

Documento de DISEÑO



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. Estilo arquitectónico.....	4
2. Diagrama de clases del diseño.....	5
3. Diagrama de paquetes estructurales.....	6
4. Estilo arquitectónico detallado.....	8
5. Identificación de las interfaces.....	10
6. Diagrama de componentes.....	16
7. Diagramas de colaboración.....	17
<i>getSiguientePedidoBar.....</i>	<i>17</i>
<i>getSiguientePedidoCocinaEnCola.....</i>	<i>18</i>
<i>getSiguientePedidoCocinaPreparandose.....</i>	<i>19</i>
<i>seleccionaBebida.....</i>	<i>20</i>
<i>seleccionaPlato.....</i>	<i>20</i>
8. Diseño de la base de datos.....	29
9. Diseño de las interfaces de usuario.....	30
<i>Interfaz de usuario de Cliente.....</i>	<i>31</i>
<i>Interfaz de usuario de Metre.....</i>	<i>33</i>
<i>Interfaz de usuario de Cocinero.....</i>	<i>34</i>
Apéndice O.1.....	35

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	02/05/10	Se ha generado el documento de diseño.

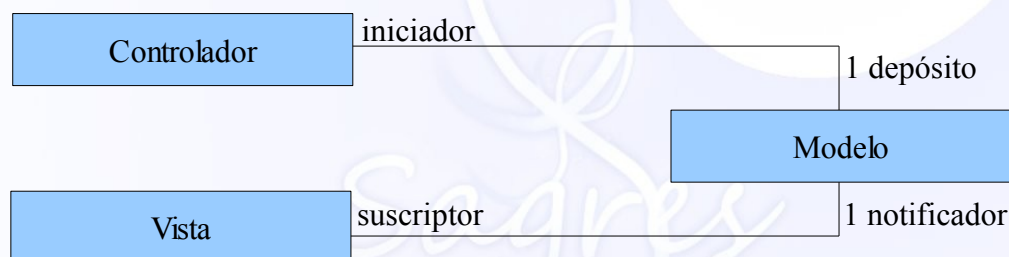
The logo for 'Sagres' features a stylized, light-colored graphic above the word 'Sagres' written in a cursive script. The background of the entire page has a faint, light blue circular design.

1. ESTILO ARQUITECTÓNICO

Para esta iteración, utilizaremos de nuevo una arquitectura basada en capas cerradas que separen la aplicación en tres niveles: interfaz de usuario, lógica de aplicación y servicios. Esto mejora la estructura del sistema y fomenta la flexibilidad en cuanto a sustitución de servicios o interfaces de usuario.

Por otra parte se usará también una arquitectura MVC (Modelo-Vista-Controlador) para desacoplar la lógica de la aplicación del interfaz de usuario. Esto facilita el diseño de distintas interfaces de usuarios adaptados a los distintos terminales, una para los clientes, otra para la cocina y otra para la barra. Con esto obtenemos un alto nivel de cohesión y un bajo nivel de acoplamiento, separando datos de todo lo demás, ya que cambios en nuestro diagrama de clases no repercutirán en cambios en la interfaz de usuario y viceversa.

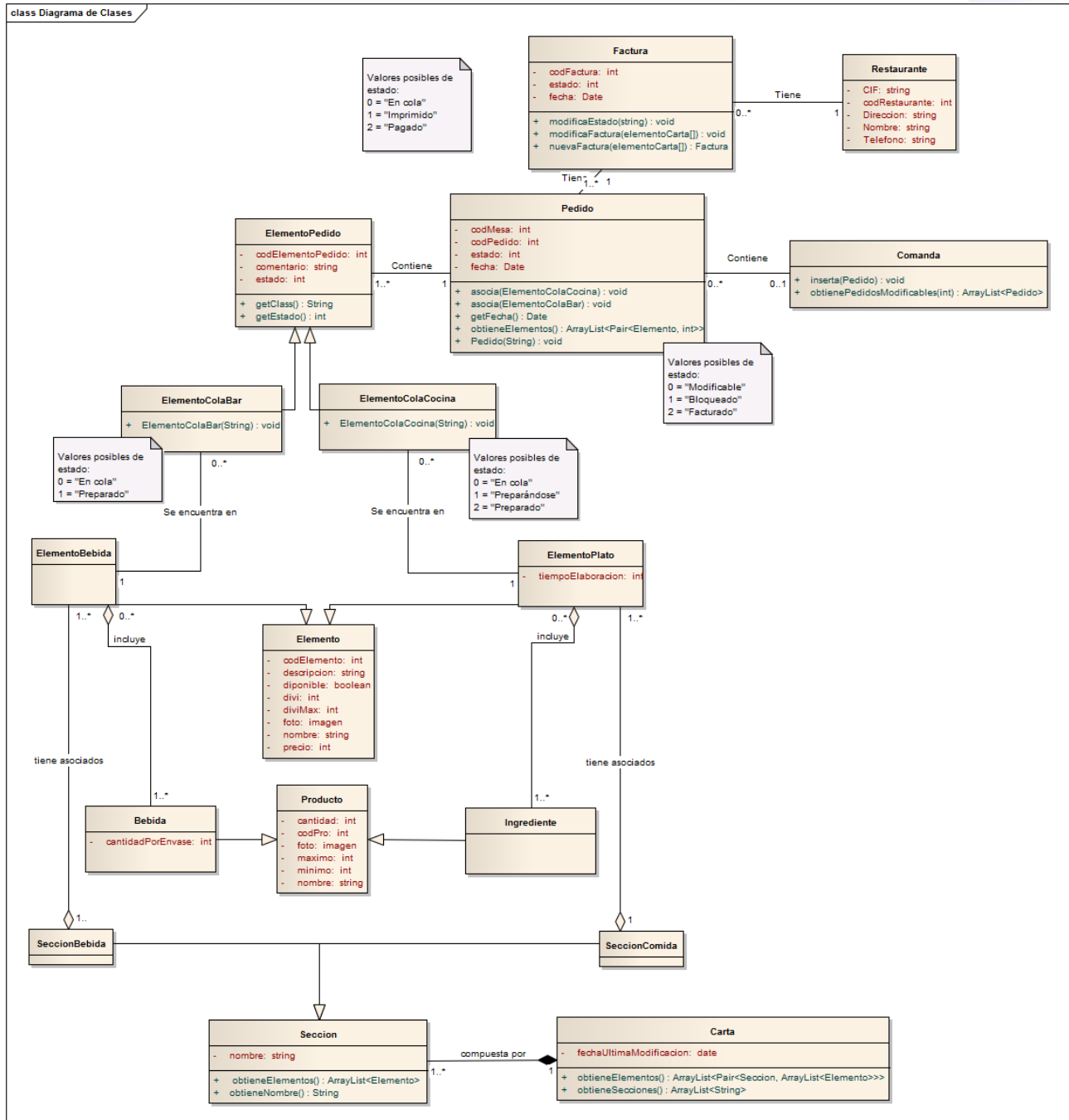
El estilo cliente-servidor podríamos utilizarlo para establecer la comunicación entre los subsistemas de cada capa que pueden estar situados en distintos nodos del sistema. Esto permite distribuir los distintos subsistemas del sistema reduciendo el acoplamiento sólo a través del protocolo de comunicaciones. Fomenta la interoperabilidad, y facilita la escalabilidad del sistema (por ejemplo, aumentar el número de clientes sin penalizar la gestión de datos).



En los apartados siguientes mostraremos, entre otras cuestiones, cómo adaptaremos esta arquitectura conforme a las necesidades de nuestro sistema.

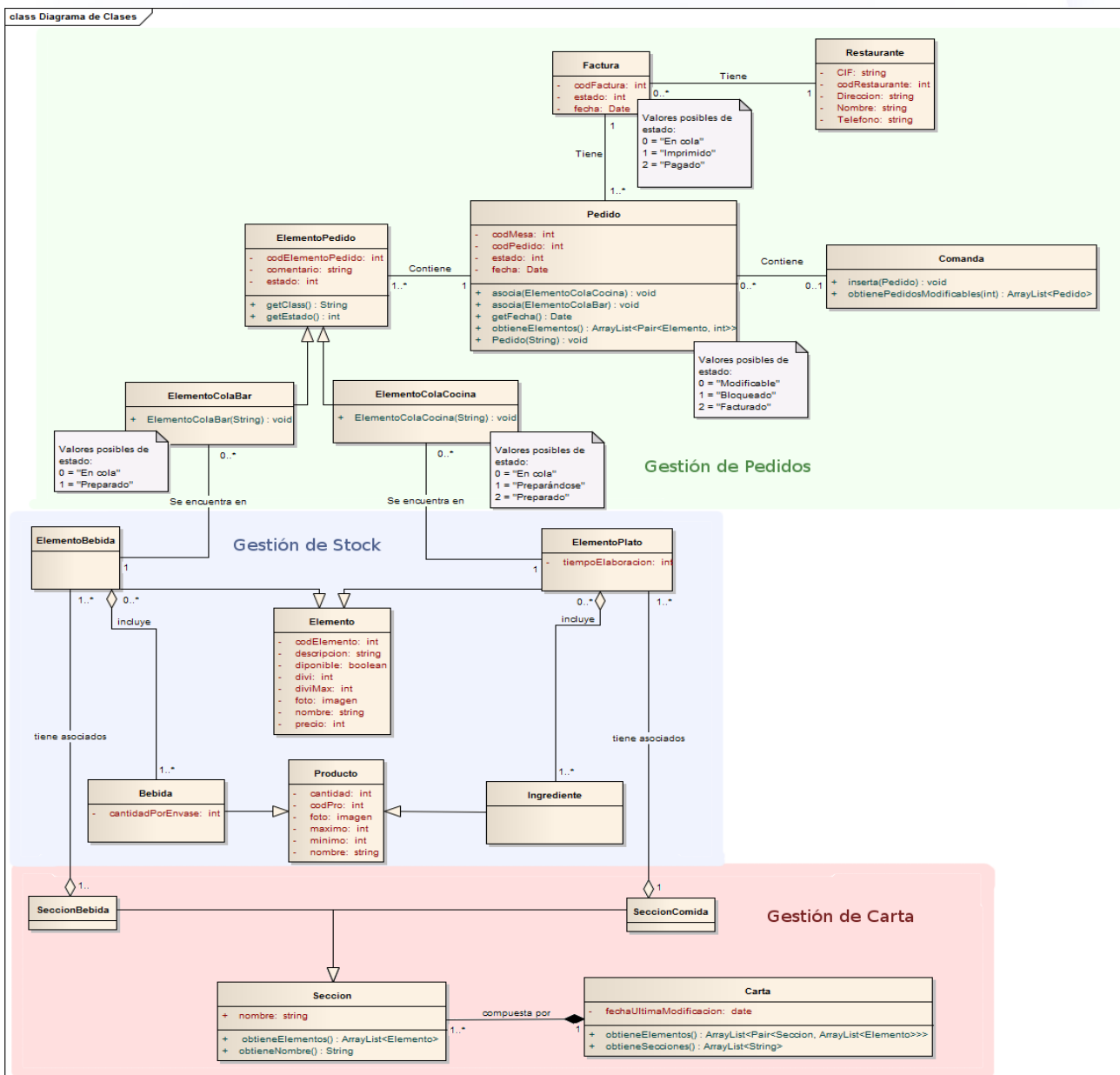
2. DIAGRAMA DE CLASES DEL DISEÑO

En este apartado mostramos el diagrama de clases de diseño, el cuál obtenemos a partir del análogo en la etapa de análisis, añadiendo a cada clase los métodos obtenidos en los diagramas de colaboración de dicho análisis.



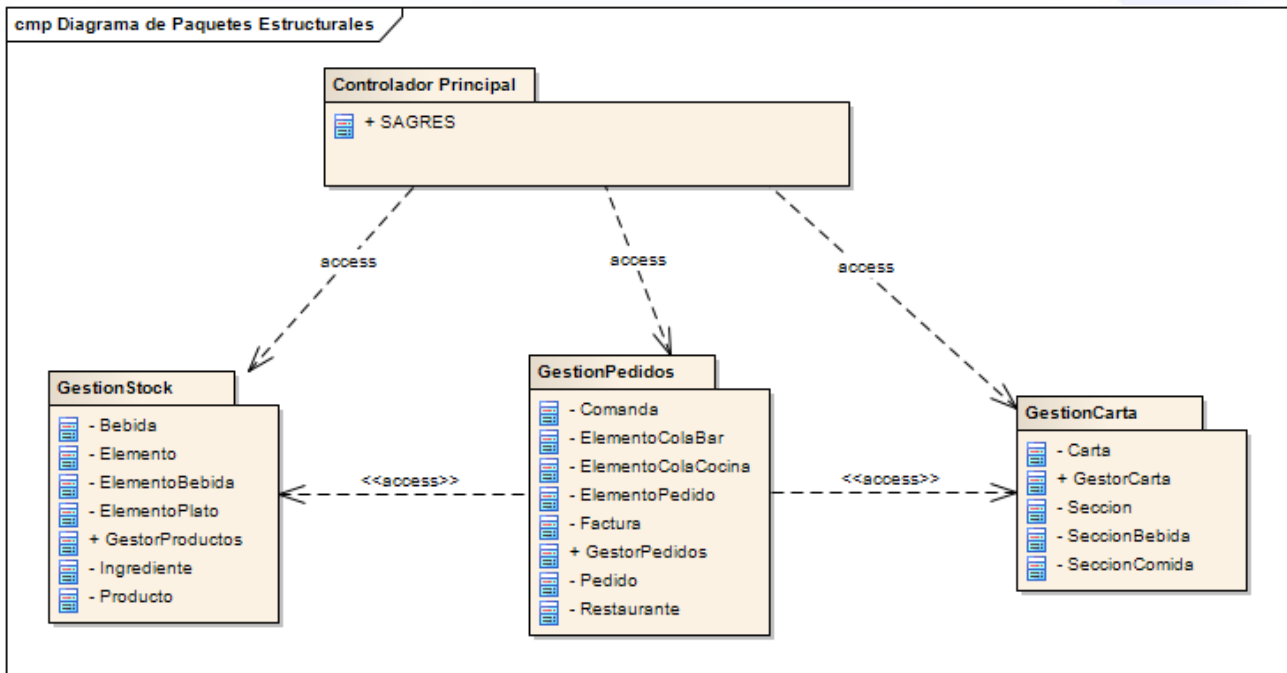
3. DIAGRAMA DE PAQUETES ESTRUCTURALES

Considerando el diagrama de clases del diseño como punto de partida, el primer paso que tomaremos es realizar la división del mismo en subsistemas estructurales. En el diagrama de paquetes funcionales obtenido en el modelado de requisitos de esta iteración concluimos que el sistema a desarrollar en esta iteración estaría compuesto por un sólo subsistema funcional, el subsistema de gestión de pedidos. Teniendo en cuenta que en el diagrama de clases tuvimos la necesidad de añadir clases procedentes de subsistemas desarrollados en la anterior iteración, la distribución en subsistemas estructurales será la siguiente:



Se observa como además del subsistema de gestión de pedidos, que es el que lleva a cabo las funcionalidades requeridas en esta iteración, se tienen en cuenta también los subsistemas de stock y de carta, ya que necesitaremos reutilizar y/o desarrollar algunas de sus funcionalidades.

A partir de esta división del diagrama de clases obtenemos el siguiente diagrama de paquetes estructurales, donde se introducen sólo aquellas conexiones entre paquetes que son necesarias para las funcionalidades a desarrollar en esta iteración:

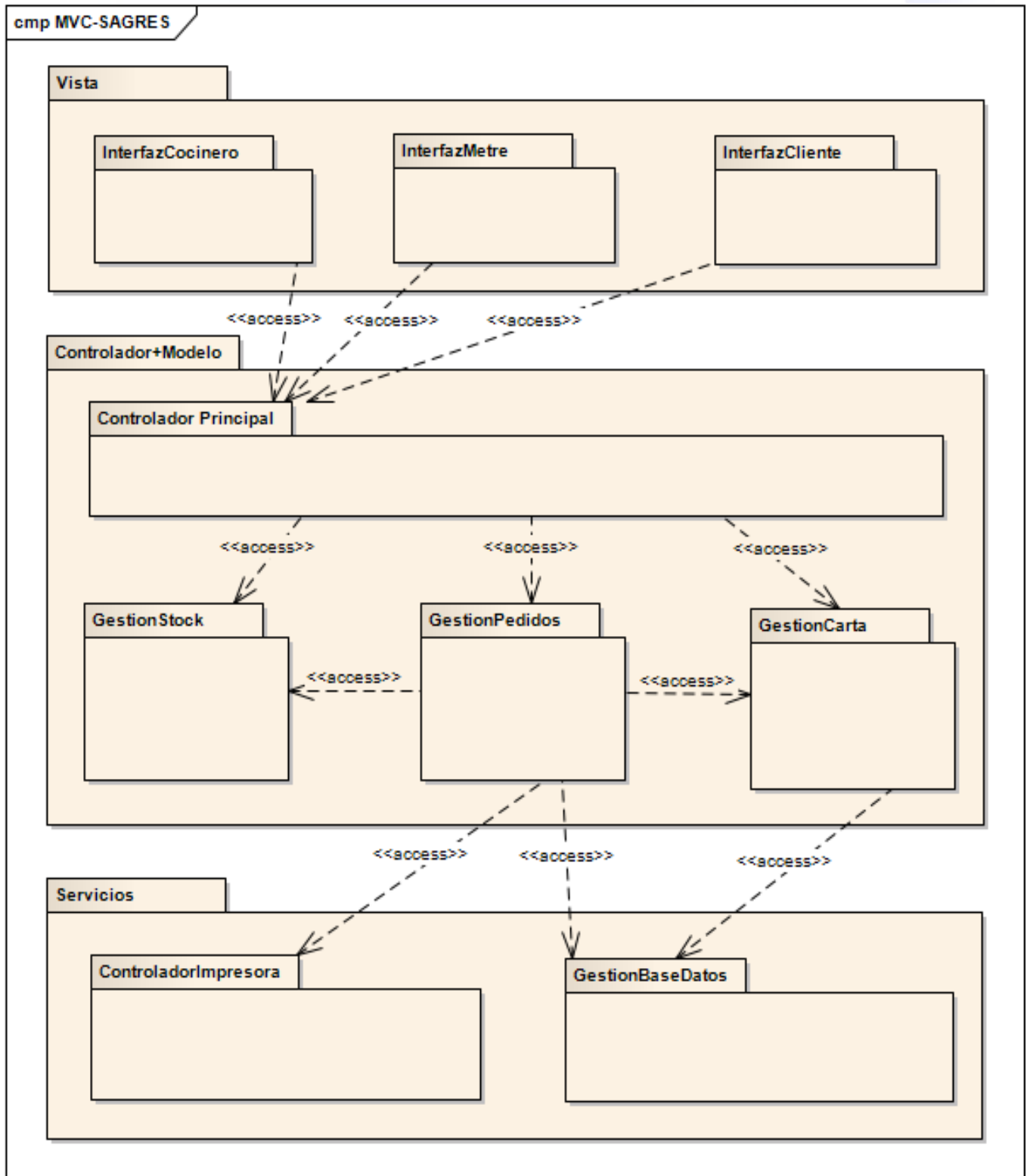


Observar que se ha introducido por vez primera vez el paquete con el controlador principal, con una clase tipo control SAGRES. En la fase de análisis se consideró en los diagramas de colaboración un controlador genérico SAGRES, para poder definir las operaciones del sistema un nivel conceptual de abstracción. En correspondencia con dicho controlador genérico, el controlador principal mostrado en este momento será en el sistema un componente más a desarrollar, ya que constituirá la fachada la lógica de nuestra aplicación.

Observar además que en los paquetes estructurales se han incluido clases gestoras. Estas clases de tipo control se encargarán de llevar a cabo los servicios que ofrecen sus respectivos paquetes. Se puede ver que el acceso de a las clases en los paquetes se restringe, de manera que para poder acceder a las funcionalidades que ofrecen hay que comunicarse con las clases de control.

4. ESTILO ARQUITECTÓNICO DETALLADO

Una vez definido el diagrama de paquetes estructurales, podemos definir de manera detallada cuál será la arquitectura del sistema. A continuación se muestra como se distribuirán los distintos subsistemas en las capas del MVC que hemos decidido adoptar:



Se observa que en la **vista** tenemos tres paquetes, cada uno de los cuales correspondiente a un grupo de interfaces con el que tendrá interacción un tipo de actor. Tendremos uno para el jefe de cocina, otro para el metre y otro para los clientes.

En el **controlador+modelo** hemos introducido todo aquello que conforma la lógica de la aplicación, en definitiva, todos los subsistemas que aparecen en el diagrama de paquetes estructurales.

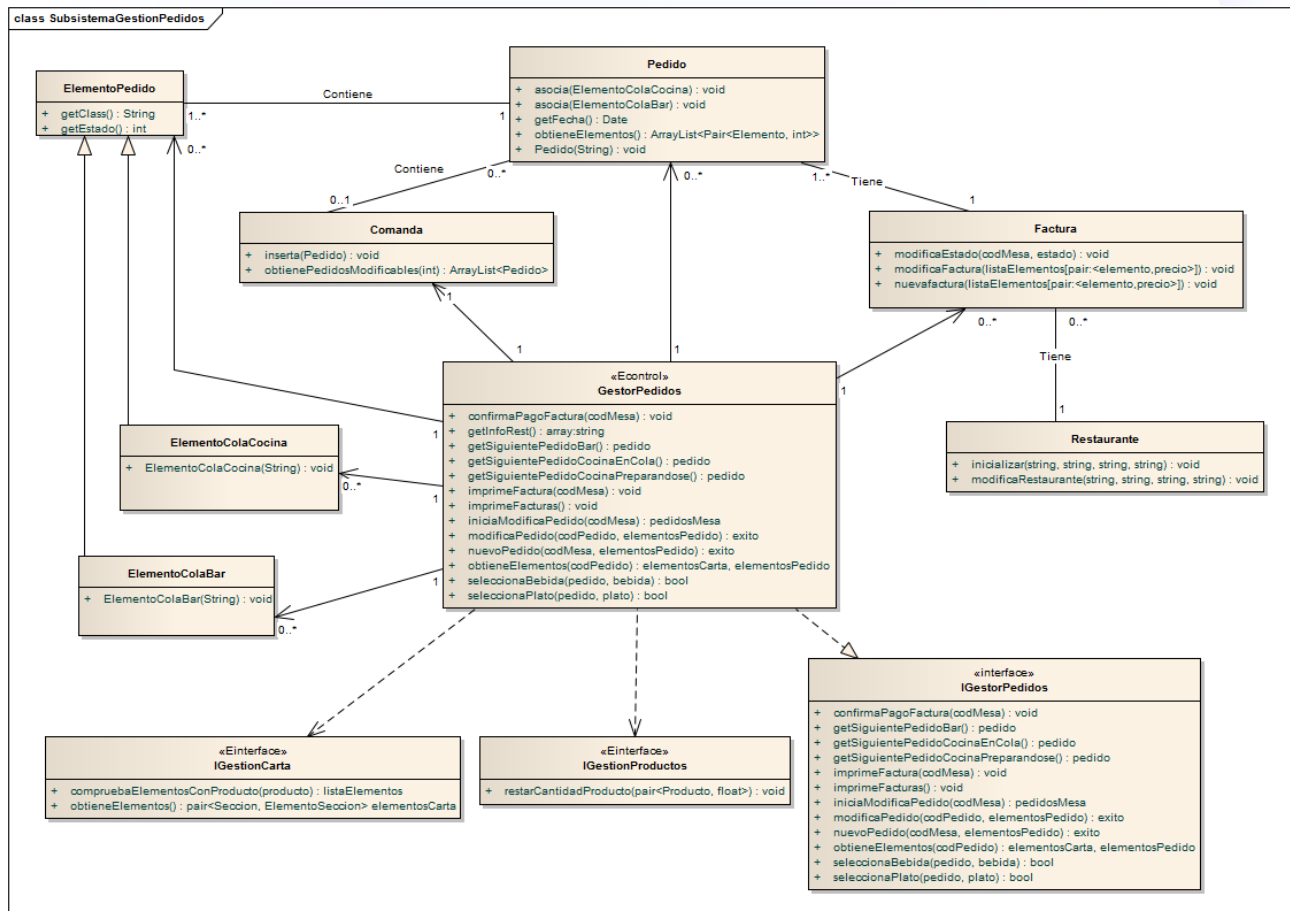
En los **servicios** incluimos todas aquellos subsistemas que escapan de la lógica de aplicación, pero que sin embargo resultan imprescindibles para llevar a cabo las funcionalidades a desarrollar. En nuestro caso necesitaremos un subsistema que permita hacer persistentes todos los cambios en los datos del sistema, y otro que nos permita llevar a cabo las impresiones de las facturas del cliente.

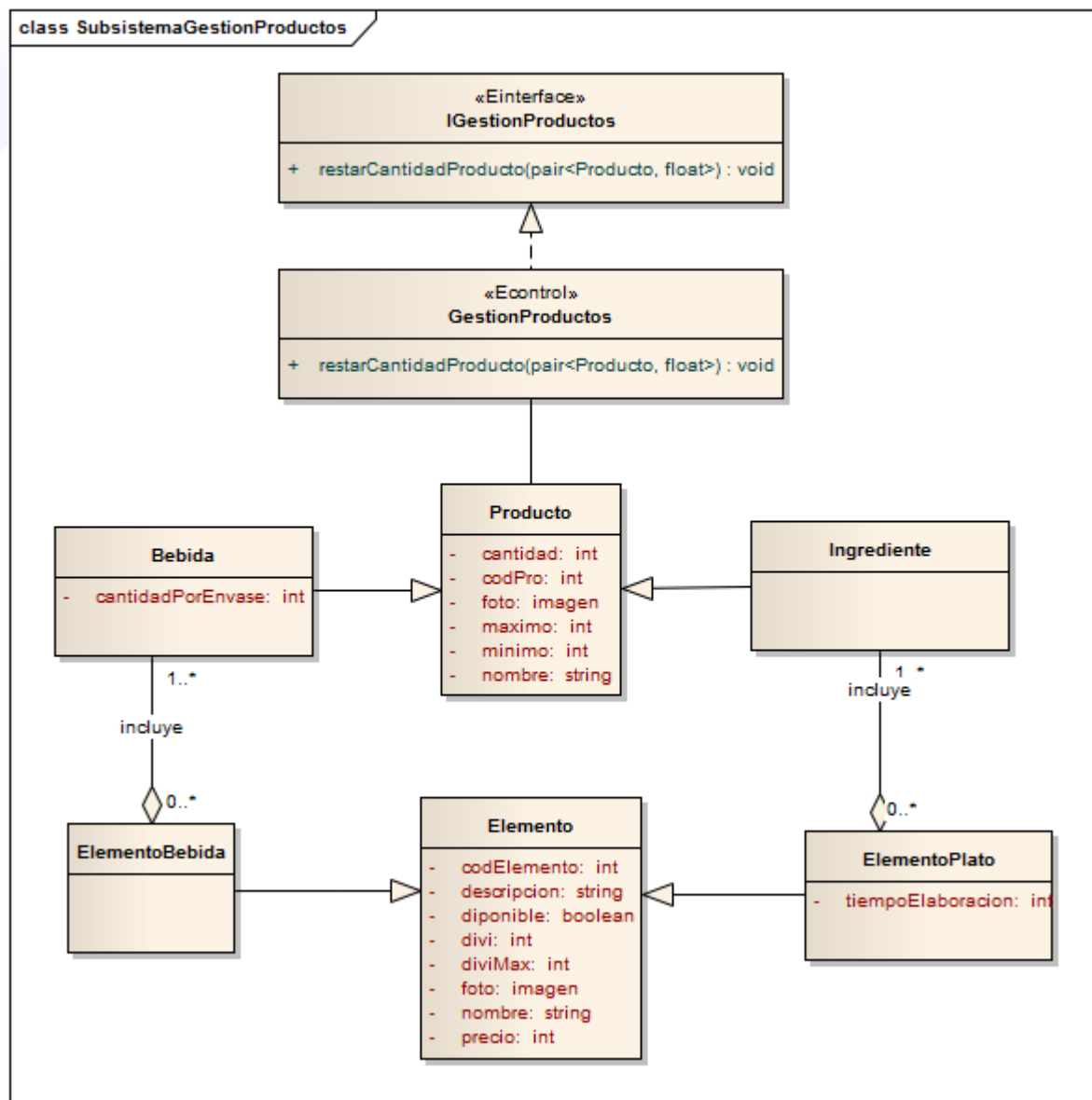
Una vez que tenemos la arquitectura detallada, debemos describir cada uno de los subsistemas subyacentes en cada una de las capas. Detallaremos, en este orden, los subsistemas de la capa controlador+modelo, los subsistemas de la capa de servicios y cada una de las interfaces de usuario pertenecientes a la vista del sistema.

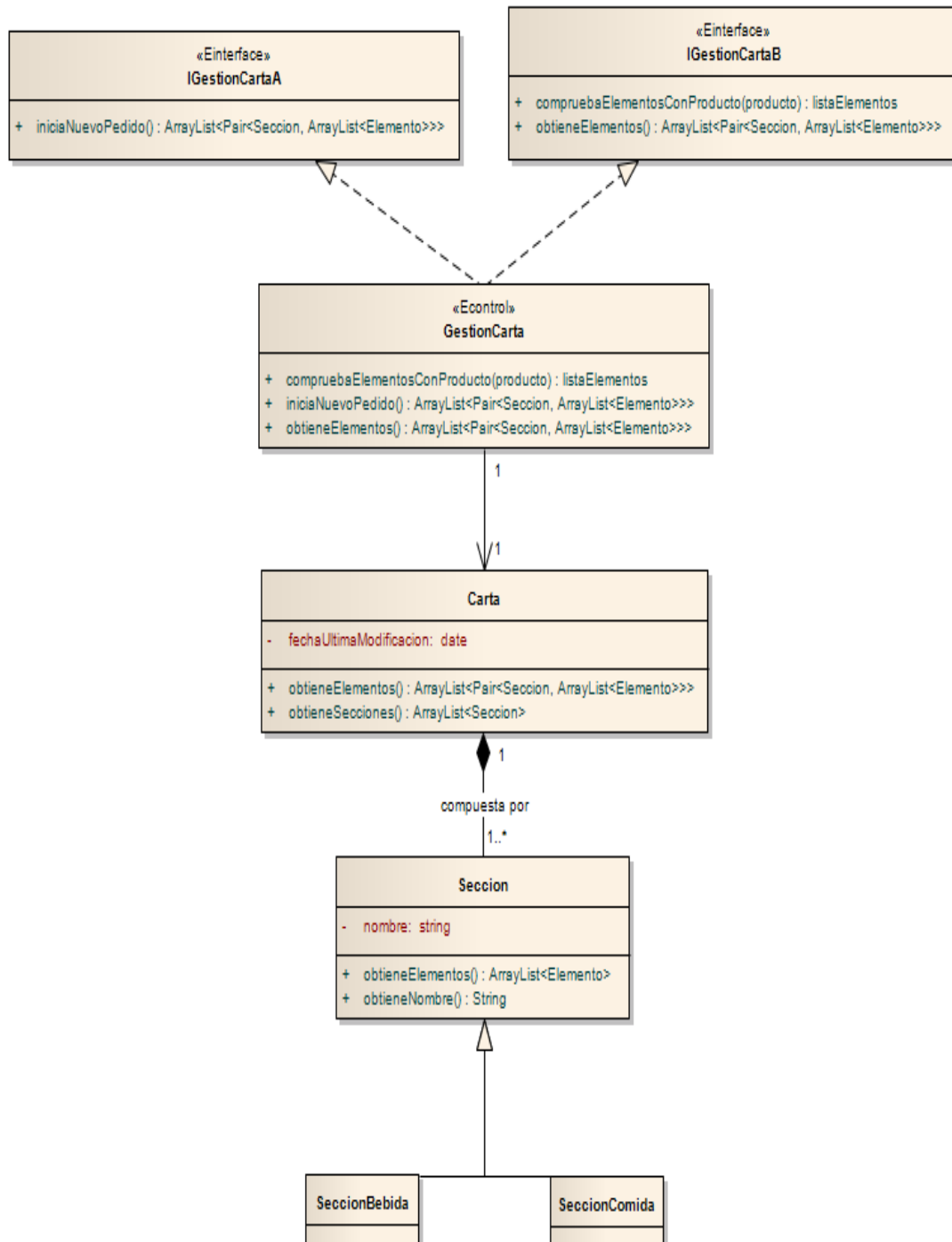


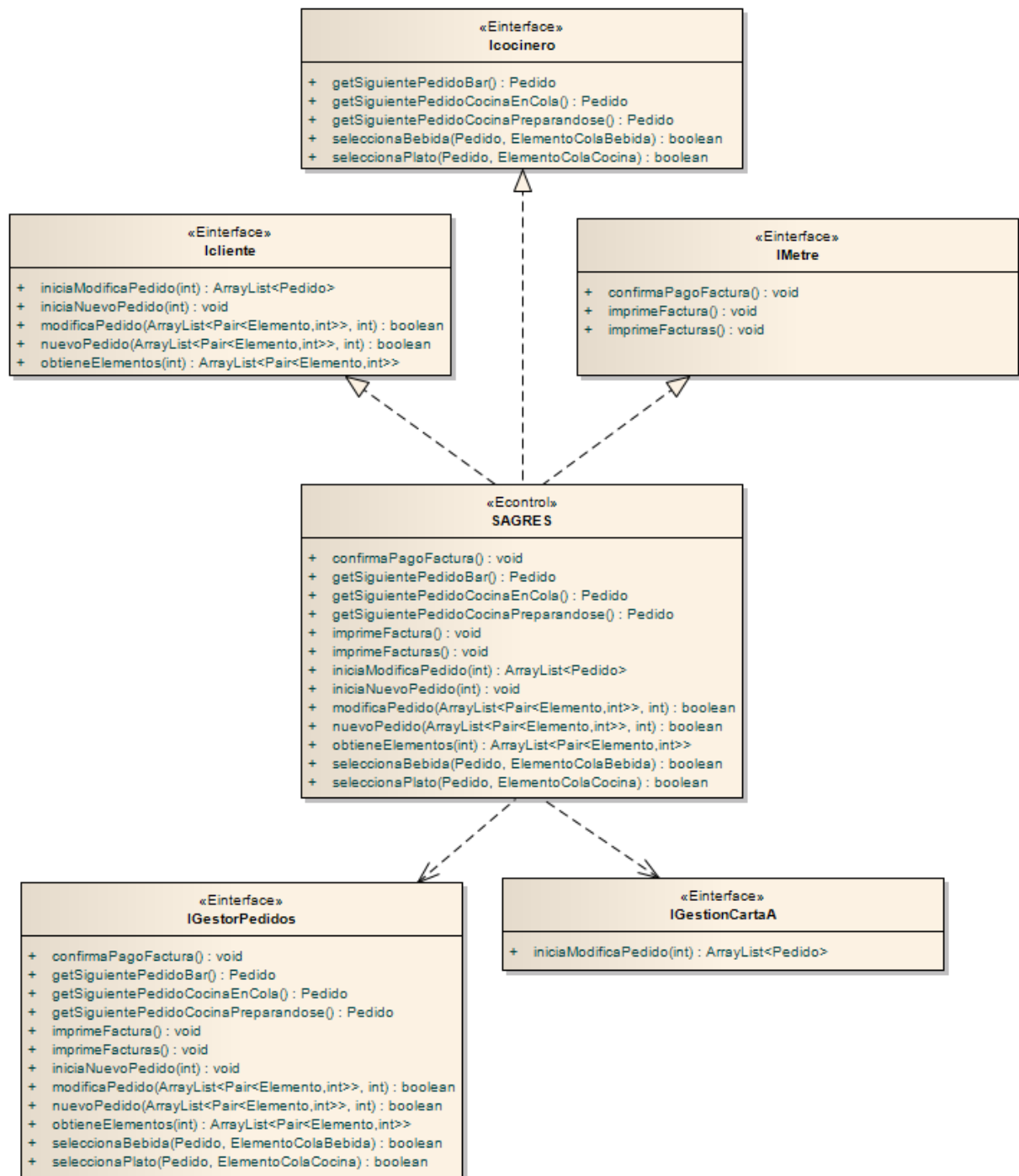
5. IDENTIFICACIÓN DE LAS INTERFACES

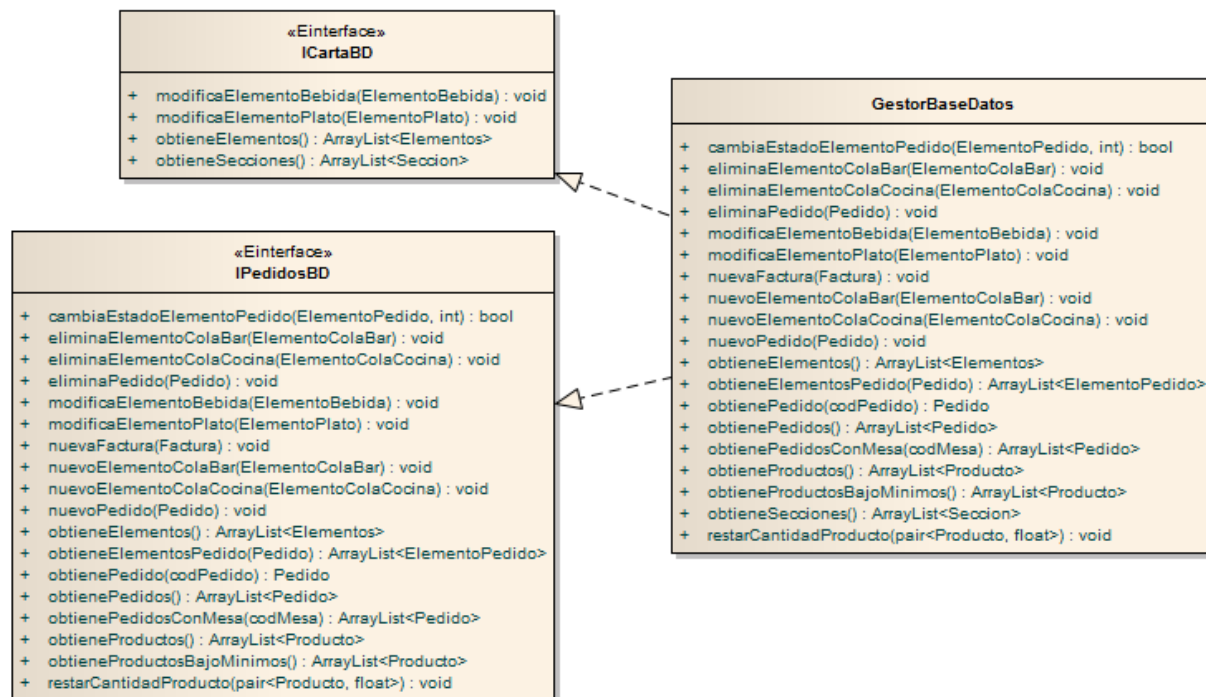
A continuación se detallarán las interfaces de los subsistemas que conforman la capa de controlador+modelo. Cada subsistema podrá tener interfaces “ofrecidas” y “requeridas”, cada una de las cuales corresponderán a un conjunto de servicios que ofrecen a otros subsistemas, en el primer caso, o bien a un conjunto de servicios que necesitan de otro subsistema.



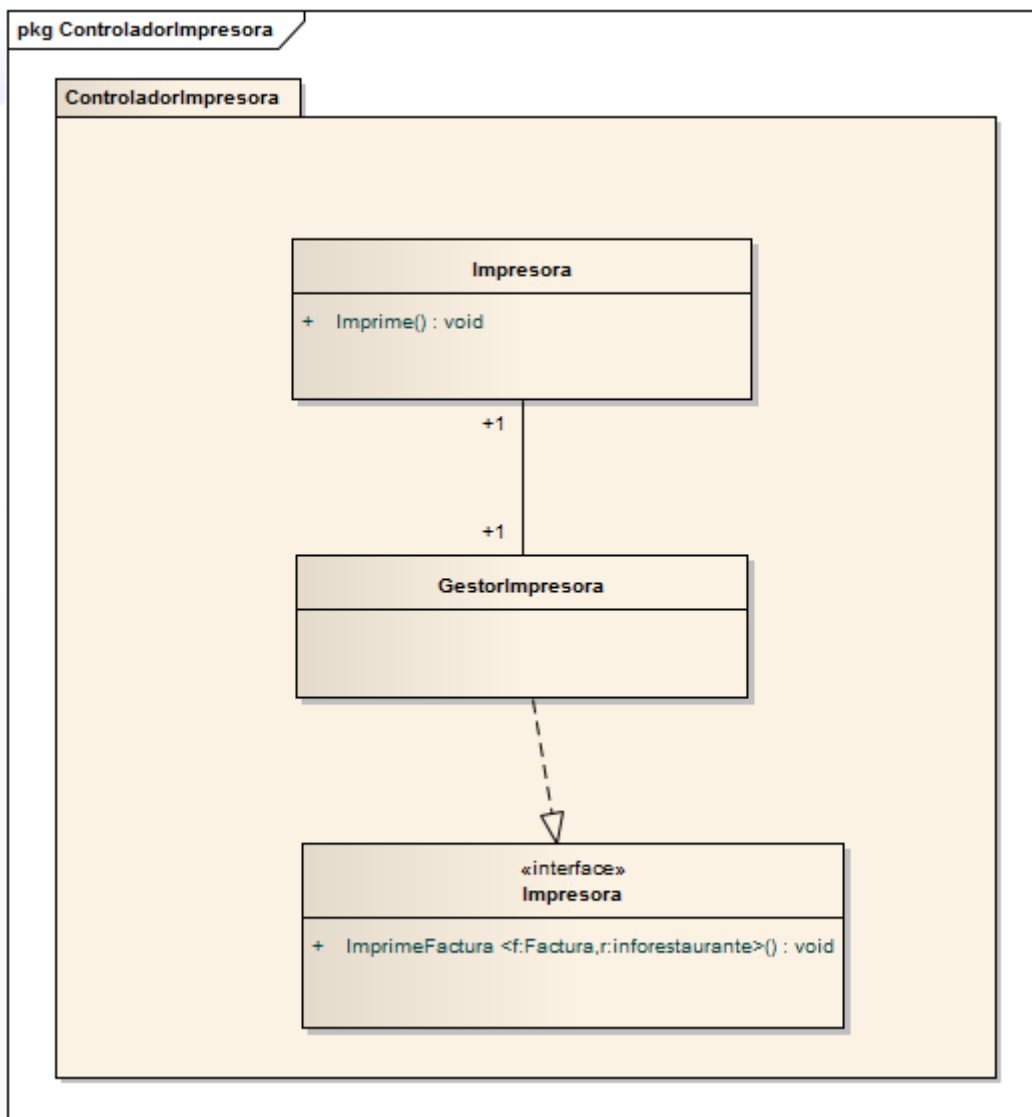






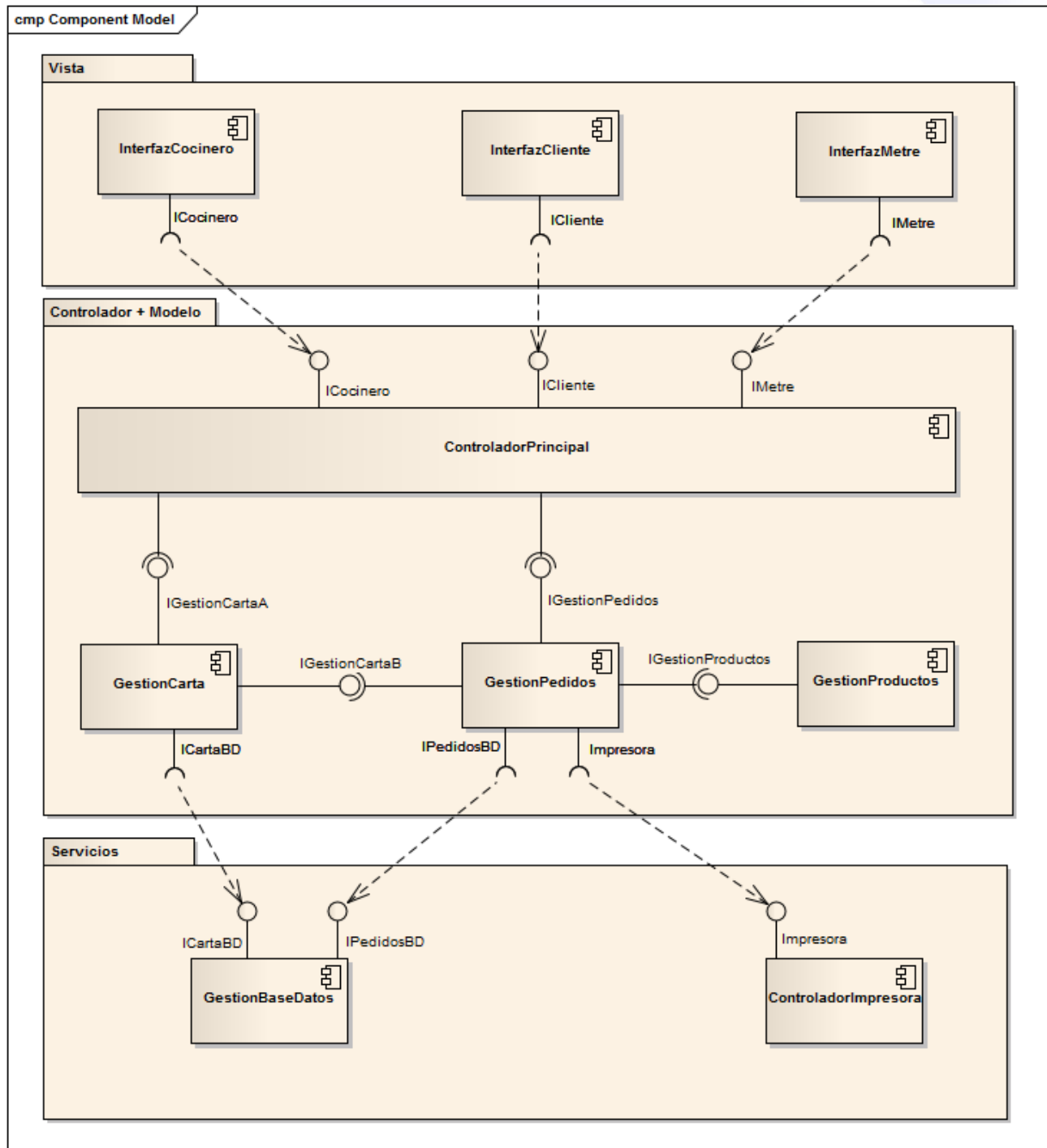


Sagres



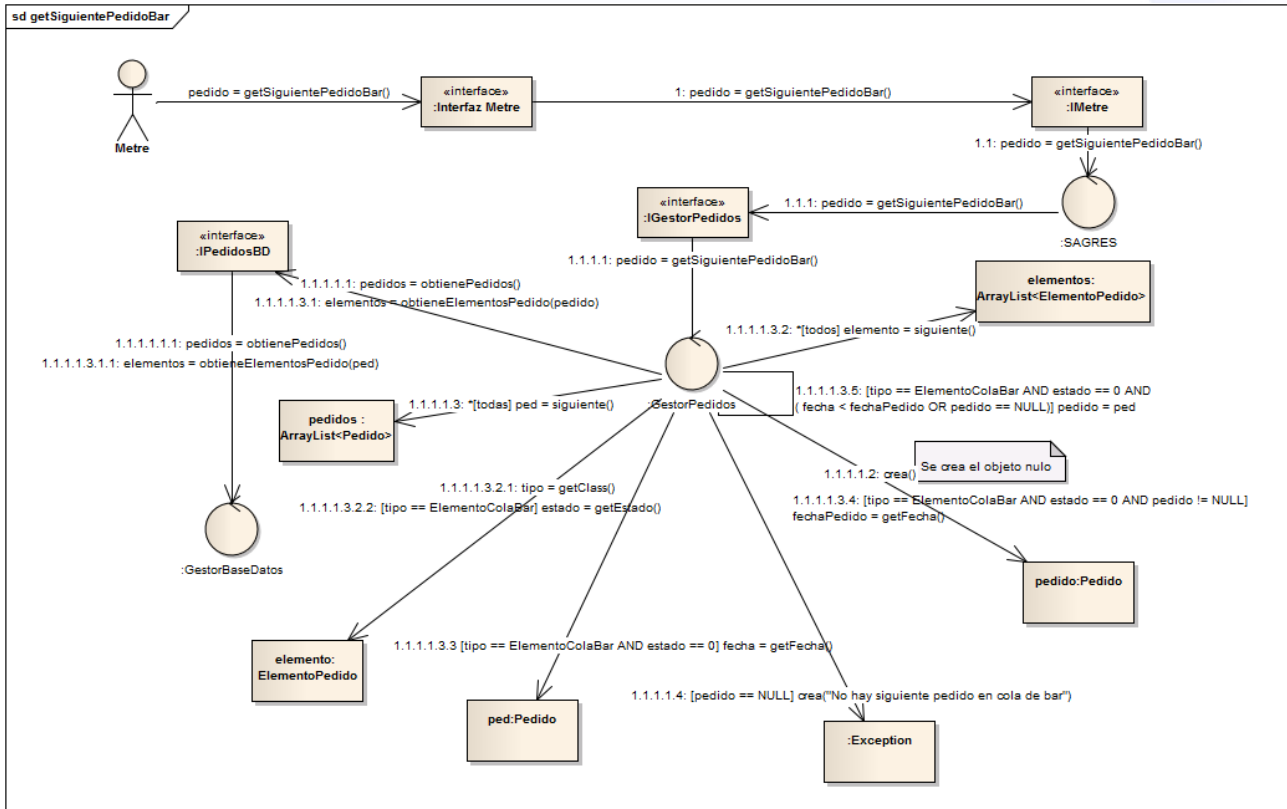
6. DIAGRAMA DE COMPONENTES

Una vez definidas las interfaces de los subsistemas, podremos definir a continuación el diagrama de componentes del sistema:



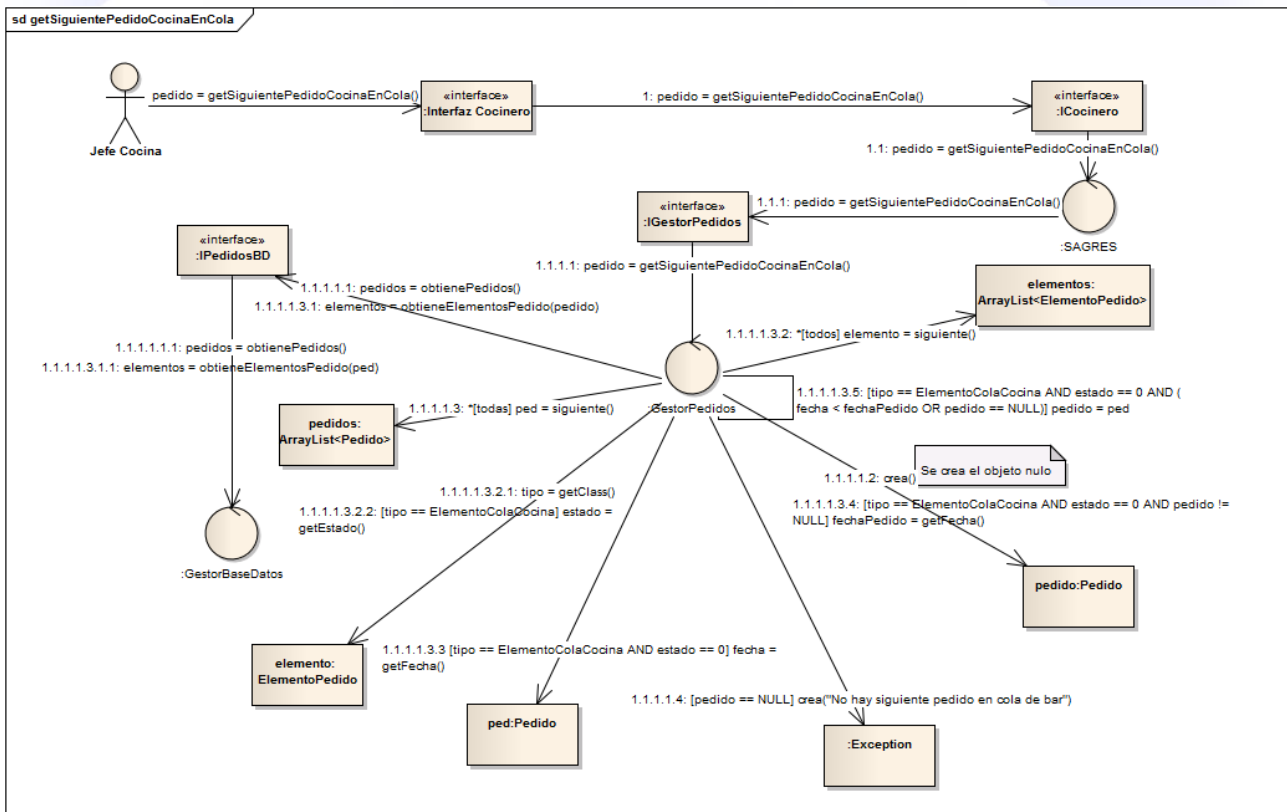
7. DIAGRAMAS DE COLABORACIÓN

getSiguientePedidoBar

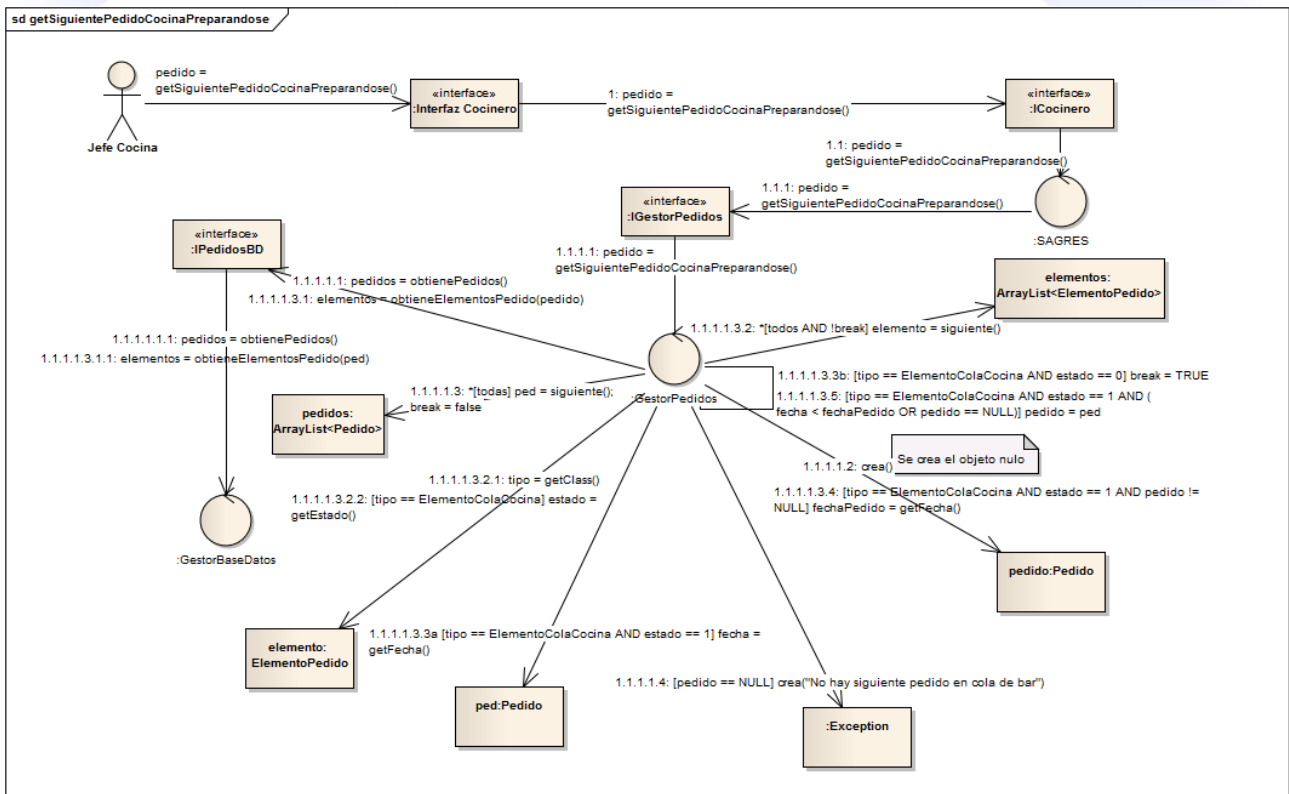


Realizado por Sergio Rodríguez Lumley

getSiguientePedidoCocinaEnCola

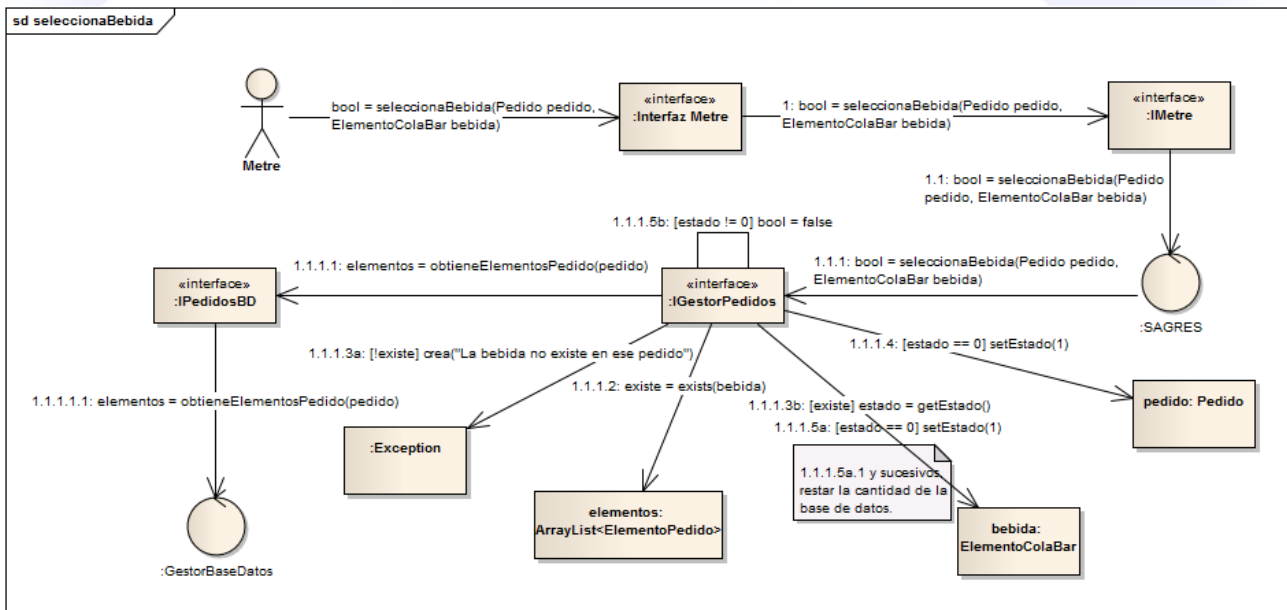


getSiguientePedidoCocinaPreparandose

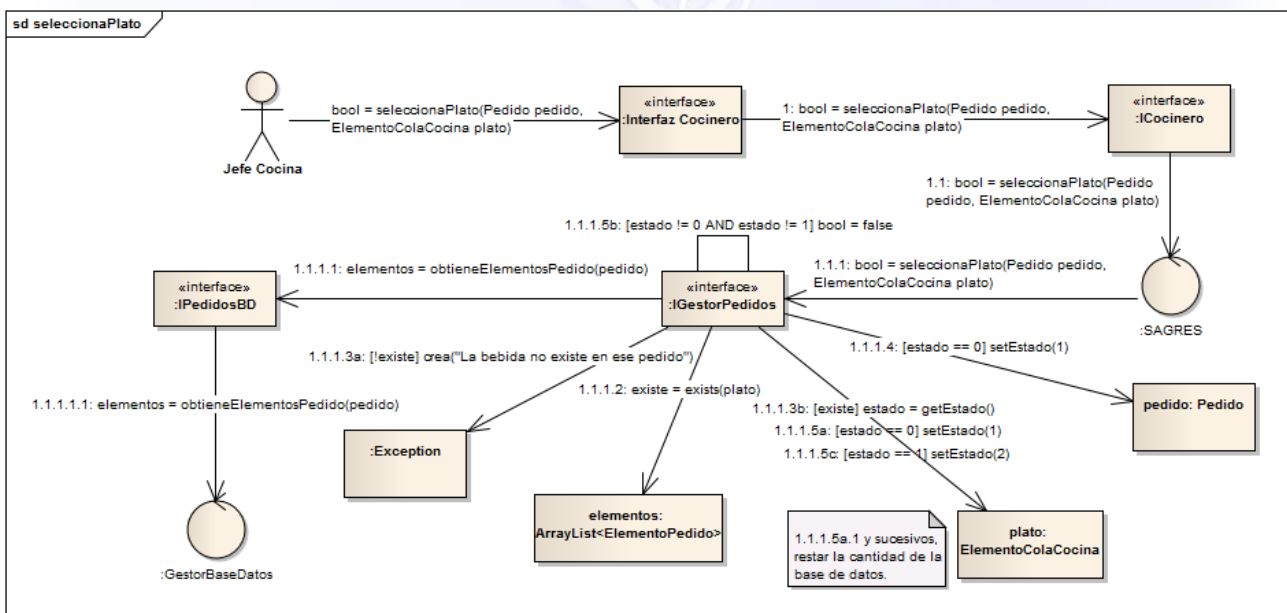


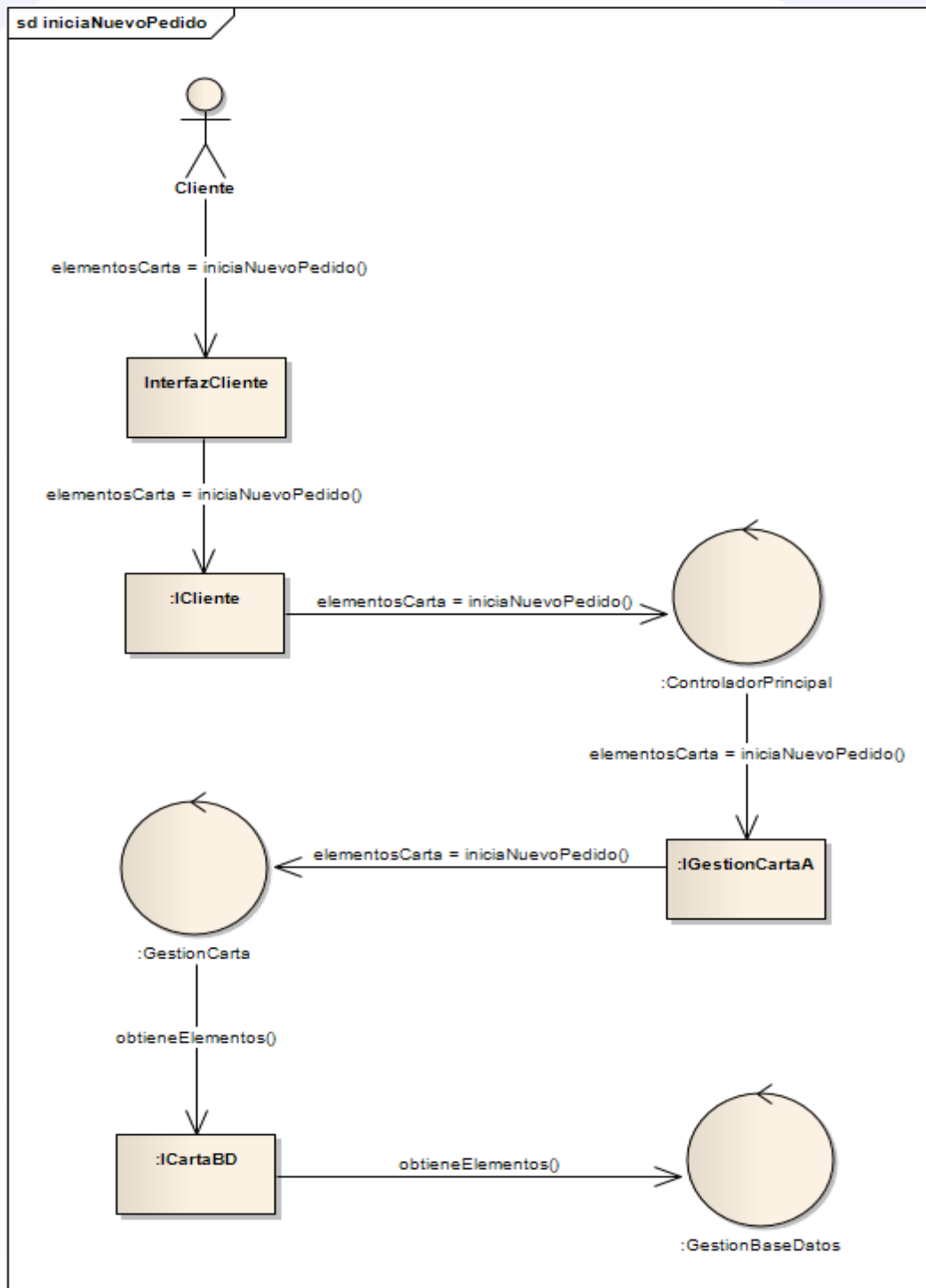
Realizado por Sergio Rodríguez Lumley

seleccionaBebida

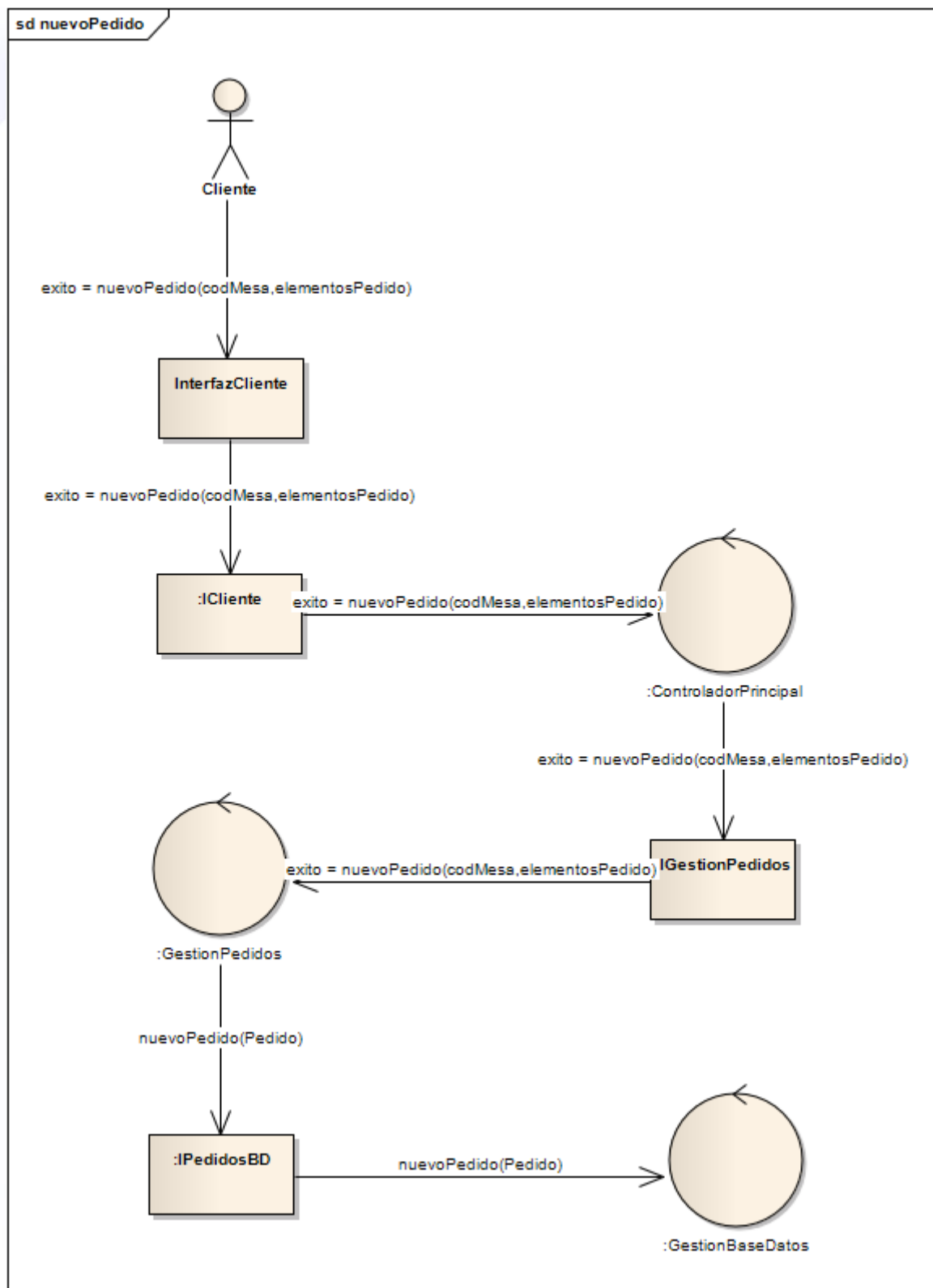


seleccionaPlato

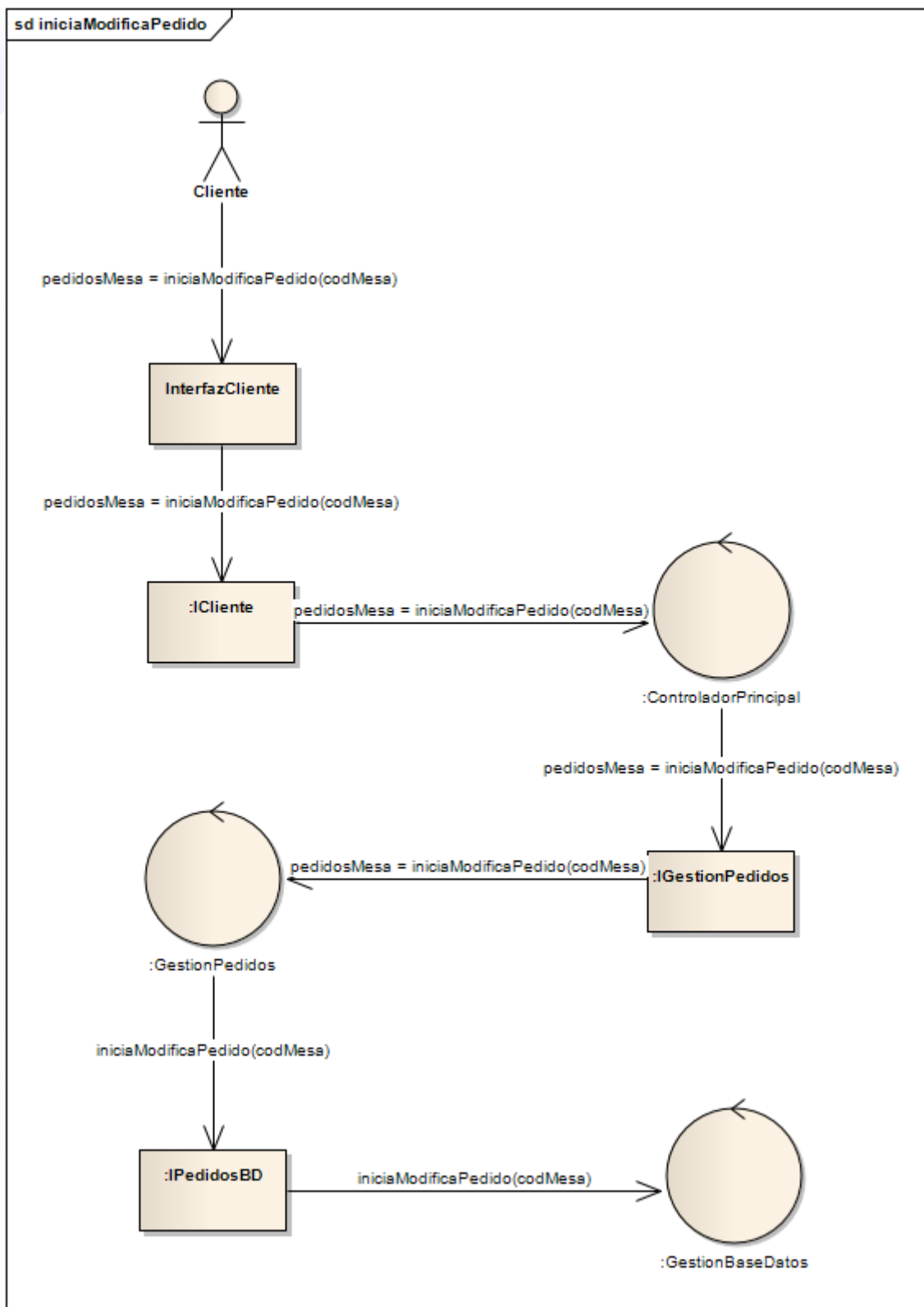




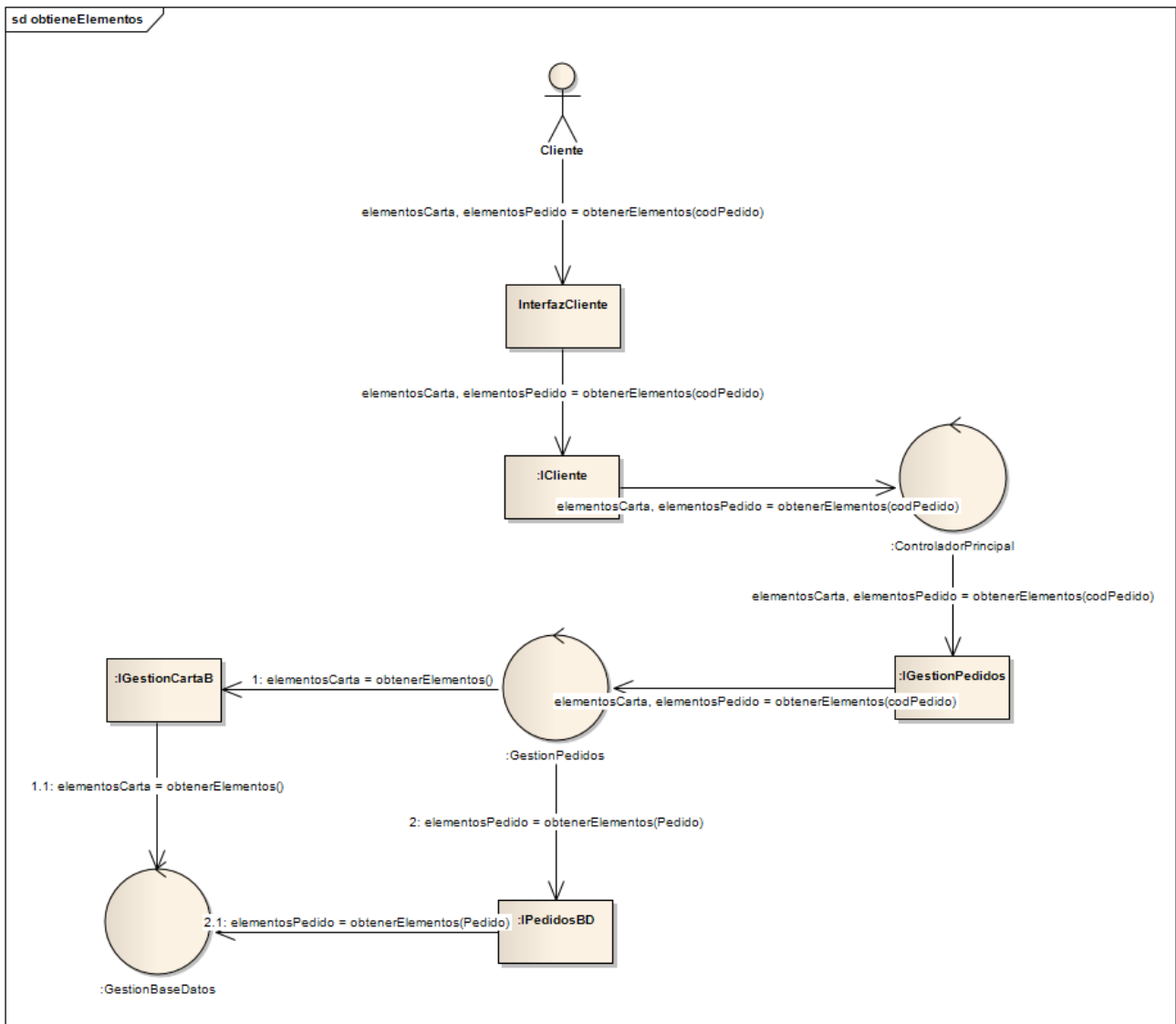
Realizado por Adrián Víctor Pérez Lopera.



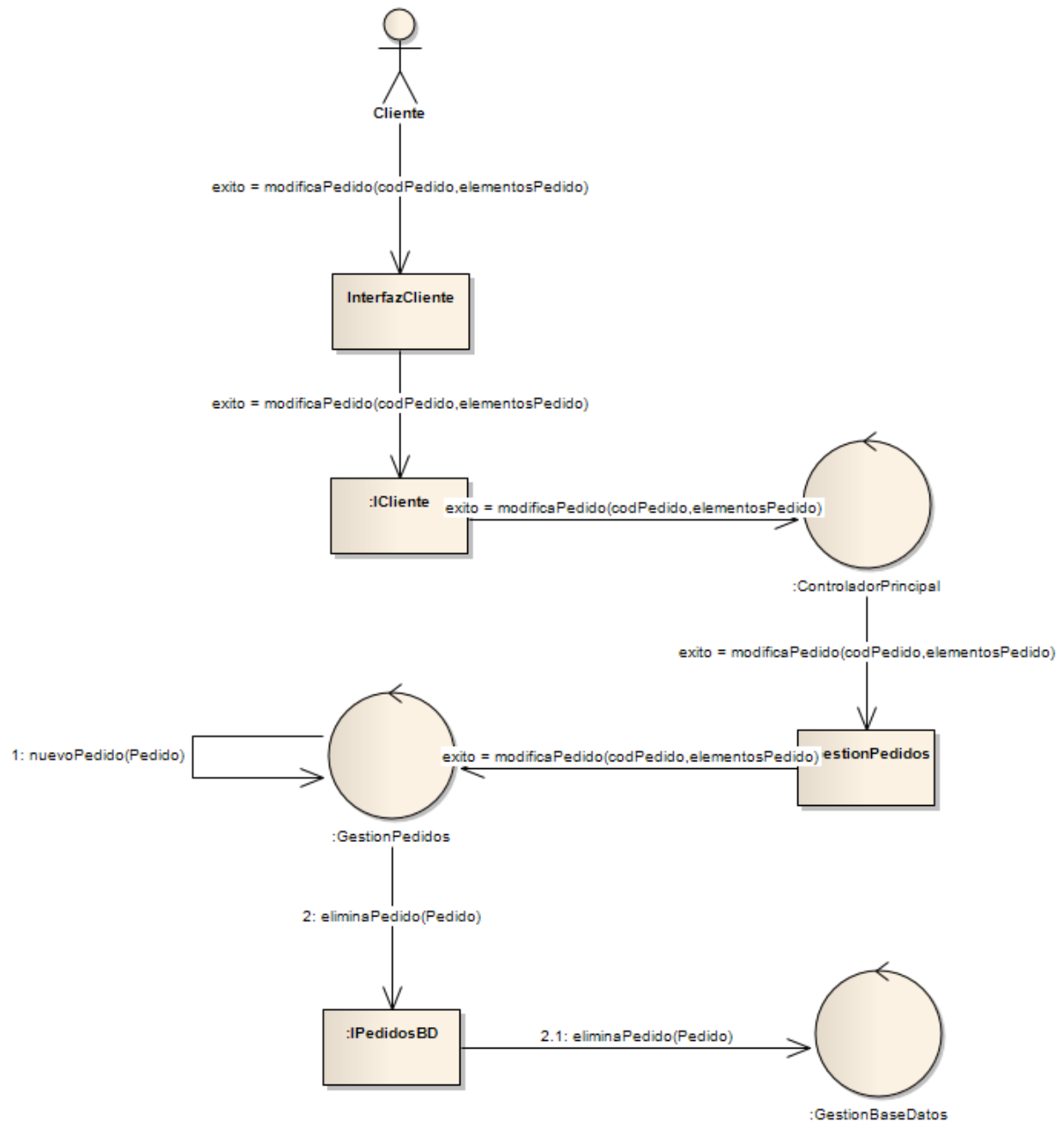
Realizado por Adrián Víctor Pérez Lopera.



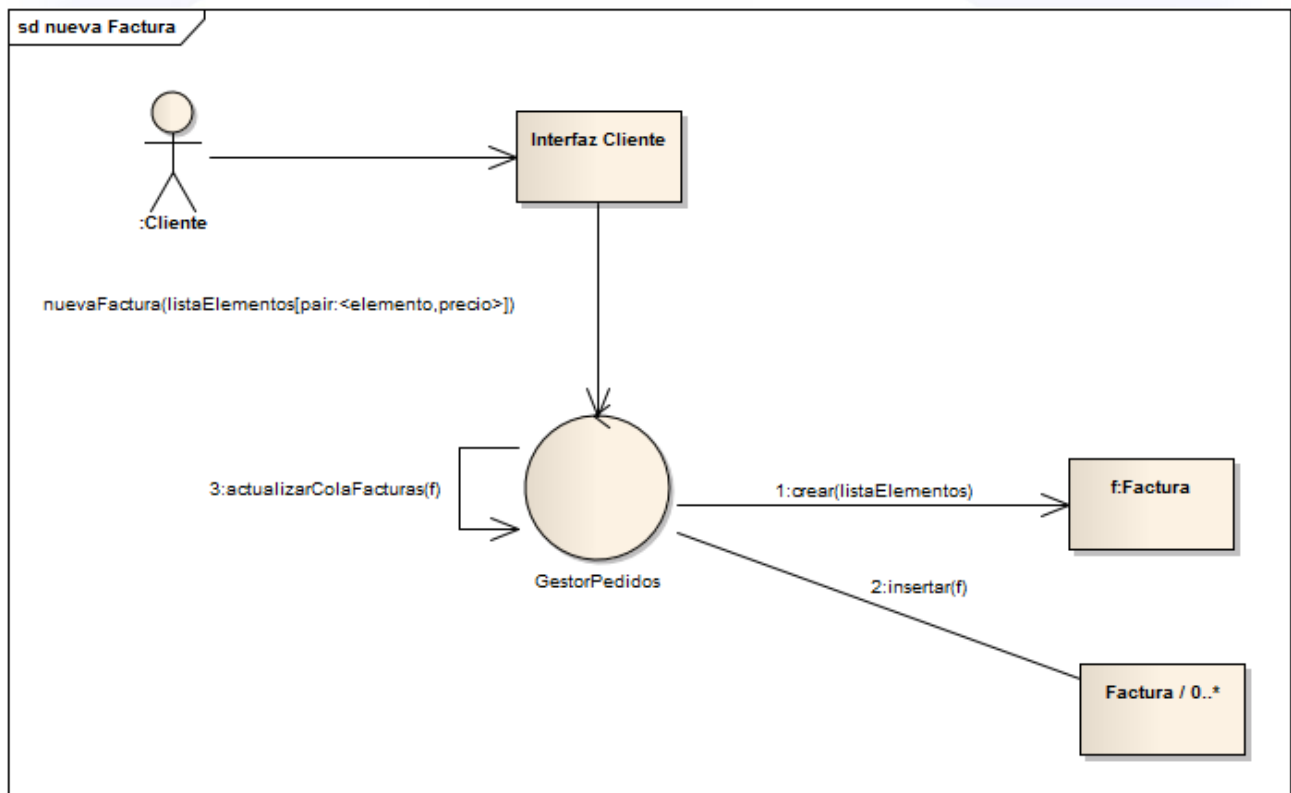
Realizado por Adrián Víctor Pérez Lopera.



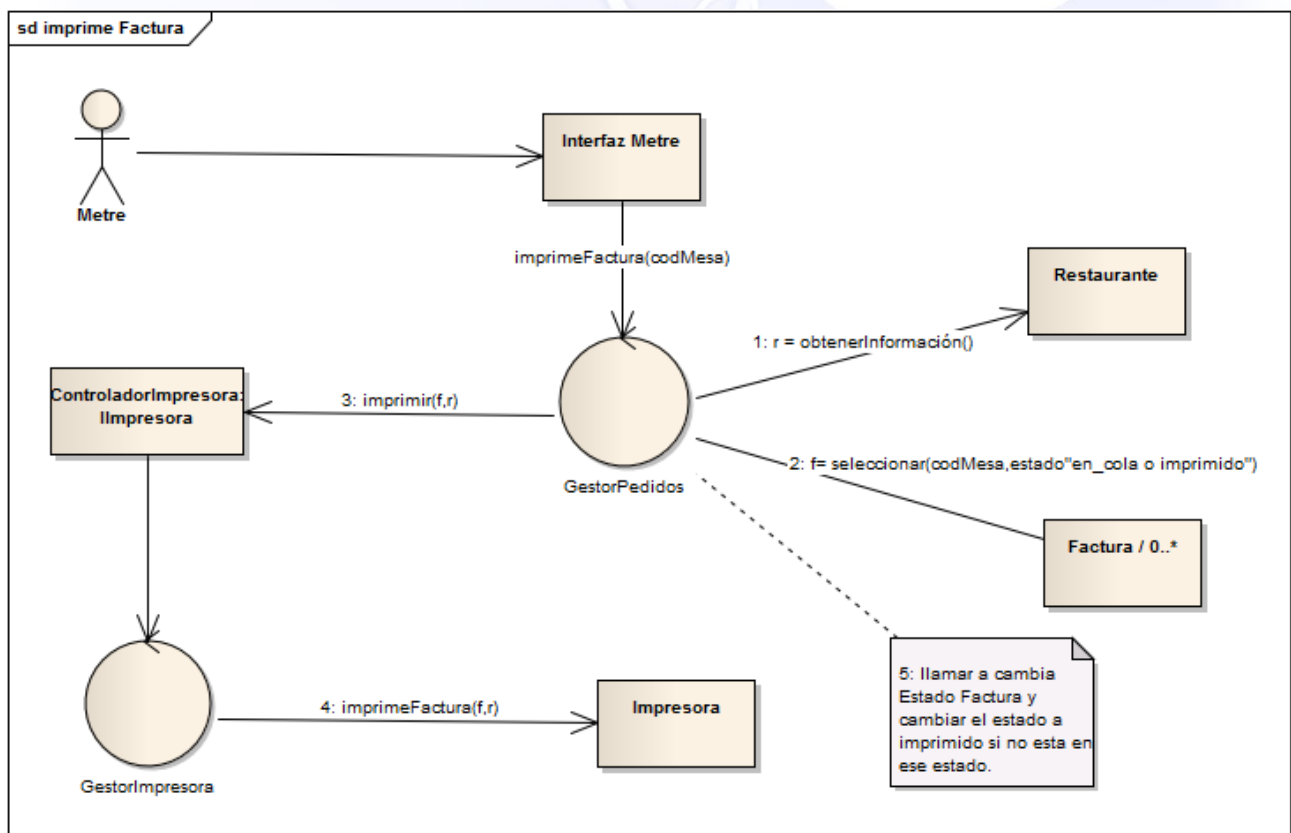
Realizado por Adrián Víctor Pérez Lopera.



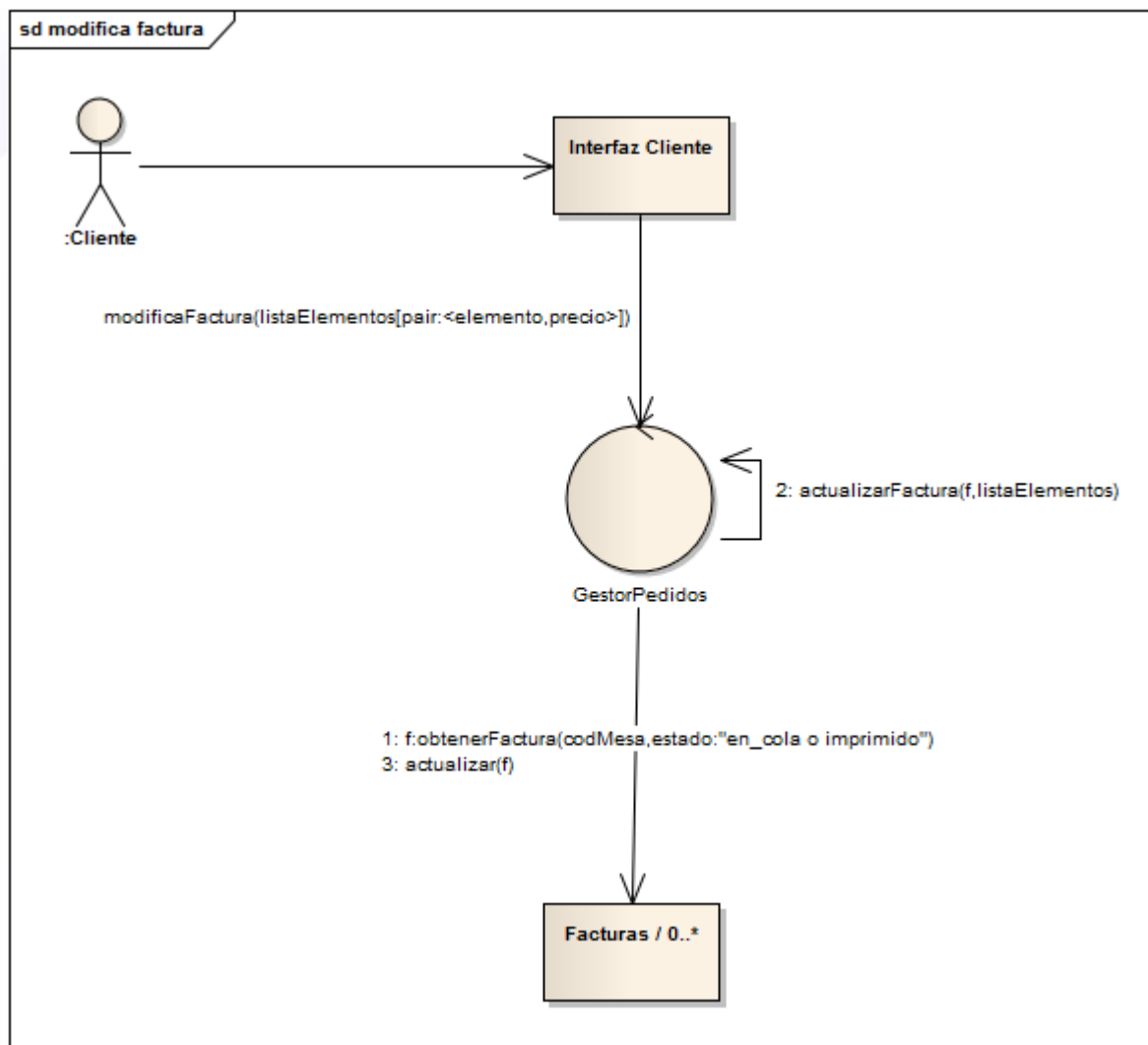
Realizado por Adrián Víctor Pérez Lopera.



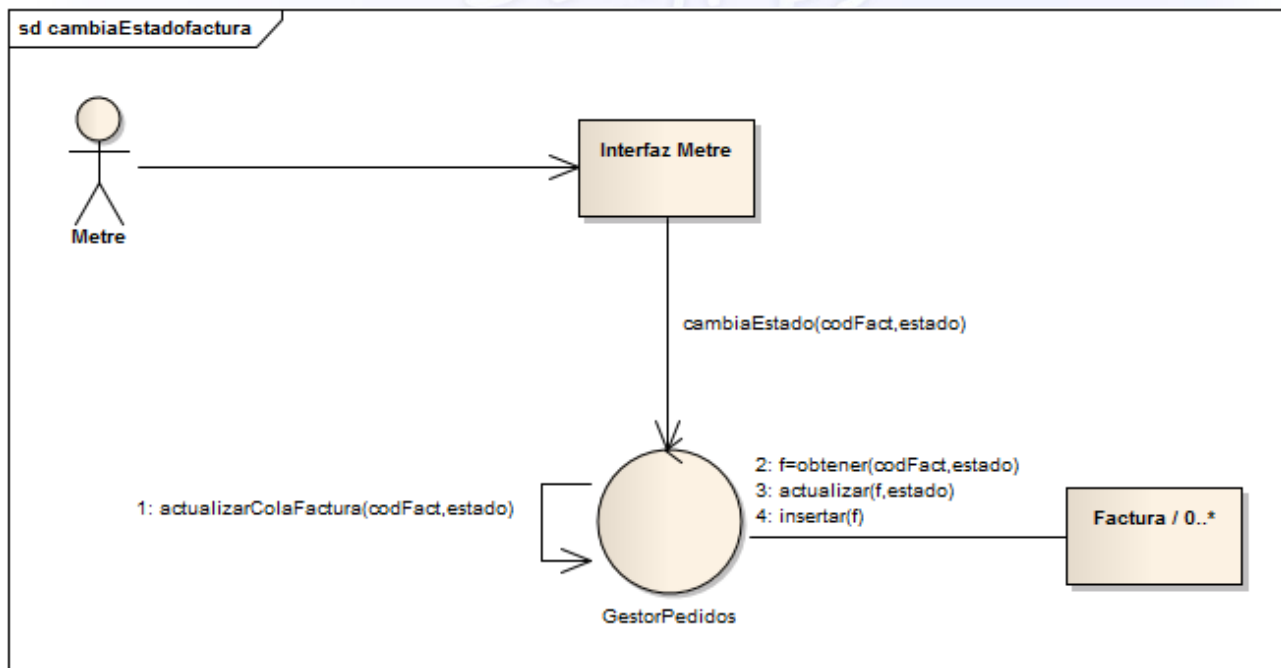
Realizado por Nabil



Realizado por Nabil

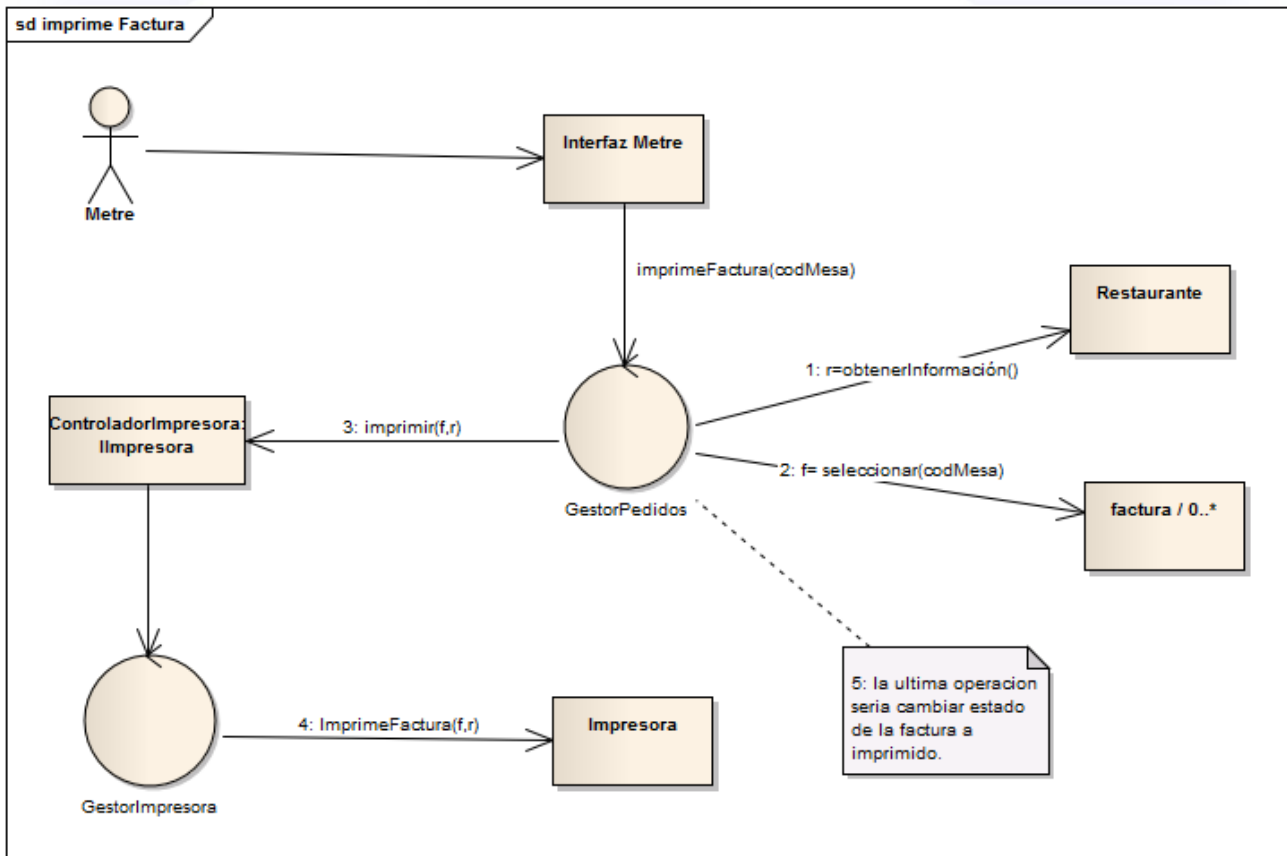


Realizado por Nabil

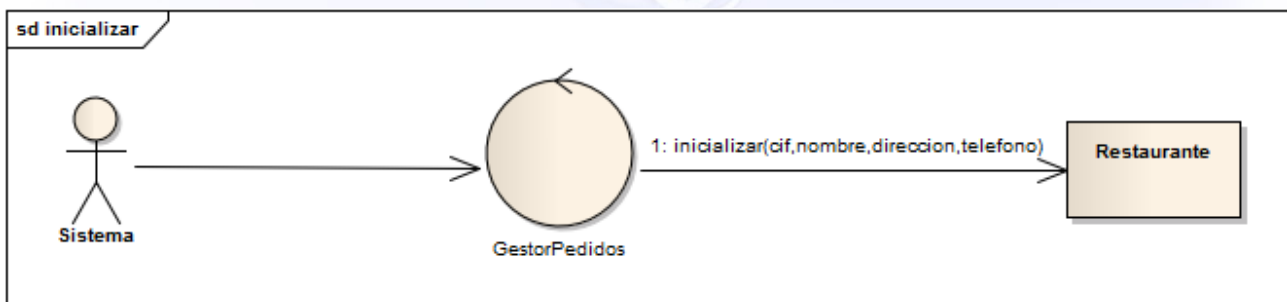


Nota: Nueva Factura se ejecuta solamente la primera vez cuando el cliente pide factura, después siempre se ejecuta modifica factura cuando el cliente hace otros pedidos y pide factura o cuando

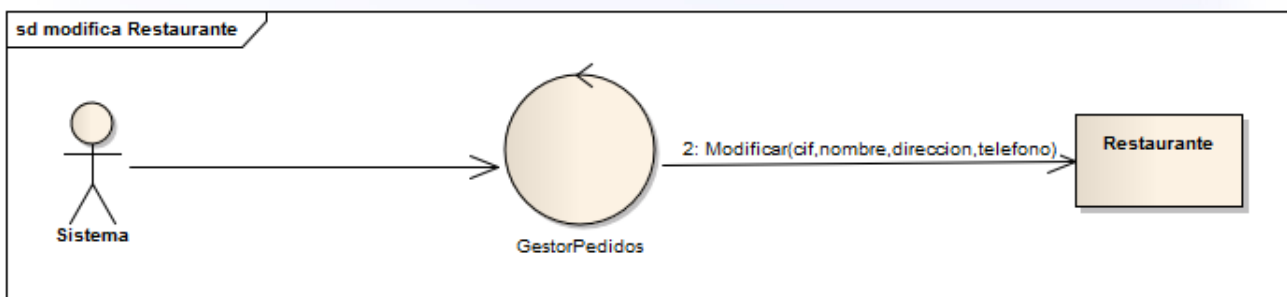
quiere pagarla. Realizado por Nabil.



Realizado por Nabil



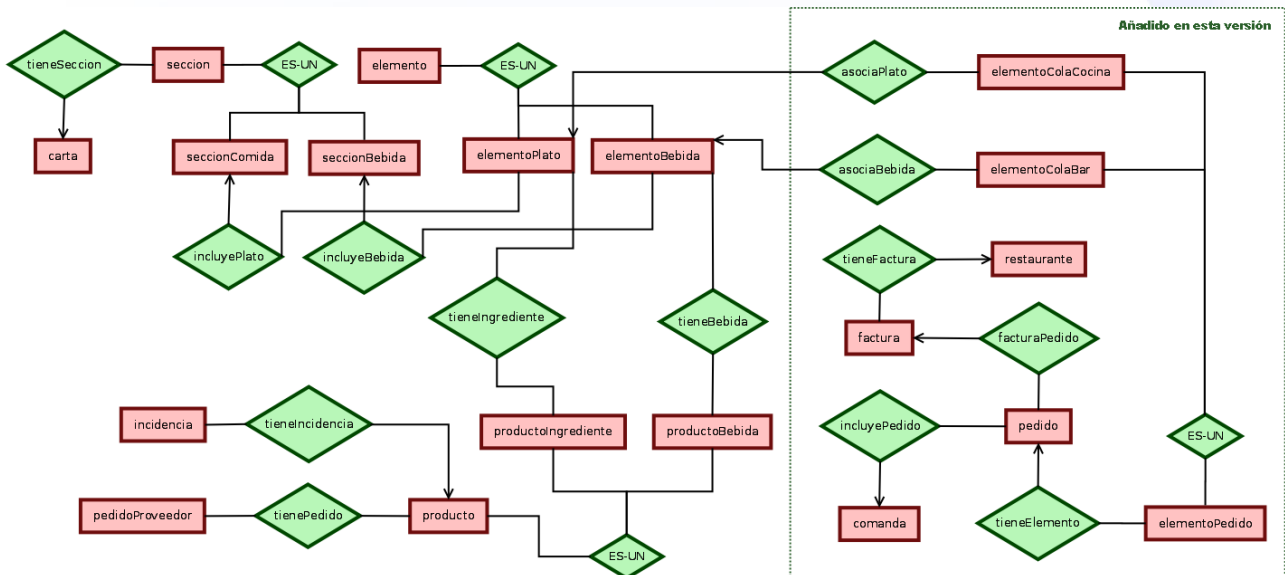
Realizado por Nabil



Realizado por Nabil

8. DISEÑO DE LA BASE DE DATOS

A continuación se muestra el diagrama Entidad-Relación, del cuál se saca el conjunto de tablas utilizadas para construir la base de datos:



9. DISEÑO DE LAS INTERFACES DE USUARIO

La interfaz de usuario se ha realizado siguiendo los diseños realizados en la primera iteración. Se han utilizado tonos suaves de azules para dar una sensación agradable al usuario durante su utilización. Los diálogos y las ventanas son simples, tratando de dejar el máximo espacio libre para no estresar al usuario con demasiada información. Se ha seguido un coloreado plano junto con degradados suaves de azul a blanco en algunos puntos, para dar continuidad y espacio a cada escritorio.

Para comenzar, mostraremos el diseño reutilizado del diálogo de confirmación.

Confirmación Indica la acción que se va a confirmar

Añadir nuevo elemento a la carta

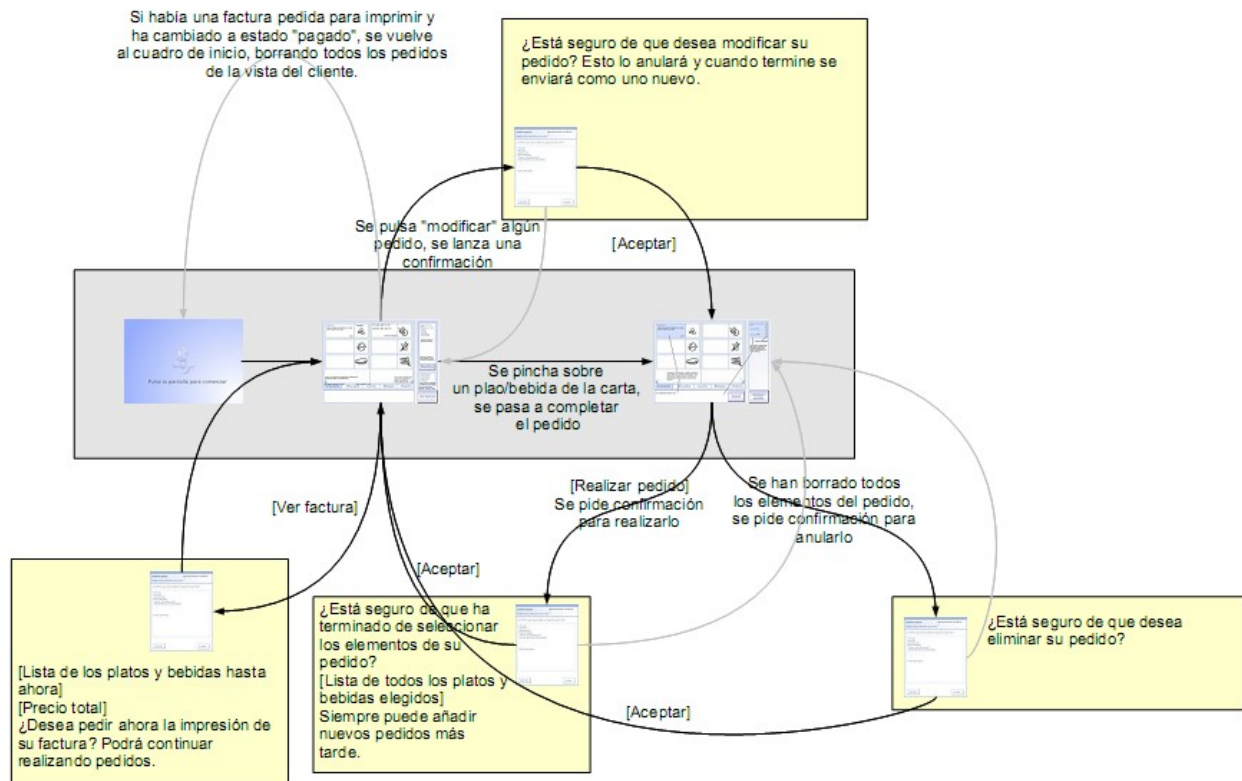
¿Confirma que desea añadir el siguiente elemento?

Sección :
Nombre:
Descripción:
Ruta a imagen:
Tiempo de elaboración:
Lista de productos asociados:
-
-
-
Precio de venta:

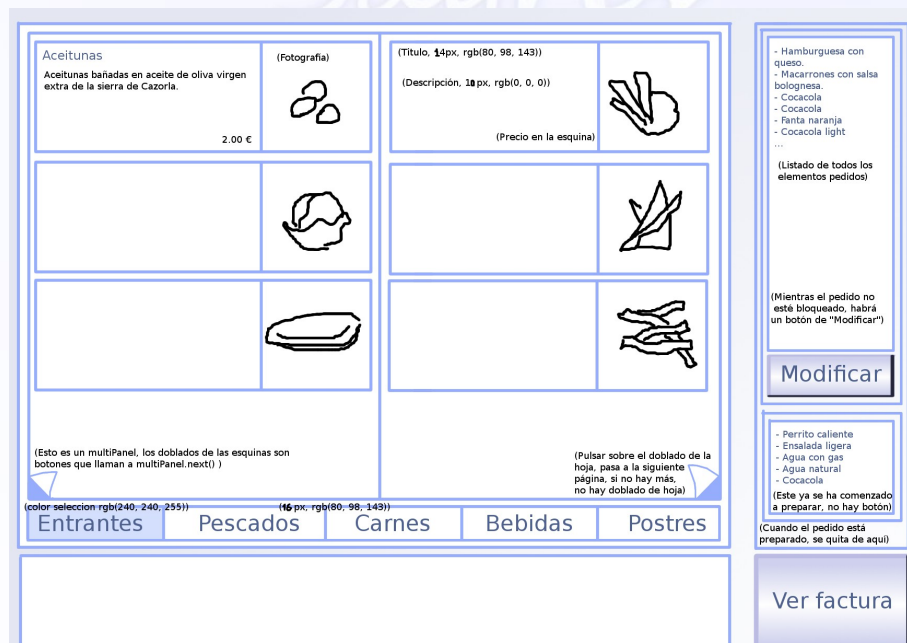
Cancelar Aceptar

A continuación mostraremos la línea de comunicación entre interfaces de usuario que sigue para cada usuario.

Interfaz de usuario de Cliente



Este es el storyboard que sigue un cliente.



Se puede ver una ventana limpia de información (salvo las anotaciones para el equipo de implementación).

Aceitunas
Aceitunas bañadas en aceite de oliva virgen extra de la sierra de Cazorla.
2.00 €

(Se selecciona un elemento, se añade el comentario abajo si se quiere y se pulsa el botón añadir, o bien se pulsa otra vez sobre el cuadro "Aceitunas" (Los dos lo añaden al pedido actual en el cuadro de la derecha))

Entrantes Pescados Carnes Bebidas Postres

Con bastante sal por fav|

Añadir

Realizar pedido

Huevo duro con patatas fritas
Lechuga fresca
CocaCola (10px)

(elimina el elemento)

(Pulsa sobre un elemento de este cuadro y se activa, el comentario aparece en el cuadro de comentarios (Se puede modificar))

En este boceto se está modificando o creando un nuevo pedido.

Sagres

Interfaz de usuario de Metre

Gestionar productos

Mesa 2, pedido 3426

Agua (10 px)

Del tiempo, por favor (10 px)

Cocacola light

Fanta naranja

Cocacola (Color de fondo rgb(211, 223, 253))

(fondo rgb(255,255,255), blanco)

(Click sobre un panel, cambia el estado de la bebida a "Preparado" y se borra de la lista, cuando es el último elemento del pedido lanza un diálogo de confirmación preguntando si se desea "Cerrar el pedido de bebida", si se acepta, se cambia el estado de la última bebida y se quita de la lista y se pone el siguiente pedido (getSiguientePedidoBebida())

(Color letra rgb(80, 98, 143)) 7 bebidas pendientes

Mesa 1

Mesa 8

Mesa 3

(Click sobre el panel, imprime la factura de la Mesa 3, se borra de esta lista y, si no estaba en la lista de "Mesas por facturar", se incluye -->)

12:36

Martes 12, Agosto 2010

(letra 14 px)

Mesas por facturar

Mesa 1

Mesa 3

Mesa 4

Mesa 5

Mesa 6

(16 px) Mesa 7

Mesa 8

Mesa 9

Mesa 10

Mesas que tienen facturas en estado "Imprimido" o "En cola" Tocar sobre cualquiera lanza un dialogo para facturar (dialogo confirmacion).

(letra 36 px)

Interfaz de usuario de Cocinero

Gestión de productos



(letra 36px)
12:36
Martes 12, Agosto 2010
(letra 14px)

Mesa 2, pedido 3426 (Fondo rgb(245,245,255))
Hamburguesa de queso (10px) (Fondo rgb(211, 223, 253))
Sin lechuga, por favor.
Pollo empanado con bechamel y con patatas fritas
iiiiCon mucha salsa alioli!!!!
Pasta Bolognesa
Pasta Bolognesa
(Pulsar sobre un panel, cambia el panel a la cola de platos preparándose, con el mismo pedido y mesa. Si era el último plato, se elimina el pedido de la cola y se carga el siguiente)

(Color letra rgb(80, 98, 143)) 7 platos pendientes

Mesa 6, pedido 3424
Mesa 4, pedido 3425
(Aquí pulsar sobre un plato lo marca como preparado y se quita. Cuando es el último plato del pedido, se pide confirmación de si "Cerrar el pedido", si se acepta, se cambia el estado del último plato a "Preparado" y se quita el pedido de la lista)
(Borde plano, rgb(150, 172, 229), 5px)
5 platos preparándose (letra 16px)

Sagres

APÉNDICE 1.0

Fecha	02/05/10
Descripción del problema	
Impacto del problema	
Soluciones adoptadas	<ul style="list-style-type: none">• Se ha generado el documento de diseño.
Anexos a la versión	

Sagres