

Documento de ANÁLISIS

Sagres

**Sistema de Administración y
Gestión de RESTaurantes**



Daniel Guerrero Martínez
Carlos Salas Morales
José David Dionisio Ruiz
Ángel Luis García Sánchez

V3.0

ÍNDICE DE CONTENIDO

Apartado de control de versiones.....	5
1. Modelado estático.....	6
<i>Descripción de clases conceptuales.....</i>	<i>6</i>
<i>Relación entre clases conceptuales.....</i>	<i>7</i>
<i>Atributos asociados a clases y/o relaciones.....</i>	<i>9</i>
<i>Diagrama de clases.....</i>	<i>11</i>
2. Modelo de comportamiento externo.....	12
<i>Descripción de contratos del sistema.....</i>	<i>12</i>
ObtieneSecciones().....	12
ObtieneProductosSeccion(sec).....	12
NuevoElemento(ele,sec).....	13
ObtieneElementosSeccion(sec).....	13
EliminaElemento(ele).....	14
ModificaElemento(ele).....	14
NuevoProducto(p).....	15
ObtieneIngredientes().....	15
ObtieneBebidas().....	16
obtieneElementosConProducto(p).....	16
EliminaProducto(p).....	17
ModificaProducto(p).....	17
ImprimeListaProductosAPedior().....	18
ObtienePedidoProveedor().....	19
NotificaRecepciónPedido(pedProveedor).....	19
CompruebaElementosInvalidos(listaProductosCantidades).....	20
NuevaIncidencia(in).....	20
GetSiguientePedidoCocinaEnCola().....	21
GetPedidosCocinaPreparandose().....	21
SeleccionaPlato(Pedido pedido, ElementoColaCocina plato).....	22
GetSiguientePedidoBar().....	23
SeleccionaBebida(Pedido pedido, ElementoColaBar bebida).....	24
GetSecciones().....	24
NuevoPedido(codMesa,,elementosPedido).....	25
GetPedidosMesaModificables(codMesa).....	25
GetElementosPedido(pedido).....	26

ModificaPedido(pedido, elementosPedido).....	26
PideFactura(codMesa, listaElementos).....	27
ImprimeFactura(codMesa).....	27
ConfirmaPagoFactura(codMesa).....	28
ObtienePlatoMasPedido(fechaInicio, fechaFin, seccion).....	28
ObtienePlatoMenosPedido(FechaInicio, FechaFin, seccion).....	29
ElementosAfectadosFaltaProductos(fechaInicio, FechaFin).....	29
GananciasPorMes.....	30
<i>Diagramas de colaboración para cada contrato.....</i>	<i>31</i>
ObtieneSecciones.....	31
ObtieneProductosSecciones(.....)	32
NuevoElemento.....	32
ObtieneElementosSeccion.....	33
ModificaElemento.....	33
NuevoProducto.....	34
ObtieneBebidas.....	34
ObtieneIngredientes.....	35
ObtieneElementosConProducto.....	36
EliminaProducto.....	36
ModificaProducto.....	36
ObtieneProductosBajoMinimos.....	37
ImprimeListaProductosAPedir.....	37
ObtienePedidoProveedor.....	38
NotificaRecepcionPedido.....	38
CompruebaElementosInvalidos.....	39
nuevaIncidencia.....	39
GetSiguientePedidoCocinaEnCola().....	40
GetPedidosCocinaPreparandose.....	40
SeleccionaPlato.....	41
GetSiguientePedidoBar.....	41
SeleccionaBebida.....	42
GetSecciones.....	43
NuevoPedido.....	43
GetPedidosMesaModificables.....	44
GetElementosPedido.....	44
ModificaPedido.....	45
ConfirmaPagoFactura.....	45

ImprimeFactura.....	46
PideFactura.....	46
ObtenerPlatoMasPedido.....	47
ObtienePlatoMenosPedido.....	48
ElementoAfectadosFaltaProductos.....	48
GananciasProMes.....	49
Apéndice 1.0.....	50
Apéndice 1.1.....	51
Apéndice 2.0.....	52
Apéndice 2.1.....	53
Apéndice 3.0.....	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	22-03-2010	Documento de análisis inicial.
V1.1	16-04-2010	Revisión del documento de análisis de la 1ª iteración
V2.0	23/04/10	Se ha creado el documento de análisis 2ª Iteración.
V2.1	07/05/10	Se han corregido fallos en algunos contratos y diagramas de colaboración.
V3.0	22/05/10	Se ha creado el documento de análisis 3ª Iteración

1. MODELADO 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

<i>Clase</i>	<i>Descripción</i>
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.
Pedido	Representa un pedido de cliente. Guarda toda la información relevante al sistema de un pedido.
Factura	En esta clase se guardan todas las facturas de los clientes.

Restaurante	En esta clase se guardan los datos del restaurante (Nombre, dirección, razón social...)
ElementoPedido	Representa un elemento de un pedido determinado seleccionado por el cliente. Almacena la información relevante del plato o bebida que representa.
ElementoColaBar	Es un elemento de pedido de tipo bebida. Representa las bebidas seleccionadas por el cliente en el pedido.
ElementoColaCocina	Es un elemento de pedido de tipo plato. Representa los platos seleccionados por el cliente en el pedido.

Relación entre clases conceptuales

<i>Clases</i>	<i>Relación</i>
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.
Pedido - Factura	Es una relación de asociación entre pedido y factura. Una factura se refiere a uno o varios pedidos.

Restaurante - Factura	Es una relación de asociación entre restaurante y factura. así la factura puede obtener la información necesaria del restaurante.
Pedido - ElementoPedido	Es una relación de asociación, la cual establece que un pedido contiene uno o muchos elementos pedidos por el cliente.
ElementoColaBar - ElementoPedido - ElementoColaCocina	Se establece una generalización entre estas tres clases en la que ElementoPedido es la clase padre y las otras dos son las que heredan de ella.
ElementoColaBar - ElementoBebida	Es una relación de asociación en la que un ninguno o muchos ElementoColaBar hacen referencia a un único ElementoBebida.
ElementoColaCocina - ElementoPlato	Es una relación de asociación en la que un ninguno o muchos ElementoColaCocina hacen referencia a un único ElementoPlato.



The logo for 'Sagres' is centered on the page. It features a stylized, light blue circular emblem above the word 'Sagres' written in a flowing, cursive script. The background of the page is a light blue gradient with a large, faint circular shape on the right side.

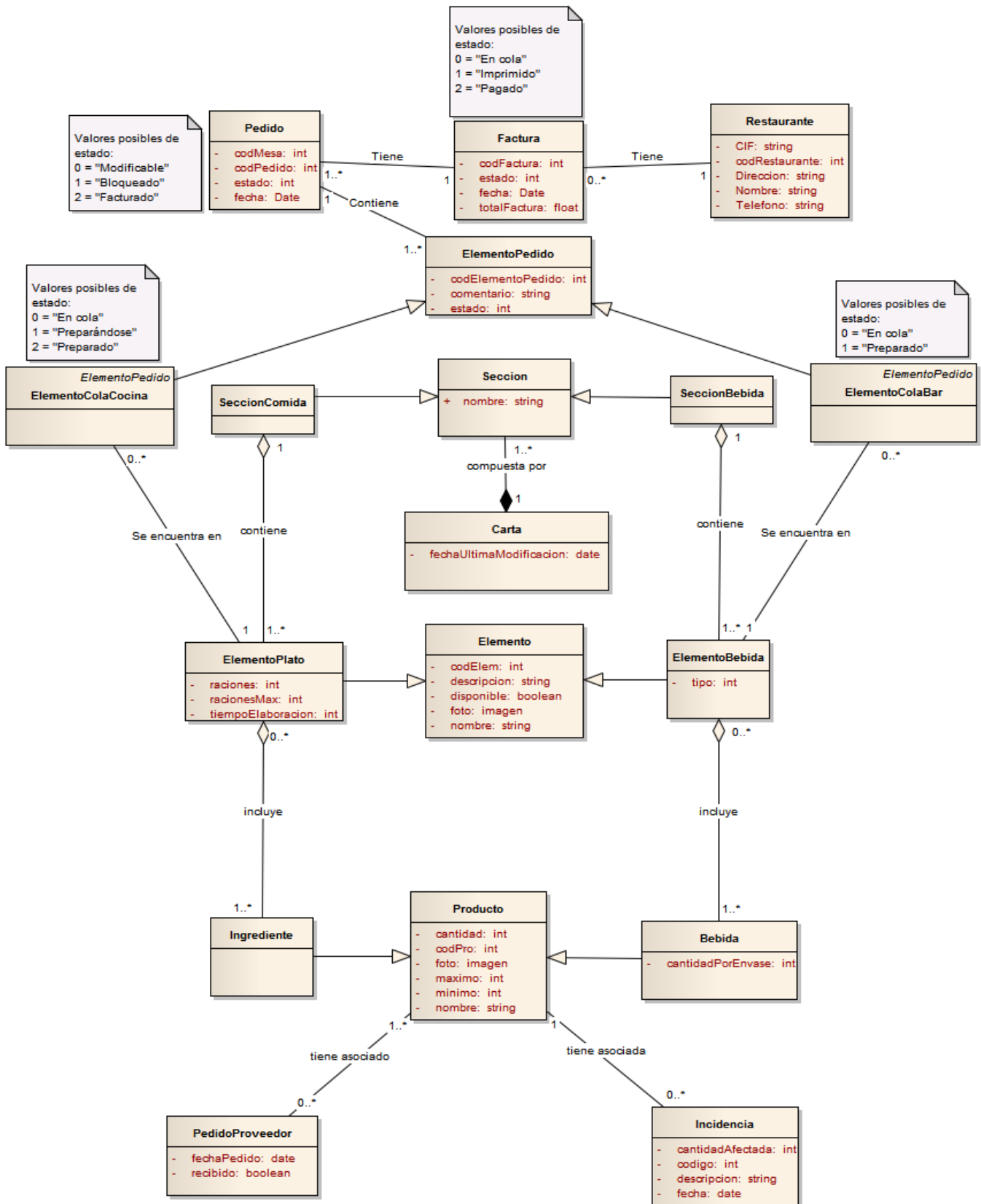
Atributos asociados a clases y/o relaciones

Clase	Atributo	Descripción
Pedido	codMesa : int	Código que identifica a una mesa. Por mesa, se entiende un terminal de cliente.
	codPedido: int	Código único que identifica a un pedido.
	estado : int	Estado de un pedido, este puede variar entre “Modificable”, “Bloqueado” o “Facturado”.
	fecha : Date	Fecha, hora, minuto y segundo en el cual fue creado el pedido.
Factura	codFactura : int	Código único identificador de la factura.
	estado : int	Indica el estado de la factura, 0=en cola, 1=imprimido, 2=pagado, 3=anulado.
	fecha : Date	La fecha en la que se ha pagado la factura.
Restaurante	CIF : string	Código de identificación fiscal del restaurante.
	CodRestaurante : int	Código del restaurante.
	Direccion : string	Dirección del restaurante.
	Nombre : string	Nombre del restaurante.
	Telefono : string	Teléfono del restaurante.
ElementoPedido	codElementoPedido : int	Código identificador del elemento de pedido.
	comentario : string	Comentario realizado por el cliente a la hora de seleccionar el pedido. Se permite que haga anotaciones al cocinero como “Sin cebolla, por favor”.
	estado : int	Estado del elemento de pedido, puede variar, según si es un plato (“En cola”, “Preparándose”, “Preparado”) o si es una bebida (“En cola”, “Preparado”).
Elemento	codElemento : int	Número que identifica a un elemento.
	descripción : int	Breve descripción del elemento.
	disponible : int	Nos dice si un elemento está disponible para su pedido o está invalidado por cualquier motivo.
	diviMax : int	Número de divisiones que se pueden hacer del elemento (en el caso de un plato raciones, en el caso de una bebida vasos).
	foto : imagen	Imagen para visualizar el elemento.
	nombre : string	Nombre del elemento que se visualizará en la carta.
	precio : int	Precio de venta al público del elemento.

ElementoPlato	tiempoElaboracion : int	Tiempo en minutos que se tarda en elaborar el plato. Solo se tiene en cuenta en el caso de que no haya raciones.
Seccion	Nombre : string	Nombre de la sección.
Carta	fechaUltimaModificacion : Date	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.
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.

Diagrama de clases

class Diagrama de clases



2. MODELO DE COMPORTAMIENTO EXTERNO

Descripción de contratos del sistema

ObtieneSecciones()

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

ObtieneProductosSeccion(sec)

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

NuevoElemento(ele,sec)

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

ObtieneElementosSeccion(sec)

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

EliminaElemento(ele)

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

ModificaElemento(ele)

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

NuevoProducto(p)

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

ObtieneIngredientes()

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

ObtieneBebidas()

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

obtieneElementosConProducto(p)

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

EliminaProducto(p)

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

ModificaProducto(p)

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

ObtieneProductosBajoMinimos()

Nombre:	ListaProductosCantidades = obtieneProductosBajoMinimos()
Responsabilidad:	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.
Tipo:	Sistema
Referencias:	Caso de uso “Imprimir lista Productos a Pedir”
Nota:	
Excepción:	<ul style="list-style-type: none">• Si no hay productos registrados en el sistema• Si no hay productos que estén bajo mínimos
Precondición:	
Postcondición:	
Salida:	<ul style="list-style-type: none">• 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.

ImprimeListaProductosAPedior()

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

ObtienePedidoProveedor()

Nombre:	PedProveedor = obtienePedidoProveedor()
Responsabilidad:	Obtener el pedido al proveedor pendiente más antiguo
Tipo:	Sistema
Referencias:	Caso de uso “Notificar Recepción de pedido del proveedor”
Nota:	
Excepción:	<ul style="list-style-type: none">• Si no hay pedidos pendientes
Precondición:	
Postcondición:	
Salida:	<ul style="list-style-type: none">• El pedido al proveedor más antiguo

NotificaRecepciónPedido(pedProveedor)

Nombre:	notificaRecepciónPedido(pedProveedor)
Responsabilidad:	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.
Tipo:	Sistema
Referencias:	Caso de uso “Notificar Recepción de pedido del proveedor”
Nota:	
Excepción:	
Precondición:	
Postcondición:	<ul style="list-style-type: none">• Se modifico el atributo Producto.cantidad de varios productos• Se modificó el atributo PedidoProveedor.recibido a TRUE.
Salida:	

CompruebaElementosInvalidos(listaProductosCantidades)

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

NuevaIncidencia(in)

Nombre:	nuevaIncidencia(in)
Responsabilidad:	Registrar una nueva incidencia en el sistema
Tipo:	Sistema
Referencias:	Caso de uso “Notificar Incidencia con Bebida” y “Notificar Incidencia con Ingrediente”
Nota:	
Excepción:	
Precondición:	
Postcondición:	<ul style="list-style-type: none">• Se creó una Incidencia• Se asoció un producto a la incidencia
Salida:	

GetSiguientePedidoCocinaEnCola()

Nombre:	Pedido = getSiguientePedidoCocinaEnCola()
Responsabilidad:	Obtener el siguiente pedido que lleve más tiempo en cola de cocina, es decir, con la fecha más lejana (el más antiguo), que tenga algún ElementoColaCocina en estado “En cola”.
Tipo:	Sistema
Referencias:	Caso de uso “Cambia Estado Plato”
Nota:	
Excepción:	<ul style="list-style-type: none">• Si no hay ningún pedido con algún ElementoColaCocina en estado “En cola”.
Precondición:	
Postcondición:	
Salida:	Un pedido con algún ElementoColaCocina en estado “En cola”, con la fecha más antigua posible.

Realizado por Sergio Rodríguez Lumley

GetPedidosCocinaPreparandose()

Nombre:	Pedido = getPedidosCocinaPreparandose()
Responsabilidad:	Obtener todos aquellos pedidos con algún plato en estado “Preparandose”.
Tipo:	Sistema
Referencias:	Caso de uso “Cambia Estado Plato”
Nota:	
Excepción:	<ul style="list-style-type: none">• Si no hay ningún pedido con algún ElementoColaCocina en estado “Preparandose”.
Precondición:	
Postcondición:	
Salida:	Todos los pedidos que tengan algún plato en estado “Preparandose”.

Realizado por Sergio Rodríguez Lumley

SeleccionaPlato(Pedido pedido, ElementoColaCocina plato)

Nombre:	Bool = seleccionaPlato(Pedido pedido, ElementoColaCocina plato)
Responsabilidad:	Seleccionar un plato del pedido “pedido” y cambiar el estado del plato. Si estaba “En cola” lo cambia a “Preparandose”, el del pedido a “Bloqueado” y actualiza la cantidad de productos asociados a ese plato. Si estaba “Preparandose” se cambiará a “Preparado”.
Tipo:	Sistema
Referencias:	Caso de uso “Cambia Estado Plato”
Nota:	
Excepción:	<ul style="list-style-type: none">• Si alguno de los parámetros recibidos es incoherente (pedido nulo, plato nulo, etc).• Si el plato no pertenece al pedido.
Precondición:	
Postcondición:	<ul style="list-style-type: none">• Se modificó un pedido.• Se modificó el estado de un plato al valor “Preparandose” o “Preparado”.• Se modificaron los ingredientes asociados al plato.
Salida:	Booleano con valor “Verdadero” si ha cambiado el estado del plato, “Falso” en caso contrario.

Realizado por Sergio Rodríguez Lumley

GetSiguientePedidoBar()

Nombre:	Pedido = getSiguientePedidoBar()
Responsabilidad:	Obtener el siguiente pedido que lleve más tiempo en cola de bar, es decir, con la fecha más lejana (el más antiguo), que tenga algún ElementoColaBar en estado “En cola”.
Tipo:	Sistema
Referencias:	Caso de uso “Cambia Estado Bebida”
Nota:	
Excepción:	<ul style="list-style-type: none">• Un pedido con algún ElementoColaBar en estado “En cola”, con la fecha más antigua posible.
Precondición:	
Postcondición:	
Salida:	Un pedido con algún ElementoColaBar en estado “En cola”, con la fecha más antigua posible.

Realizado por Sergio Rodríguez Lumley

Sagres

SeleccionaBebida(Pedido pedido, ElementoColaBar bebida)

Nombre:	Bool = seleccionaBebida(Pedido pedido, ElementoColaBar bebida)
Responsabilidad:	Seleccionar una bebida del pedido “pedido”, cambiar el estado del pedido a “Bloqueado” y también el de la bebida. Si esta estaba “En cola” se cambiará a “Preparado” y reducirá la cantidad de bebidas en stock.
Tipo:	Sistema
Referencias:	Caso de uso “Cambia Estado Bebida”
Nota:	
Excepción:	<ul style="list-style-type: none">• Si alguno de los parámetros recibidos es incoherente (pedido nulo, bebida nula, etc).• Si la bebida no pertenece al pedido.
Precondición:	
Postcondición:	<ul style="list-style-type: none">• Se modificó un pedido.• Se modificó el estado de una bebida al valor “Preparado”.• Se modificaron las bebidas asociadas al producto.
Salida:	Booleano con valor “Verdadero” si ha cambiado el estado de la bebida, “Falso” en caso contrario.

Realizado por Sergio Rodríguez Lumley

GetSecciones()

Nombre:	elementosCarta = getSecciones()
Responsabilidad:	Operación encargada de devolver las secciones de la carta con sus correspondientes elementos.
Tipo:	Sistema.
Referencias:	Nuevo Pedido, Modifica Pedido.
Nota:	
Excepción:	
Precondición:	
Postcondición:	
Salida:	<ul style="list-style-type: none">• Una lista de elementos tipo sección de carta.

Realizado por Adrián Víctor Pérez Lopera

NuevoPedido(codMesa,,elementosPedido)

Nombre:	nuevoPedido (codMesa, elementosPedido)
Responsabilidad:	Operación encargada de incluir en el sistema un nuevo pedido (con sus elementos) asociado a una mesa.
Tipo:	Sistema.
Referencias:	Nuevo Pedido.
Nota:	El parámetro “codMesa” de la operación, se especificaba en el DMR como “mesa”. Se ha cambiado ya que en este punto ya sí sabemos que las mesas van a ser referenciadas por un código.
Excepción:	
Precondición:	<ul style="list-style-type: none">• El parámetro “codMesa” es correcto.• Los miembros del parámetro “elementosPedido” son correctos.
Postcondición:	<ul style="list-style-type: none">• Se creó un pedido con estado “modificable”.• Se crearon elementos de pedido con estado “en cola”.• Se crearon enlaces entre el pedido y los elementos de pedido.• Se crearon enlaces entre los elementos de pedido y los elementos plato y/o elementos bebida de la carta.
Salida:	

Realizado por Adrián Víctor Pérez Lopera

GetPedidosMesaModificables(codMesa)

Nombre:	pedidosMesa = getPedidosMesaModificables(codMesa)
Responsabilidad:	Operación encargada de obtener los pedidos con estado “modificable” de una mesa.
Tipo:	Sistema.
Referencias:	Modifica Pedido.
Nota:	El parámetro “codMesa” de la operación, se especificaba en el DMR como “mesa”. Se ha cambiado ya que en este punto ya sí sabemos que las mesas van a ser referenciadas por un código.
Excepción:	
Precondición:	<ul style="list-style-type: none">• El parámetro “codMesa” es correcto.
Postcondición:	
Salida:	<ul style="list-style-type: none">• Una lista de pedidos.

Realizado por Adrián Víctor Pérez Lopera

GetElementosPedido(pedido)

Nombre:	elementosPedido = getElementosPedido(pedido)
Responsabilidad:	Operación encargada de obtener los elementos de un pedido con su cantidad.
Tipo:	Sistema.
Referencias:	Modifica Pedido.
Nota:	
Excepción:	
Precondición:	<ul style="list-style-type: none">• El parámetro “pedido” es correcto.
Postcondición:	
Salida:	<ul style="list-style-type: none">• Una lista de elementos de pedido.

Realizado por Adrián Víctor Pérez Lopera

ModificaPedido(pedido, elementosPedido)

Nombre:	modificaPedido (pedido, elementosPedido)
Responsabilidad:	Operación encargada de destruir el pedido al que hace referencia “pedido” y crear un nuevo pedido con los elementos de “elementosPedido”.
Tipo:	Sistema.
Referencias:	Modifica Pedido.
Nota:	
Excepción:	
Precondición:	<ul style="list-style-type: none">• El parámetro “pedido” es correcto.• Los miembros del parámetro “elementosPedido” son correctos.
Postcondición:	<ul style="list-style-type: none">• Se eliminó un pedido y sus elementos asociados.• Se creó un pedido con estado “modificable”.• Se crearon elementos de pedido con estado “en cola”.• Se crearon enlaces entre el pedido y lo elementos de pedido.• Se crearon enlaces entre los elementos de pedido y los elementos plato y/o elementos bebida.
Salida:	

Realizado por Adrián Víctor Pérez Lopera

PideFactura(codMesa, listaElementos)

Nombre:	Confirmación = pideFactura(codMesa,listaElementos)
Responsabilidad:	Inserta la petición de imprimir la facturas de pedido(s) que contiene listaElementos en la cola de facturas.
Tipo:	Sistema
Referencias:	Caso de uso “Pide factura”
Nota:	
Excepción:	<ul style="list-style-type: none">• El código de mesa es incorrecto.• La lista de de elementos está vacía.
Precondición:	<ul style="list-style-type: none">• El cliente ha hecho un pedido.
Postcondición:	<ul style="list-style-type: none">• Inserción de la petición en la cola de facturas.
Salida:	

Realizado por nabil

ImprimeFactura(codMesa)

Nombre:	imprimeFactura(codMesa)
Responsabilidad:	Imprime la factura de la mesa con el código (codMesa)
Tipo:	sistema
Referencias:	Caso de uso “Cambia Estado Factura”
Nota:	
Excepción:	Código de mesa incorrecto.
Precondición:	
Postcondición:	
Salida:	

Realizado por nabil

ConfirmaPagoFactura(codMesa)

Nombre:	ConfirmaPagoFactura(codMesa)
Responsabilidad:	Confirmar el pago de una factura de la mesa con el código (codMesa) y buscar si hay otro(s) elementos(s) servido(s) después de la impresión de esta factura. En caso de que esto ocurra se imprima la factura nueva que contiene todos los elementos con el total.
Tipo:	sistema
Referencias:	Caso de uso “Cambia Estado Factura”
Excepción:	
Precondición:	
Postcondición:	<ul style="list-style-type: none">Cola de facturas actualizada.
Salida:	

Realizado por nabil

ObtienePlatoMasPedido(fechaInicio, fechaFin, seccion)

Nombre:	Elemento = obtenienePlatoMasPedido(fechaInicio, fechaFin, seccion)
Responsabilidad:	Obtiene el plato que más ha sido pedido en el restaurante durante un plazo de tiempo. También es posible indicar la sección en que queremos que se centre.
Tipo:	Sistema
Referencias:	Caso de uso “Consultar Estadísticas”
Nota:	Si no se especifica sección se realiza la consulta sobre todas las secciones. Si no se especifican fechas se toma la fecha como nula, en este caso: <ul style="list-style-type: none">Si es la fechaInicio nula se comenzará desde el primer pedidoSi es la fechaFin nula se llegará hasta el último pedido.Si ambas son nulas se actúa sobre todos los pedidos.
Excepción:	<ul style="list-style-type: none">No se han realizado pedidos aún en el restaurante.
Precondición:	<ul style="list-style-type: none">fechaInicio debe ser menor que fechaFin o ser nula.fechaFin debe ser menor o igual que la fecha actual o ser nula.Seccion debe existir o ser nula
Postcondición:	
Salida:	Un objeto Elemento

Realizado por Carlos Salas

ObtienePlatoMenosPedido(FechaInicio, FechaFin, seccion)

Nombre:	Elemento = obtenienePlatoMenosPedido(fechaInicio, fechaFin, seccion)
Responsabilidad:	Obtiene el plato que menos ha sido pedido en el restaurante durante un plazo de tiempo. También es posible indicar la sección en que queremos que se centre.
Tipo:	Sistema
Referencias:	Caso de uso “Consultar Estadísticas”
Nota:	Si no se especifica sección se realiza la consulta sobre todas las secciones. Si no se especifican fechas se toma la fecha como nula, en este caso: <ul style="list-style-type: none">• Si es la fechaInicio nula se comenzará desde el primer pedido• Si es la fechaFin nula se llegará hasta el último pedido.• Si ambas son nulas se actúa sobre todos los pedidos.
Excepción:	<ul style="list-style-type: none">• No se han realizado pedidos aún en el restaurante.
Precondición:	<ul style="list-style-type: none">• fechaInicio debe ser menor que fechaFin o ser nula.• fechaFin debe ser menor o igual que la fecha actual o ser nula.• Seccion debe existir o ser nula
Postcondición:	
Salida:	Un objeto Elemento

Realizado por Carlos Salas

ElementosAfectadosFaltaProductos(fechaInicio, FechaFin)

Nombre:	dataSet = elementosAfectadosFaltaProductos(fechaInicio, fechaFin)
Responsabilidad:	Obtiene los datos de los elementos que se han visto implicados por la falta de algún producto durante un determinado plazo de tiempo.
Tipo:	Sistema
Referencias:	Caso de uso “Consultar Estadísticas”
Excepción:	<ul style="list-style-type: none">• Si no ha faltado ningún producto durante el tiempo determinado .
Precondición:	<ul style="list-style-type: none">• fechaInicio debe ser menor que fechaFin.• fechaFin debe ser menor o igual que la fecha actual.
Postcondición:	
Salida:	Un objeto DefaultCategoryDataset con los datos de los elementos que se han visto afectados por la falta de cada producto.

Realizado por Angel

GananciasPorMes

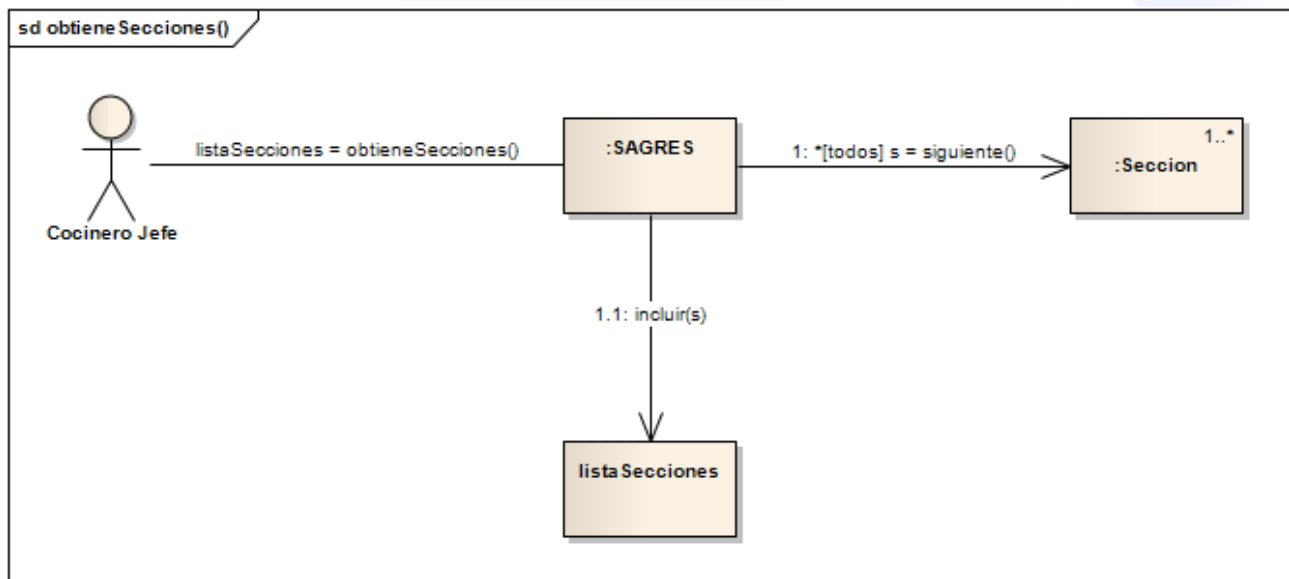
Nombre:	ganancias = gananciasPorMes(fechaInicio, fechaFin)
Responsabilidad:	Obtiene las ganancias totales divididas por mes y año.
Tipo:	Sistema
Referencias:	Caso de uso "Consultar Estadísticas"
Nota:	
Excepción:	
Precondición:	<ul style="list-style-type: none">• fechaInicio debe ser menor que fechaFin.• fechaFin debe ser menor o igual que la fecha actual.
Postcondición:	
Salida:	Devuelve un objeto DefaultCategoryDataset con las ganancias distribuidas por mes y año, indicados con el formato MM/YYYY

Realizado por Dioni

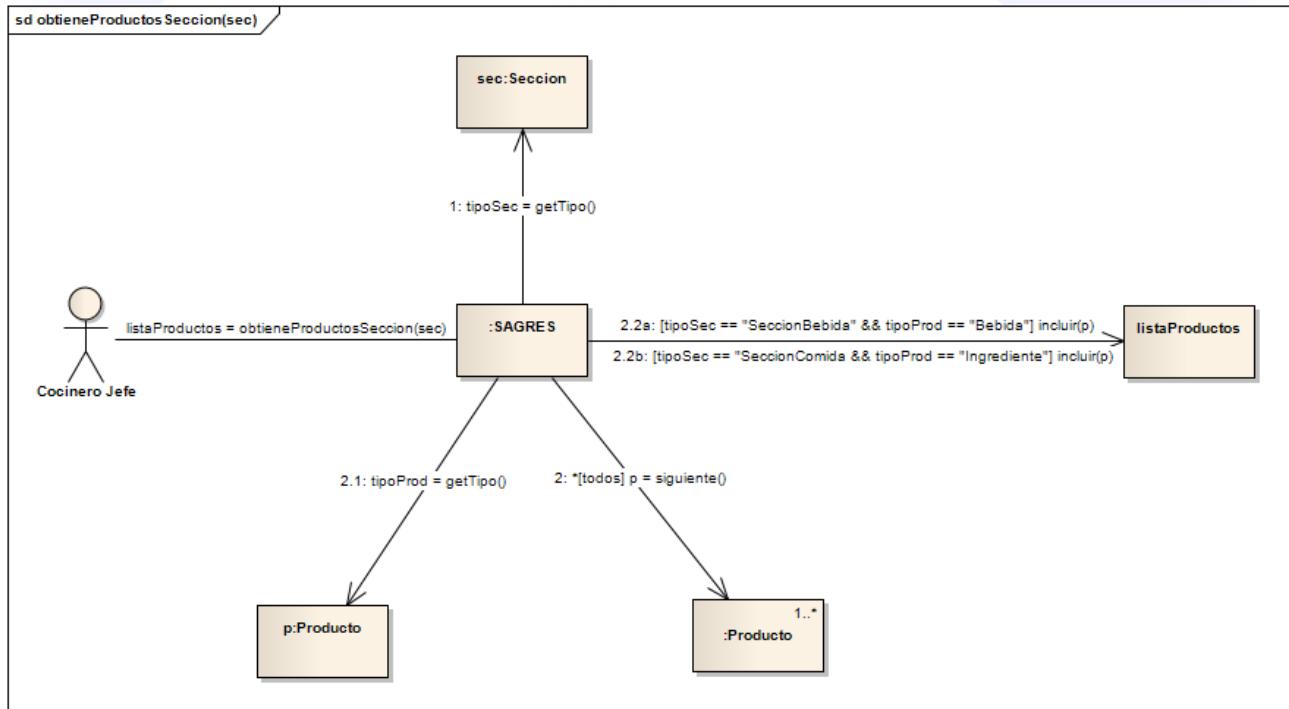
Sagres

Diagramas de colaboración para cada contrato

ObtieneSecciones

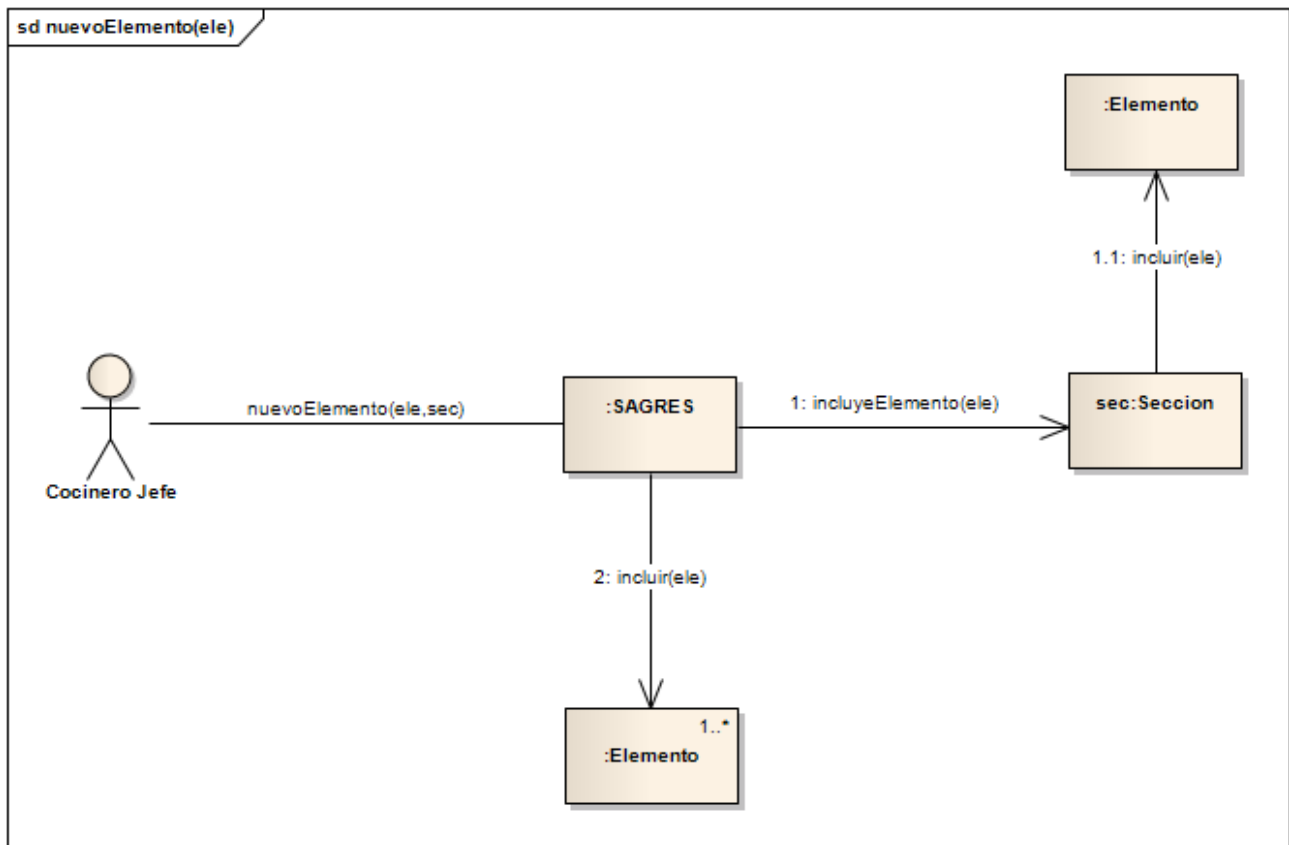


ObtieneProductosSecciones()

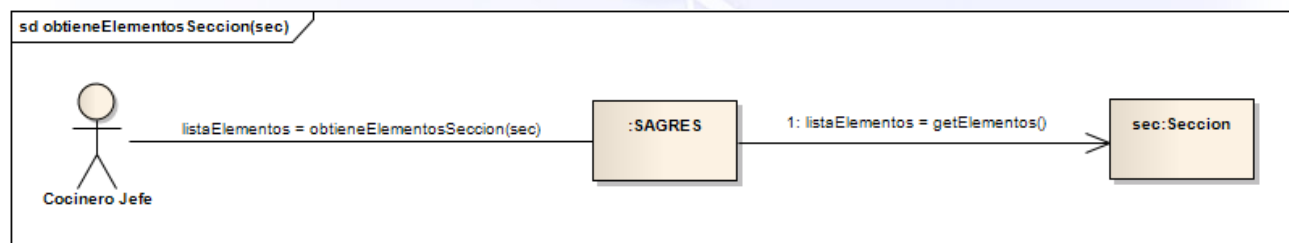


NuevoElemento

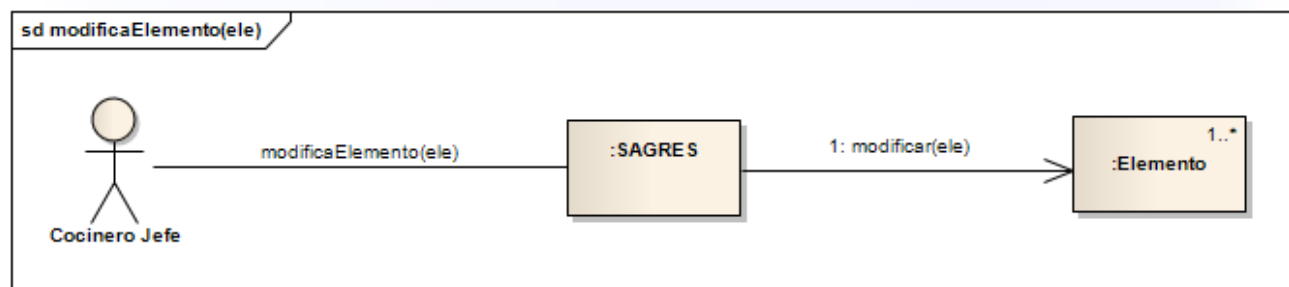
Sagres



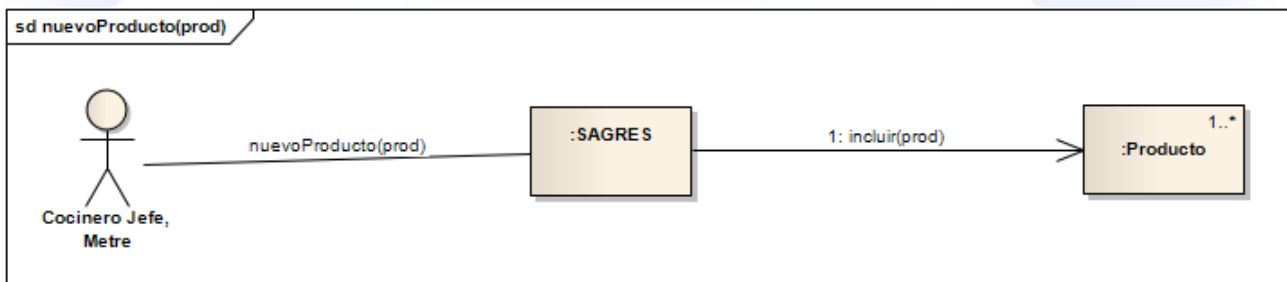
ObtieneElementosSeccion



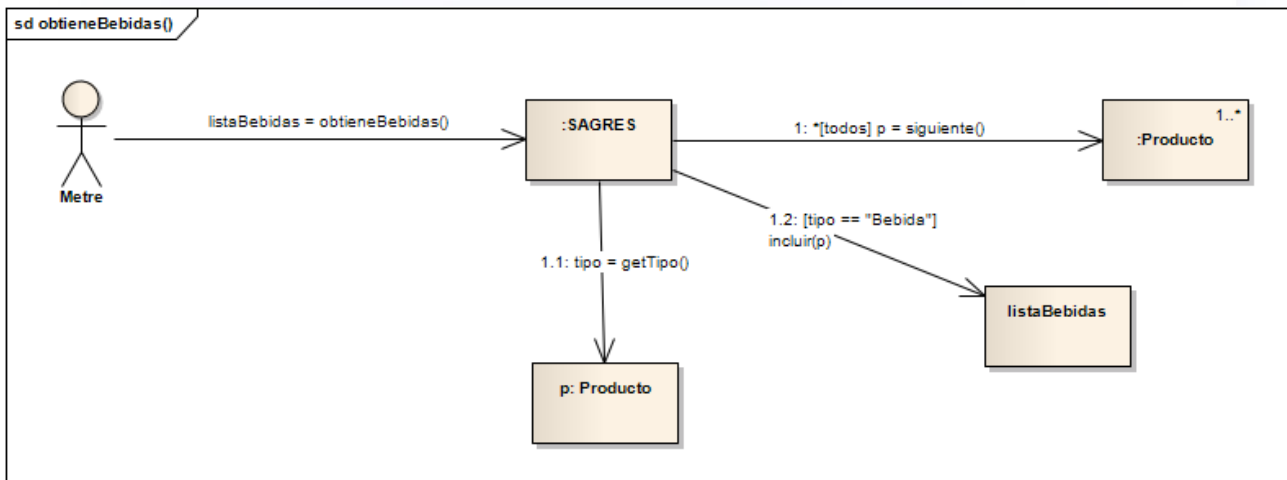
ModificaElemento



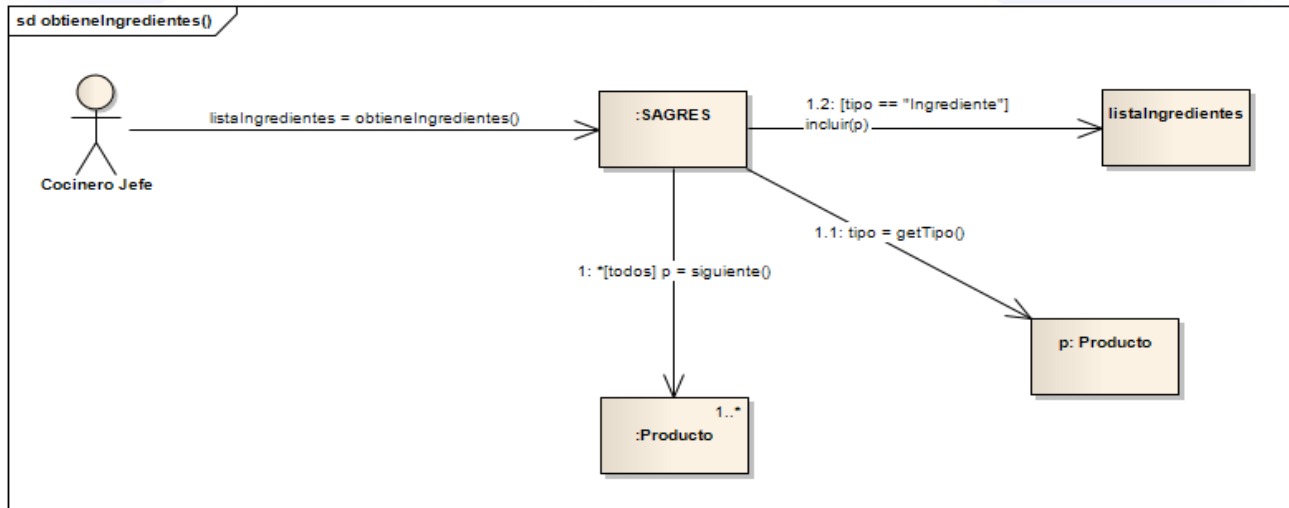
NuevoProducto



ObtieneBebidas

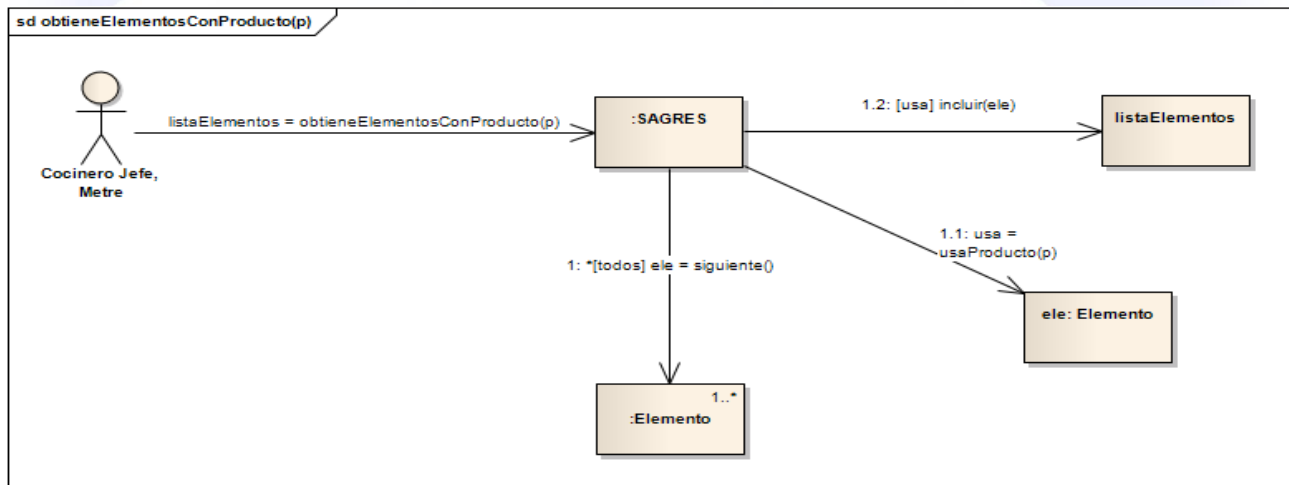


ObtieneIngredientes

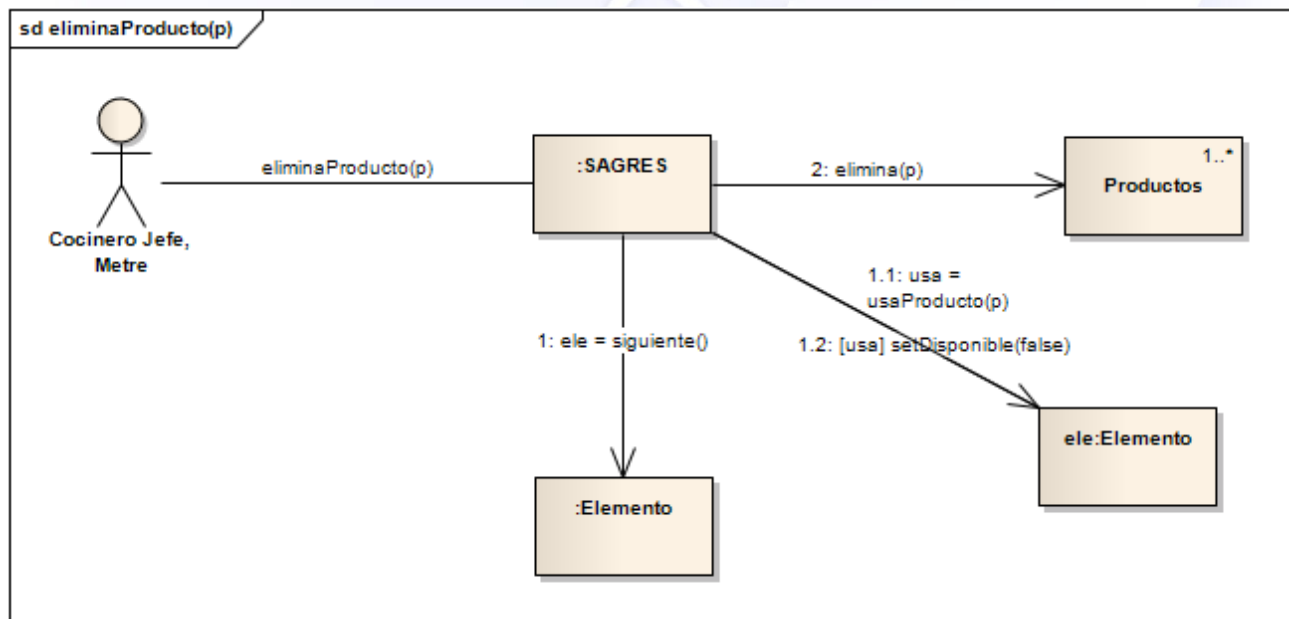


Sagres

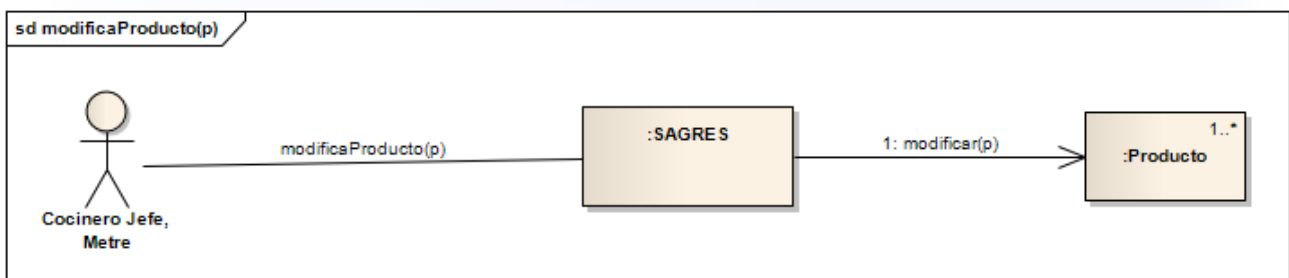
ObtieneElementosConProducto



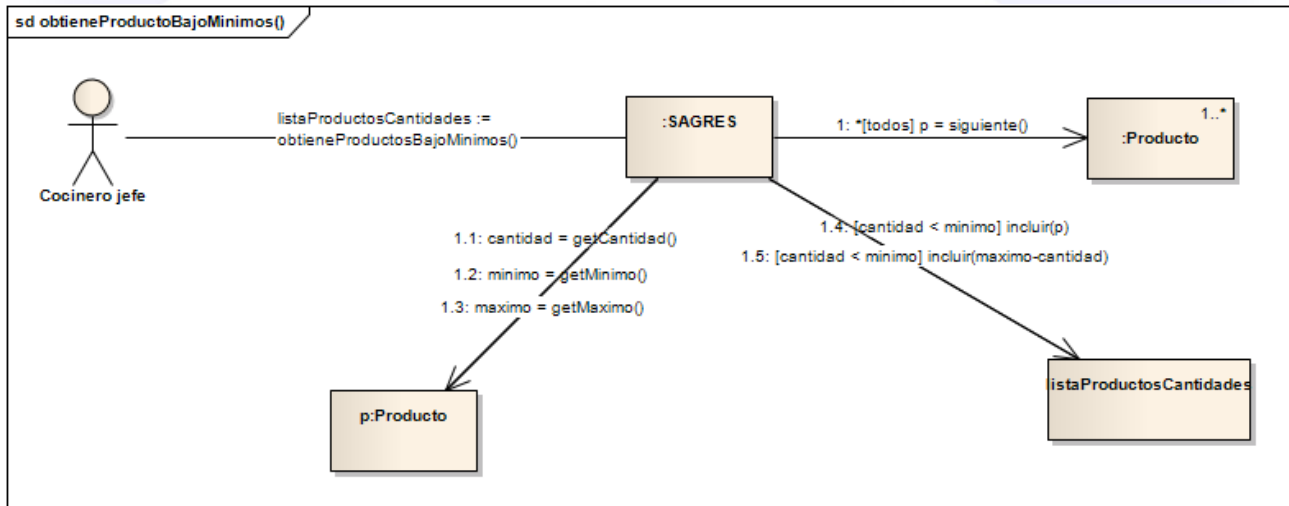
EliminaProducto



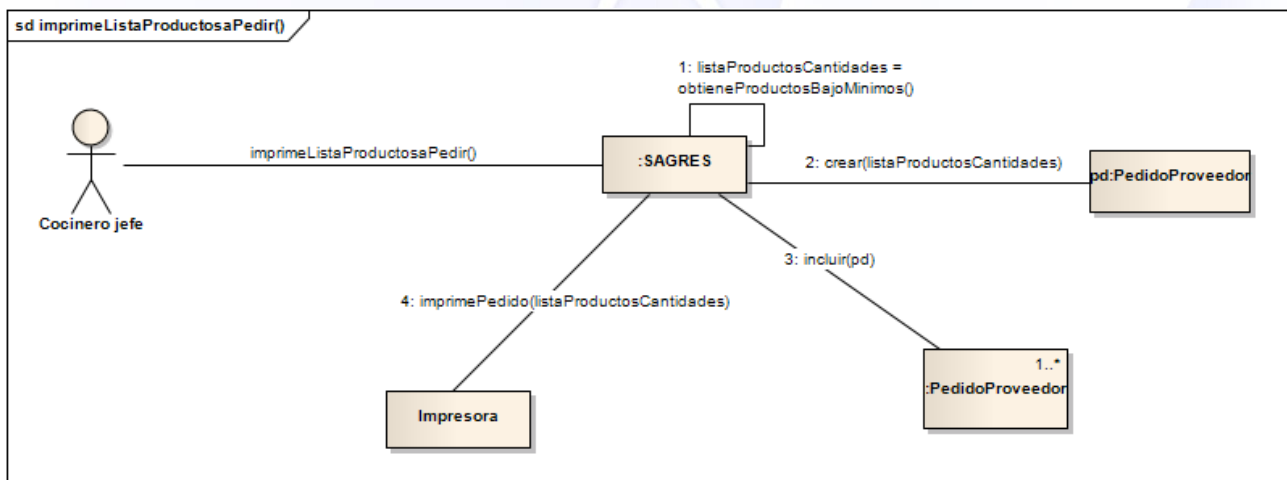
ModificaProducto



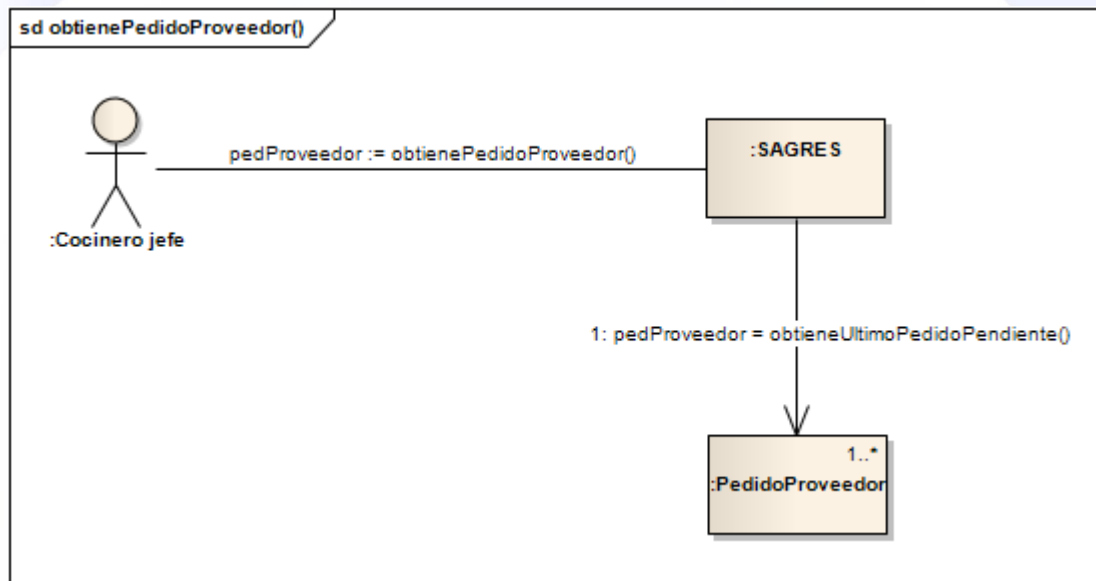
ObtieneProductosBajoMinimos



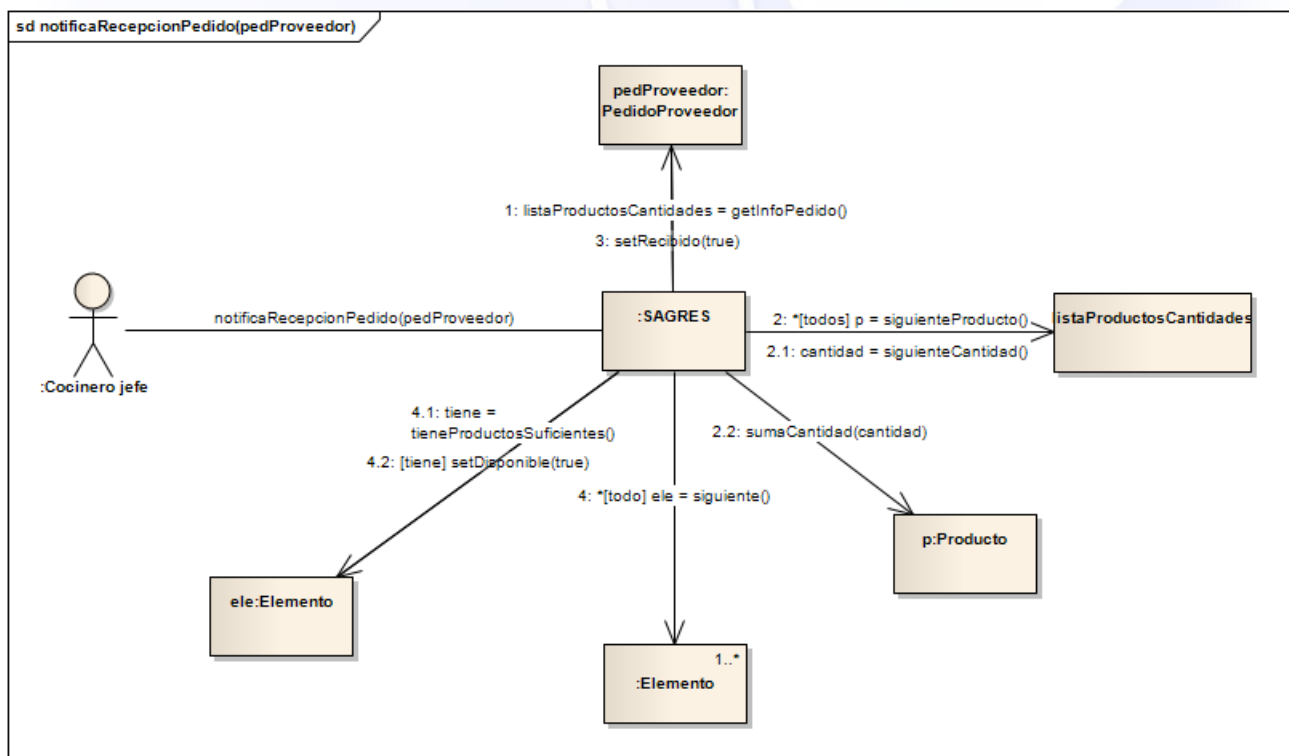
ImprimeListaProductosAPedir



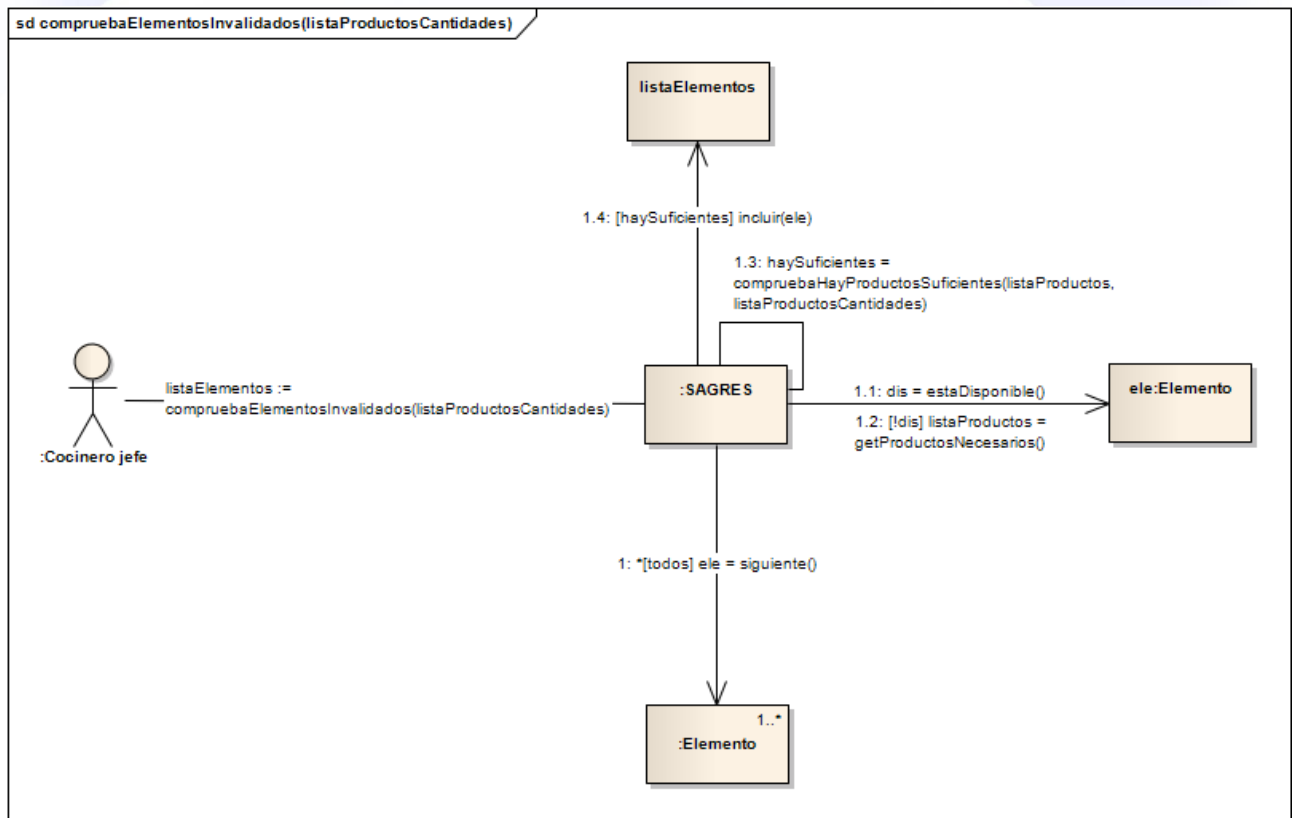
ObtienePedidoProveedor



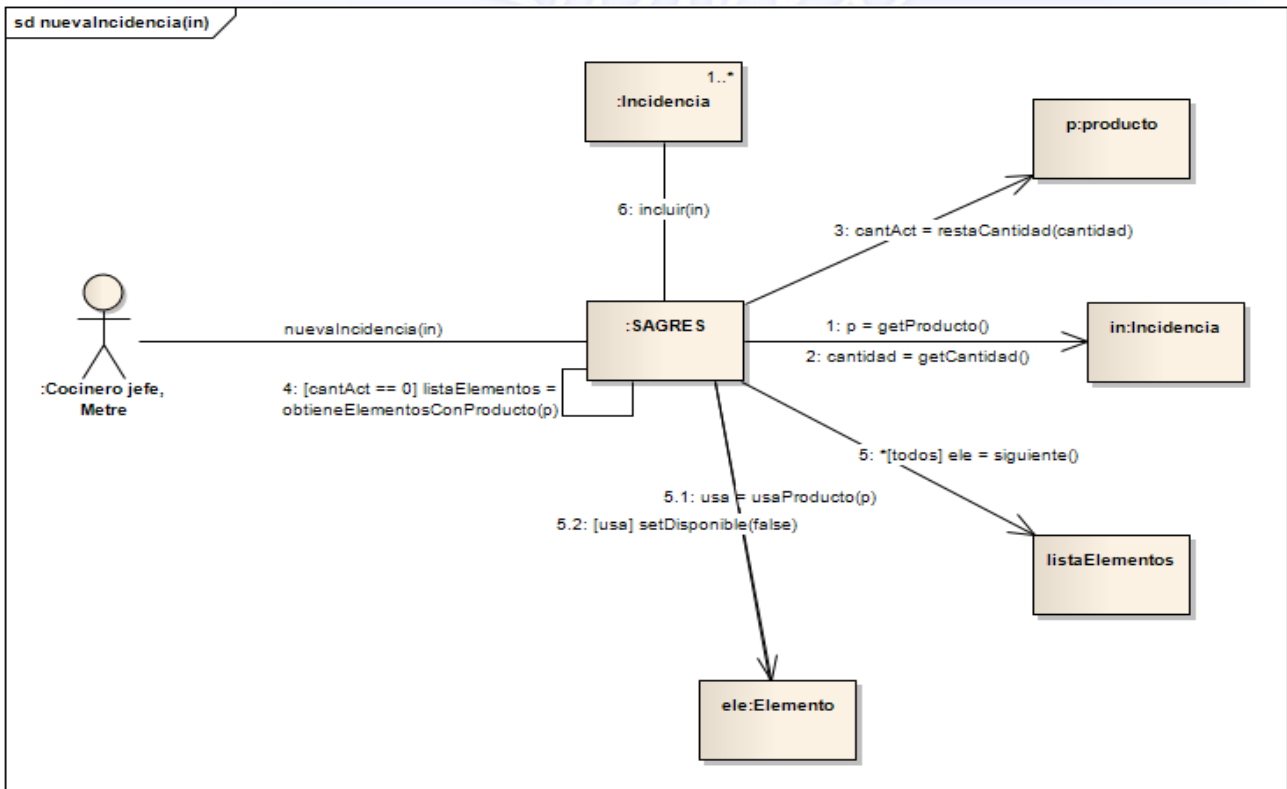
NotificaRecepcionPedido



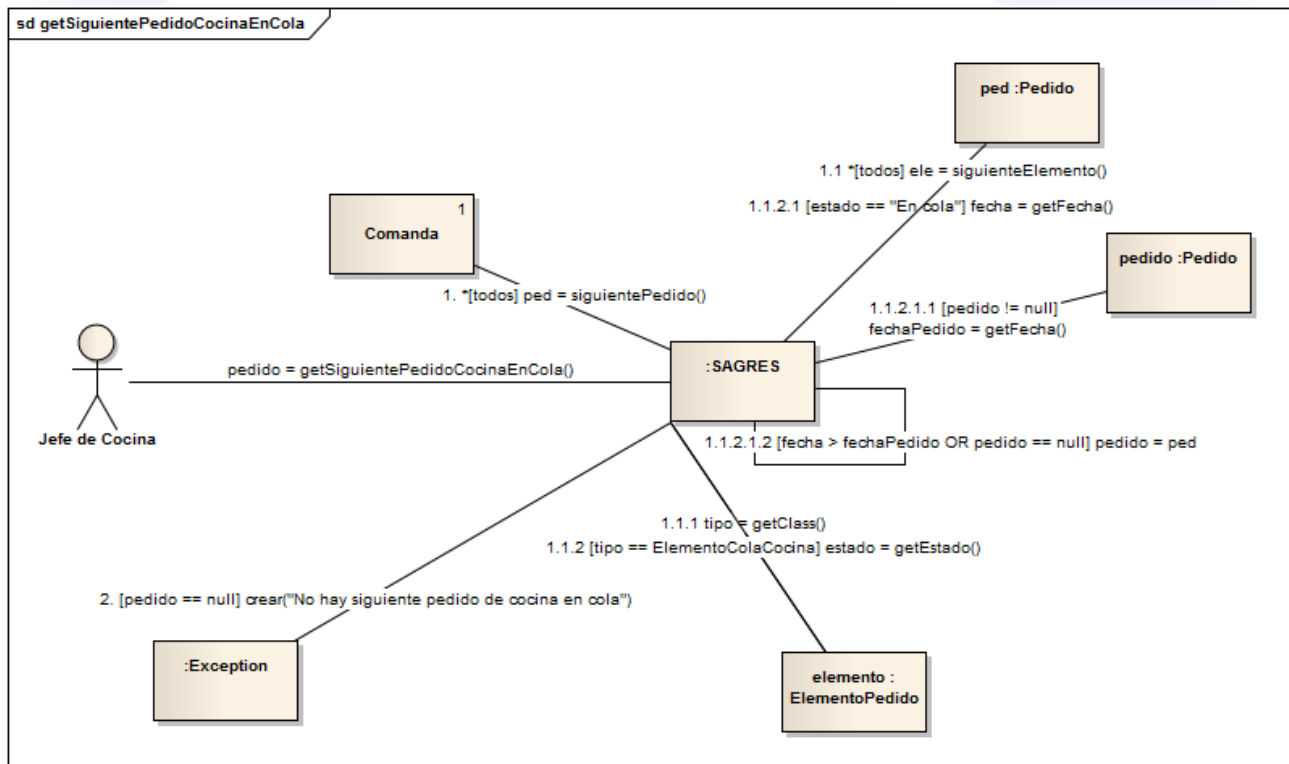
CompruebaElementosInvalidos



nuevaIncidencia

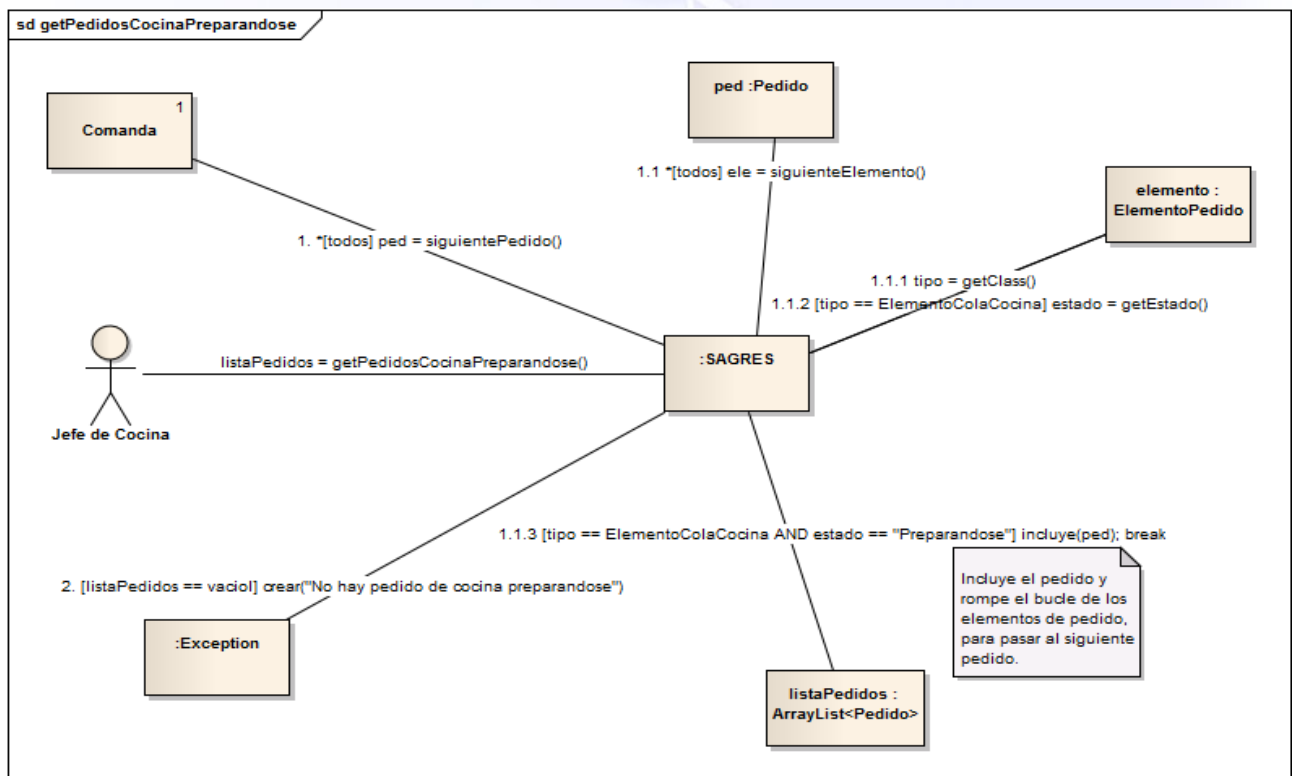


GetSiguientePedidoCocinaEnCola()



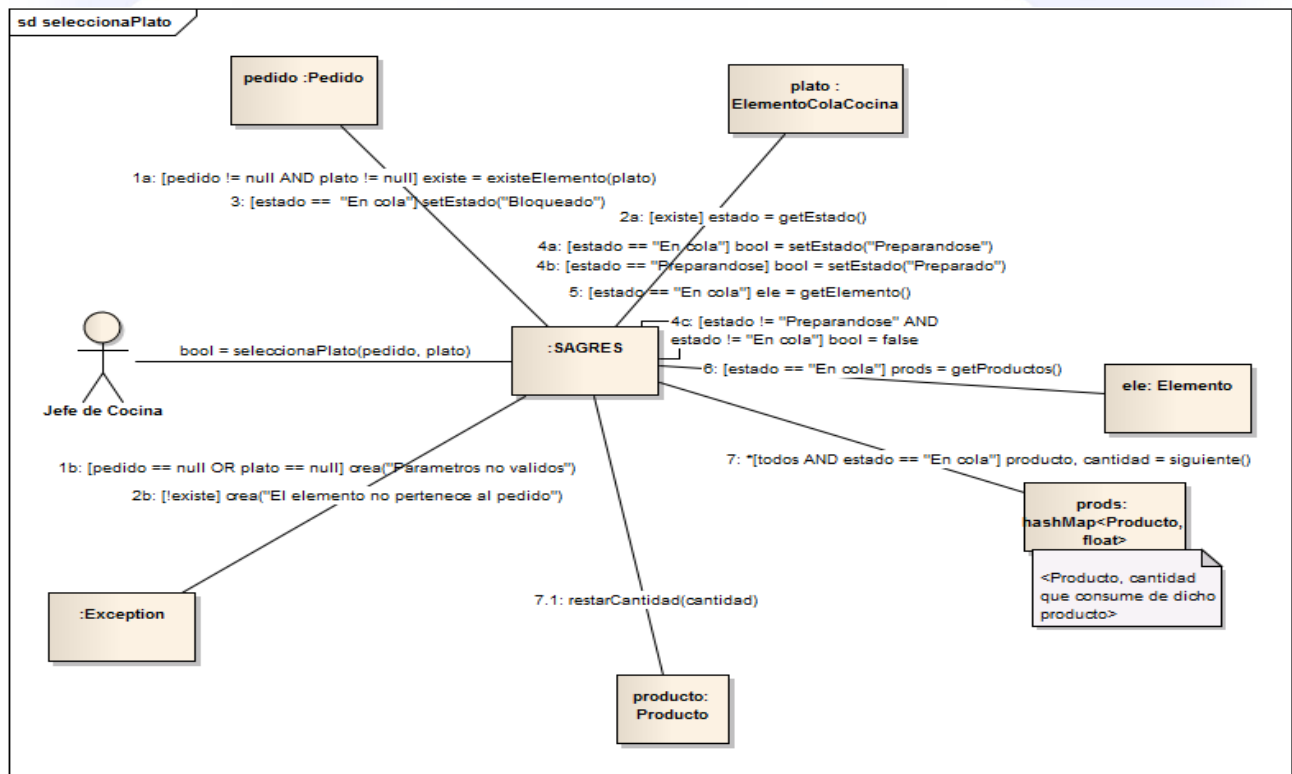
Realizado por Sergio Rodríguez Lumley

GetPedidosCocinaPreparandose



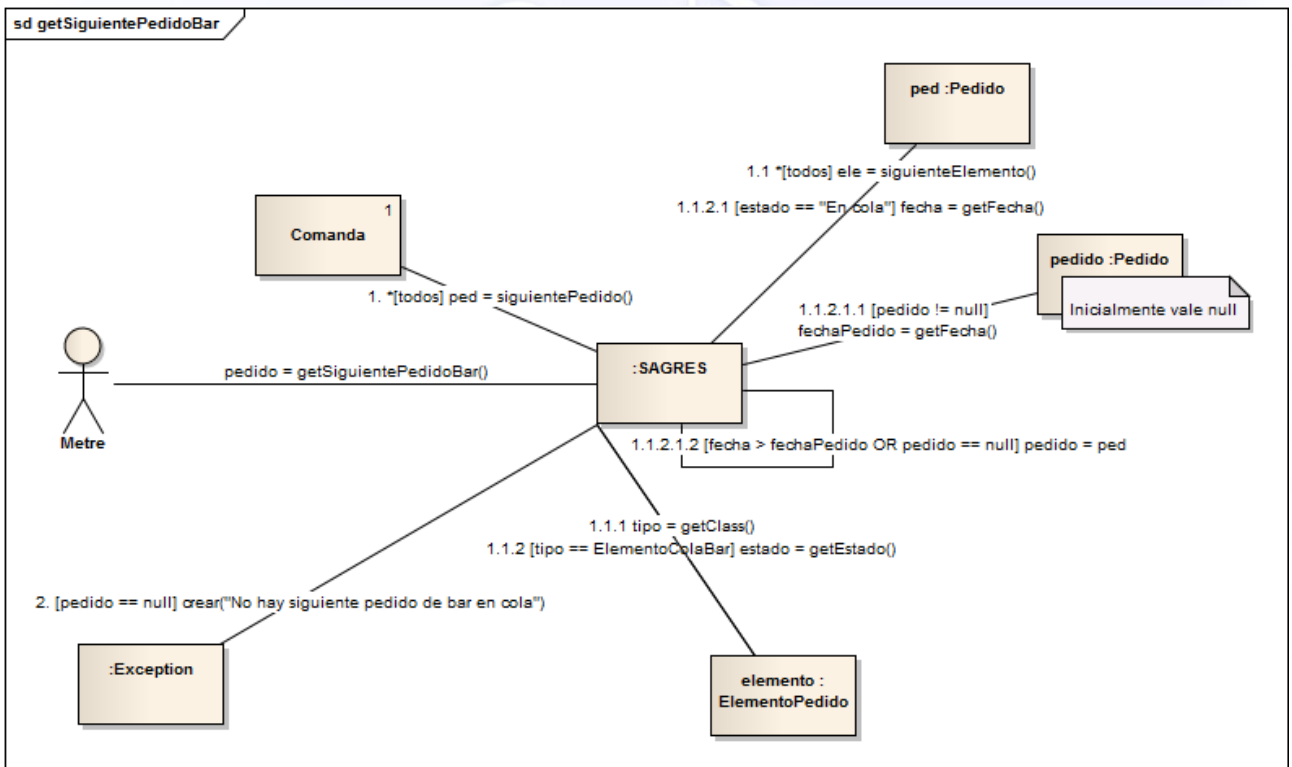
Realizado por Sergio Rodríguez Lumley

SeleccionaPlato



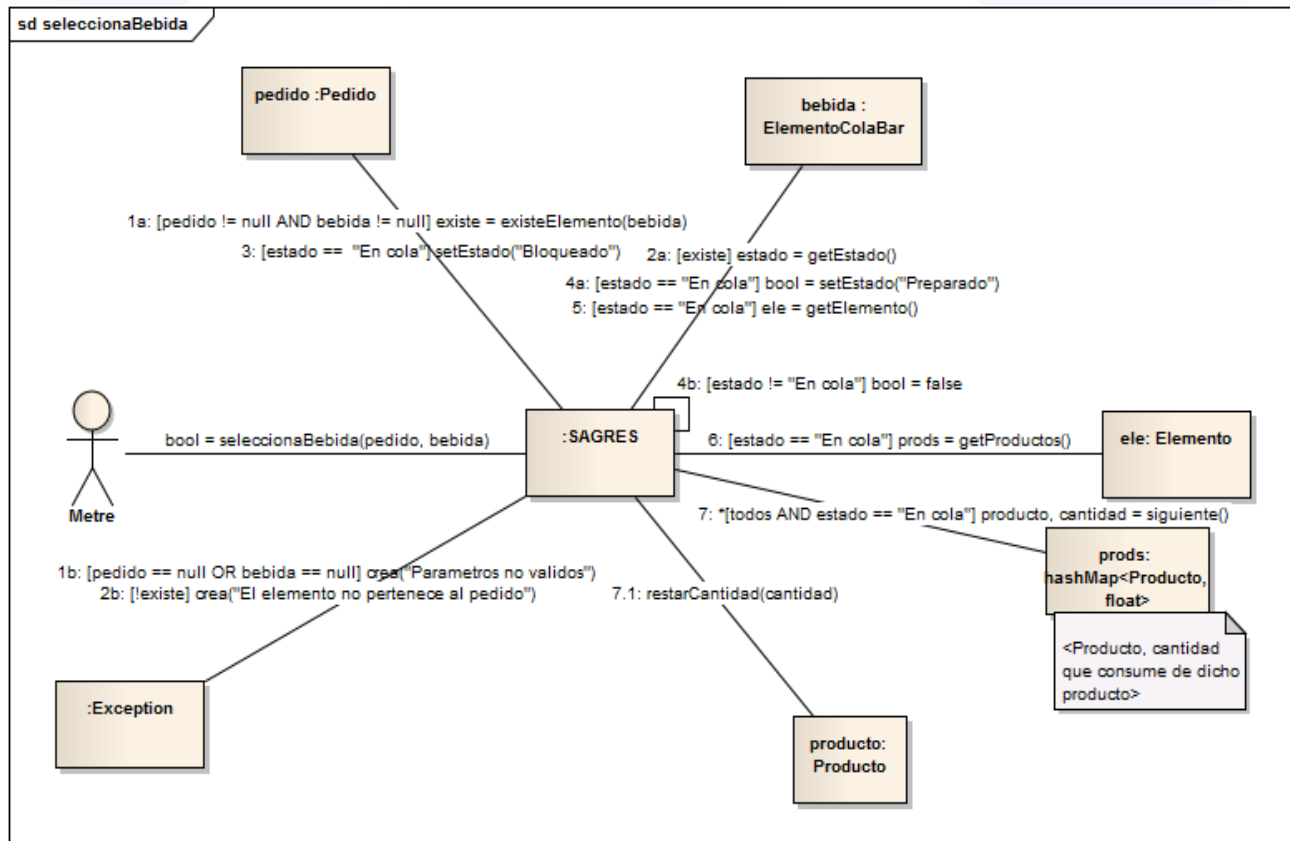
Realizado por Sergio Rodríguez Lumley

GetSiguientePedidoBar



Realizado por Sergio Rodríguez Lumley

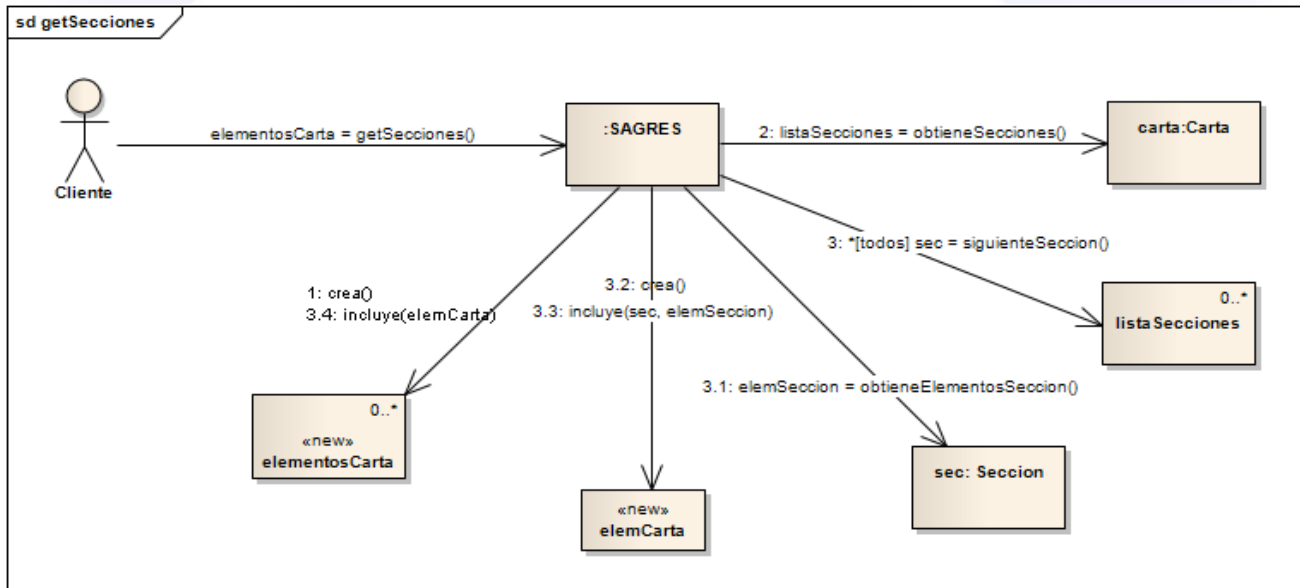
SeleccionaBebida



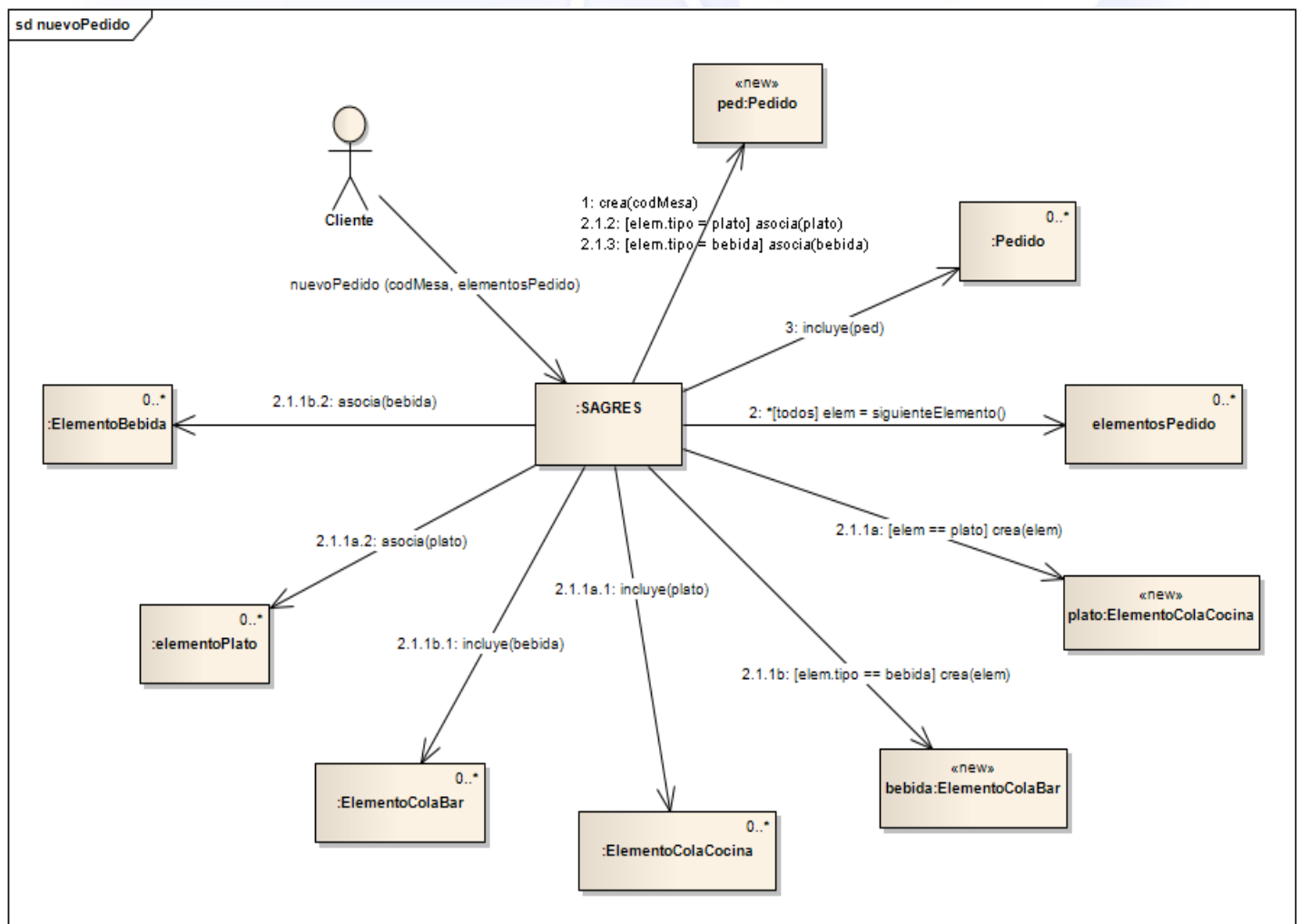
Realizado por Sergio Rodríguez Lumley

Sagres

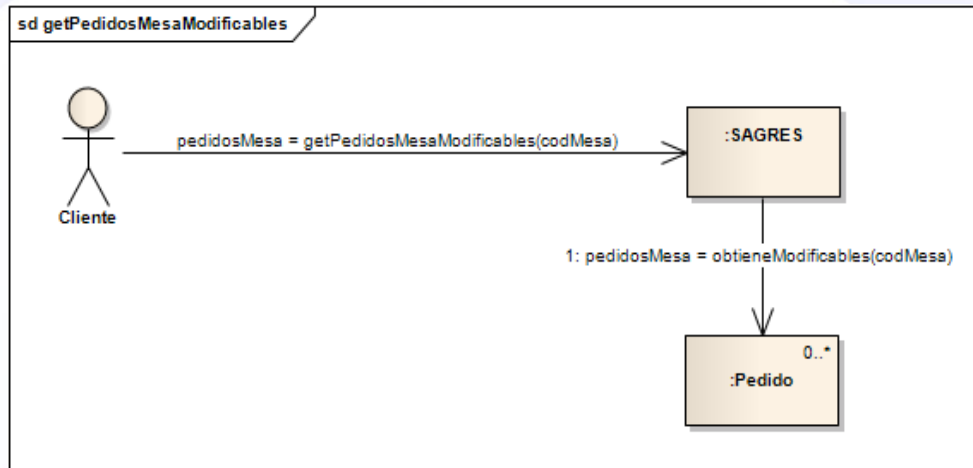
GetSecciones



NuevoPedido

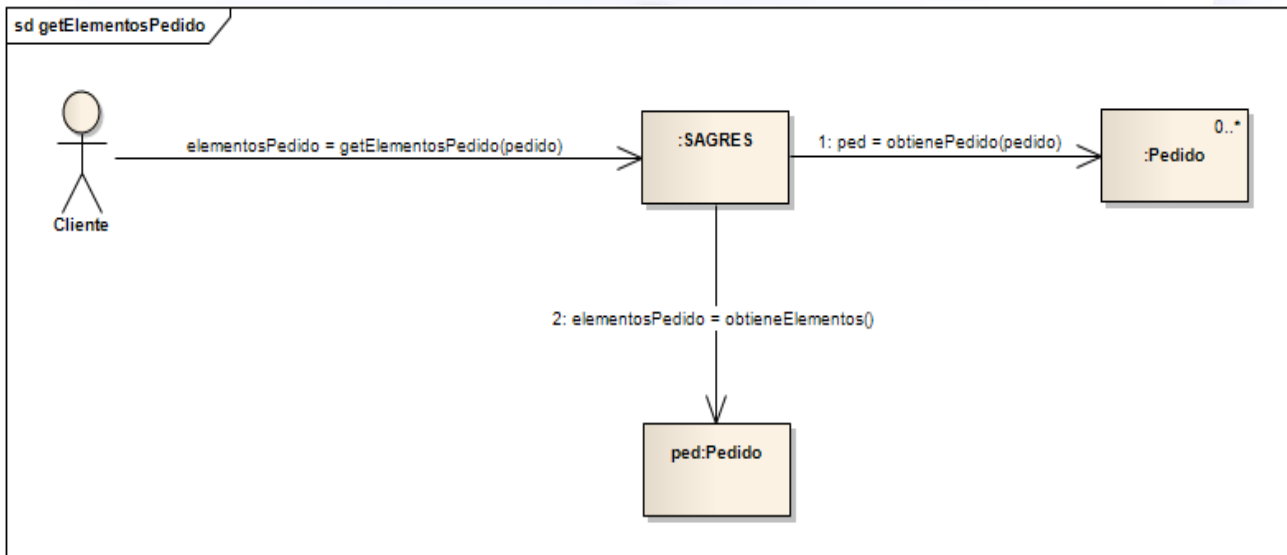


GetPedidosMesaModificables



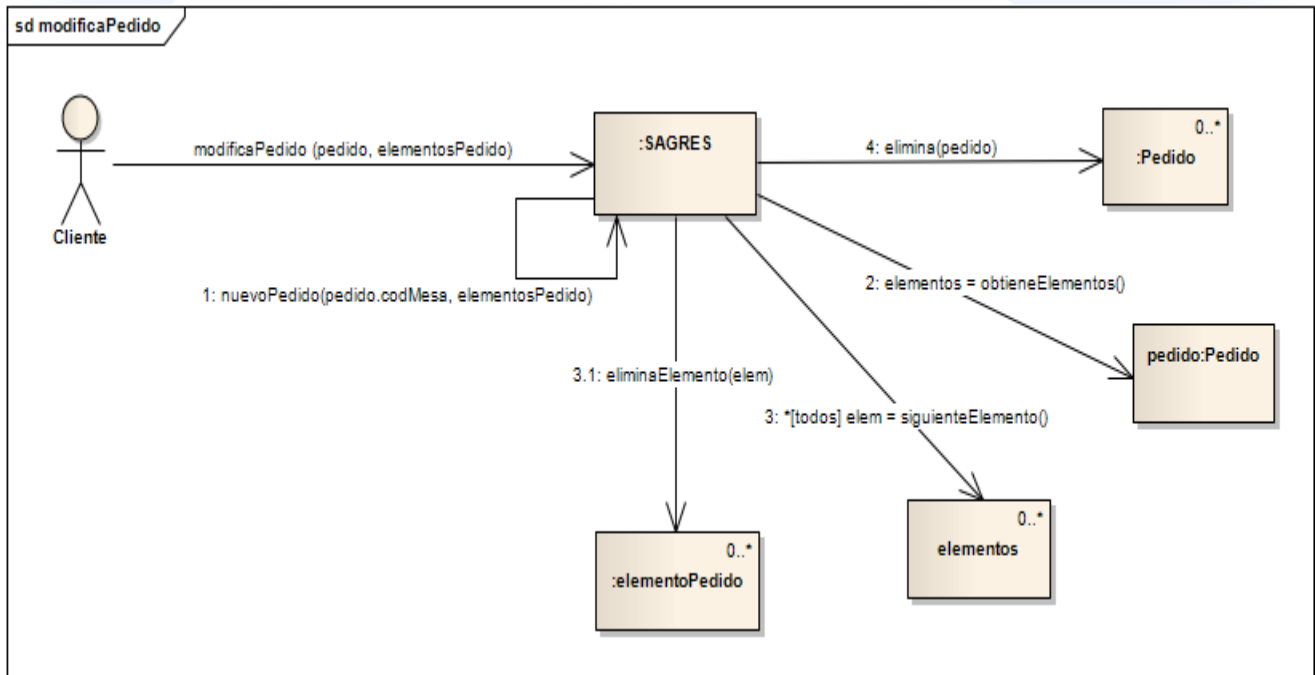
Realizado por Adrián Víctor Pérez Lopera

GetElemntosPedido



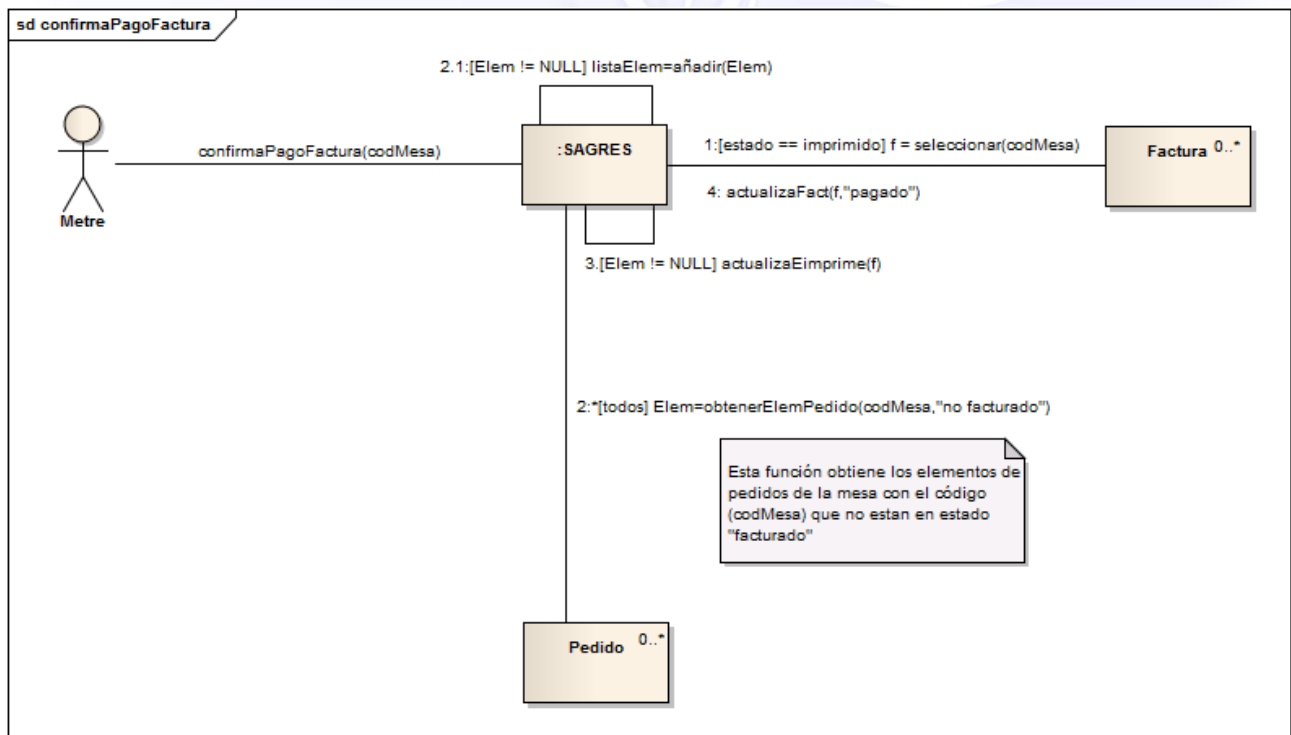
Realizado por Adrián Víctor Pérez Lopera

ModificaPedido



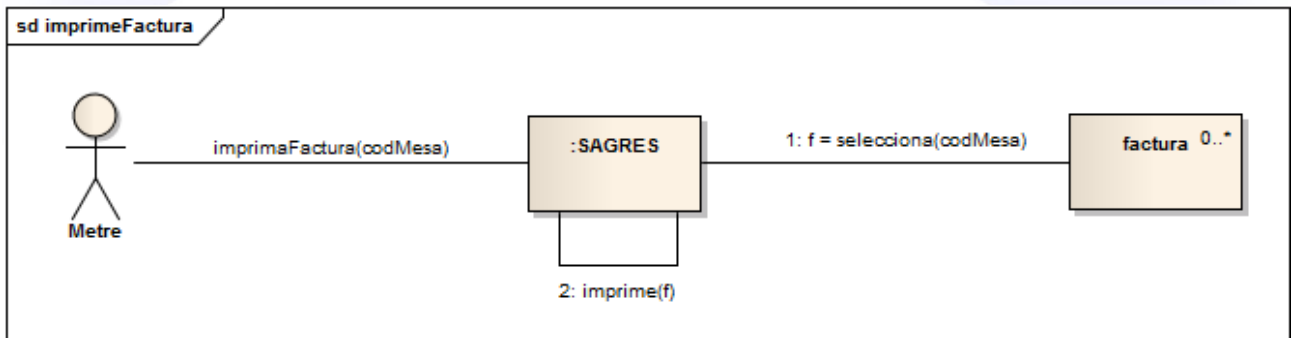
Realizado por Adrián Víctor Pérez Lopera

ConfirmaPagoFactura



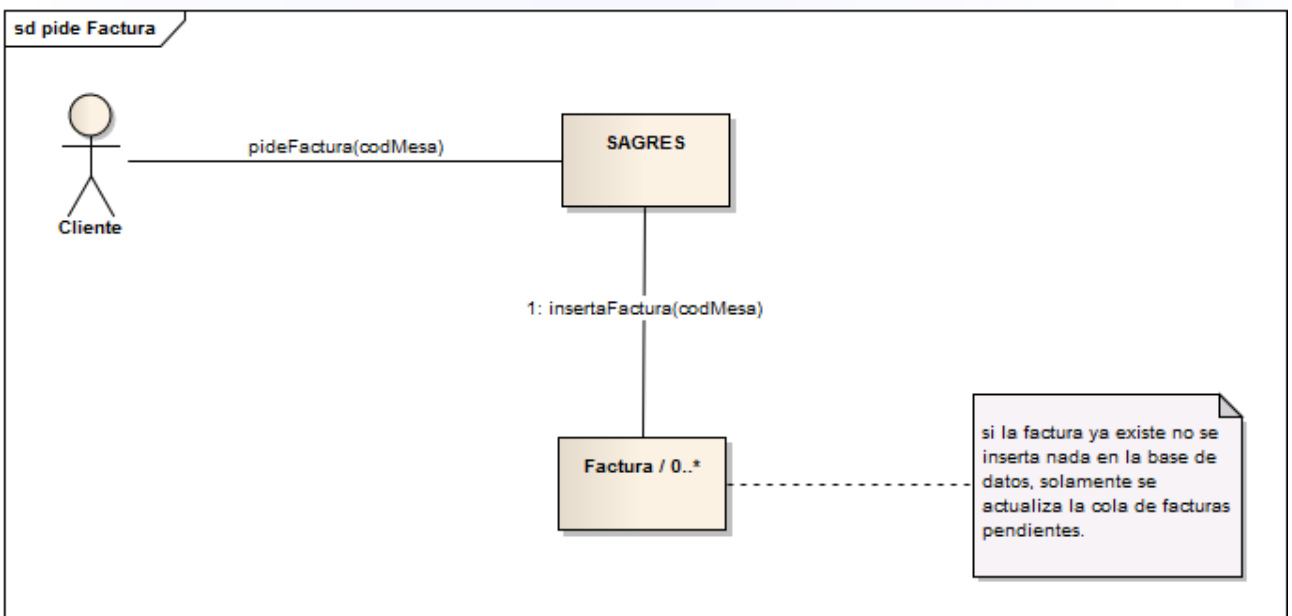
Realizado por Nabil Sabeg

ImprimeFactura



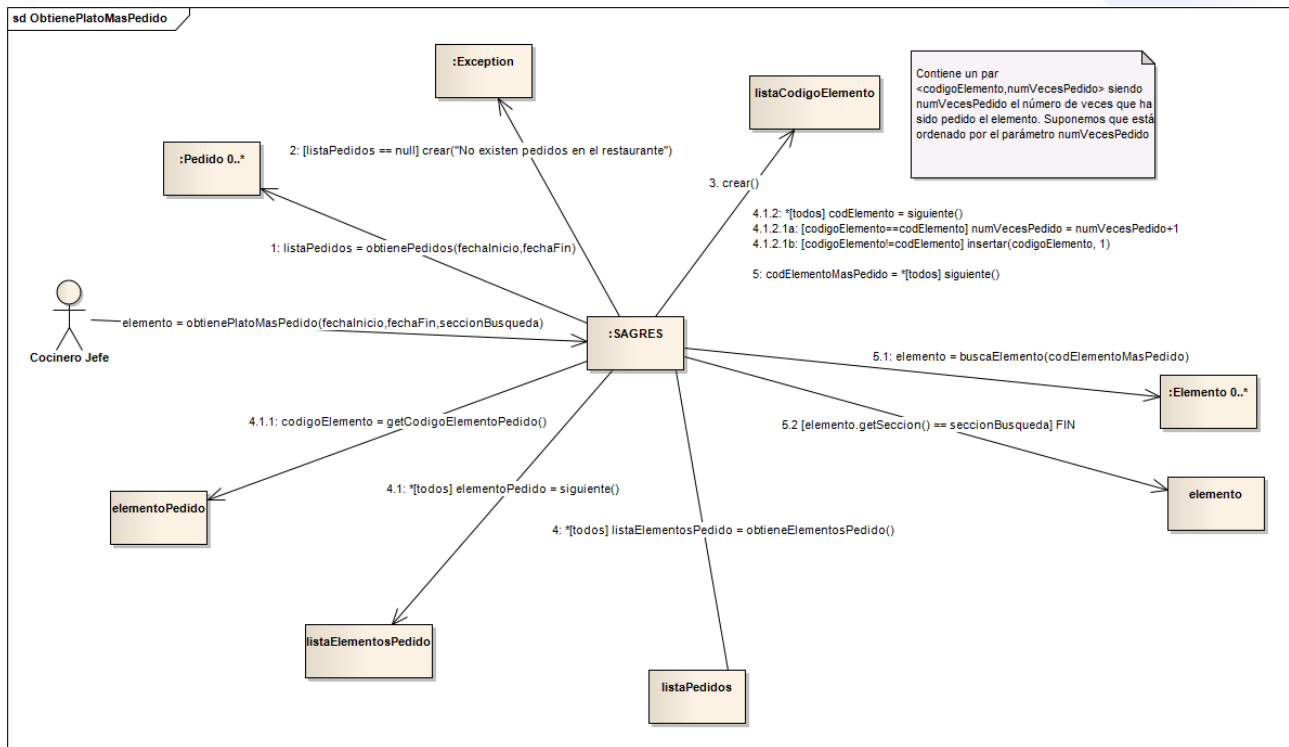
Realizado por Nabil Sabeg

PideFactura



Realizado por Nabil Sabeg

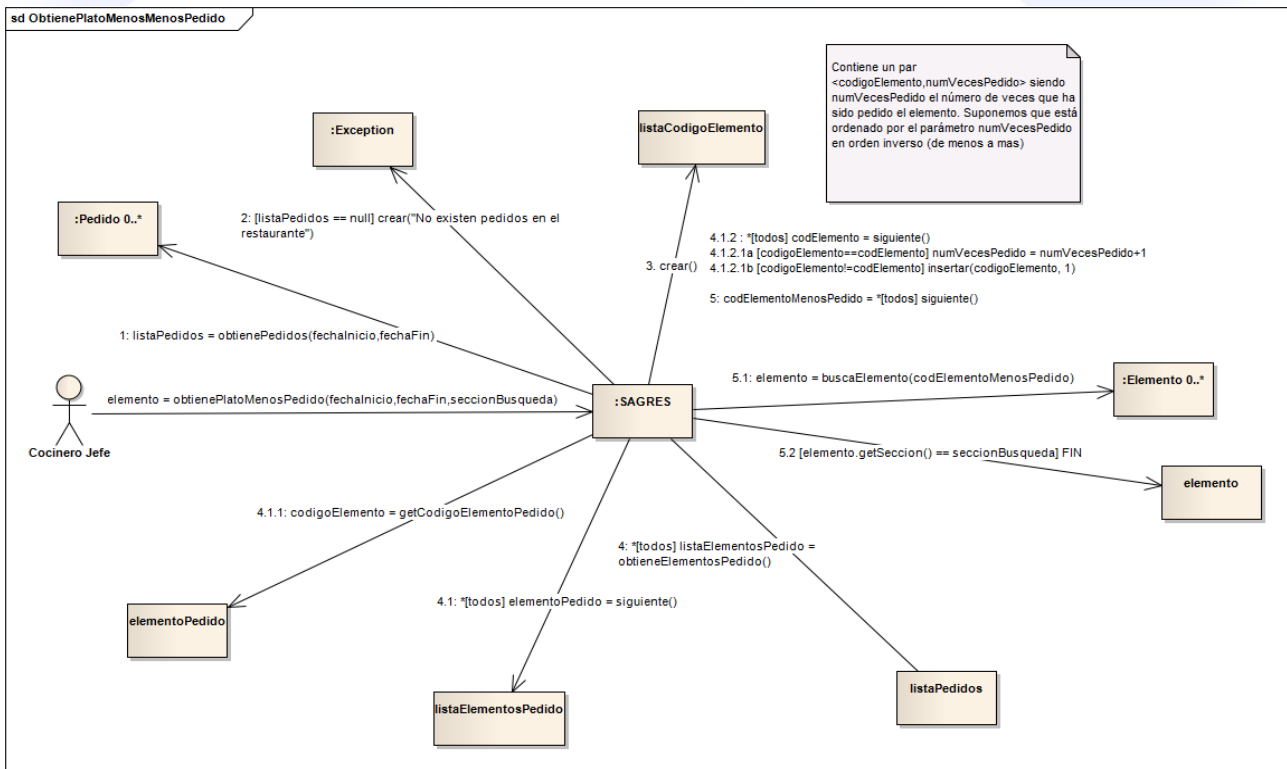
ObtenerPlatoMasPedido



Realizado por Carlos Salas

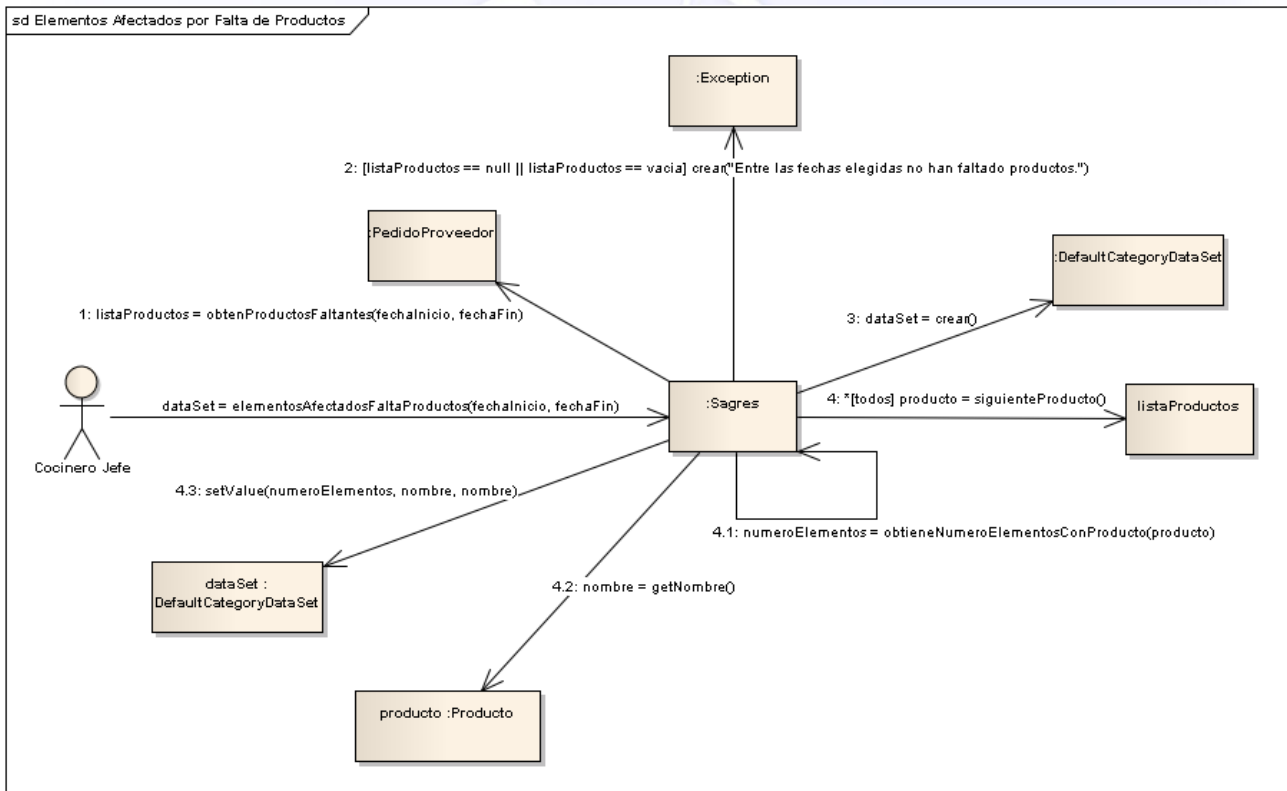
Sagres

ObtienePlatoMenosPedido



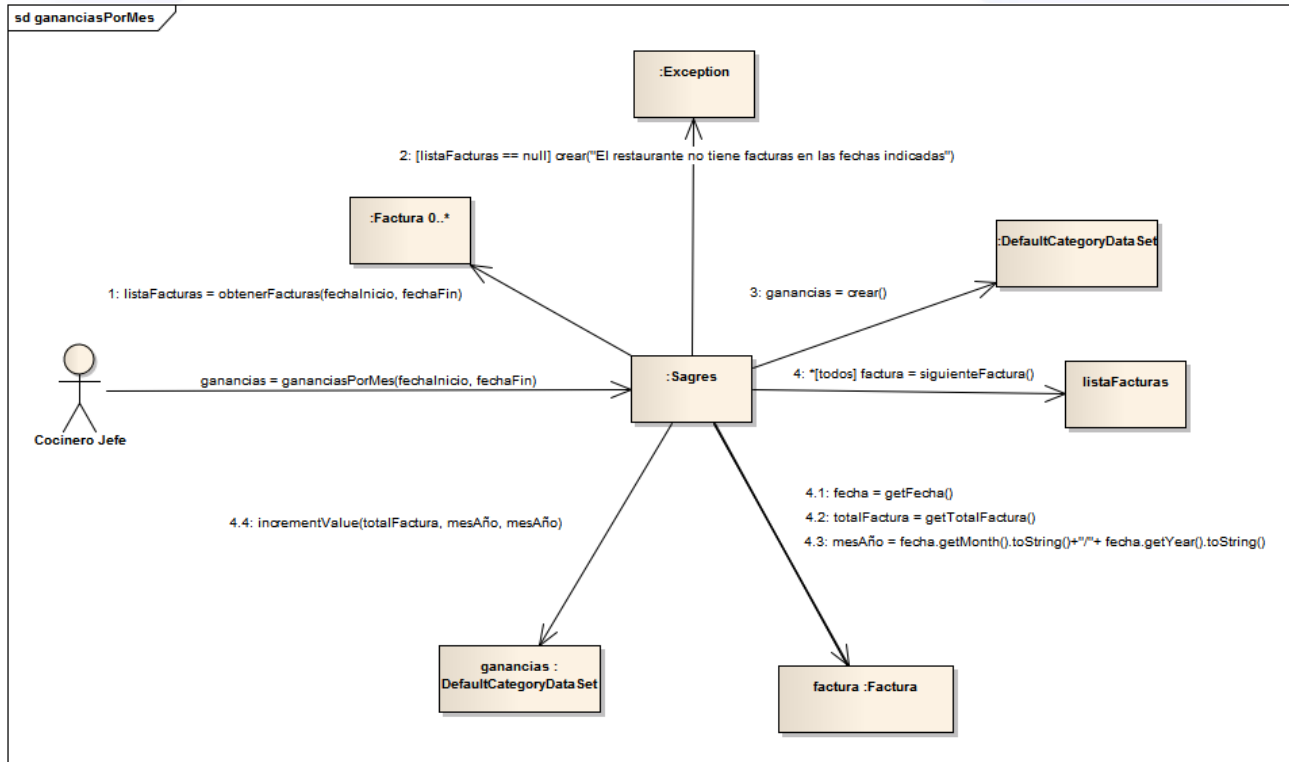
Realizado por Carlos Salas

ElementoAfectadosFaltaProductos



Realizado por Angel

GananciasProMes



Realizado por Dioni

Sagres

APÉNDICE 1.0

Fecha	22/03/2010
Descripción del problema	
Impacto del problema	
Soluciones adoptadas	<ul style="list-style-type: none">• Se ha generado el documento de análisis inicial.(1ª Iteración)
Anexos a la versión	

Sagres

APÉNDICE 1.1

<i>Fecha</i>	16/04/2010
<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">• Se han modificado los contratos de las operaciones y los diagramas de colaboración
<i>Anexos a la versión</i>	

APÉNDICE 2.0

Fecha	23/04/10
Descripción del problema	
Impacto del problema	
Soluciones adoptadas	<ul style="list-style-type: none">• Se ha creado el documento de análisis. (2ª Iteración)
Anexos a la versión	

Sagres

APÉNDICE 2.1

Fecha	07/05/10
Descripción del problema	<ol style="list-style-type: none">1. Los contratos y los diagramas de colaboración correspondientes a la realización de nuevos pedidos y a la modificación de los mismos por parte del cliente tenían fallos que afectaban a la lógica de la aplicación.2. No se mostraban en los diagramas de colaboración de colas parte de la funcionalidad de las operaciones, al igual que en sus contratos.3. Los cambios hechos en el modelado de requisitos llevaban a eliminar el contrato y diagrama de colaboración de “imprime Facturas”.4. Se ha añadido el diagrama de colaboración “pide factura”5. La clase “Comanda” no aparece en ninguno de los diagramas de colaboración, por lo que no tiene ninguna utilidad.6. Los cambios hechos en el DMR llevaban a eliminar los contratos y diagramas de colaboración de “iniciaNuevoPedido”, “iniciaMofidicaPedido” y “obtieneElementos”, además de añadir los contratos y diagramas de colaboración de “obtieneElementosCarta”, “obtienePedidosMesaModificables” y “obtieneElementosPedido”.
Impacto del problema	<ol style="list-style-type: none">1. Se arrastraban incoherencias en ciertas operaciones a la hora de realizar el diseño.2. Imposibilidad de conocer la funcionalidad correcta y posible error en la especificación del diseño.3. Aparece una clase en el diagrama de clases que no se utiliza, con lo que se están introduciendo incoherencias en el sistema.4. El análisis y el DMR no coincidían.
Soluciones adoptadas	<ul style="list-style-type: none">• Se han corregido los contratos y diagramas de colaboración pertinentes.• Se ha añadido la funcionalidad completa a los diagramas de colaboración de colas, así como actualizadas sus funciones y contratos.• Se ha eliminado la clase “Comanda”.• Se ha quitado el diagrama de colaboración y contrato de “imprime Facturas”• Se ha añadido el diagrama de colaboración “pide factura”• Se han quitado los diagramas de colaboración y los contratos de “iniciaNuevoPedido”, “iniciaModificaPedido” y “obtieneElementos”, y se han añadido los diagramas de colaboración y los contratos de “obtieneElementosCarta”, “obtienePedidosMesaModificables” y “obtieneElementosPedido”.
Anexos a la versión	

APÉNDICE 3.0

Fecha	22/05/10
Descripción del problema	
Impacto del problema	
Soluciones adoptadas	<ul style="list-style-type: none">• Se ha creado el documento de análisis. (3ª Iteración)
Anexos a la versión	

Sagres