

# Documento de MODELADO DE REQUISITOS



**Sistema de Administración y  
Gestión de RESTaurantes**



Adrián Víctor Pérez Lopera  
Sergio Rodríguez Lumley  
Nabil Sabeg

V1.1

# ÍNDICE DE CONTENIDO

Apartado de control de versiones.....	4
Introducción.....	5
1. Modelo funcional.....	6
<i>Identificación de actores.....</i>	<i>6</i>
Cocinero Jefe.....	6
Metre.....	6
Cliente.....	6
<i>Identificación de los requerimientos funcionales.....</i>	<i>6</i>
<i>Diagrama de Casos de Uso.....</i>	<i>8</i>
<i>Descripción de los casos de uso.....</i>	<i>9</i>
Cola de cocina.....	9
Cambia estado plato.....	9
Cola de bar.....	10
Cambia estado bebida.....	10
Pedidos de cliente.....	11
Nuevo pedido.....	11
Modifica pedido.....	12
Cambia estado Factura.....	13
Pide Factura.....	14
2. Subsistemas funcionales.....	15
<i>Identificación de los subsistemas funcionales.....</i>	<i>15</i>
<i>Diagrama de paquetes.....</i>	<i>15</i>
<i>Diagrama de casos de uso por división de reparto.....</i>	<i>16</i>
3. Requisitos no funcionales.....	17
<i>Identificación y descripción de los requisitos no funcionales del sistema.....</i>	<i>17</i>
Facilidad de uso.....	17
Fiabilidad.....	17
Rendimiento.....	17
Soporte.....	17
Implementación.....	17
Interfaz.....	17
Operaciones.....	17
Empaquetamiento.....	17

Legales.....	17
<b>4. Operaciones del sistema.....</b>	<b>18</b>
<i>Diagramas de secuencia del sistema.....</i>	<i>18</i>
Cola de cocina.....	18
Cola de bar.....	19
Nuevo Pedido.....	20
Modifica Pedido.....	21
Subsistema de gestión de facturas.....	22
<i>Identificación de operaciones del sistema.....</i>	<i>24</i>
Alta de pedidos.....	24
Modificar pedidos.....	25
Cola de cocina.....	26
Cola de bar.....	26
Facturas.....	27
<b>Apéndice 1.0.....</b>	<b>28</b>
<b>Apéndice 1.1.....</b>	<b>29</b>

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>
V1.0	15/04/10	Se ha generado el documento de Modelado de Requisitos.
V1.1	22/04/10	Se han corregido los errores mencionados en el fichero de revisión y cambios en las colas de bar y cocina.

The logo for 'Sagres' is centered on the page. It features a stylized, light-colored graphic above the word 'Sagres' written in a cursive, handwritten-style font.

## INTRODUCCIÓN

---

En esta segunda iteración, hemos decidido abordar con todos los subsistemas que hemos podido identificar salvo el subsistema de gestión de pedidos. Con ello, los subsistemas que trataremos en esta iteración son:

- Subsistema de pedidos de cliente
- Subsistema de cola de platos
- Subsistema de cola de bebidas
- Subsistema de facturación

El subsistema de pedidos de cliente se encarga de que los clientes puedan realizar, modificar y anular pedidos.

El subsistema de cola de platos se encargará de gestionar los platos de pedidos que se deben cocinar en cocina, englobando los estados posibles por los que pasa un plato.

El subsistema de cola de bebidas gestionará las bebidas que deberán ser servidas.

El subsistema de facturación recibe una lista de elementos, generará una factura y la apuntará a la cola de facturas del metre, con el estado "En cola", o similar, al igual que en la base de datos. La factura deberá ser emitida por el metre y entregada por un camarero. Una vez pagada, se elimina la factura de la lista de pendientes y se almacena en la base de datos como factura cerrada.

Todos estos subsistemas por funcionalidad, quedan recogidos en un único subsistema, ya que todos se encargan de tratar con pedidos, con lo que el subsistema que trataremos quedará definido como **Subsistema de gestión de pedidos**.

## 1. MODELO FUNCIONAL

---

### Identificación de actores

#### Cocinero Jefe

Es el encargado de cambiar el estado de los platos que se reciben para ser preparados. Según si los platos están a la espera de ser cocinados, cocinándose o preparados el cocinero notificará su estado.

#### Metre

Es el encargado de cambiar el estado de las bebidas que se reciben para ser servidas. También se ocupa de modificar el estado de las facturas, cuando un cliente quiera una factura, le llegará una notificación al metre y este cambiará su estado cuando la imprima para ser entregada y cuando sea pagada.

#### Cliente

El cliente pide pedidos, los cuales contienen tanto bebidas como platos. Puede elegir modificarlos mientras no se estén atendiendo, es decir, que puede quitar y añadir elementos de un pedido. Además en cualquier momento puede pedir la factura de todos los pedidos que ha hecho.

### Identificación de los requerimientos funcionales

Requerimientos funcionales aplicados a los subsistemas que se tratarán en esta iteración.

Identificador	Descripción
RF1	El sistema contempla a tres tipos de usuario: cliente, cocinero jefe y metre.
RF2	El cocinero jefe ha de poder cambiar el estado de un pedido a medida que vaya cocinando los platos.
RF3	El metre ha de poder cambiar el estado de un pedido a medida que vaya teniendo listas las bebidas de este.
RF4	El sistema ha de notificar al metre la petición de una factura.
RF5	El metre ha de poder modificar el estado de una factura a medida que se complete la entrega y transacción.
RF6	Las facturas se imprimen primero luego se pagan.
RF7	Las facturas deben tener un formato específico que contiene los elementos pedidos de una mesa mas la información del restaurante mas el total a pagar.
RF8	Un cliente debe poder elegir los elementos que desee para realizar un pedido.

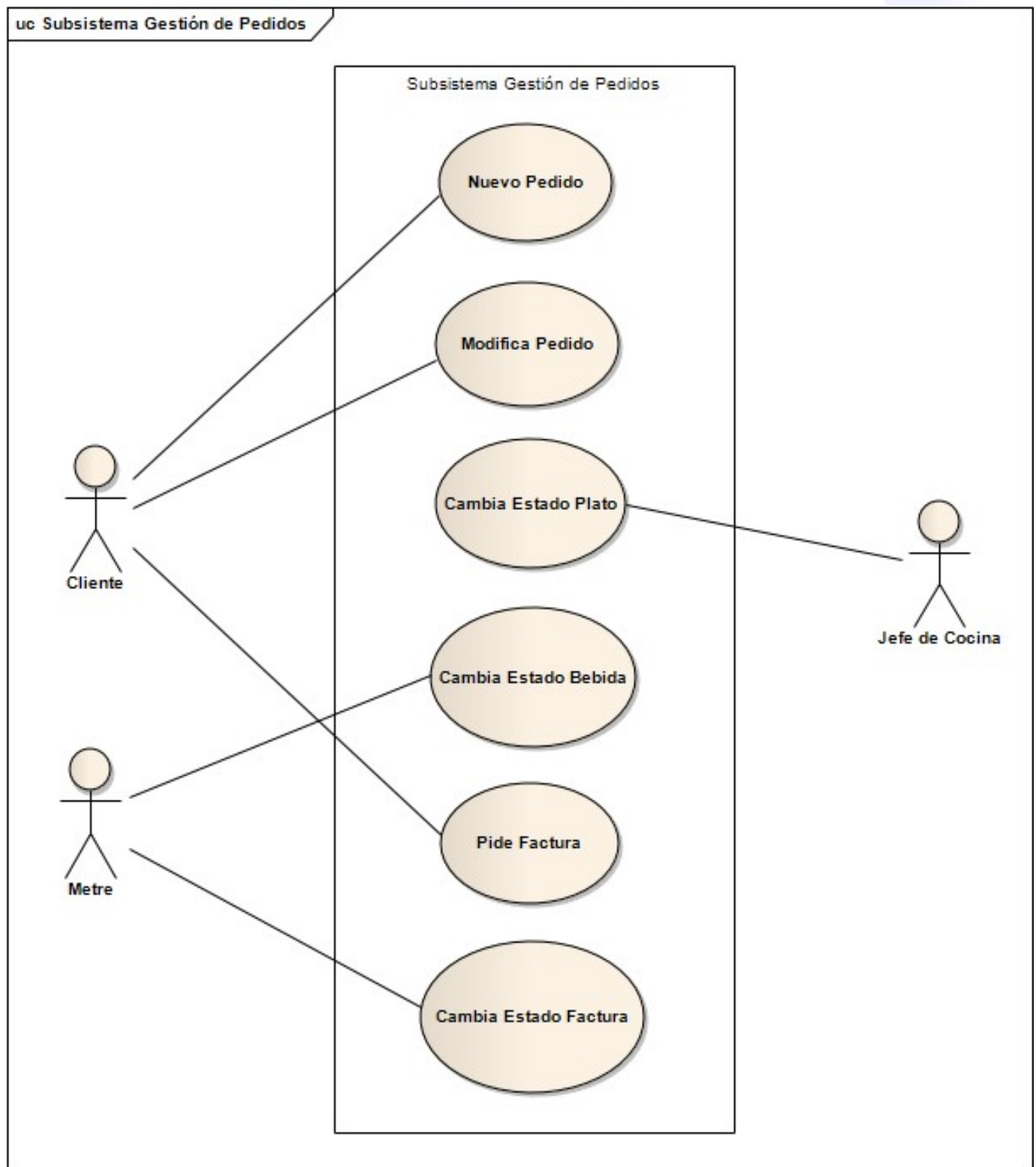
RF9	Un cliente debe poder modificar y anular cualquiera de los elementos de su pedido, o todos ellos, siempre y cuando este no se haya comenzado a atender.
RF10	Un cliente debe poder pedir una factura en el momento que desee.
RF12	El metre puede anular una factura pendiente de pagar si el cliente que la pidió se fue sin pagarla.

The logo for Sagres, featuring a stylized white outline of a wine bottle with a cork, positioned above the word "Sagres" written in a white, elegant cursive script. The background is a light blue gradient with a large, faint white circle on the right side.



## Diagrama de Casos de Uso

A continuación mostramos el diagrama de caso de uso, así como de algunos detalles respecto a este.





## Descripción de los casos de uso

### Cola de cocina

#### Cambia estado plato

Nombre del caso:	Cambia Estado Plato
Resumen:	Este caso de uso permite al jefe de cocina modificar el estado del siguiente plato de la cola de platos no atendidos, o bien el estado de un plato cualquiera de la lista de platos en preparación.
Dependencias	
Actores:	Jefe de cocina.
Precondiciones:	<ul style="list-style-type: none"><li>Existe algún plato, en la cola de platos no atendidos o en la lista de platos en preparación.</li></ul>
Postcondiciones:	<ul style="list-style-type: none"><li>Se cambió el estado de un plato.</li><li>Se cambió el estado del pedido.</li></ul>
Curso normal:	<ol style="list-style-type: none"><li>El usuario selecciona un plato del siguiente pedido que le muestra el sistema.</li><li>Si el plato se encuentra en estado “En cola”, el sistema cambia su estado a “Preparándose” y cambia el estado del pedido a “Bloqueado”.</li><li>Si el plato se encuentra en estado “Preparándose”, el sistema comprueba si era el último pedido por cambiar de estado:<ol style="list-style-type: none"><li>Si el plato era el último que quedaba del pedido por cambiar de estado, el sistema pide confirmación al usuario para cerrar el pedido.</li><li>El usuario confirma la acción.</li><li>El sistema cambia el estado del plato a “Preparado”.</li></ol></li></ol>
Cursos alternativos:	<ol style="list-style-type: none"><li>3.1a No era el último plato que quedaba del pedido por cambiar de estado, se cambia el estado del plato a “Preparado”.</li><li>3.2b El usuario cancela que ha finalizado el servicio del pedido.</li><li>3.3b El sistema no modifica el estado del plato, termina el caso.</li></ol>
Observaciones:	
Requisitos no funcionales específicos:	

*Especificado por Sergio Rodríguez Lumley*

## Cola de bar

### Cambia estado bebida

Nombre del caso:	Cambia Estado Bebida
Resumen:	Este caso de uso permite al metre cambiar el estado de la siguiente bebida de la cola de bebidas no atendidas.
Dependencias	
Actores:	Metre.
Precondiciones:	<ul style="list-style-type: none"><li>Existe alguna bebida en la cola de bebidas no atendidas.</li></ul>
Postcondiciones:	<ul style="list-style-type: none"><li>Se cambió el estado de una bebida.</li><li>Se cambió el estado del pedido.</li></ul>
Curso normal:	<ol style="list-style-type: none"><li>4 El metre selecciona una bebida del siguiente pedido de la cola de bebidas no atendidas que le muestra el sistema.</li><li>5 El sistema cambia el estado del pedido a “bloqueado” y comprueba si es la última bebida para cambiar de estado<ol style="list-style-type: none"><li>5.1 Si la bebida era la última que quedaba del pedido por cambiar de estado, el sistema pide confirmación al usuario para cerrar el pedido de bebidas.</li><li>5.2 El usuario confirma que ha finalizado el servicio del pedido.</li><li>5.3 El sistema cambia el estado de la bebida a “Preparado”.</li></ol></li></ol>
Cursos alternativos:	<ol style="list-style-type: none"><li>2.1a No era la última bebida que quedaba del pedido por cambiar de estado, se cambia el estado de la bebida a “Preparado”.</li><li>2.2b El usuario cancela que ha finalizado el servicio del pedido.</li><li>2.3b El sistema no modifica el estado de la bebida, termina el caso.</li></ol>
Observaciones:	
Requisitos no funcionales específicos:	

*Especificado por Sergio Rodríguez Lumley*

## Pedidos de cliente

### Nuevo pedido

<b>Nombre del caso:</b>	Nuevo Pedido
<b>Resumen:</b>	Este caso de uso permite al cliente realizar un nuevo pedido.
<b>Dependencias</b>	
<b>Actores:</b>	Cliente.
<b>Precondiciones:</b>	<ul style="list-style-type: none"><li>Existe en la carta algún plato o bebida disponible.</li></ul>
<b>Postcondiciones:</b>	<ul style="list-style-type: none"><li>Se creó un nuevo pedido asociado a la mesa.</li></ul>
<b>Curso normal:</b>	<ol style="list-style-type: none"><li>El cliente inicia el caso de uso.</li><li>El sistema le muestra al cliente la carta.</li><li>Mientras el cliente quiera modificar elementos del pedido:<ol style="list-style-type: none"><li>Si añade un nuevo elemento:<ol style="list-style-type: none"><li>El sistema añade el elemento del pedido.</li><li>El sistema muestra al cliente la carta y los elementos actuales que tiene su pedido.</li></ol></li><li>Si elimina un elemento antes añadido:<ol style="list-style-type: none"><li>El sistema elimina el elemento del pedido.</li><li>El sistema muestra al cliente la carta y los elementos actuales que tiene su pedido.</li></ol></li></ol></li><li>El cliente confirma el pedido.</li><li>El sistema registra el pedido y avisa al cliente que el pedido ha sido realizado con éxito.</li></ol>
<b>Cursos alternativos:</b>	4.a El cliente cancela el pedido. Se cancela el caso de uso.
<b>Observaciones:</b>	Cuando se confirma el pedido, los elementos de este deben encolarse en la cola de cocina y en la cola de bar.
<b>Requisitos no funcionales específicos:</b>	

*Especificado por Adrián Víctor Pérez Lopera*

### *Modifica pedido*

<b>Nombre del caso:</b>	Modifica Pedido
<b>Resumen:</b>	Este caso de uso permite al cliente modificar un pedido en el cual ninguno de sus elementos están en preparación.
<b>Dependencias</b>	
<b>Actores:</b>	Cliente.
<b>Precondiciones:</b>	
<b>Postcondiciones:</b>	<ul style="list-style-type: none"><li>• Se modificó un pedido asociado a la mesa.</li></ul>
<b>Curso normal:</b>	<ol style="list-style-type: none"><li>1 El cliente inicia el caso de uso.</li><li>2 El sistema le muestra al cliente los pedidos que hay asociados a su mesa y que pueden modificarse.</li><li>3 El cliente selecciona el pedido.</li><li>4 El sistema muestra al cliente la carta y los elementos actuales que tiene su pedido.</li><li>5 Mientras el cliente quiera modificar elementos del pedido:<ol style="list-style-type: none"><li>5.1 Si añade un nuevo elemento:<ol style="list-style-type: none"><li>5.1.1 El sistema añade el elemento del pedido.</li><li>5.1.2 El sistema muestra al cliente la carta y los elementos actuales que tiene su pedido.</li></ol></li><li>5.2 Si elimina un elemento antes añadido:<ol style="list-style-type: none"><li>5.2.1 El sistema elimina el elemento del pedido.</li><li>5.2.2 El sistema muestra al cliente la carta y los elementos actuales que tiene su pedido.</li></ol></li></ol></li><li>6 El cliente confirma la modificación.</li><li>7 El sistema registra los cambios y avisa al cliente que el pedido ha sido modificado con éxito.</li></ol>
<b>Cursos alternativos:</b>	6.a El cliente cancela la modificación. Se cancela el caso de uso.
<b>Observaciones:</b>	Cuando se modifica un pedido, los elementos de este pasan a estar en último lugar tanto en la cola de cocina como en la de bar.
<b>Requisitos no funcionales específicos:</b>	

*Especificado por Adrián Víctor Pérez Lopera*

## Cambia estado Factura

<b>Nombre del caso:</b>	Cambia Estado Factura
<b>Resumen:</b>	El Metre utiliza este caso de uso para cambiar estado de las facturas.
<b>Dependencias:</b>	
<b>Actores:</b>	Metre
<b>Precondiciones:</b>	
<b>Postcondiciones:</b>	
<b>Curso normal:</b>	<ol style="list-style-type: none"> <li>1 El caso de uso comienza cuando el usuario quiere actualizar el estado de una o varias facturas.</li> <li>2 El usuario selecciona: <ol style="list-style-type: none"> <li>2.1 Imprimir todas las facturas pendientes. <ol style="list-style-type: none"> <li>2.1.1 El sistema imprime todas las facturas pendientes.</li> </ol> </li> <li>2.2 Cambiar estado de factura(s) de una mesa: <ol style="list-style-type: none"> <li>2.2.1 El usuario selecciona la mesa. <ol style="list-style-type: none"> <li>2.2.1.1 El usuario selecciona la factura: <ol style="list-style-type: none"> <li>2.2.1.1.1 Si se quiere imprimir la factura seleccionada: <ol style="list-style-type: none"> <li>2.2.1.1.1.1 El sistema imprime la factura.</li> </ol> </li> <li>2.2.1.1.2 Si se quiere pagar la factura: <ol style="list-style-type: none"> <li>2.2.1.1.2.1 El usuario confirma el pago.</li> </ol> </li> </ol> </li> </ol> </li> <li>2.3 Anular factura no pagada: <ol style="list-style-type: none"> <li>2.3.1 el usuario selecciona la mesa. <ol style="list-style-type: none"> <li>2.3.1.1 El usuario selecciona la factura no pagada.</li> <li>2.3.1.2 El usuario confirma al sistema que la factura no va a ser pagada.</li> </ol> </li> </ol> </li> </ol> </li> </ol> </li></ol>
<b>Cursos alternativos:</b>	2. El usuario sale del caso del uso.
<b>Observaciones:</b>	Las facturas van a ser ordenadas en la cola de factura por mesas, un cliente puede hacer varios pedidos, cuando el cliente quiere pagar se visualiza todas la facturas de esa mesa, si hay facturas que el cliente no ha pedido imprimir las imprime. Si se confirma que una factura pendiente en la mesa no ha sido pagada se marca como no pagada.
<b>Requisitos no funcionales específicos:</b>	

*Especificado por Nabil Sabeg*

## *Pide Factura*

<b>Nombre del caso:</b>	Pide Factura
<b>Resumen:</b>	El usuario pide que se imprime su factura.
<b>Dependencias:</b>	
<b>Actores:</b>	Cliente
<b>Precondiciones:</b>	
<b>Postcondiciones:</b>	Cola de facturas pendiente de imprimir actualizada.
<b>Curso normal:</b>	<ol style="list-style-type: none"><li>1 El caso de uso comienza cuando el cliente quiere la factura,</li><li>2 El sistema actualiza la cola de facturas.</li></ol>
<b>Cursos alternativos:</b>	
<b>Observaciones:</b>	La información que debe tener la factura es los pedidos con sus elementos y el código de mesa correspondiente.
<b>Requisitos no funcionales específicos:</b>	

*Especificado por Nabil Sabeg*

## 2. SUBSISTEMAS FUNCIONALES

---

### *Identificación de los subsistemas funcionales*

Debido a la brevedad de los subsistemas funcionales que quedarían en caso de repartir los casos de usos considerados, asumiremos que todo el sistema a desarrollar en esta iteración se encuentra dentro un sólo subsistema, al que denominaremos Subsistema Gestión de Pedidos. Dentro de este subsistema se realizan funciones de **gestión de facturación**, de **gestión de pedidos del cliente** y de **gestión de cola de cocina y cola de bar**.

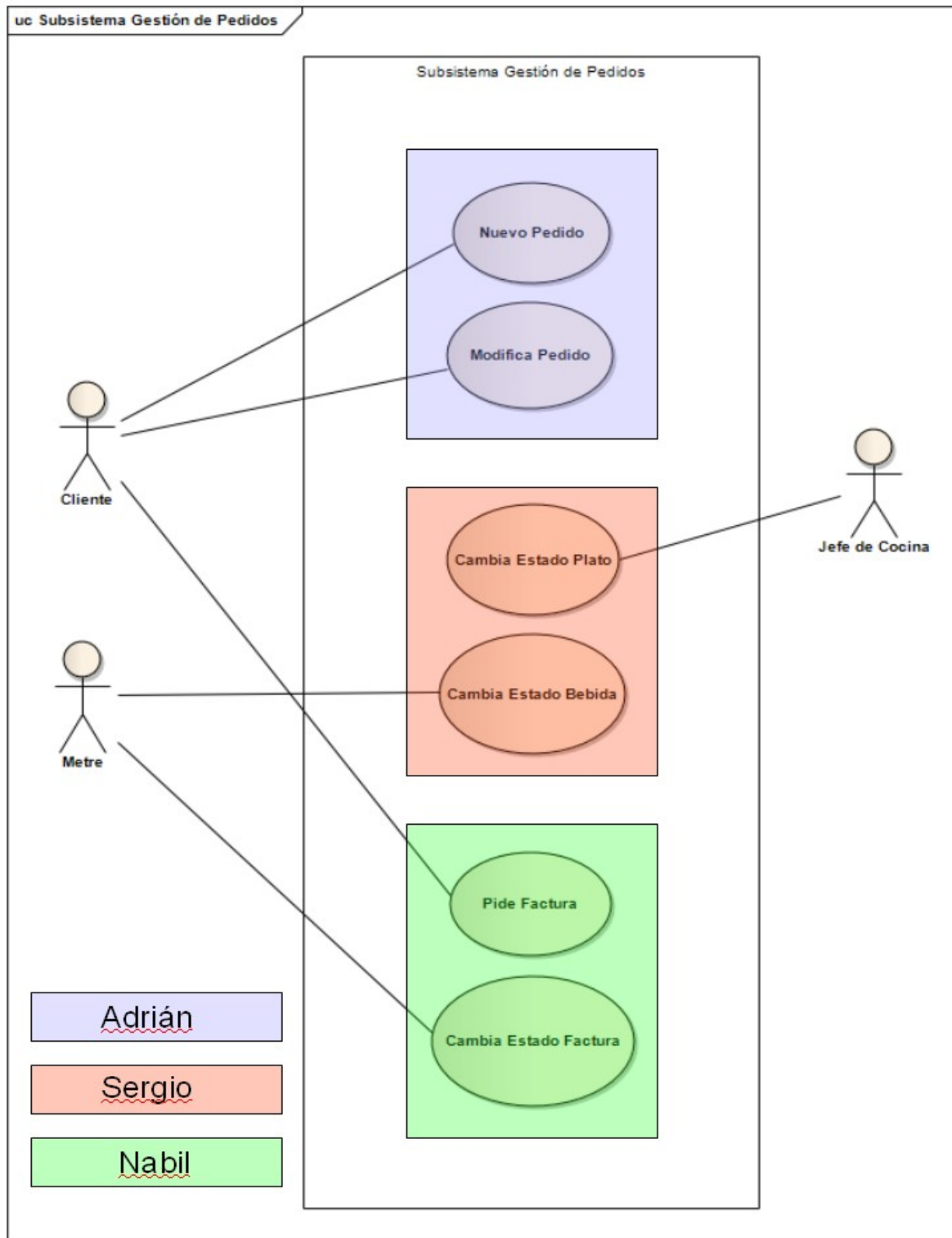
### *Diagrama de paquetes*

En este documento no incluiremos diagrama de paquetes ya que suponemos un único paquete, con lo que no hay ninguna información adicional que aportar en este campo.





## Diagrama de casos de uso por división de reparto



### 3. REQUISITOS NO FUNCIONALES

#### Identificación y descripción de los requisitos no funcionales del sistema

<i>Facilidad de uso</i>	<ul style="list-style-type: none"><li>• La interfaz debe ser bastante intuitiva para que la utilice cualquier usuario correctamente.</li><li>• Dado que se utilizara una pantalla táctil, el diseño debe ser adecuado para utilizarlo fácilmente.</li></ul>
<i>Fiabilidad</i>	<ul style="list-style-type: none"><li>• El sistema debe ser fiable y robusto.</li><li>• Se deben hacer copias de seguridad cada cierto tiempo para poder recuperarse de posibles caídas o fallos.</li></ul>
<i>Rendimiento</i>	<ul style="list-style-type: none"><li>• El sistema debe ser eficiente y rápido.</li><li>• El sistema debe sincronizar las peticiones simultaneas de los cliente.</li></ul>
<i>Soporte</i>	<ul style="list-style-type: none"><li>• El sistema debe ejecutarse sobre cualquier equipo usando la maquina virtual de java.</li><li>• El gestor de la base de datos debe ser compatible con estos equipos.</li></ul>
<i>Implementación</i>	<ul style="list-style-type: none"><li>• Los lenguajes que se van a utilizar son:<ul style="list-style-type: none"><li>◦ java</li><li>◦ php</li><li>◦ sql</li></ul></li><li>• los equipos utilizados son terminales táctiles con conexión permanente con el servidor.</li></ul>
<i>Interfaz</i>	<ul style="list-style-type: none"><li>• El sistema no interactuara con otro sistema externo,todas las operaciones serán totalmente internas al sistema.</li></ul>
<i>Operaciones</i>	<ul style="list-style-type: none"><li>• El sistema dispondrá de toda la información contenida en la base de datos al iniciarse.</li></ul>
<i>Empaquetamiento</i>	<ul style="list-style-type: none"><li>• La base de datos debe ser instalada y configurada para su uso por parte del sistema en el local.</li><li>• El sistema se ejecuta directamente usando la maquina virtual de java en caso de los terminales del restaurante , o bien usando pagina web en caso de terminales disponibles en la habitaciones del hotel.</li></ul>
<i>Legales</i>	<ul style="list-style-type: none"><li>• El sistema debe cumplir las disposiciones recogidas en la Ley Orgánica de Datos Personales y en el Reglamento de medidas de seguridad.</li><li>• Al usar software gratuito para el desarrollo del proyecto no será necesaria la compra de licencias de ningún tipo.</li></ul>

## 4. OPERACIONES DEL SISTEMA

### Diagramas de secuencia del sistema

#### Cola de cocina

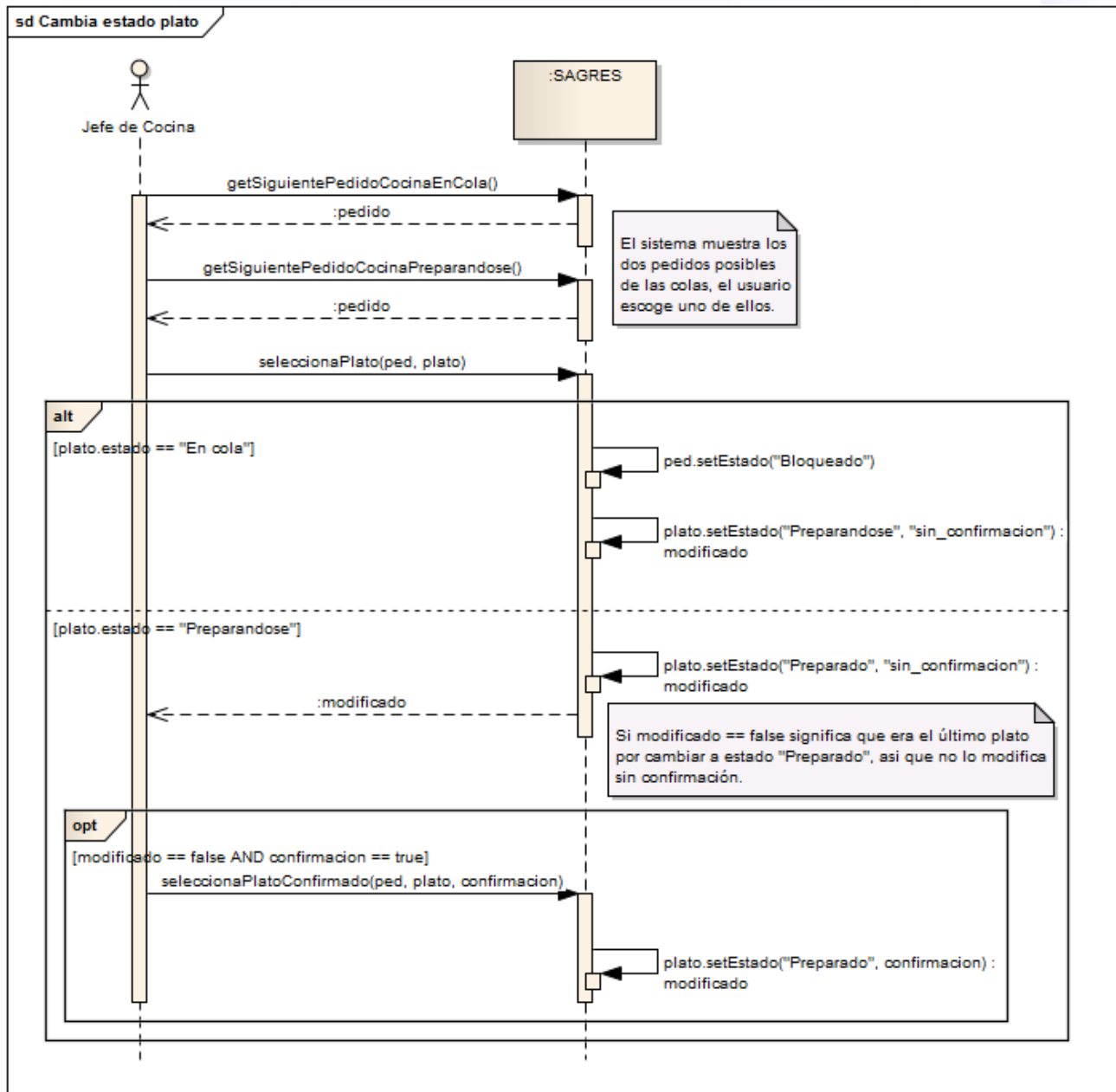


Diagrama realizado por Sergio Rodríguez Lumley

## Cola de bar

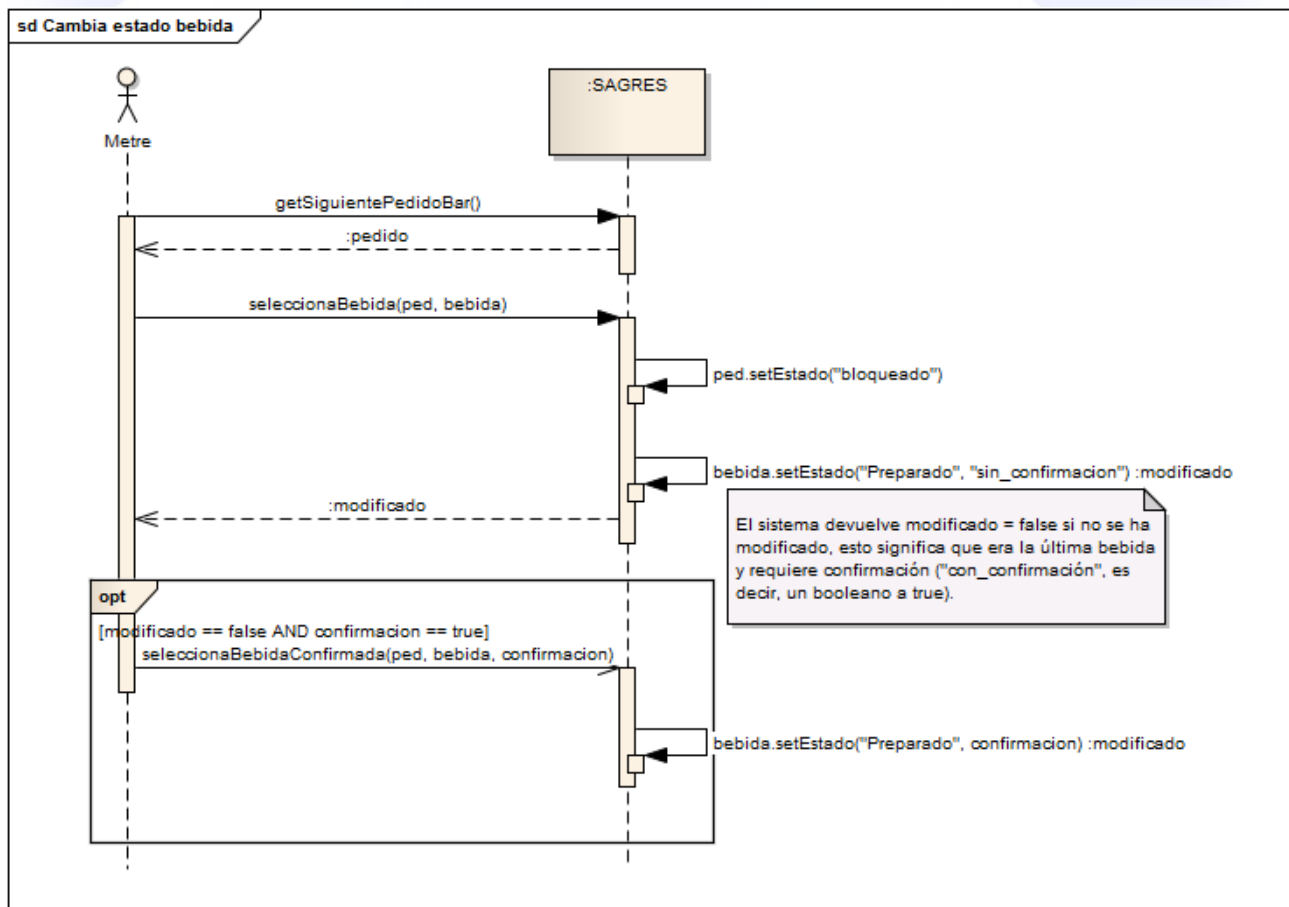


Diagrama realizado por Sergio Rodríguez Lumley

## Nuevo Pedido

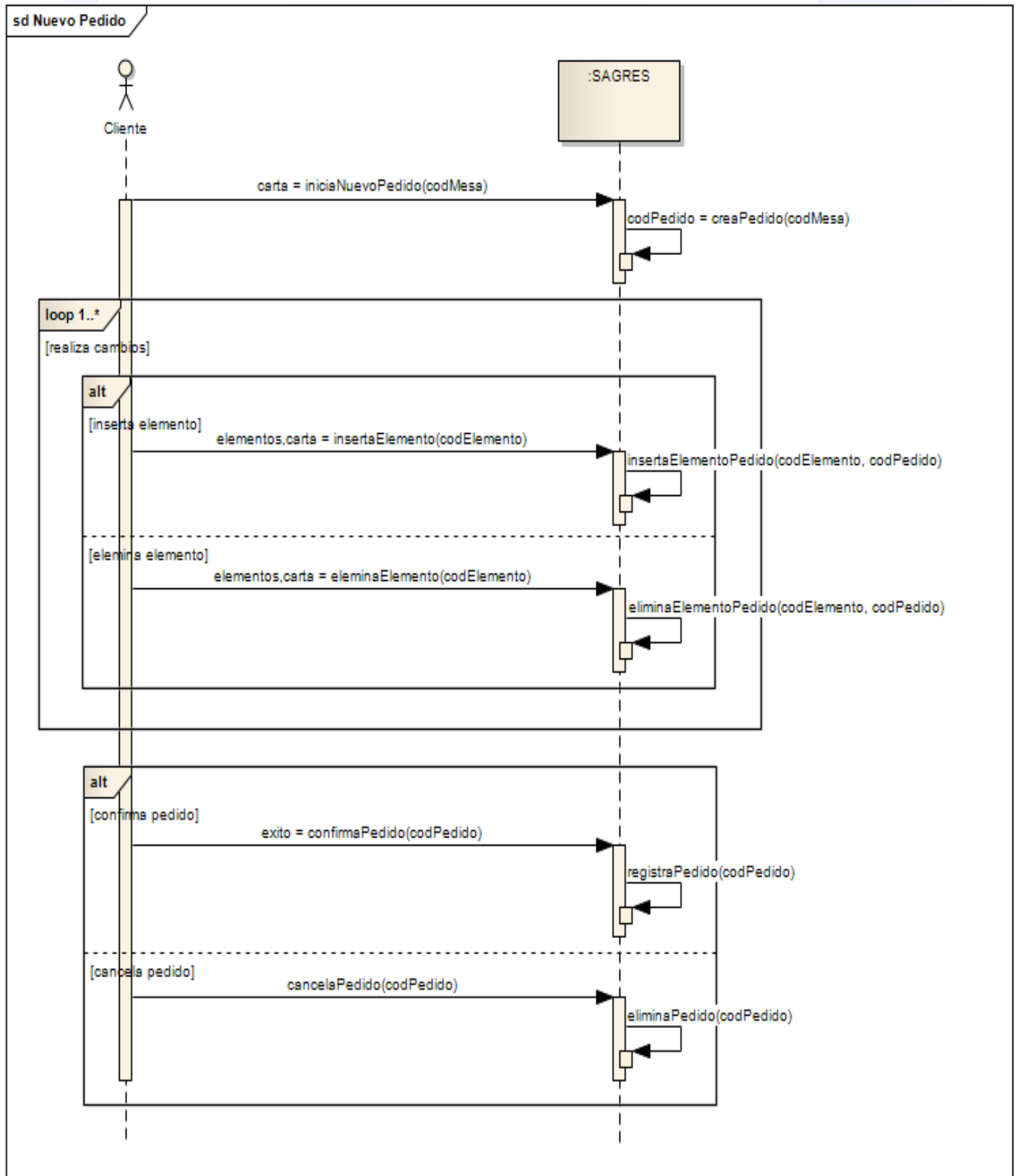


Diagrama realizado por Adrián Víctor Pérez Lopera

## Modifica Pedido

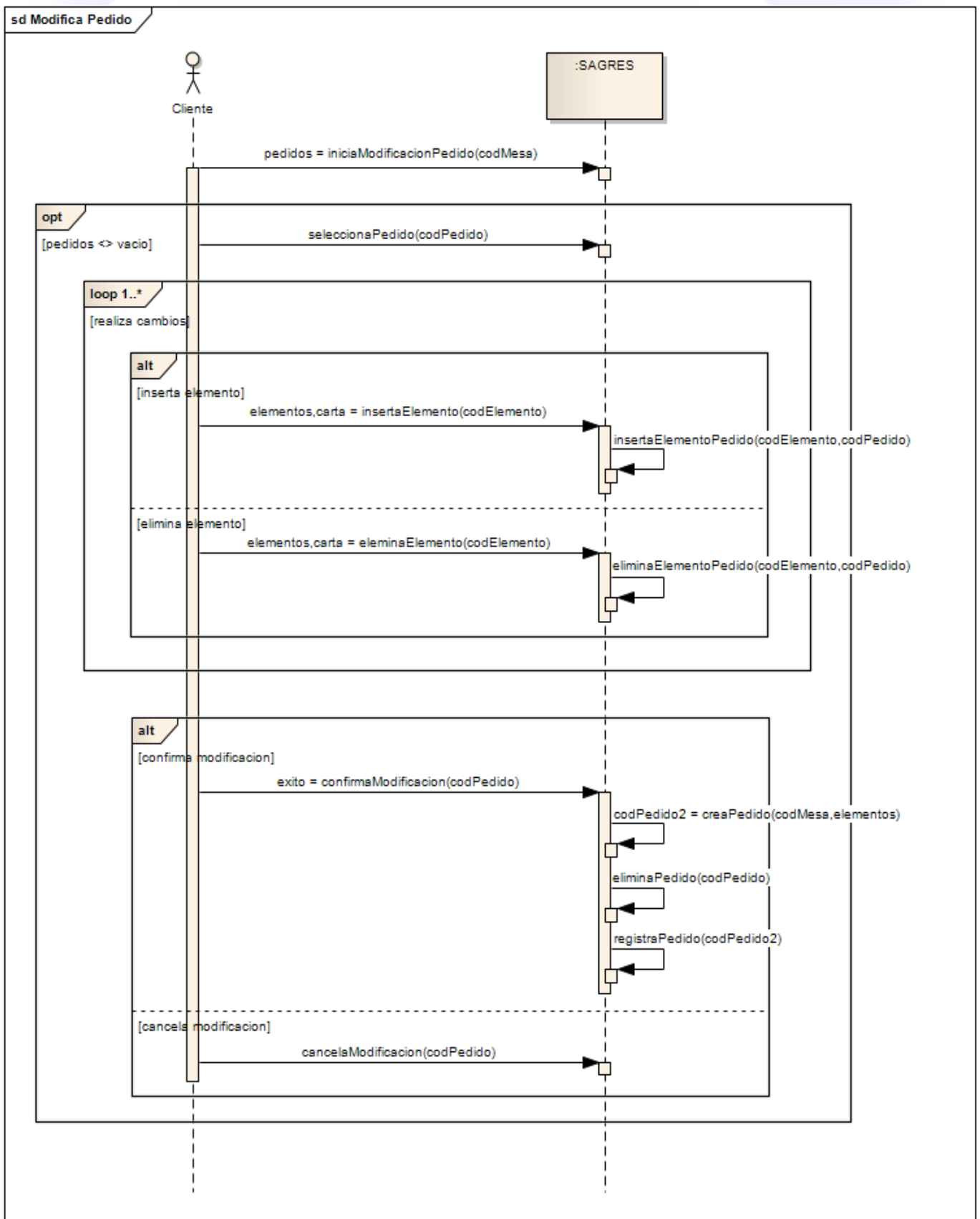


Diagrama realizado por Adrián Víctor Pérez Lopera

## Subsistema de gestión de facturas

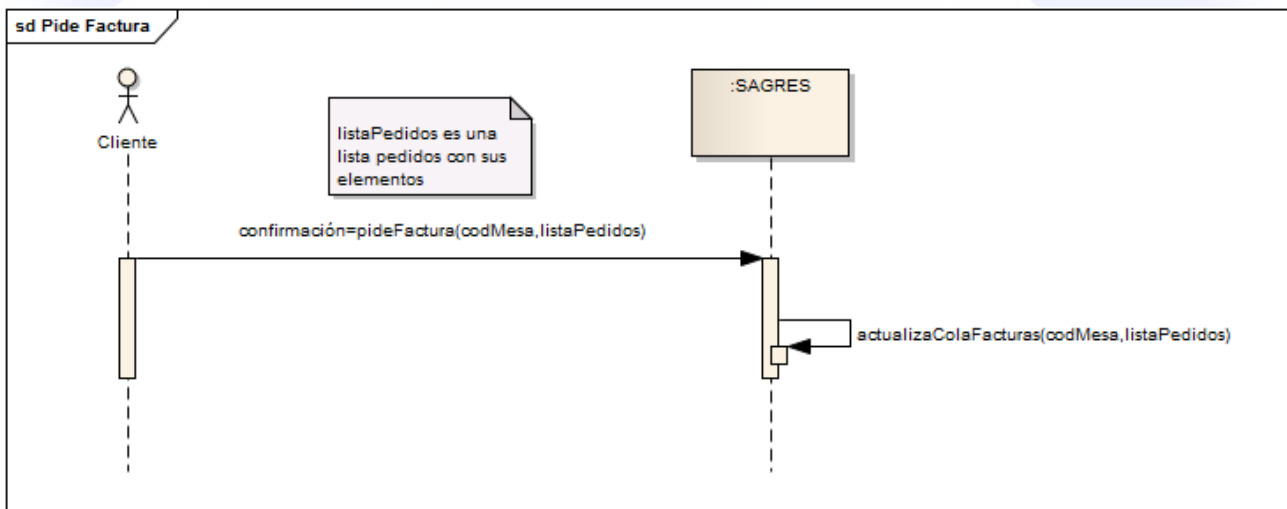
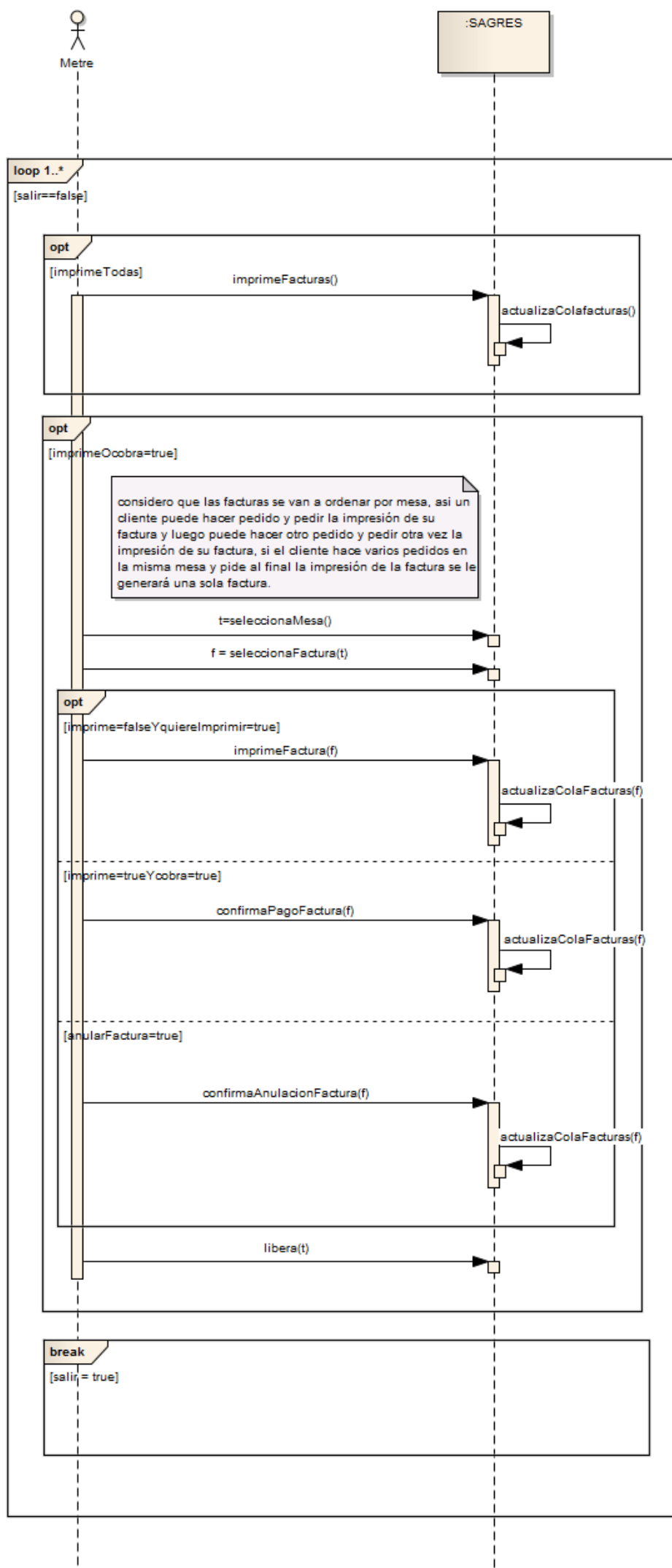


Diagrama realizado por Nabil Sabeg

Sagres





## Identificación de operaciones del sistema

En este apartado vamos a enumerar las operaciones identificadas en el sistema ordenadas por el subsistema en el que se utilizan. Esto significa que puede haber operaciones que se repiten en distintos subsistemas.

### Alta de pedidos

Operación	Descripción
carta = iniciaNuevoPedido(codMesa)	Inicia el caso de uso introduciendo la mesa (referenciada por “codMesa”) y retorna los elementos de la carta.
codPedido = creaPedido(codMesa)	Crea un pedido asociado a la mesa que hace referencia “codMesa” y retorna el código del pedido.
elementos, carta = insertaElemento(codElemento)	Inserta un nuevo elemento, referenciado por “codElemento”, y retorna los elementos actuales del pedido y los elementos de la carta.
insertaElementoPedido(codElemento, codPedido)	Inserta el elemento que hace referencia “codElemento” en el pedido que hace referencia “codPedido”.
elementos, carta = eliminaElemento(codElemento)	Elimina un elemento, referenciado por “codElemento”, y retorna los elementos actuales del pedido y los elementos de la carta.
eliminaElementoPedido(codElemento, codPedido)	Elimina el elemento que hace referencia “codElemento” en el pedido que hace referencia “codPedido”.
exito = confirmaPedido(codPedido)	Realiza confirmación del pedido que hace referencia “codPedido”, y retorna en “exito” el valor “true” si el pedido ha sido registrado correctamente, y “false” en caso contrario.
registraPedido(codPedido)	Registra el pedido en el sistema, y dirige los elementos del mismo a las colas de bar y de cocina.
cancelaPedido(codPedido)	Cancela el pedido referenciado por “codPedido”.
eliminaPedido(codPedido)	Elimina el pedido referenciado por “codPedido” del sistema.

*Operaciones definidas por Adrián Víctor Pérez Lopera*

## Modificar pedidos

Operación	Descripción
pedidos = iniciaModificacionPedido(codMesa)	Inicia el caso de uso y retorna la lista de pedidos modificables asociados con la mesa referenciada por “codMesa”
seleccionaPedido(codPedido)	Selecciona un pedido, referenciado por “codPedido”.
elementos, carta = insertaElemento(codElemento)	Utilizado también en Nuevo Pedido. Inserta un nuevo elemento, referenciado por “codElemento”, y retorna los elementos actuales del pedido y los elementos de la carta.
insertaElementoPedido(codElemento, codPedido)	Utilizado también en Nuevo Pedido. Inserta el elemento que hace referencia “codElemento” en el pedido que hace referencia “codPedido”.
elementos, carta = eliminaElemento(codElemento)	Utilizado también en Nuevo Pedido. Elimina un elemento, referenciado por “codElemento”, y retorna los elementos actuales del pedido y los elementos de la carta.
eliminaElementoPedido(codElemento, codPedido)	Utilizado también en Nuevo Pedido. Elimina el elemento que hace referencia “codElemento” en el pedido que hace referencia “codPedido”.
exito = confirmaModificacion(codPedido)	Realiza confirmación de la modificación del pedido referenciado por “codPedido”, y retorna en “exito” el valor “true” si el pedido ha sido modificado correctamente, y “false” en caso contrario.
codPedido2 = creaPedido(codMesa, elementos)	Utilizado también en Nuevo Pedido. Crea un pedido asociado a la mesa que hace referencia “codMesa”, con los elementos “elementos” y retorna el código del pedido.
eliminaPedido(codPedido)	Utilizado también en Nuevo Pedido. Elimina el pedido referenciado por “codPedido” del sistema.
registraPedido(codPedido2)	Utilizado también en Nuevo Pedido. Registra el pedido en el sistema, y dirige los elementos del mismo a las colas de bar y de cocina.
cancelaModificacion(codPedido)	Cancela la modificación del pedido que referencia “codPedido”.

Operaciones definidas por Adrián Víctor Pérez Lopera

## Cola de cocina

Operación	Descripción
Pedido = getSiguientePedidoCocinaEnCola();	Devuelve el pedido con la fecha más lejana (es decir, el más antiguo) que tenga algún ElementoColaCocina en estado “En cola”.
Pedido = getSiguientePedidoCocinaPreparandose();	Devuelve el pedido con la fecha más lejana (es decir, el más antiguo) que tenga algún ElementoColaCocina en estado “Preparandose” y ninguno “En cola”.
Bool = seleccionaPlato(Pedido pedido, ElementoColaCocina plato);	Selecciona un “plato” del Pedido “pedido”, cambia el estado del pedido a “Bloqueado” y también el del plato. Si estaba “En cola” lo cambia a “Preparandose”. Si estaba “Preparandose” se cambiará a “Preparado” si no era el último elemento por cambiar a “Preparado” devolviendo “True”, o “False” si era el último (y no cambia el estado).
seleccionaPlatoConfirmado(Pedido pedido, ElementoColaCocina plato, Bool confirmacion);	Selecciona un “plato” del Pedido “pedido”, cambia el estado del pedido a “Bloqueado” y también el del plato. Si estaba “En cola” lo cambia a “Preparandose”. Si estaba “Preparandose” se cambiará “Preparado” si no era el último elemento o bien si lo era y confirmación tiene de valor “True”.

*Operaciones definidas por Sergio Rodríguez Lumley*

## Cola de bar

Operación	Descripción
Pedido = getSiguientePedidoBar();	Devuelve el pedido con la fecha más lejana (es decir, el más antiguo) que tenga algún ElementoColaBar en estado “En cola”.
Bool = seleccionaBebida(Pedido pedido, ElementoColaBar bebida);	Selecciona una “bebida” del Pedido “pedido”, cambia el estado del pedido a “Bloqueado” y también el del plato. Si estaba “En cola” se cambiará a “Preparado” si no era el último elemento por cambiar a “Preparado” devolviendo “True”, o “False” si era el último (y no cambia el estado).
seleccionaBebidaConfirmada(Pedido pedido, ElementoColaBar bebida, Bool confirmacion);	Selecciona una “bebida” del Pedido “pedido”, cambia el estado del pedido a “Bloqueado” y también el de la bebida. Si estaba “En cola” se cambiará “Preparado” si no era el último elemento o bien si lo era y confirmación tiene de valor “True”.

*Operaciones definidas por Sergio Rodríguez Lumley*

## Facturas

Operación	Descripción
Confirmacion=pideFactura(Cod Mesa,listaPedidos)	Manda al sistema la petición de imprimir una factura, listaPedidos es una lista que puede contener un pedidos o varios. Esta función insertara una sola factura en la cola.
actualizaColaFacturas(CodMesa, listaPedidos)	Actualiza la cola de facturas.
imprimeFacturas()	Imprime todas las facturas pendientes de imprimir en la cola de facturas.
actualizaColafacturas()	Actualiza la cola de facturas, se llama a esta función cuando se elige la opción imprimeFacturas().
t=seleccionaMesa()	Selecciona una mesa a asigna su referencia a t.
f=seleccionafacturas(t)	Selecciona una factura de la mesa t y asigna su referencia a f.
imprimeFactura(f)	Imprime la factura f.
confirmaPagoFctura(f)	Confirma el pago de la factura f
confirmaAnulacionfactura(f)	Confirma la anulación de la factura f.
actualizacolaFactura(f)	Actualiza el estado de la factura f en la cola de facturas: <ul style="list-style-type: none"> <li>• se la factura f se imprime se actualiza su estado a imprimida.</li> <li>• si la factura f es la ultima en la mesa sin pagar y se paga, se quita de la cola la facturas.</li> <li>• Si la factura f se anula se quila de la cola de facturas.</li> </ul>
libera(t)	Libera la mesa referenciada por t.

*Operaciones definidas por sabeg nabil*

## APÉNDICE 1.0

<i>Fecha</i>	15/04/10
<i>Descripción del problema</i>	Se ha generado el documento de Modelado de Requisitos.
<i>Impacto del problema</i>	
<i>Soluciones adoptadas</i>	<ul style="list-style-type: none"><li>•</li></ul>
<i>Anexos a la versión</i>	

Sagres

## APÉNDICE 1.1

<i>Fecha</i>	22/04/10
<i>Descripción del problema</i>	<ol style="list-style-type: none"><li>1. Se especificó en la introducción que las facturas se guardan en la base de datos únicamente cuando se cierra la factura.</li><li>2. La especificación de los casos de uso “Cambia estado plato” y “Cambia estado bebida” estaban poco detallados. En consecuencia, los diagramas de secuencia y sus operaciones correspondientes contenían errores.</li><li>3. Se han especificado una serie de puntos en el documento de revisión del “Documento de Modelado de Requisitos v1.0”.</li></ol>
<i>Impacto del problema</i>	<ol style="list-style-type: none"><li>1. Puede provocar inconsistencia en la persistencia de datos.</li><li>2. Falta de entendimiento del sistema de colas de bar y de cocina, lo que puede dar lugar a un mal análisis y diseño del mismo.</li><li>3. Posibles errores de interpretación.</li></ol>
<i>Soluciones adoptadas</i>	<ol style="list-style-type: none"><li>1. Se ha especificado que las facturas se guardan en todo momento.</li><li>2. Se han vuelto a especificar los casos de uso “Cambia estado plato” y “Cambia estado bebida”, se han vuelto a generar sus diagramas de secuencia y sus operaciones correspondientes.</li><li>3. Se han corregido varios de estos puntos.</li></ol>
<i>Anexos a la versión</i>	