

# Documento de ANÁLISIS

Sagres

**Sistema de Administración y  
Gestión de RESTaurantes**



Adrián Víctor Pérez Lopera  
Sergio Rodríguez Lumley  
Nabil Sabeg

V1.0

## ÍNDICE DE CONTENIDO

Apartado de control de versiones.....	3
1. Modelado estático.....	4
<i>Descripción de clases conceptuales.....</i>	<i>4</i>
<i>Relación 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 de 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.....	25



## **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>
01/01/00	23/04/10	Se ha creado el documento de análisis.

The logo for 'Sagres' features a stylized, light-colored graphic above the word 'Sagres' written in a cursive script. The background of the page has a faint, light blue circular design.

## 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>
<b>Pedido</b>	Representa un pedido de cliente. Guarda toda la información relevante al sistema de un pedido.
<b>Factura</b>	En esta clase se guardan todas las facturas de los clientes.
<b>Restaurante</b>	En esta clase se guardan los datos del restaurante (Nombre, dirección, razón social...)
<b>Comanda</b>	Es una clase contenedor, almacena los pedidos de cliente que quedan por servir.
<b>ElementoPedido</b>	Representa un elemento de un pedido determinado seleccionado por el cliente. Almacena la información relevante del plato o bebida que representa.
<b>ElementoColaBar</b>	Es un elemento de pedido de tipo bebida. Representa las bebidas seleccionadas por el cliente en el pedido.
<b>ElementoColaCocina</b>	Es un elemento de pedido de tipo plato. Representa los platos seleccionados por el cliente en el pedido.
<b>ElementoBebida</b>	Elemento de tipo bebida. Representa a las bebidas que se ofrecen en la carta.
<b>ElementoPlato</b>	Elemento de tipo plato. Representa a los platos de comida que se forman con la combinación de distintos ingredientes.
<b>Elemento</b>	Representa a cada elemento de la carta del restaurante. Almacena la información asociada al plato o bebida que representa.
<b>SeccionComida</b>	Es un tipo de sección. Su función es organizar los elementos de la carta que son platos de comida.
<b>SeccionBebida</b>	Es un tipo de sección. Su función es organizar los elementos de la carta que son bebidas.
<b>Seccion</b>	La carta del restaurante se encuentra dividida en secciones. Está clase se encarga de dividir los distintos elementos que componen la carta según el tipo de cada uno.
<b>Carta</b>	Esta clase es la encargada de almacenar, controlar y gestionar toda la información referente a la carta del restaurante.

## Relación entre clases conceptuales

<i>Clases</i>	<i>Relación</i>
<b>Pedido - Factura</b>	Es una relación de asociación entre pedido y factura. Una factura se refiere a uno o varios pedidos.
<b>Restaurante - Factura</b>	Es una relación de asociación entre restaurante y factura. así la factura puede obtener la información necesaria del restaurante.
<b>Pedido - ElementoPedido</b>	Es una relación de asociación, la cual establece que un pedido contiene uno o muchos elementos pedidos por el cliente.
<b>Pedido - Comanda</b>	Es una relación de asociación, indica que una comanda puede tener cero o muchos pedidos contenidos, sin embargo los pedidos pueden pertenecer a una única comanda, o ninguna.
<b>ElementoColaBar - ElementoPedido - ElementoColaCocina</b>	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.
<b>ElementoColaBar - ElementoBebida</b>	Es una relación de asociación en la que un ninguno o muchos ElementoColaBar hacen referencia a un único ElementoBebida.
<b>ElementoColaCocina - ElementoPlato</b>	Es una relación de asociación en la que un ninguno o muchos ElementoColaCocina hacen referencia a un único ElementoPlato.
<b>ElementoBebida - Elemento - ElementoPlato</b>	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.
<b>ElementoPlato - SeccionComida</b>	Está claro que una sección de la carta organiza a una serie de elementos de un tipo concreto. En este caso, una sección 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.
<b>ElementoBebida - SeccionBebida</b>	Se da la misma situación que para la relación anterior.
<b>SeccionComida - Seccion - SeccionBebida</b>	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.
<b>Carta - Seccion</b>	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.

## Atributos asociados a clases y/o relaciones

Clase	Atributo	Descripción
<b>Pedido</b>	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.
<b>Factura</b>	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.
<b>Restaurante</b>	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.
<b>ElementoPedido</b>	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”).
<b>Elemento</b>	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.



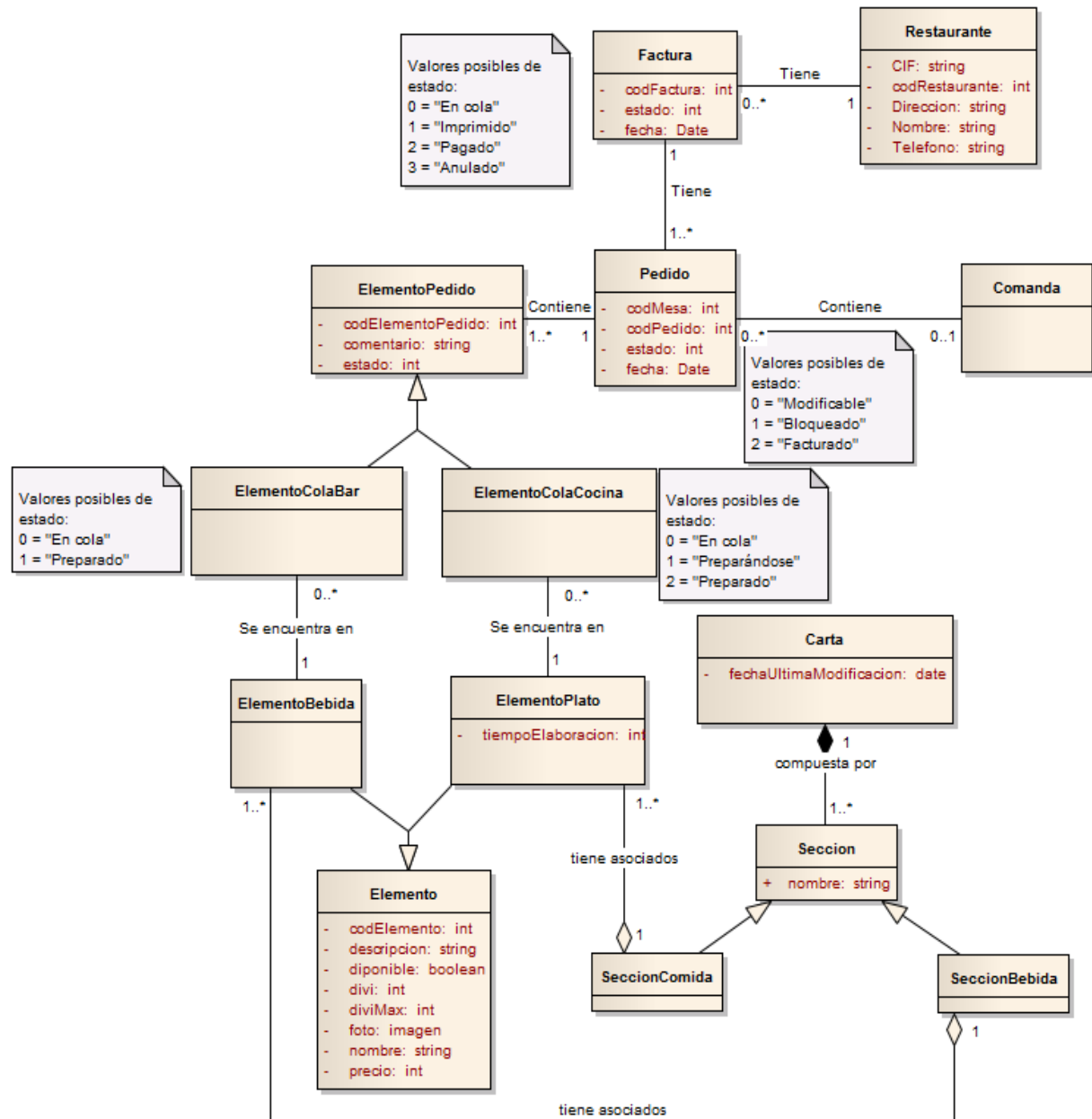
	precio : int	Precio de venta al público del elemento.
<b>ElementoPlato</b>	tiempoElaboracion : int	Tiempo en minutos que se tarda en elaborar el plato. Solo se tiene en cuenta en el caso de que no haya raciones.
<b>Seccion</b>	Nombre : string	Nombre de la sección.
<b>Carta</b>	fechaUltimaModificacion : 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.



The logo for 'Sagres' is centered on the page. It features a stylized, white, cursive script for the word 'Sagres'. Above the text is a decorative white swirl or ribbon-like element that loops around the 'S' and extends upwards.

# Diagrama de clases

class Diagrama de Clases





## 2. MODELO DE COMPORTAMIENTO EXTERNO

### Descripción de contratos del sistema

<b>Nombre:</b>	Pedido = getSiguientePedidoCocinaEnCola()
<b>Responsabilidad:</b>	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”.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Cambia Estado Plato”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"><li>• Si no hay ningún pedido con algún ElementoColaCocina en estado “En cola”.</li></ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	Un pedido con algún ElementoColaCocina en estado “En cola”, con la fecha más antigua posible.

*Realizado por Sergio Rodríguez Lumley*

<b>Nombre:</b>	Pedido = getSiguientePedidoCocinaPreparandose()
<b>Responsabilidad:</b>	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 “Preparandose” y ninguno “En cola”.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Cambia Estado Plato”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si no hay ningún pedido con algún ElementoColaCocina en estado “Preparandose” y ninguno “En cola”.</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	Un pedido con algún ElementoColaCocina en estado “Preparandose” y ninguno “En cola”, con la fecha más antigua posible.

*Realizado por Sergio Rodríguez Lumley*

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

*Realizado por Sergio Rodríguez Lumley*

<b>Nombre:</b>	Pedido = getSiguientePedidoBar()
<b>Responsabilidad:</b>	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”.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Cambia Estado Bebida”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Un pedido con algún ElementoColaBar en estado “En cola”, con la fecha más antigua posible.</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	Un pedido con algún ElementoColaBar en estado “En cola”, con la fecha más antigua posible.

*Realizado por Sergio Rodríguez Lumley*

<b>Nombre:</b>	Bool = seleccionaBebida(Pedido pedido, ElementoColaBar bebida)
<b>Responsabilidad:</b>	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”.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Cambia Estado Bebida”
<b>Nota:</b>	
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• Si alguno de los parámetros recibidos es incoherente (pedido nulo, bebida nula, etc).</li> <li>• Si la bebida no pertenece al pedido.</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se modificó un pedido.</li> <li>• Se modificó el estado de una bebida al valor “Preparado”.</li> </ul>
<b>Salida:</b>	Booleano con valor “Verdadero” si ha cambiado el estado de la bebida, “Falso” en caso contrario.

*Realizado por Sergio Rodríguez Lumley*

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

*Realizado por Adrián Víctor Pérez Lopera*

<b>Nombre:</b>	exitos = nuevoPedido(codMesa, elementosPedido)
<b>Responsabilidad:</b>	Operación encargada de incluir un nuevo pedido en el sistema.
<b>Tipo:</b>	Sistema.
<b>Referencias:</b>	Nuevo Pedido.
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>El parámetro codMesa es correcto.</li> <li>Los miembros del parámetro elementosPedido son correctos.</li> </ul>
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>Se creó un pedido con estado “modificable”.</li> <li>Se crearon elementos de pedido con estado “en cola”.</li> <li>Se creó un enlace entre el pedido y la comanda.</li> <li>Se crearon enlaces entre el pedido y los elementos de pedido.</li> <li>Se crearon enlaces entre los elementos de pedido y los elementos plato y/o elementos bebida.</li> </ul>
<b>Salida:</b>	<ul style="list-style-type: none"> <li>True si el pedido fue realizado con éxito, false en caso contrario.</li> </ul>

*Realizado por Adrián Víctor Pérez Lopera*

<b>Nombre:</b>	pedidosMesa = iniciaModificaPedido(codMesa)
<b>Responsabilidad:</b>	Operación obtener los pedidos con estado “modificable” de la mesa a la que hace referencia codMesa.
<b>Tipo:</b>	Sistema.
<b>Referencias:</b>	Modifica Pedido.
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>El parámetro codMesa es correcto.</li> </ul>
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>Una lista de pedidos.</li> </ul>

*Realizado por Adrián Víctor Pérez Lopera*

<b>Nombre:</b>	elementosCarta, elementosPedido = obtieneElementos(codPedido)
<b>Responsabilidad:</b>	Operación obtener los elementos de la carta y los elementos del pedido al que hace referencia codPedido.
<b>Tipo:</b>	Sistema.
<b>Referencias:</b>	Modifica Pedido.
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>El parámetro codPedido es correcto.</li> </ul>
<b>Postcondición:</b>	
<b>Salida:</b>	<ul style="list-style-type: none"> <li>Una lista de elementos tipo par nombre de sección de carta, lista de elementos de la sección.</li> <li>Una lista de elementos de pedido.</li> </ul>

*Realizado por Adrián Víctor Pérez Lopera*

<b>Nombre:</b>	exito = modificaPedido(codPedido, elementosPedido)
<b>Responsabilidad:</b>	Operación encargada de modificar los elementos asociados a un pedido de la mesa a la que hace referencia codMesa. Destruye el pedido que tenía el
<b>Tipo:</b>	Sistema.
<b>Referencias:</b>	Nuevo Pedido.
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>• El parámetro codMesa es correcto.</li> <li>• Los miembros del parámetro elementosPedido son correctos.</li> </ul>
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Se creó un pedido con estado “modificable”.</li> <li>• Se crearon elementos de pedido con estado “en cola”.</li> <li>• Se creó un enlace entre el pedido y la comanda.</li> <li>• Se crearon enlaces entre el pedido y lo elementos de pedido.</li> <li>• Se crearon enlaces entre los elementos de pedido y los elementos plato y/o elementos bebida.</li> </ul>
<b>Salida:</b>	<ul style="list-style-type: none"> <li>• True si el pedido fue realizado con éxito, false en caso contrario.</li> </ul>

*Realizado por Adrián Víctor Pérez Lopera*

*Sagres*



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

*Realizado por nabil*

<b>Nombre:</b>	imprimeFacturas()
<b>Responsabilidad:</b>	Imprime todas las facturas pendiente de imprimir en la cola de facturas.
<b>Tipo:</b>	Sistema
<b>Referencias:</b>	Caso de uso “Cambia Estado Factura”
<b>Nota:</b>	Esta operación es para facilitar al metre la impresión de todas las facturas si le llegan viarias peticiones al mismo tiempo.
<b>Excepción:</b>	<ul style="list-style-type: none"> <li>• No hay facturas en la cola de facturas.</li> </ul>
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Cambiar el estado de todas las factura a imprimido.</li> </ul>
<b>Salida:</b>	Las facturas imprimidas.

*Realizado por nabil*



<b>Nombre:</b>	imprimeFactura()
<b>Responsabilidad:</b>	Selecciona una factura y la imprime.
<b>Tipo:</b>	sistema
<b>Referencias:</b>	Caso de uso “Cambia Estado Factura”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	
<b>Salida:</b>	

*Realizado por nabil*

<b>Nombre:</b>	ConfirmaPagoFactura()
<b>Responsabilidad:</b>	Confirmar el pago de una factura y buscar si hay otro(s) pedido(s) servido(s) después de la impresión de esta factura del mismo cliente que va a pagar. En caso de que esto ocurra se imprima la factura nueva del resto de pedidos y se confirma su pago también.
<b>Tipo:</b>	sistema
<b>Referencias:</b>	Caso de uso “Cambia Estado Factura”
<b>Nota:</b>	
<b>Excepción:</b>	
<b>Precondición:</b>	
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>Cola de facturas actualizada.</li> </ul>
<b>Salida:</b>	

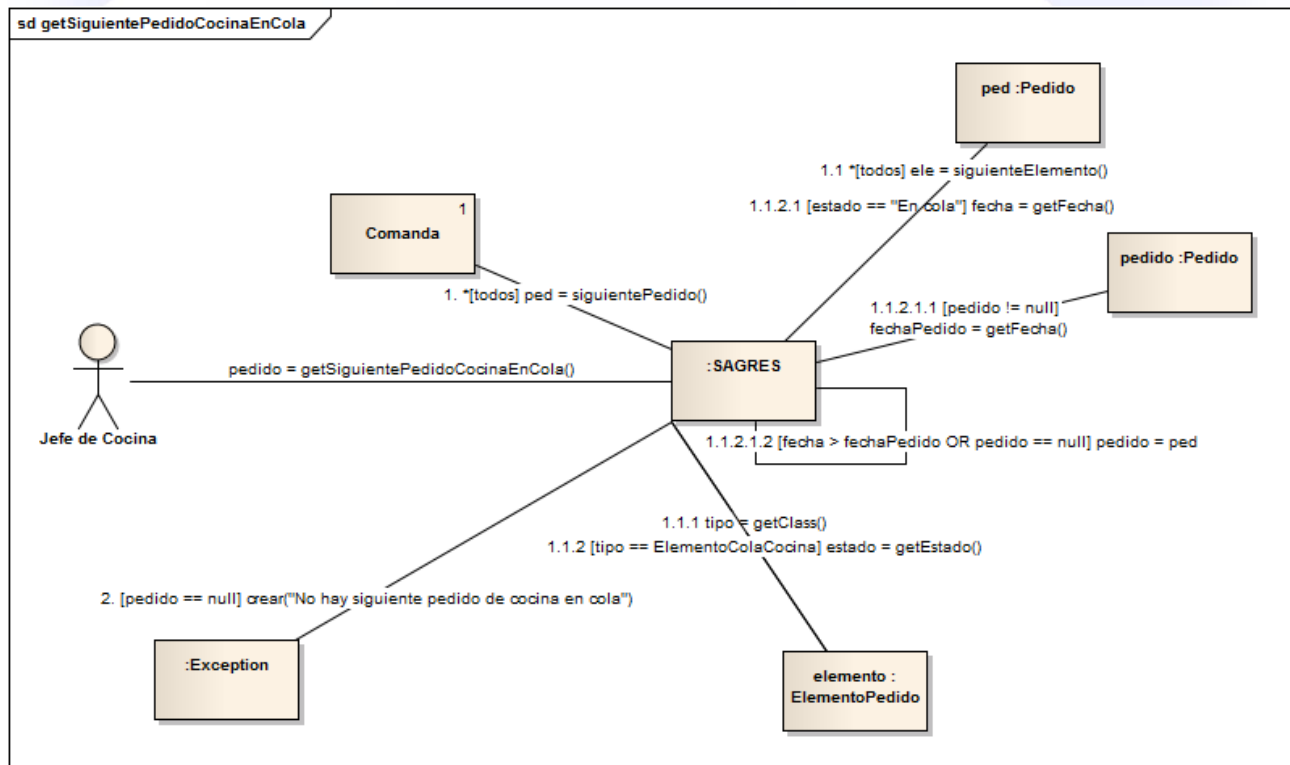
*Realizado por nabil*

<b>Nombre:</b>	confirmaAnulacionFactura()
<b>Responsabilidad:</b>	Confirmación de anulación de una factura.
<b>Tipo:</b>	sistema
<b>Referencias:</b>	
<b>Nota:</b>	Puede ocurrir que una factura no se ha pagado por que el cliente que la pidió se fue sin pagarla.
<b>Excepción:</b>	
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>• Factura pendiente de pagar.</li> </ul>
<b>Postcondición:</b>	<ul style="list-style-type: none"> <li>• Cola de facturas actualizada.</li> </ul>
<b>Salida:</b>	

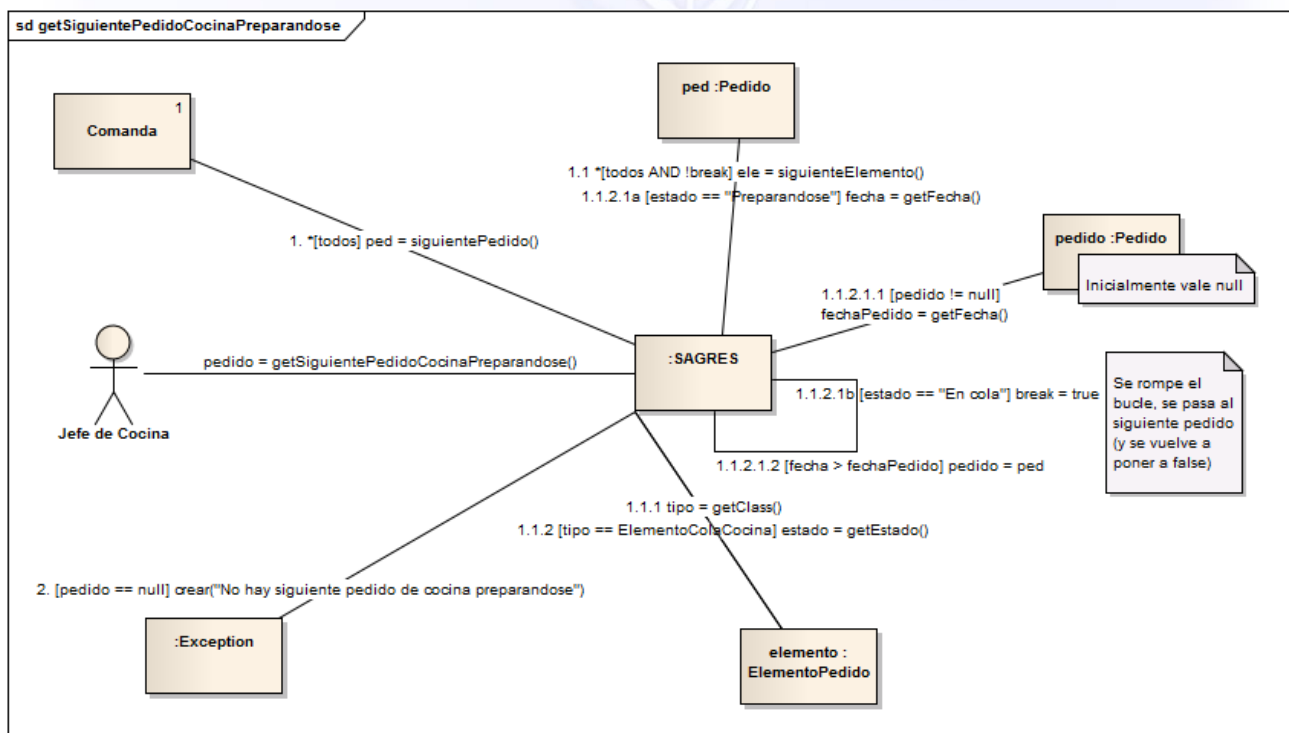
*Realizado por nabil*

*Sagres*

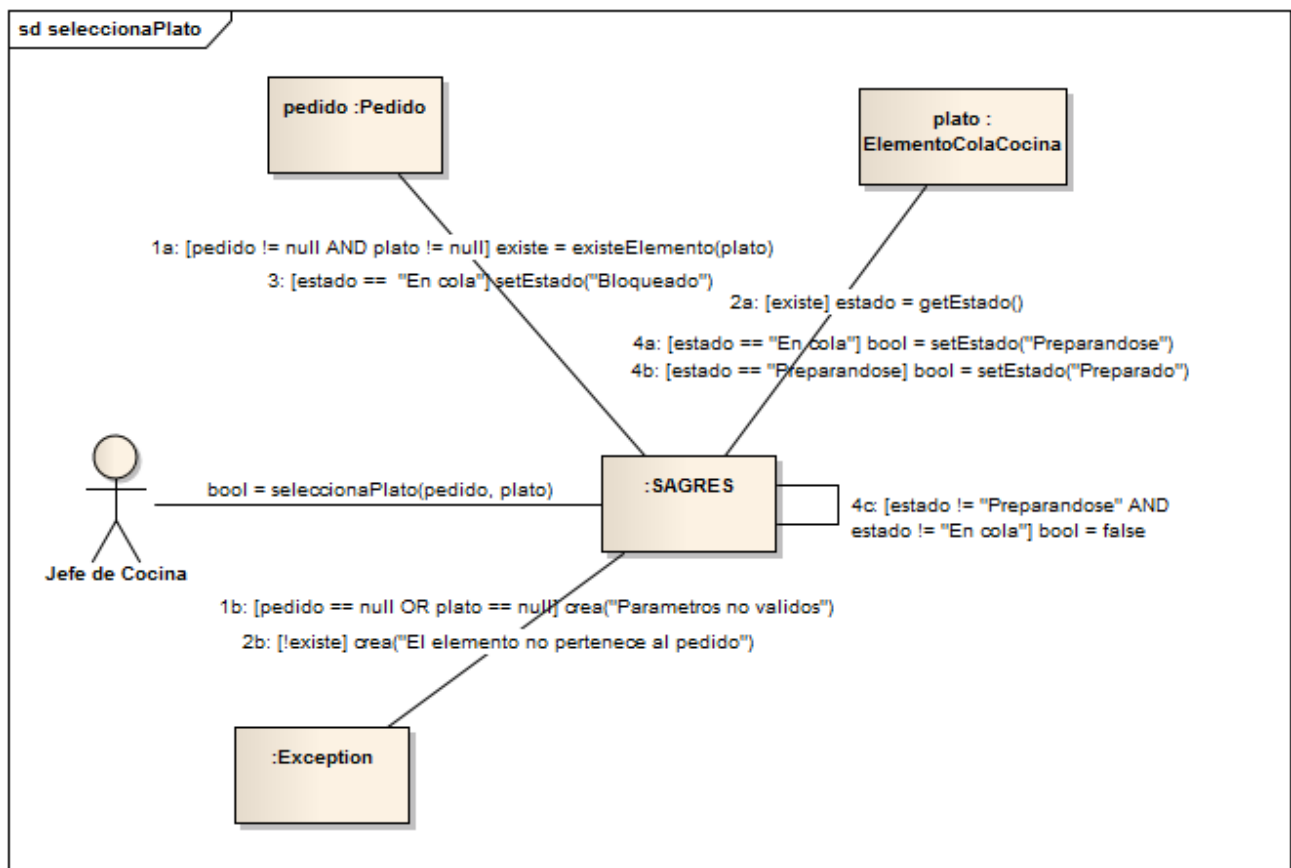
## Diagramas de colaboración para cada contrato



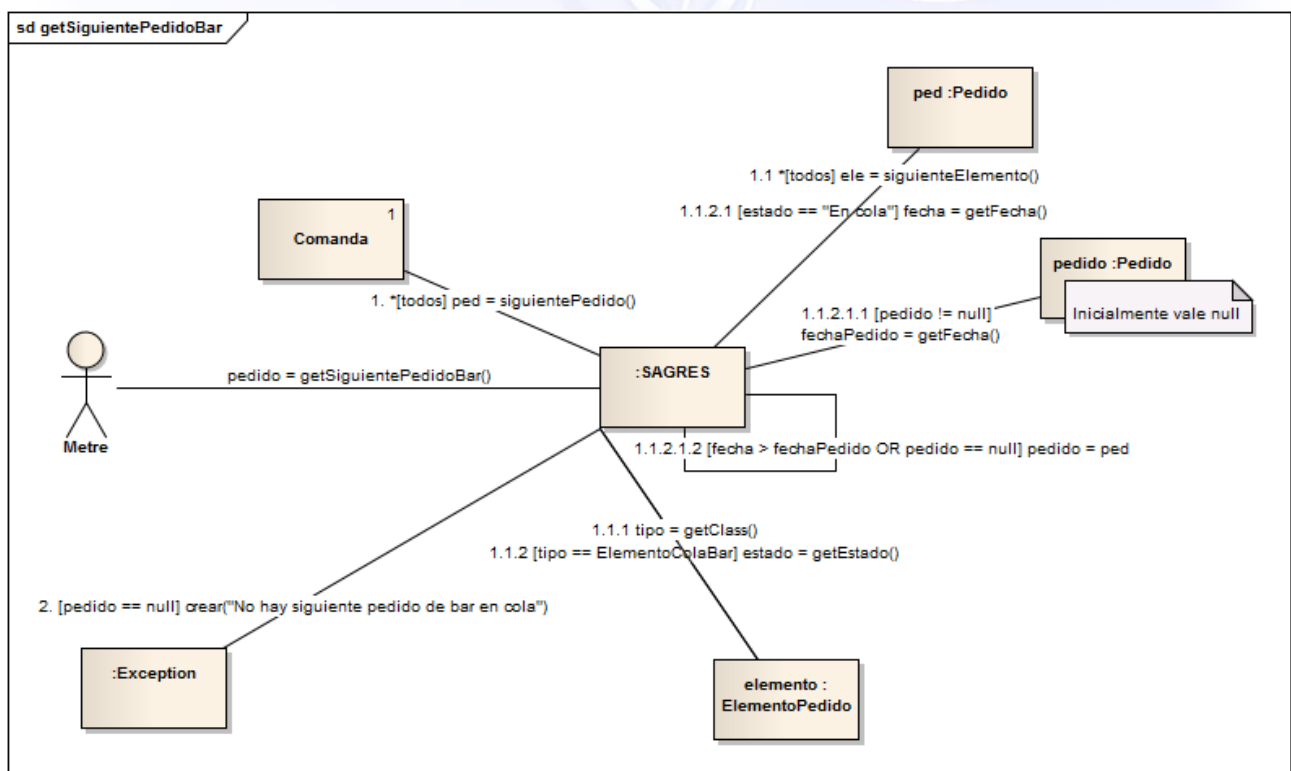
Realizado por Sergio Rodríguez Lumley



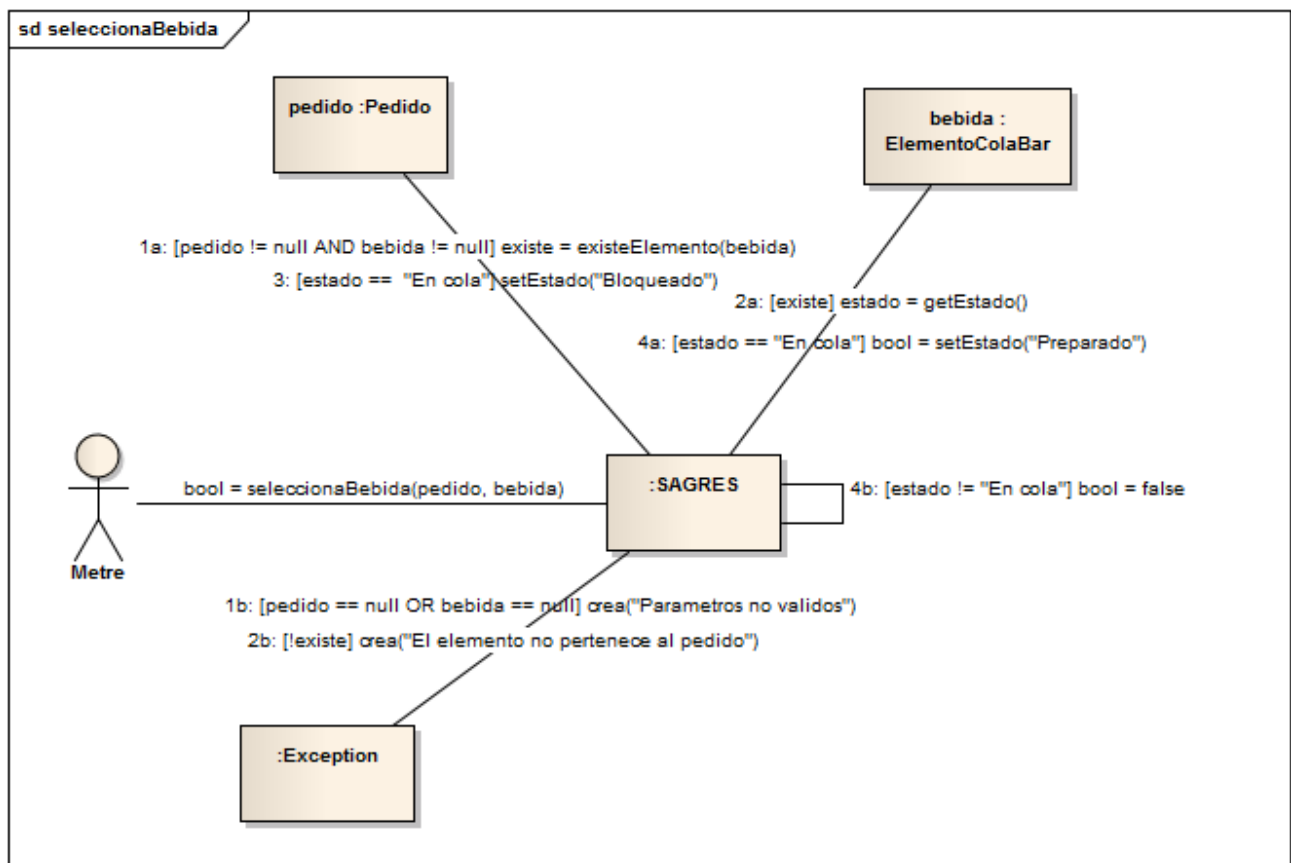
Realizado por Sergio Rodríguez Lumley



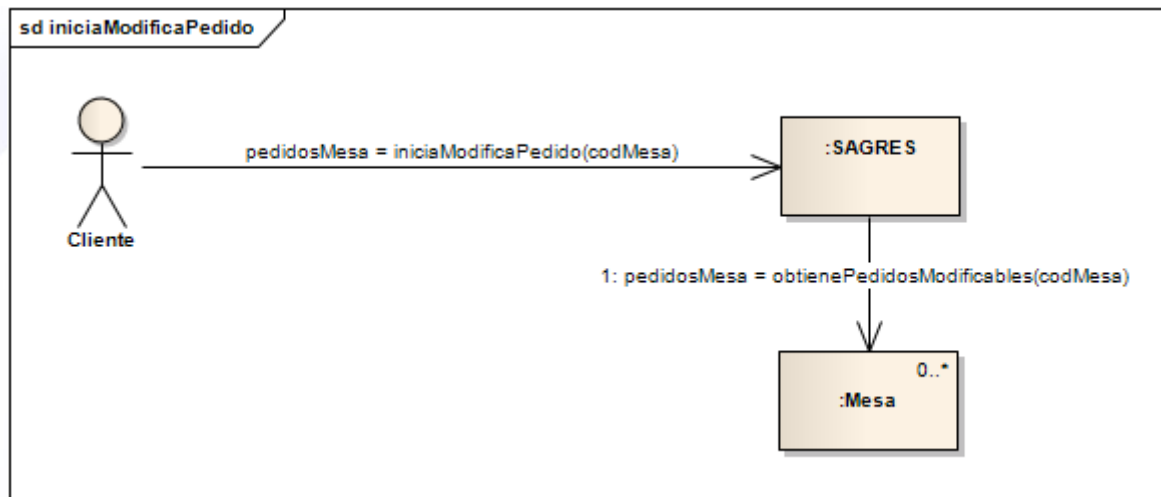
*Realizado por Sergio Rodríguez Lumley*



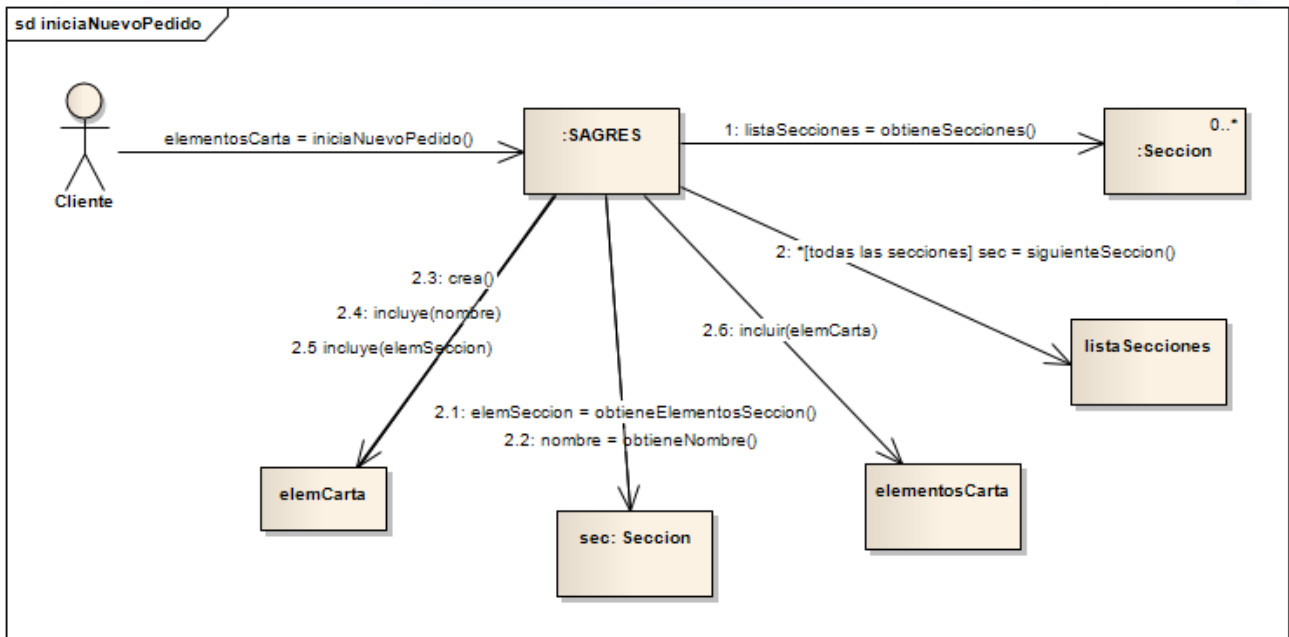
*Realizado por Sergio Rodríguez Lumley*



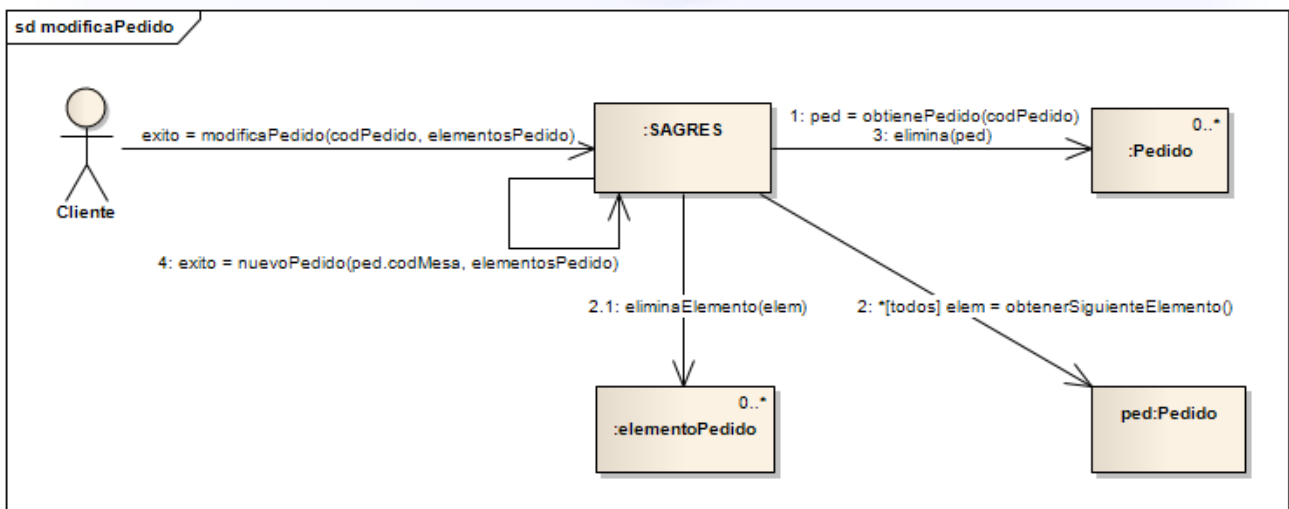
*Realizado por Sergio Rodríguez Lumley*



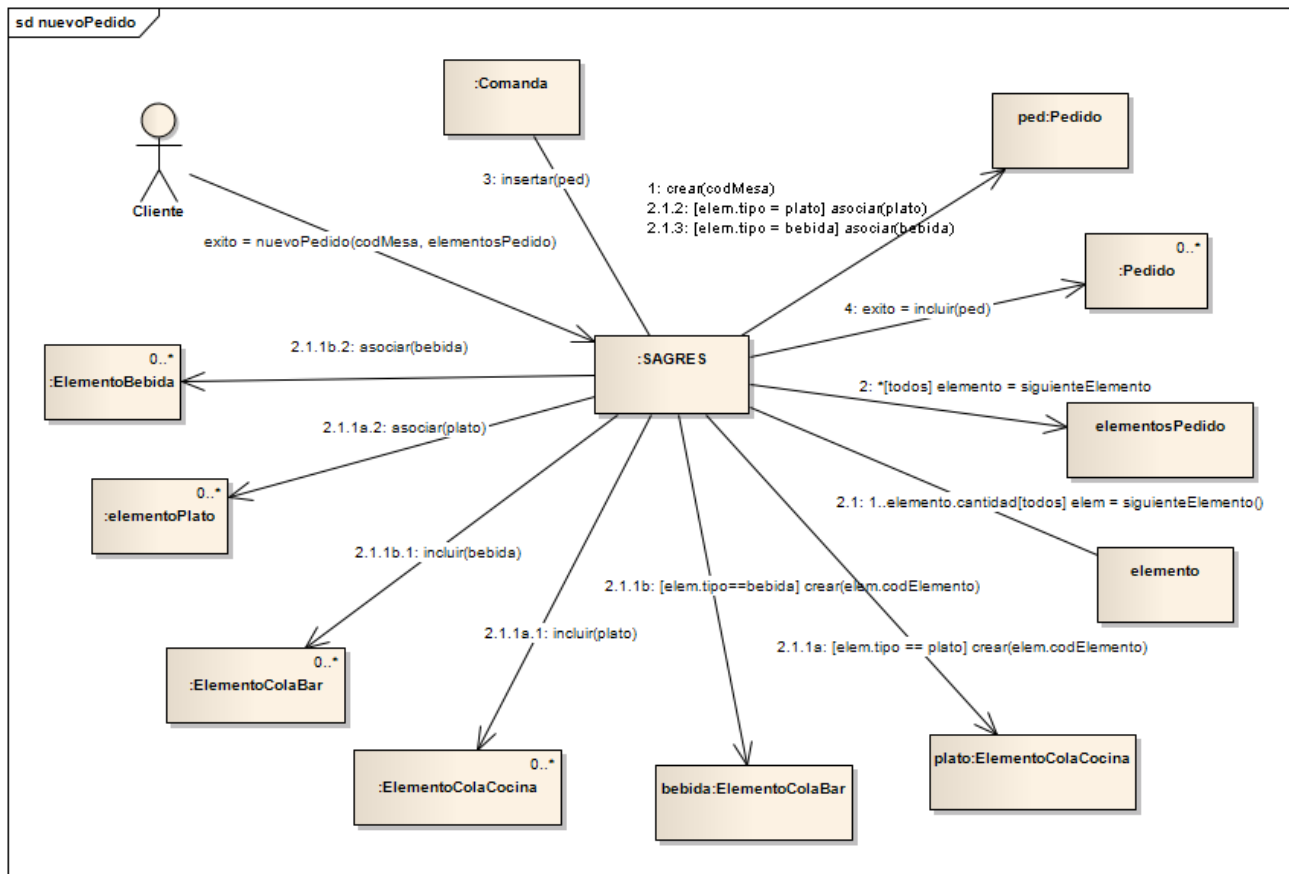
*Realizado por Adrián Víctor Pérez Lopera*



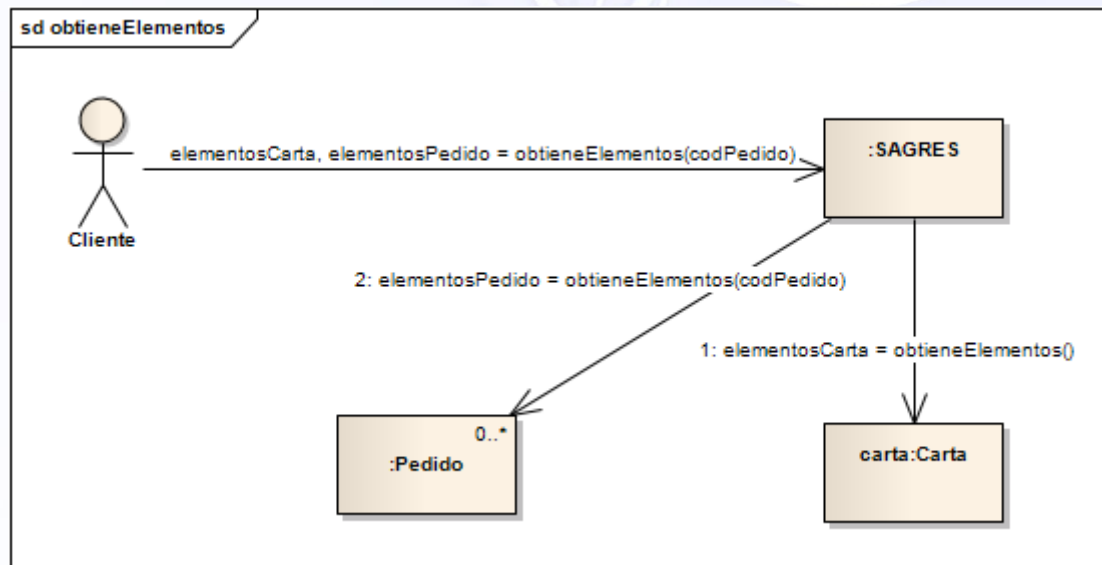
*Realizado por Adrián Víctor Pérez Lopera*



*Realizado por Adrián Víctor Pérez Lopera*

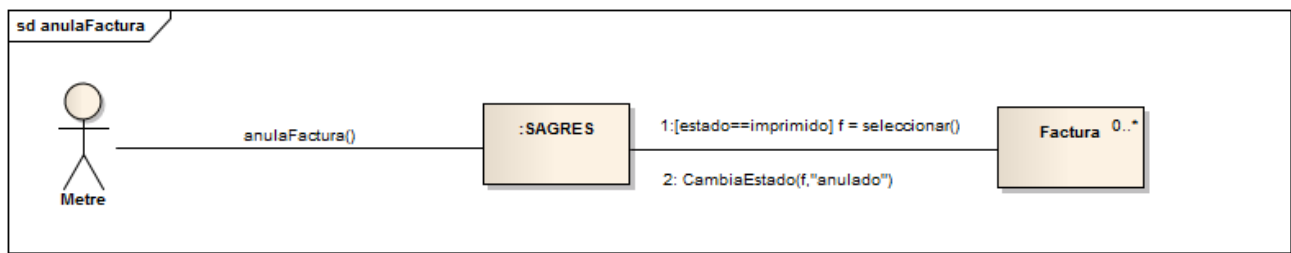


*Realizado por Adrián Víctor Pérez Lopera*

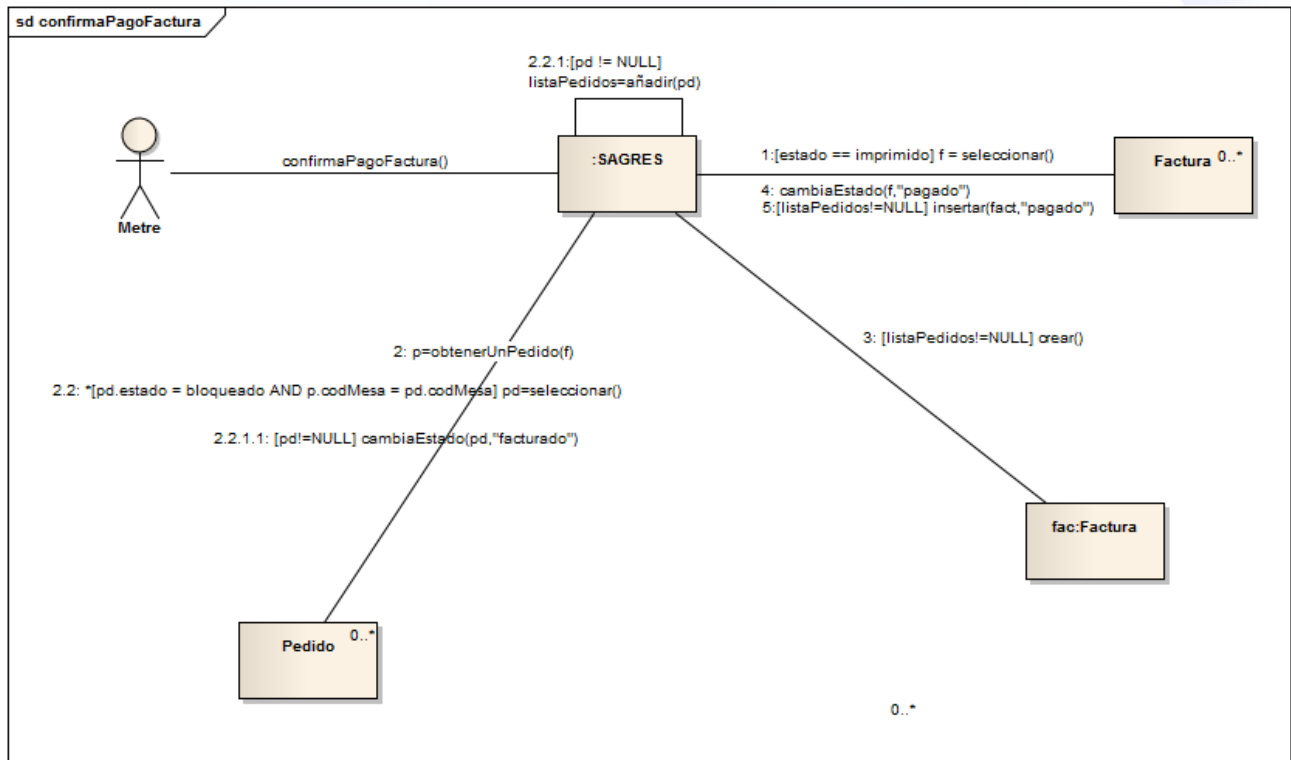


*Realizado por Adrián Víctor Pérez Lopera*

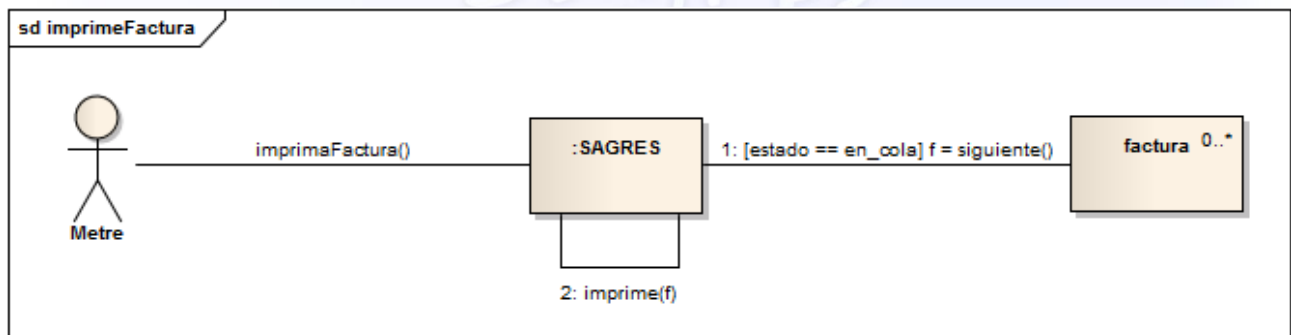




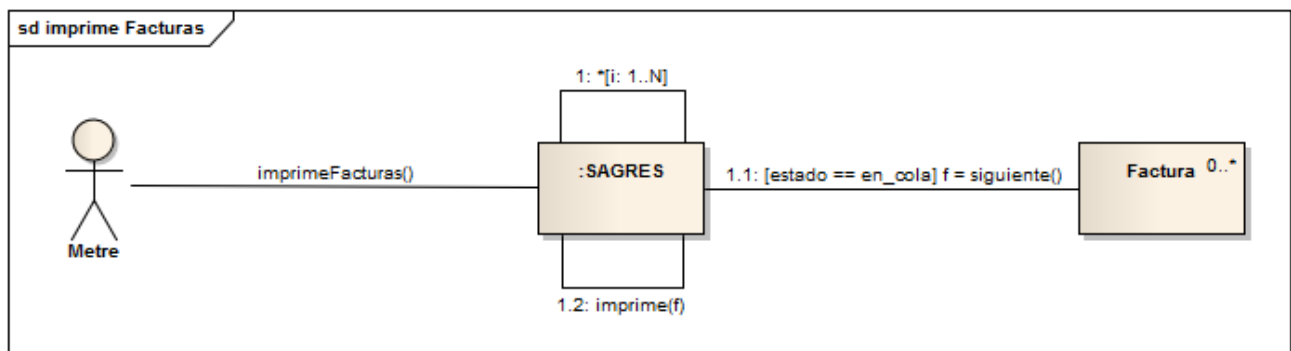
*Realizado por Nabil Sabeg*



*Realizado por Nabil Sabeg*



*Realizado por Nabil Sabeg*



*Realizado por Nabil Sabeg*

Sagres

## APÉNDICE 1.0

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

Sagres