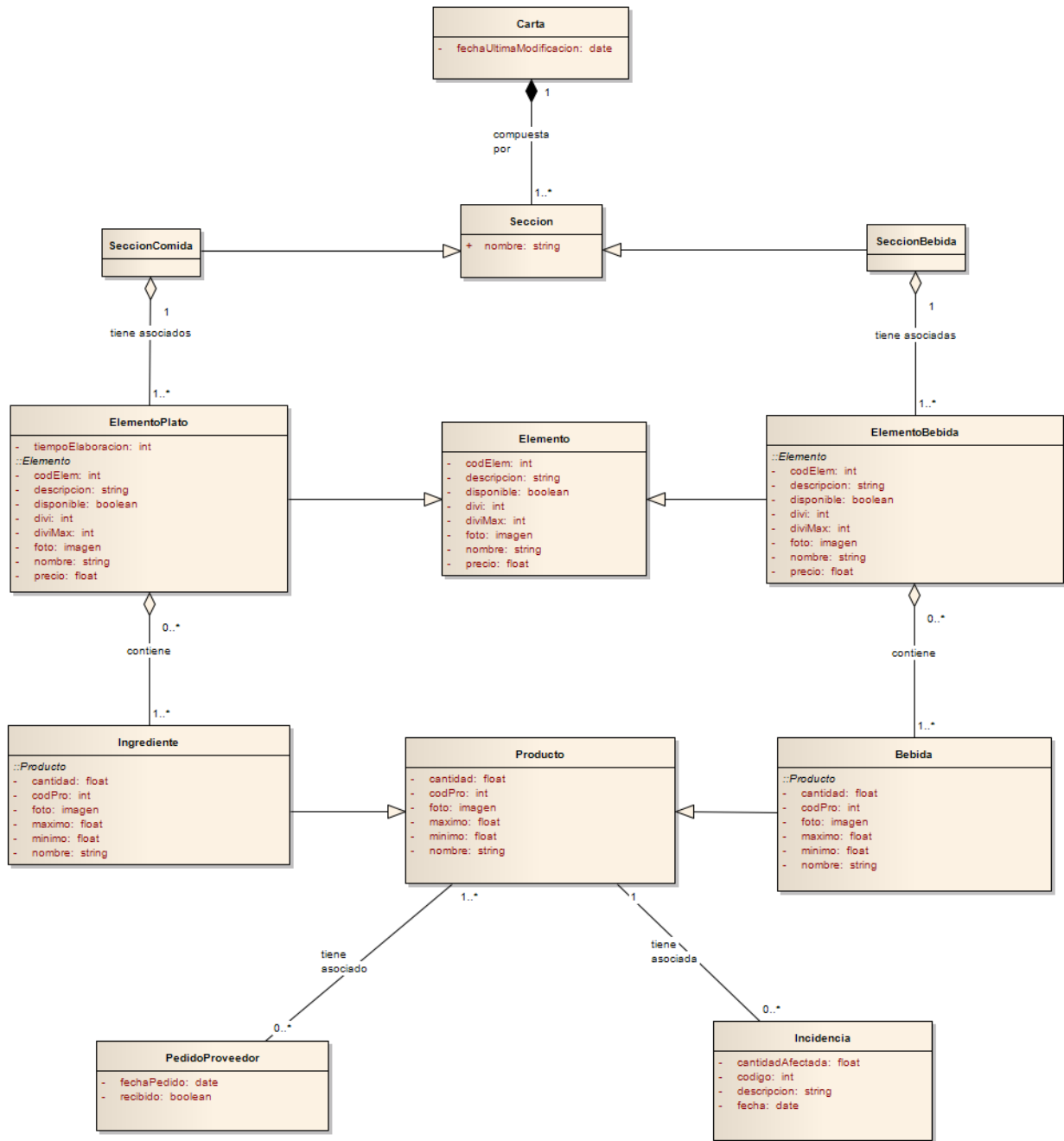
- **Carta**
  - fechaUltimaModificación: Indica la fecha más reciente en la que la carta fue modificada. Este atributo se utilizará para controlar que la carta no sufra cambios importantes hasta pasados los 6 meses desde la última modificación.
- **Sección**
  - nombre: Nombre de la sección.
- **Elemento**
  - 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.
  - diviMax: número de divisiones que se pueden hacer del elemento (en el caso de un plato raciones, en el caso de una bebida vasos).
  - divi: número de porciones que quedan del elemento.
- **ElementoPlato**
  - tiempoElaboracion: tiempo en minutos que se tarda en elaborar el plato. Solo se tiene en cuenta en el caso de que no haya raciones.
- **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.
- **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.

A stylized logo featuring a large, white, cursive letter 'S' that loops around itself. Below the 'S', the word 'Sagres' is written in a white, cursive script font. The entire logo is set against a light blue background with a large white circle on the right side.

# Diagrama de clases

class Diagrama de clases





## 2.MODELO DEL COMPORTAMIENTO EXTERNO

### Descripción de contratos del sistema

<b>Nombre:</b>	listaSecciones = obtieneSecciones()
<b>Responsabilidad:</b>	Obtener una lista con las secciones registradas en el sistema
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Casos de uso “Añadir elemento a la carta”, “Eliminar elemento de la carta”, “Modificar elemento de la carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"><li>• Si no existen secciones</li></ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"><li>• Una lista con las secciones registradas en el sistema</li></ul>

<b>Nombre:</b>	listaProductos = obtieneProductosSeccion(sec)
<b>Responsabilidad:</b>	Obtener una lista de productos para la sección especificada
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir elemento a la carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"><li>• Si no existen productos para la sección especificada</li></ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"><li>• La lista de productos para la sección especificada</li></ul>

<b>Nombre:</b>	nuevoElemento(ele, sec)
<b>Responsabilidad:</b>	Registrar un nuevo elemento en el sistema
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir elemento a la carta”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó un Elemento</li> <li>• Se asociaron varios productos a un elemento</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	listaElementos = obtieneElementosSeccion(sec)
<b>Responsabilidad:</b>	Obtener una lista de elementos para la sección especificada
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Casos de uso “Eliminar elemento de la carta” y “Modificar elemento de la carta”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no existen elementos para la sección especificada</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• La lista de elementos para la sección especificada</li> </ul>

<b>Nombre:</b>	eliminaElemento(ele)
<b>Responsabilidad:</b>	Eliminar un elemento de la lista de elementos del sistema
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Eliminar elemento de la carta”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se eliminó un Elemento</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	modificaElemento(ele)
<b>Responsabilidad:</b>	Modificar un elemento de la lista de elementos del sistema
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Modificar elemento a la carta”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se modificó un Elemento</li> </ul>
<b>Salida:</b>	

<b>Nombre:</b>	nuevoProducto(p)
<b>Responsabilidad:</b>	Registra un nuevo producto en el sistema
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Añadir Bebida” y “Añadir Ingrediente”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó un Producto</li> </ul>
<b>Salida:</b>	



The logo for 'Sagres' is displayed in a light blue, cursive script. Above the text is a stylized graphic element consisting of a large, flowing loop that resembles a ribbon or a stylized letter 'S'. The background is a light blue gradient with a large, white, semi-circular shape on the right side.

<b>Nombre:</b>	listaIngredientes = obtieneIngredientes()
<b>Responsabilidad:</b>	Obtiene una lista de los productos de tipo ingrediente que hay registrados en el sistema.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Casos de uso “Eliminar Ingrediente”, “Modificar Ingrediente” y “Notificar Incidencia con Ingrediente”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no hay ningún ingrediente registrado en el sistema</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• Una lista con los ingredientes registrados en el sistema</li> </ul>



<b>Nombre:</b>	ListaBebidas = obtieneBebidas()
<b>Responsabilidad:</b>	Obtiene una lista de los productos de tipo bebida que hay registrados en el sistema.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Casos de uso “Eliminar bebida”, “Modificar bebida” y “Notificar Incidencia con Bebida”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no hay ninguna bebida registrada en el sistema.</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• Una lista con las bebidas registradas en el sistema</li> </ul>





<b>Nombre:</b>	ListaElementos = obtieneElementosConProducto(p)
<b>Responsabilidad:</b>	Obtiene una lista de Elementos, los cuales llevan entre sus productos asociados el producto p.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Eliminar Ingrediente” y “Eliminar Bebida”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no hay elementos registrados en el sistema</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• Una lista con los elementos que tienen asociado el producto p</li> </ul>



The logo for 'Sagres' is displayed in a light blue, cursive script. Above the text is a stylized graphic element consisting of a large, flowing loop that resembles a ribbon or a stylized letter 'S'. The background of the entire page is a light blue gradient with a subtle, larger-scale pattern.

<b>Nombre:</b>	eliminaProducto(p)
<b>Responsabilidad:</b>	Elimina un producto
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Elimina Ingrediente” y “Elimina Bebida”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se eliminó un Producto</li> </ul>
<b>Salida:</b>	



The logo for 'Sagres' is displayed in a white, elegant script font. Above the text is a stylized graphic element consisting of a large, flowing loop that resembles a ribbon or a calligraphic flourish. The entire logo is set against a light blue background with a subtle gradient and a faint, larger-scale circular pattern.

<b>Nombre:</b>	modificaProducto(p)
<b>Responsabilidad:</b>	Modificar los atributos de un producto registrado en el sistema
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Modificar Ingrediente” y “Modificar Bebida”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se modificó un producto.</li> </ul>
<b>Salida:</b>	



The logo for 'Sagres' is displayed in a white, elegant script font. Above the text is a stylized graphic element consisting of a large, flowing loop that resembles a ribbon or a calligraphic flourish. The entire logo is set against a light blue background with a subtle gradient and a faint, larger-scale circular pattern.

<b>Nombre:</b>	ListaProductosCantidades = obtieneProductosBajoMinimos()
<b>Responsabilidad:</b>	Obtener una lista con los productos cuya cantidad actual en estos se encuentra por debajo de su mínimo y la cantidad necesaria para que esta cantidad se sitúe en su máximo.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Imprimir lista Productos a Pedir”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no hay productos registrados en el sistema</li> <li>• Si no hay productos que estén bajo mínimos</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• Una lista con los productos cuya cantidad actual en estos se encuentra por debajo de su mínimo y la cantidad necesaria para que esta cantidad se sitúe en su máximo.</li> </ul>



Sagres

<b>Nombre:</b>	ImprimeListaProductosaPedir()
<b>Responsabilidad:</b>	Guardar una copia del pedido al proveedor e imprimirla
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Imprimir lista Productos a Pedir”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no hay productos registrados en el sistema</li> <li>• Si no hay productos que estén bajo mínimos</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó un PedidoProveedor</li> <li>• Se asociaron uno o más productos al pedido</li> </ul>
<b>Salida:</b>	



The logo for 'Sagres' is displayed in a large, white, cursive script font. Above the text is a stylized white graphic element consisting of a large loop and a smaller circle, resembling a stylized 'S' or a decorative flourish. The background is a light blue gradient with a faint, larger-scale circular pattern.

<b>Nombre:</b>	PedProveedor = obtienePedidoProveedor()
<b>Responsabilidad:</b>	Obtener el pedido al proveedor pendiente más antiguo
<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 pedidos pendientes</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• El pedido al proveedor más antiguo</li> </ul>



The logo for 'Sagres' is displayed in a light blue, cursive script. Above the text is a stylized graphic element consisting of a large, white, circular shape with a smaller, white, circular shape inside it, resembling a stylized 'S' or a drop.



<b>Nombre:</b>	notificaRecepciónPedido(pedProveedor)
<b>Responsabilidad:</b>	Actualizar las cantidades de los productos que estaban referidos en el pedido y habilitar los elementos de la carta invalidados que dispongan de todos sus productos.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Notificar Recepción de pedido del proveedor”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se modifico el atributo Producto.cantidad de varios productos</li> <li>• Se modificó el atributo PedidoProveedor.recibido a TRUE.</li> </ul>
<b>Salida:</b>	



Sagres

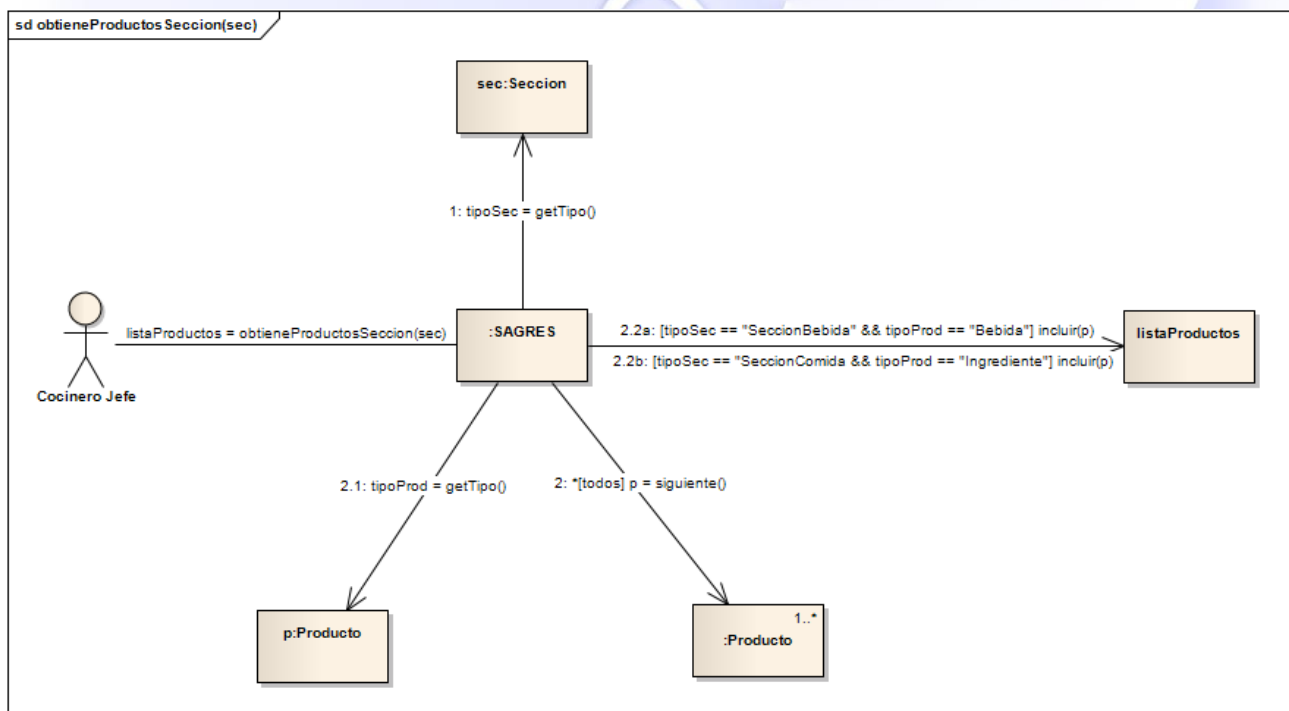
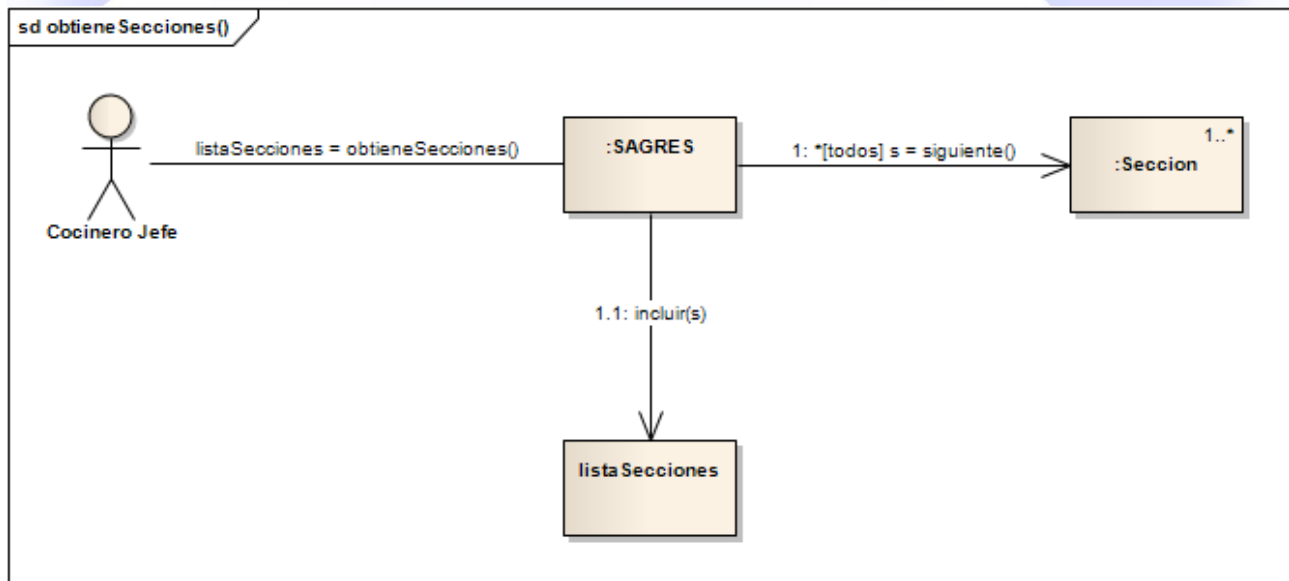
<b>Nombre:</b>	listaElementos = compruebaElementosInvalidados(listaProductosCantidades)
<b>Responsabilidad:</b>	Obtener una lista con los elementos que se verían habilitados tras la actualización de las cantidades de los productos reflejados en listaProductosCantidades
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Notificar Recepción de pedido del proveedor”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• Una lista con los elementos que se verían habilitados tras la notificación de la recepción del pedido del proveedor</li> </ul>

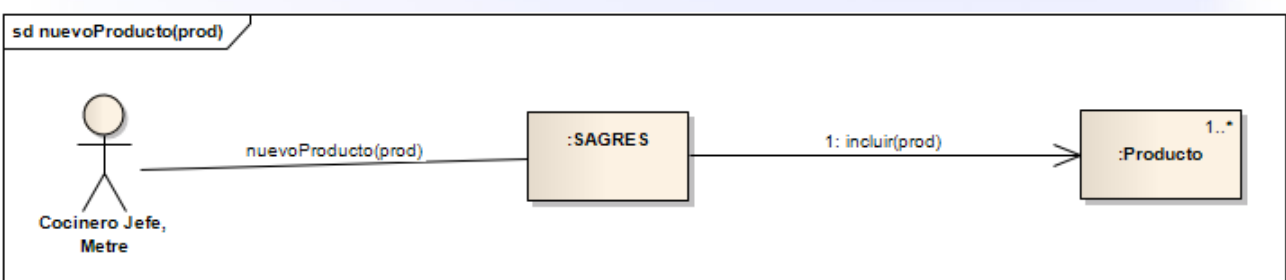
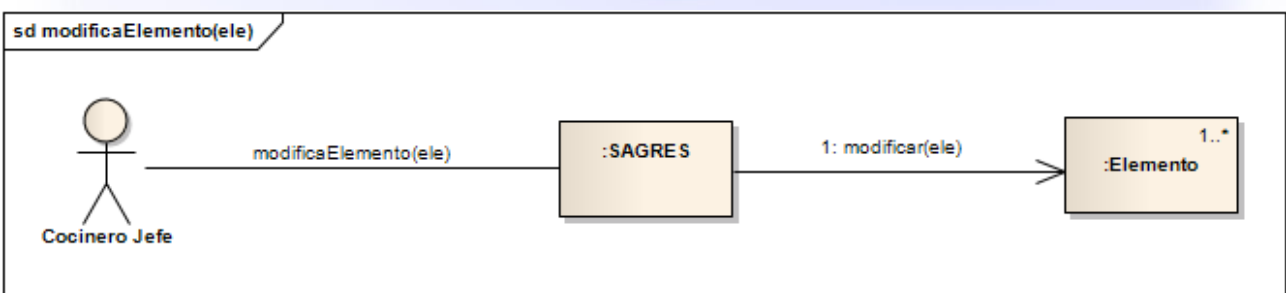
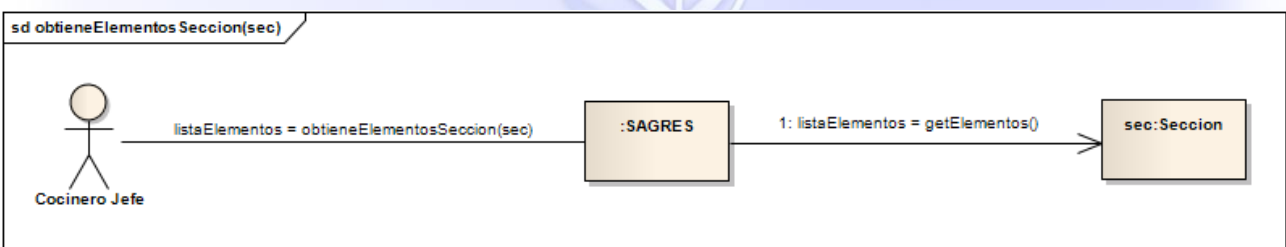
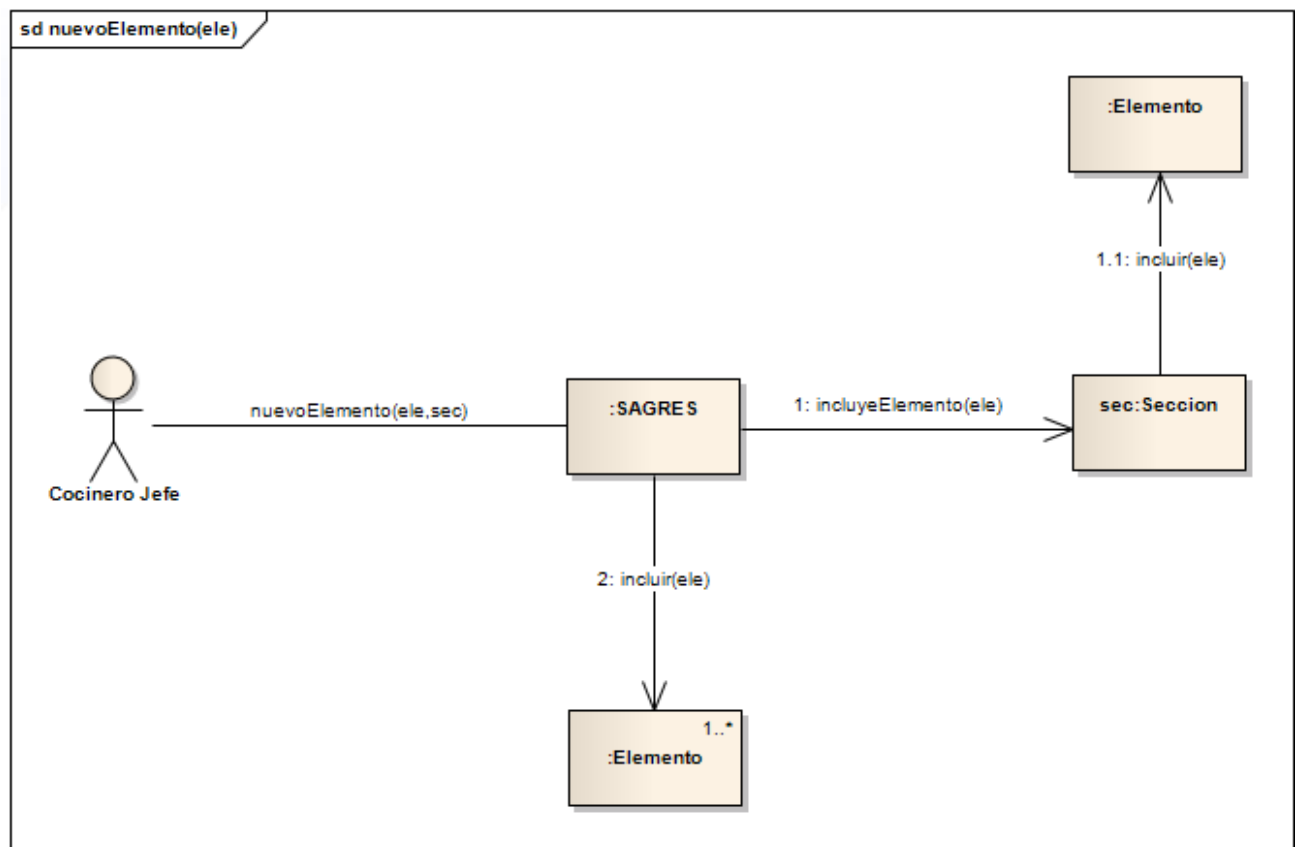
Sagres

<b>Nombre:</b>	nuevaIncidencia(in)
<b>Responsabilidad:</b>	Registrar una nueva incidencia en el sistema
<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>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó una Incidencia</li> <li>• Se asoció un producto a la incidencia</li> </ul>
<b>Salida:</b>	

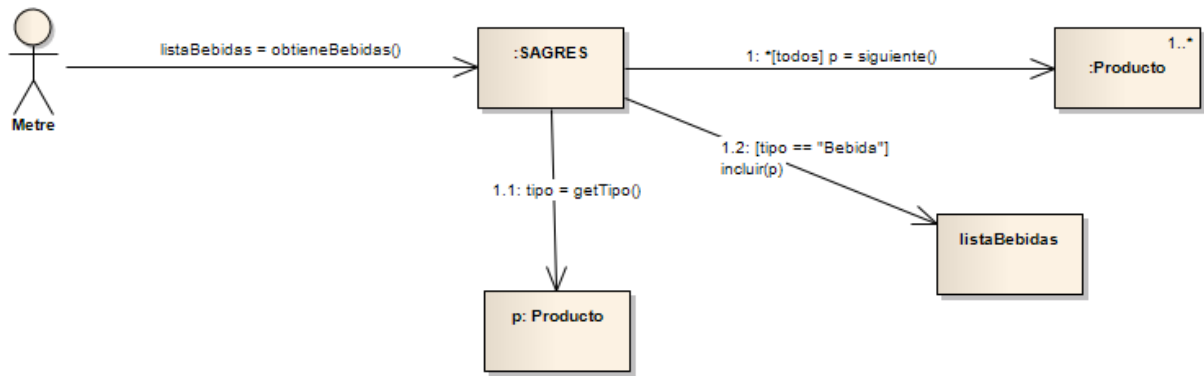


## Diagramas de colaboración para cada contrato

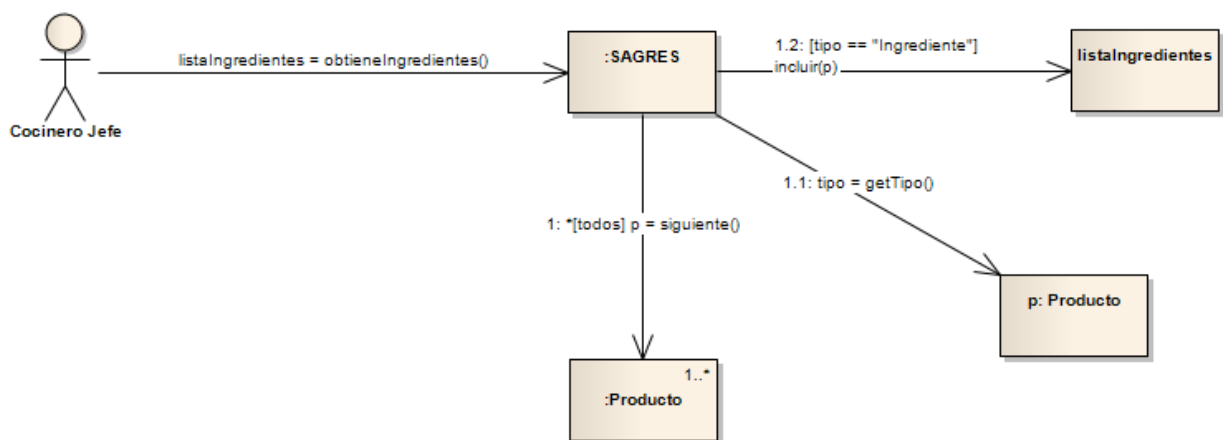




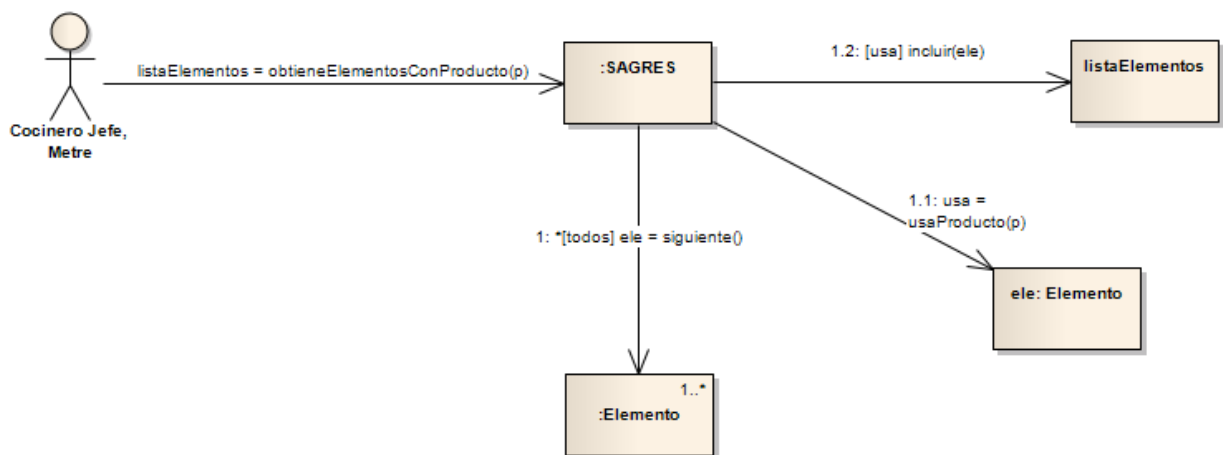
#### sd obtieneBebidas()



#### sd obtieneIngredientes()

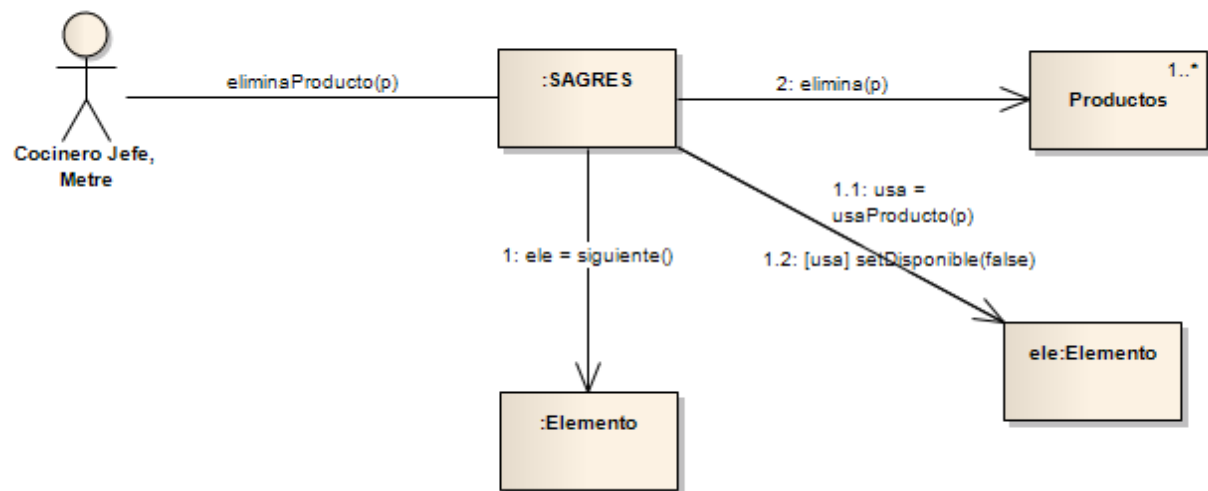


#### sd obtieneElementosConProducto(p)

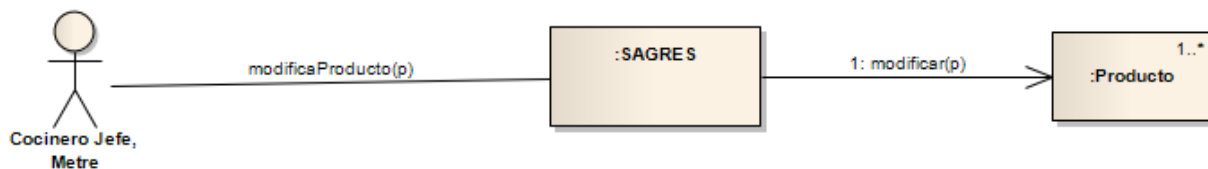




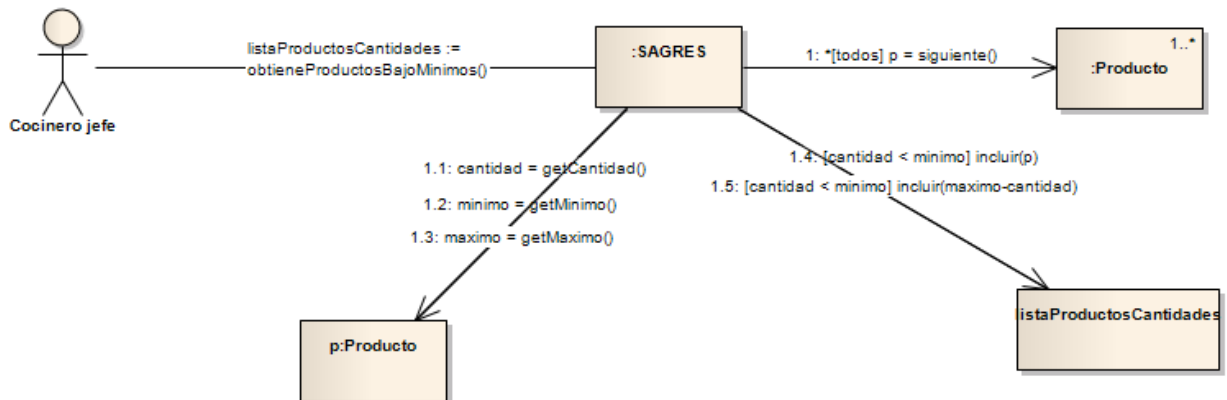
### sd eliminaProducto(p)



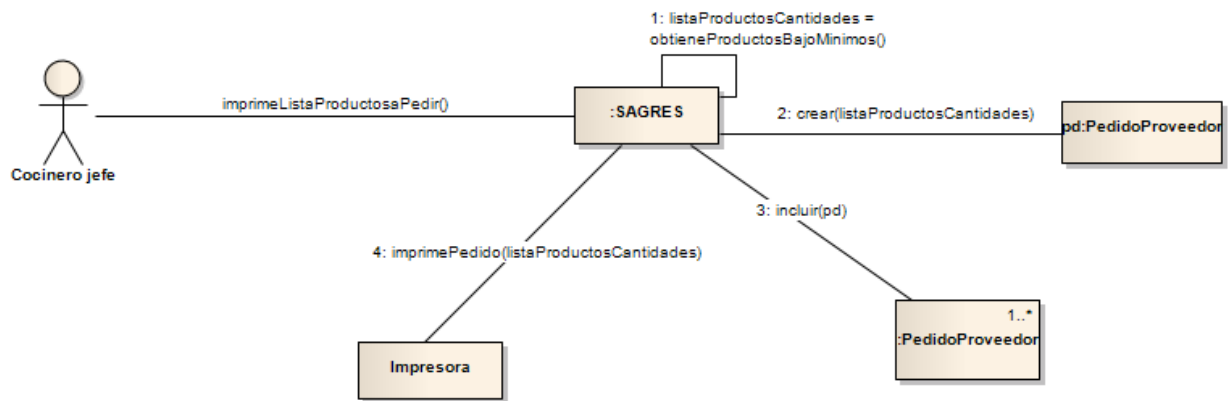
### sd modificaProducto(p)



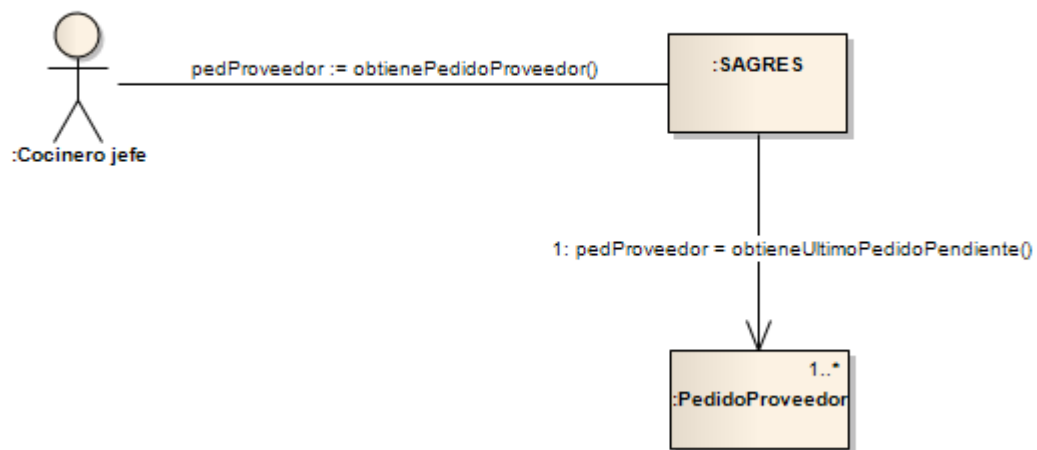
### sd obtieneProductoBajoMinimos()



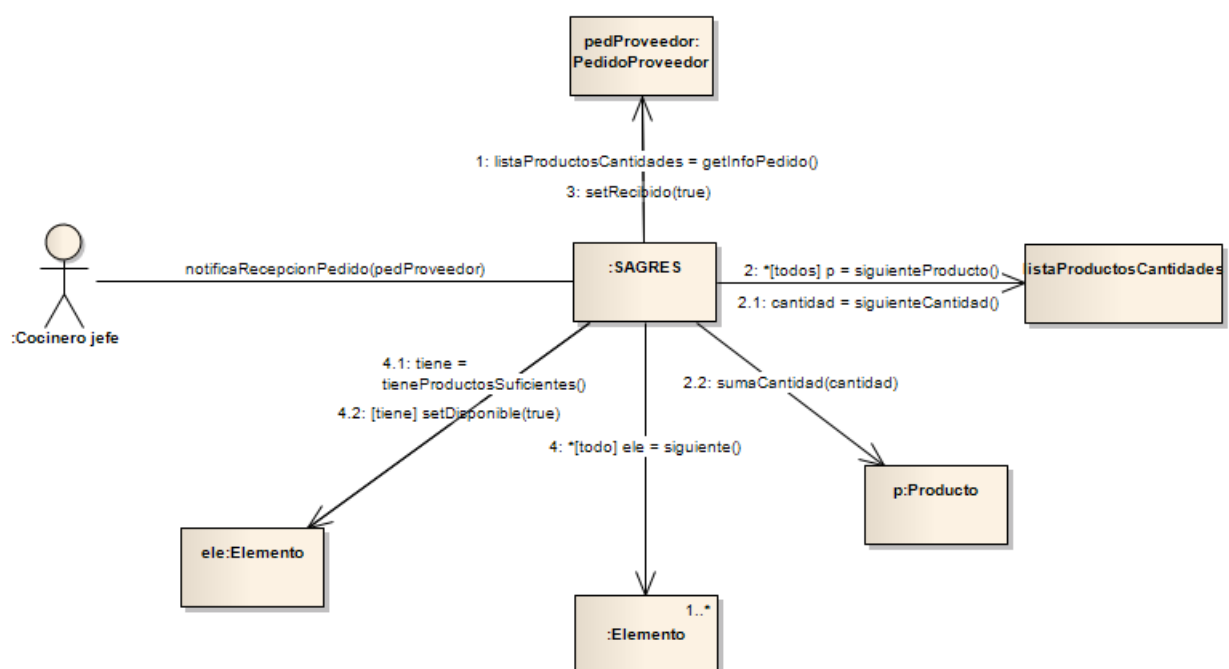
### sd imprimeListaProductosaPedir()



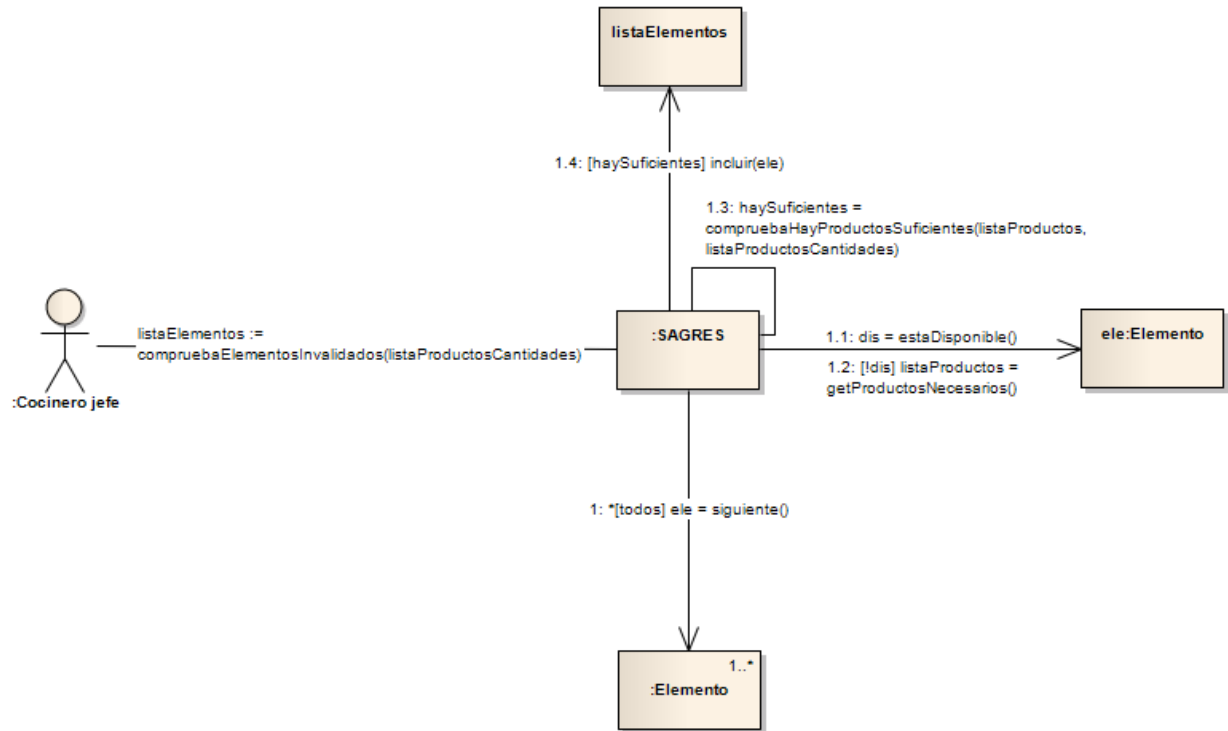
### sd obtienePedidoProveedor()



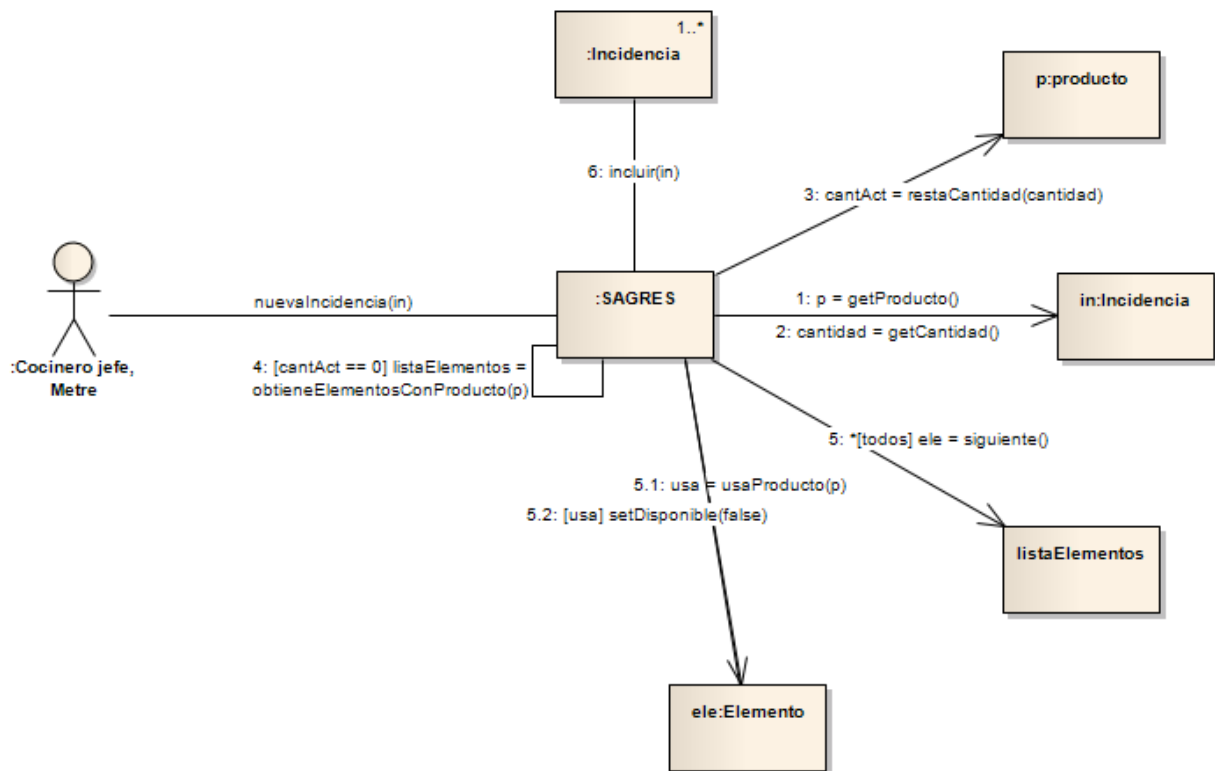
### sd notificaRecepcionPedido(pedProveedor)



sd compruebaElementosInvalidados(listaProductosCantidades)



sd nuevaIncidencia(in)



## 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

## APÉNDICE 1.1

<i>Fecha</i>	<b>16/04/2010</b>
<i>Descripción del problema</i>	Tras la revisión del documento de modelado de requisitos de la 1ª iteración han aparecido nuevos cambios en las operaciones del sistema.
<i>Impacto del problema</i>	La nueva revisión del documento de modelado de requisitos nos ha obligado a rehacer el documento de análisis para poder llevar a cabo una correcta revisión del diseño.
<i>Soluciones adoptadas</i>	<ul style="list-style-type: none"><li>• Se han modificado los contratos de las operaciones y los diagramas de colaboración</li></ul>
<i>Anexos a la versión</i>	

Sagres