

# Documento UML

*v4.0.1*

Ingeniería del Software



PiKey Team- P<sub>K</sub>T:

Jesús Aguirre Pemán  
Enrique Ballesteros Horcajo  
Jaime Dan Porras Rhee  
Ignacio Iker Prado Rujas  
Alejandro Villarín Prieto

3 de Junio de 2013



# Índice general

<b>I</b>	<b>Introducción</b>	<b>2</b>
1.	Propósito . . . . .	3
2.	Audiencia . . . . .	3
3.	Alcance . . . . .	3
4.	Definiciones, acrónimos y abreviaturas . . . . .	3
5.	Referencias . . . . .	3
<b>II</b>	<b>Requisitos</b>	<b>6</b>
1.	Modelo de Dominio . . . . .	8
2.	Diagramas de actividad . . . . .	8
<b>III</b>	<b>Análisis</b>	<b>10</b>
1.	Diagramas de paquetes . . . . .	11
2.	Diagramas de comunicación . . . . .	11
2.1.	Menú del restaurante . . . . .	11
2.2.	Reservas del restaurante . . . . .	13
2.3.	Comandas del restaurante . . . . .	14
2.4.	Factura del restaurante . . . . .	15
2.5.	Incidencias de mantenimiento . . . . .	15
3.	Diagramas de clases de análisis . . . . .	15
3.1.	Menú del restaurante . . . . .	15
3.2.	Reservas del restaurante . . . . .	15
3.3.	Comandas del restaurante . . . . .	16
3.4.	Factura del restaurante . . . . .	17
<b>IV</b>	<b>Diseño</b>	<b>18</b>
1.	Diagramas de secuencia . . . . .	19
1.1.	Menú del restaurante . . . . .	19
1.2.	Reservas del restaurante . . . . .	22
1.3.	Comandas del restaurante . . . . .	23
1.4.	Factura del restaurante . . . . .	25

1.5.	Recuento de existencias . . . . .	25
2.	Diagramas de clases de diseño . . . . .	26
2.1.	Menú del restaurante . . . . .	26
2.2.	Reservas del restaurante . . . . .	28
2.3.	Comandas del restaurante . . . . .	29
2.4.	Factura del restaurante . . . . .	31
2.5.	Cargadores de ficheros . . . . .	31
3.	Diagramas de componentes . . . . .	31
4.	Diagramas de despliegue . . . . .	32
 <b>V Patrones de diseño</b>		 <b>33</b>

**Parte I**

**Introducción**

## 1. Propósito

Este documento pretende recopilar de manera clara y concisa todos los diagramas UML del Software KIKE HOSTELERIA ®, generados con la herramienta Bouml. Se pueden dividir en tres partes, cada una de ellas más cercana al código del lenguaje de programación, y con mayor grado de detalle. Éstas son: Requisitos, análisis y diseño, partes fundamentales de este documento.

## 2. Audiencia

La aplicación está pensada para hoteles y/o restaurantes de capacidad media/baja, pero podría ampliarse para capacidades mayores sin dificultad.

El documento está pensado para dos audiencias distintas. En las primeras partes, sirve como lazo de comunicación entre el equipo (nosotros) y la empresa hostelera. Según se entra más en materia, centrándose en el Software, está pensado para el equipo de diseñadores y programadores.

## 3. Alcance

Todos los diagramas que se han requerido para la posterior codificación de la aplicación. Se añade, además, una breve explicación de los mismos, poniéndolos en contexto e indicando su naturaleza y significado.

Todo el documento debe entenderse dentro del ámbito del UML y del modelo del Proceso Unificado, seguido durante todo el proyecto.

## 4. Definiciones, acrónimos y abreviaturas

Todo lo referente a esta sección se encuentra en el documento anexo **Definiciones, acrónimos y abreviaturas**. Ahí se detallan las palabras poco comunes, los anglicismos y abreviaturas que se emplean.

## 5. Referencias

Se han utilizado los siguientes recursos:

- Nuestro propio documento de Documento de casos de uso, así como Definiciones, acrónimos y abreviaturas
- Apuntes de Clase de la asignatura “Ingeniería del Software” de la Universidad Complutense de Madrid, Curso 2012-2013
- **Booch, G.; Rumbaugh, J.; Jacobson, I.** . El lenguaje unificado de modelado (2006, 2ª edición).

- **Larman, C..** UML y patrones (2003, 2ª edición).
- **Ambler, S. W.** The Elements of UML Style (2003).





# Parte II

## Requisitos

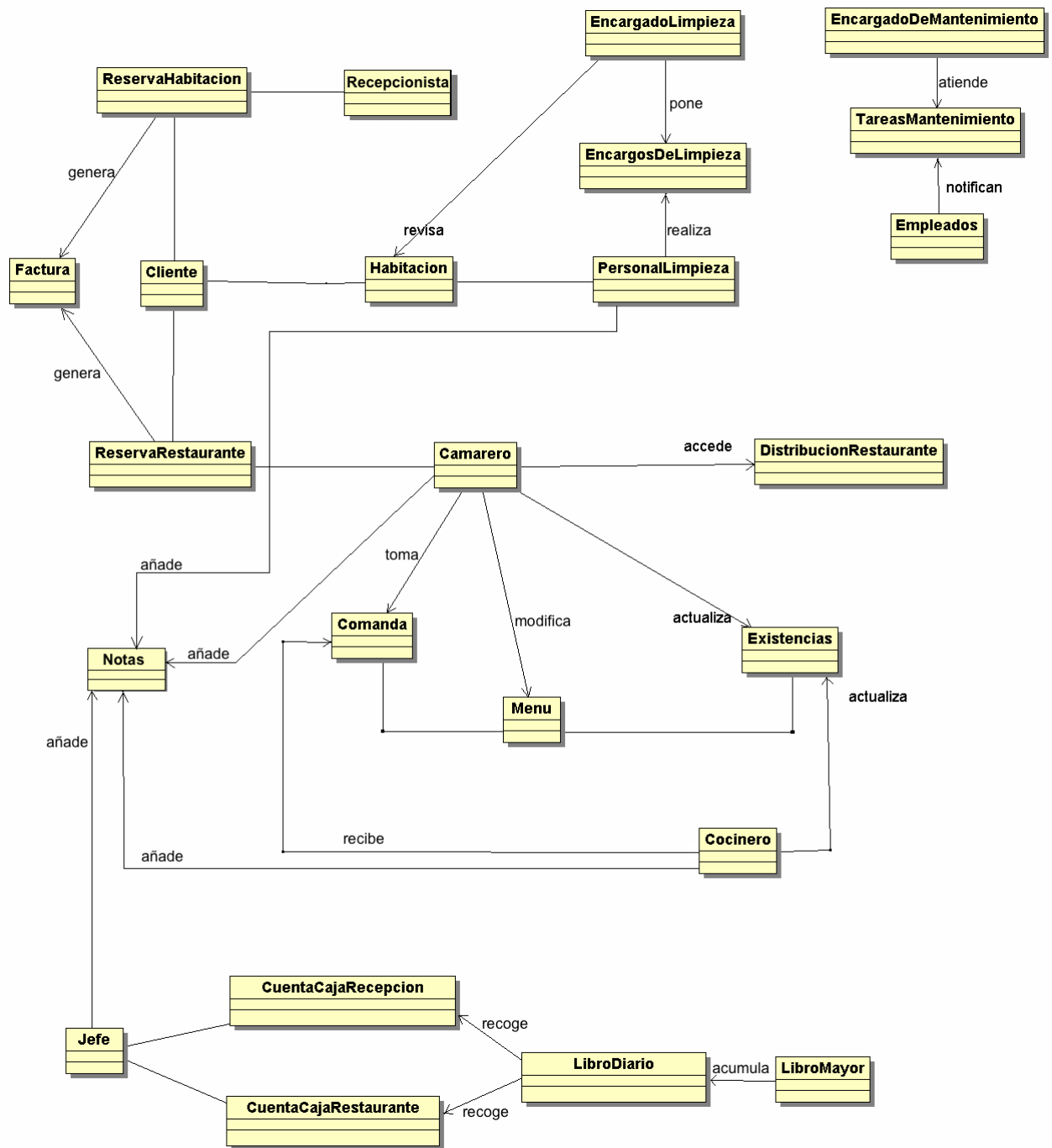


Figura 1: Modelo de dominio

## 1. Modelo de Dominio

En esta sección se recoge el modelo de dominio de toda la aplicación. De izquierda a derecha, podemos ver la relación entre el cliente y las reservas que puede hacer, tanto del restaurante como del hotel. A partir de ellas se puede generar una factura.

Las reservas del hotel son atendidas por el recepcionista, y las del restaurante por el camarero (o en su defecto, el maître). El camarero, además tiene varias posibilidades, puede acceder y modificar la distribución de las mesas del restaurante, tomar o modificar comandas, ver o modificar el menú y ver o modificar las existencias de la cocina y el bar.

En cuanto a la limpieza, el encargado asigna tareas de limpieza al personal de limpieza. Posteriormente, cuando se suponen completadas, el encargado revisa si se han cumplido los objetivos en dichos encargos, dando su visto bueno.

Además, el encargado de mantenimiento puede ver las tareas que tiene pendientes por resolver en el hotel o el restaurante. Estas son notificadas por el resto de empleados (o el mismo, al ser un empleado más) cuando las descubren. Otro medio de comunicación entre los empleados son las notas. Cualquiera puede escribir una y cualquiera puede verlas. Su fin es distinto al de las obligaciones de mantenimiento, ya que pueden tratarse temas más triviales o que no tocan dichas tareas.

Por último, el jefe del negocio tiene acceso a las cuentas de las cajas de la recepción y del restaurante, que se recogen en el libro diario de la aplicación. Éste, a su vez, se resume en el libro mayor.

## 2. Diagramas de actividad

No consideramos necesarios estos diagramas, ya que creemos que la información que aportan ya la dan los diagramas de secuencia junto con la documentación.



## Parte III

# Análisis

## 1. Diagramas de paquetes

En la aplicación hay una clara brecha entre la lógica y la interfaz de la misma, más nítida por el uso del patrón Modelo-Vista-Controlador (ver Parte V para más información). Esto se refleja en el diagrama de paquetes. Además, en la lógica se distinguen a su vez dos partes, lo relativo a los empleados y lo relativo a las cuentas del negocio. Cada una de estas partes incluye lo que indica el diagrama.

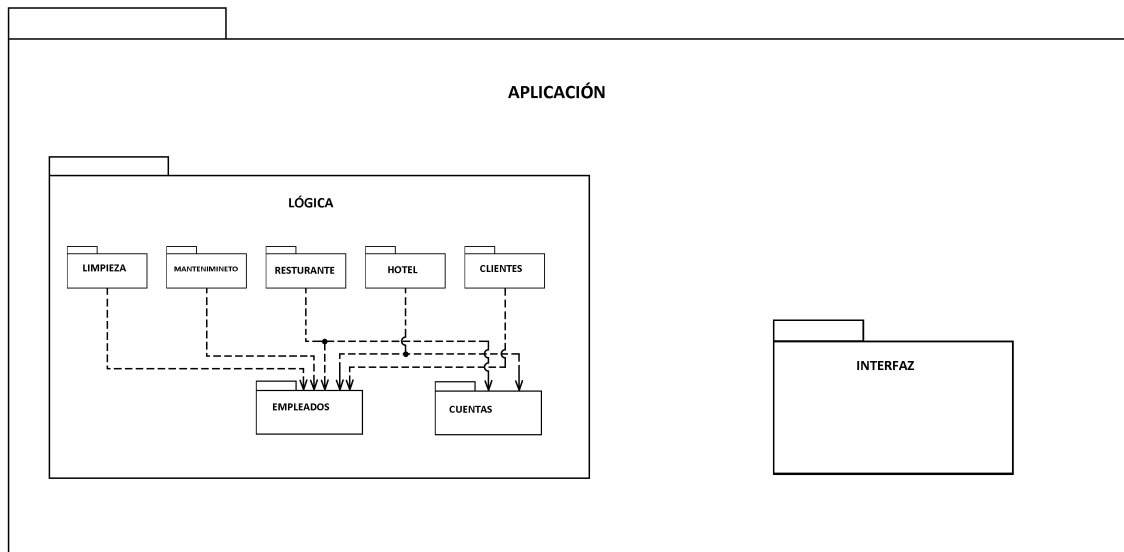


Figura 2: Diagrama de paquetes

## 2. Diagramas de comunicación

### 2.1. Menú del restaurante

Desde el menú principal se accede a la vista del menú que le solicita los datos a la clase Menú. Requiere los tipos de plato existentes y las consumiciones de cada tipo de plato.

Se accede a la pestaña de ver menú que reúne los datos del Menú y te permite editar los campos de las consumiciones que se quiera a través de una interfaz.

Accediendo a la pestaña ver menú, que reúne los datos de la clase Menú, te permite añadir consumiciones a ésta a través de una interfaz.

A través de la interfaz de ver menú que reúne las consumiciones del menú, te permite eliminar cualquier plato de este.

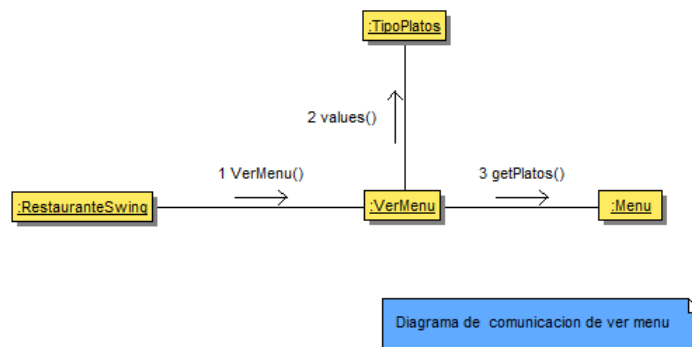


Figura 3: Diagrama de comunicación del menú

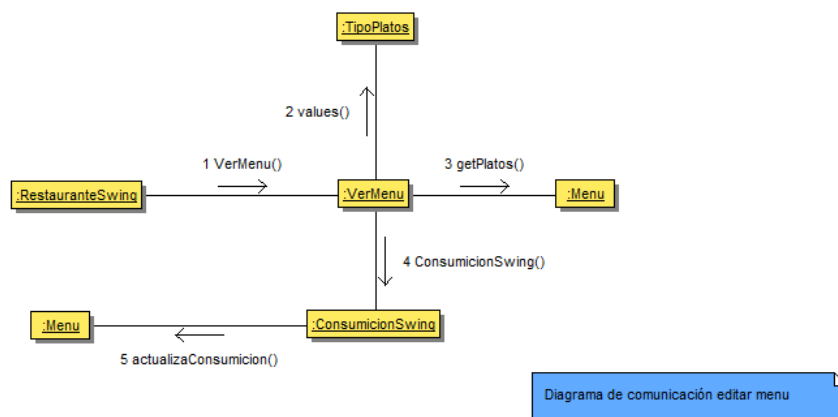


Figura 4: Diagrama de comunicación de modificar el menú

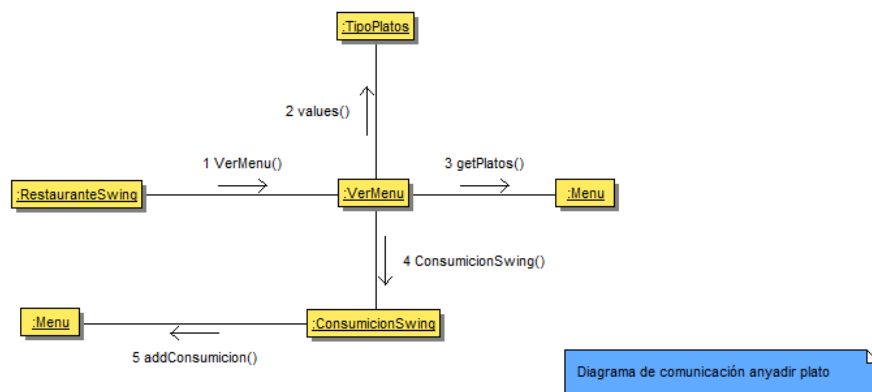


Figura 5: Diagrama de comunicación de añadir plato al menú

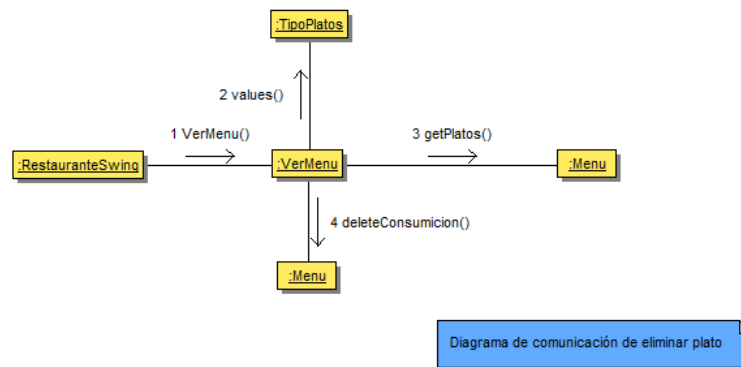


Figura 6: Diagrama de comunicación de eliminar consumición

## 2.2. Reservas del restaurante

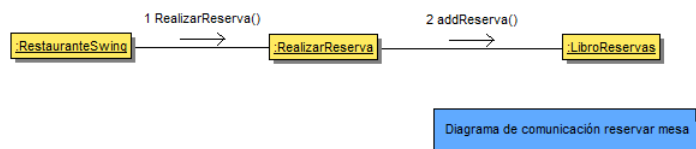


Figura 7: Diagrama de comunicación de reservar una mesa

A través de la interfaz realizar reserva, que te permite rellenar los campos de una, se añade al libro de reservas que acumula todas.

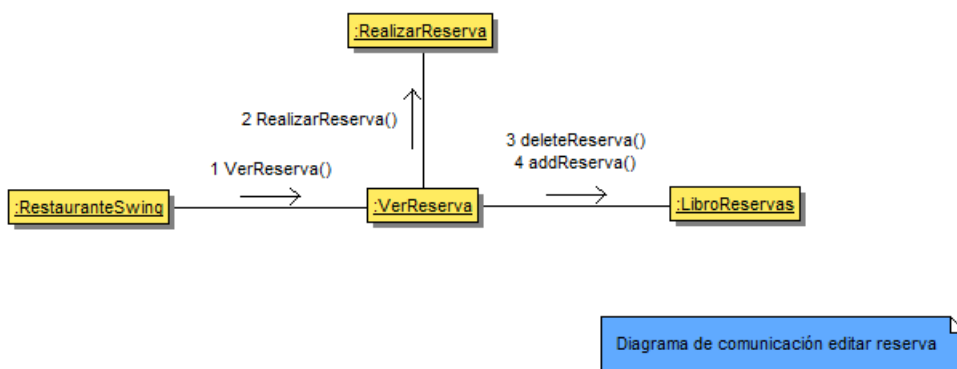


Figura 8: Diagrama de comunicación de editar reserva del restaurante



Accediendo a ver reserva se crea una nueva reserva inicialmente con los campos de la reserva a editar. Una vez creada en realizar reserva se devuelve y se añade a la lista de reservas, se la que se ha borrado previamente la anterior.

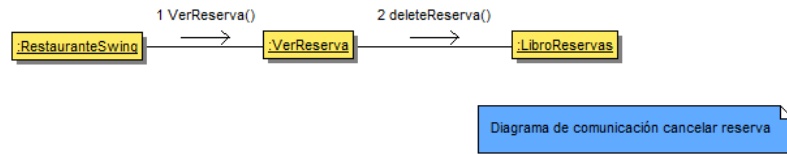


Figura 9: Diagrama de comunicación de anular reserva del restaurante

Accediendo desde la ventana principal, ver reserva te ofrece la posibilidad de eliminar la reserva deseada del libro de reservas.

### 2.3. Comandas del restaurante

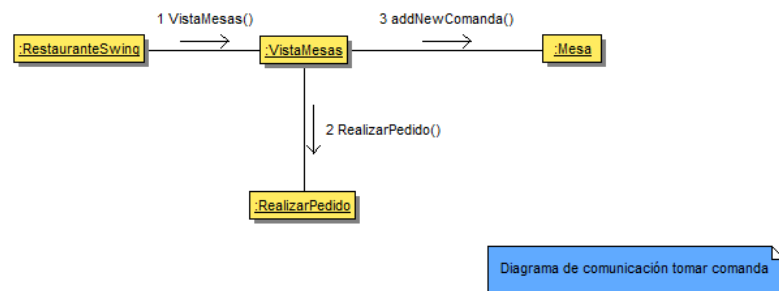


Figura 10: Diagrama de comunicación de tomar comanda

Desde la ventana principal se accede a la vista de las mesas, que tras haber seleccionado una contacta con realizar pedido que te permite crear una comanda. Una vez creada, la vista de las mesas se la envía a la mesa para que la almacene.

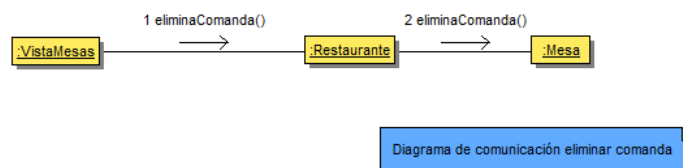


Figura 11: Diagrama de comunicación de eliminar comanda

Desde la vista de las mesas se selecciona una comanda y se notifica a la mesa su borrado.

## 2.4. Factura del restaurante

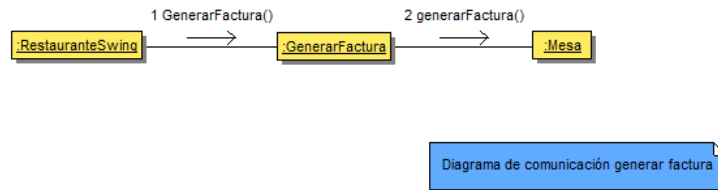


Figura 12: Diagrama de comunicación de generar factura del restaurante

Desde la ventana principal se accede a generar factura, donde se selecciona la mesa de la cual se quiere cobrar la factura, a quien se le solicita ésta para gestionarla desde generar factura.

## 2.5. Incidencias de mantenimiento

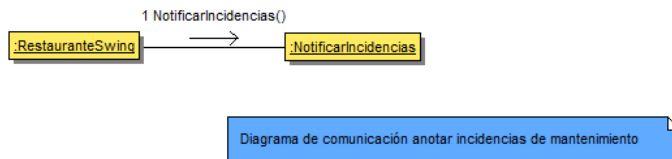


Figura 13: Diagrama de comunicación de mantenimiento

Se crea una nota de texto al acceder a notificar incidencias que te permite indicar tu problema.

## 3. Diagramas de clases de análisis

### 3.1. Menú del restaurante

El camarero puede ver los elementos del menú, así como modificarlos (que implica verlos). Para ello se accede al menú, que lleva los distintos tipos de consumiciones, que son bebidas, primeros, segundos y postres.

### 3.2. Reservas del restaurante

El maître, o el camarero, puede añadir, modificar o eliminar reservas al libro de reservas, que es el sitio donde están todas las reservas. Cada reserva lleva asociada una fecha.

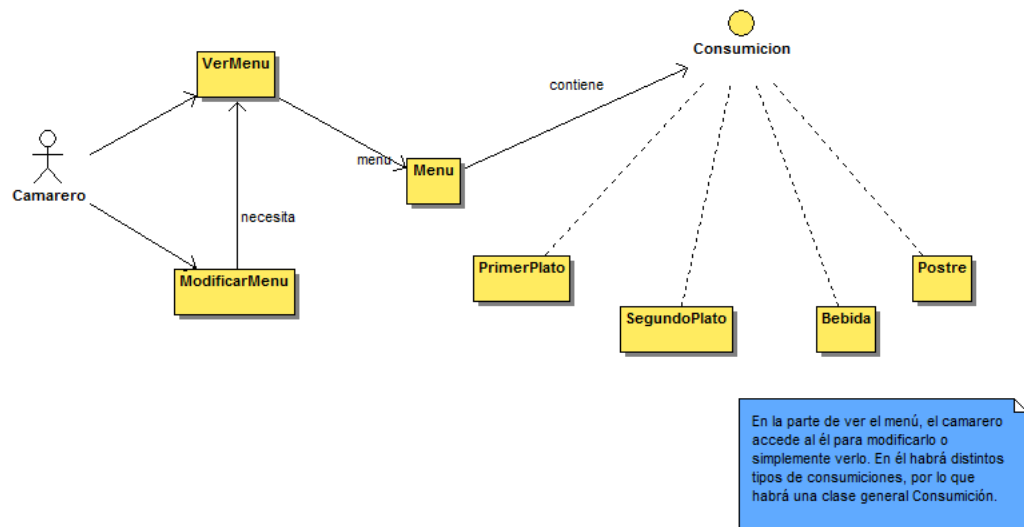


Figura 14: Diagrama de clases de análisis del menú

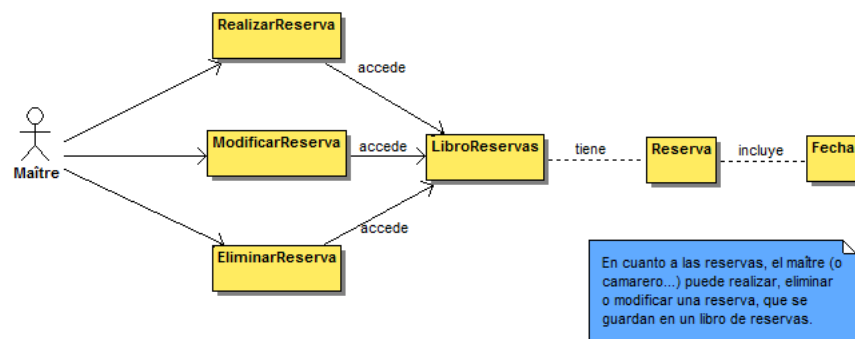


Figura 15: Diagrama de clases de análisis de las reservas del restaurante

### 3.3. Comandas del restaurante

El camarero puede realizar o modificar una comanda. Ese pedido se le asocia a una mesa determinada. La comanda en cuestion lleva un conjunto de consumiciones que se extraen del menú.

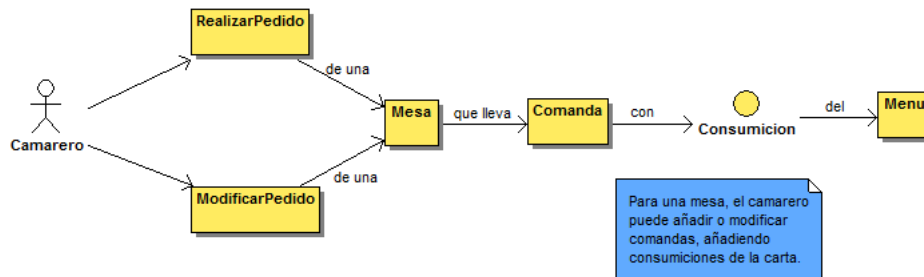


Figura 16: Diagrama de clases de análisis de las comandas

### 3.4. Factura del restaurante

Esta operación es sencilla. Llegado el momento de pagar, se pide que se genere una factura de una determinada mesa. Entonces se recopilan las consumiciones de esa mesa y se extrae el total a pagar.

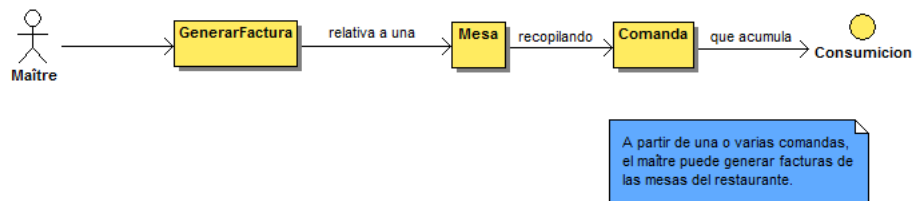


Figura 17: Diagrama de clases de análisis de las facturas

**Parte IV**

**Diseño**

## 1. Diagramas de secuencia

### 1.1. Menú del restaurante

La acción se inicia en la interfaz RestauranteSwing, que le pide al menú que desea verlo. Este le devolverá una lista con la información de qué tipos de consumiciones hay, para mostrarlos. Posteriormente, se notificará qué tipo de consumición se desea ver. En el bucle se le pide uno a uno los nombres de cada consumición de ese tipo, y opcionalmente se pedirá que muestre la descripción de alguno.

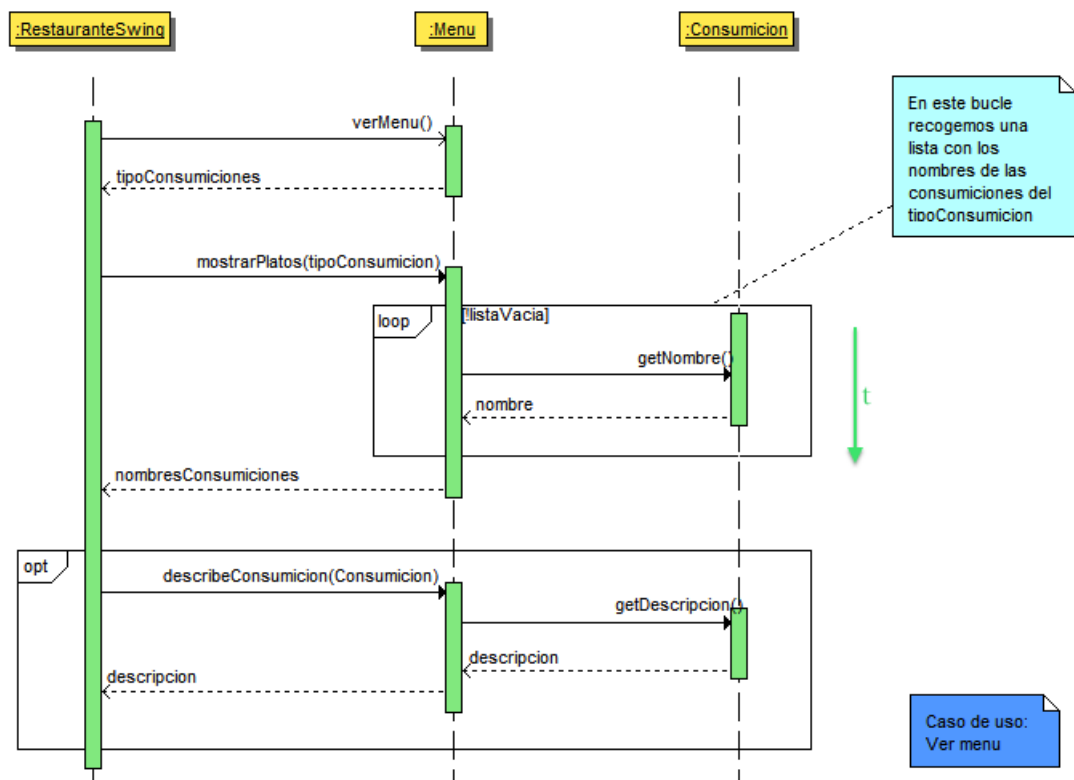


Figura 18: Diagrama de secuencia del menú

También se permite, como se ha dicho numerosas veces, modificar el menú. Hay tres opciones distintas para ello, a saber, añadir una nueva consumición, eliminar una existente o modificarla. Por claridad, cada una de estas opciones está en un diagrama distinto. El camino siempre es el mismo: de la interfaz al controlador, y de éste al menú. Cabe destacar que al modificar una consumición puedes cambiar el nombre, la descripción o el precio.

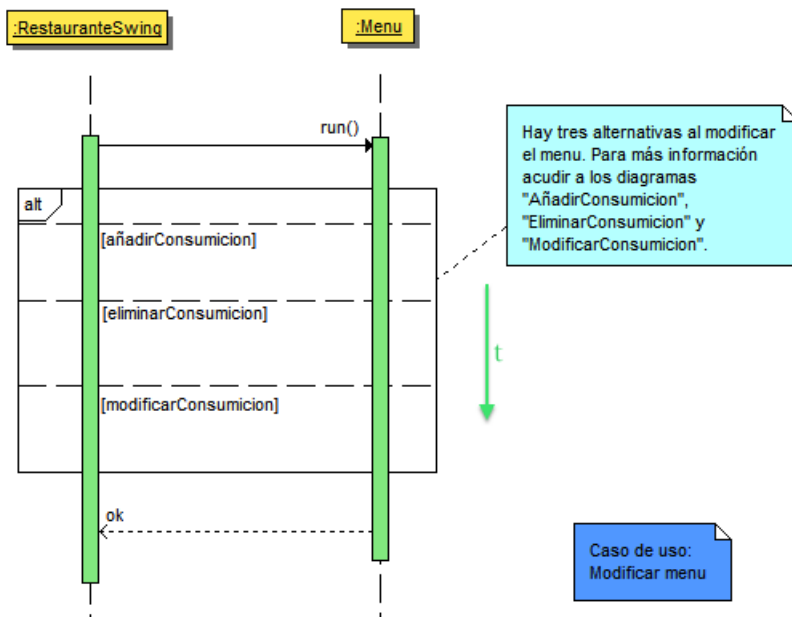


Figura 19: Diagrama de secuencia de modificar menú

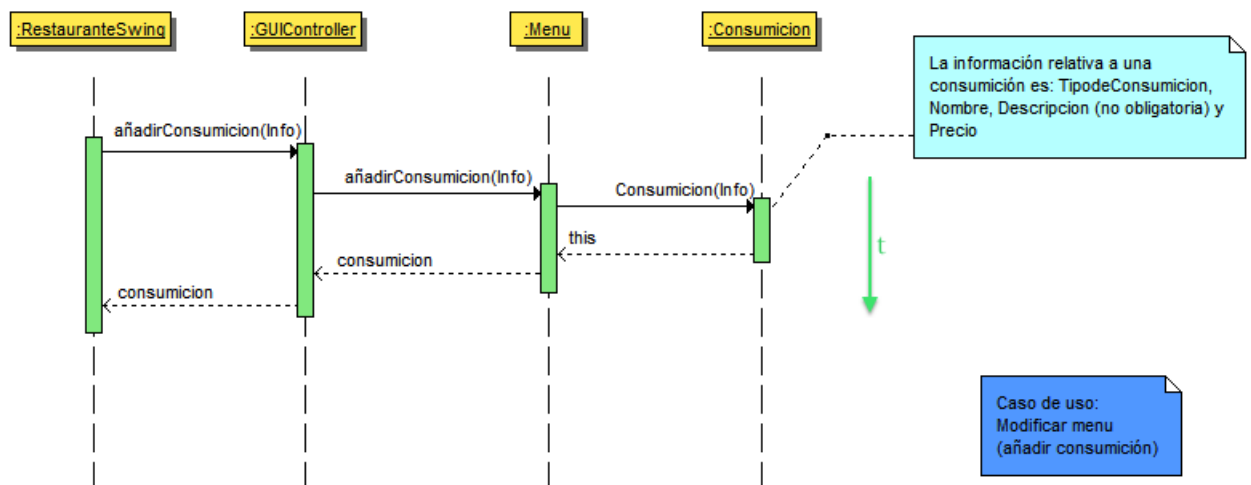


Figura 20: Diagrama de secuencia de modificar menú (añadir consumición)

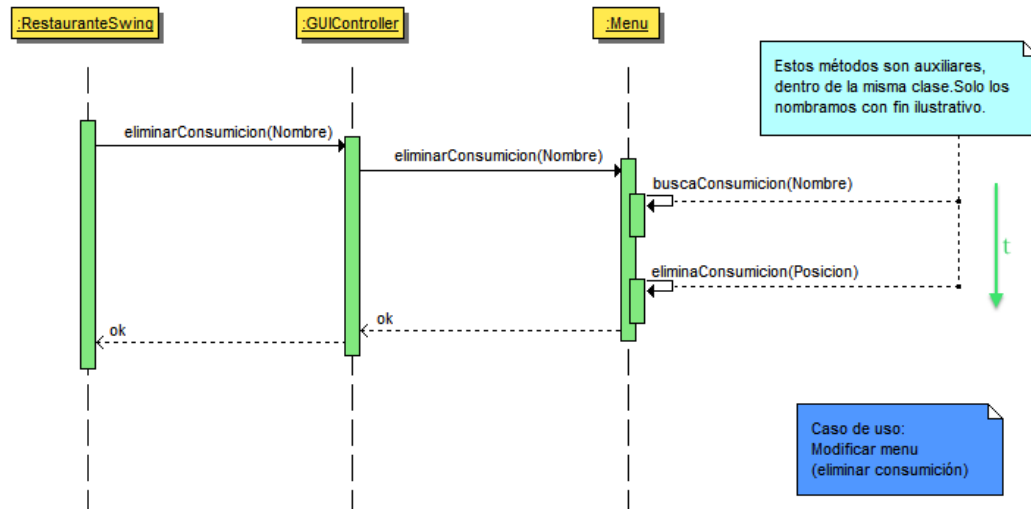


Figura 21: Diagrama de secuencia de modificar menú (eliminar consumición)

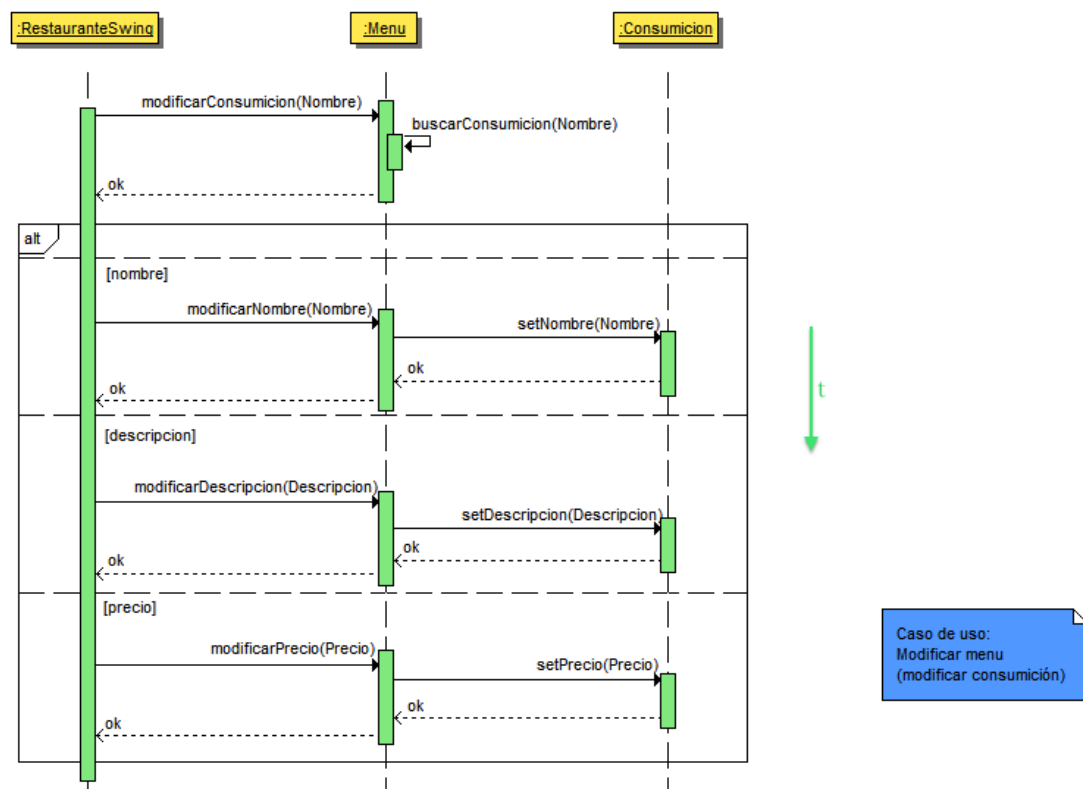


Figura 22: Diagrama de secuencia de modificar menú (modificar consumición)



## 1.2. Reservas del restaurante

Para ver las reservas, como siempre hacemos debido al M-V-C, la interfaz `RestauranteSwing` se lo pide al controlador, y este se lo pide al `Restaurante`, que lleva una lista con las reservas.

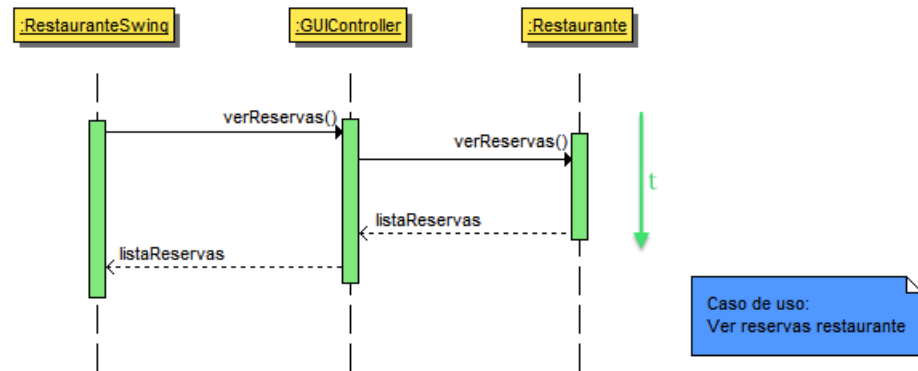


Figura 23: Diagrama de secuencia de ver las reservas del restaurante

Para hacer una reserva, el proceso es el mismo, sólo que se llama además al constructor de la reserva. Una vez creada se añade si es posible (puede que para la fecha solicitada se haya llenado ya el cupo de reservas). La anulación de una reserva es análoga.

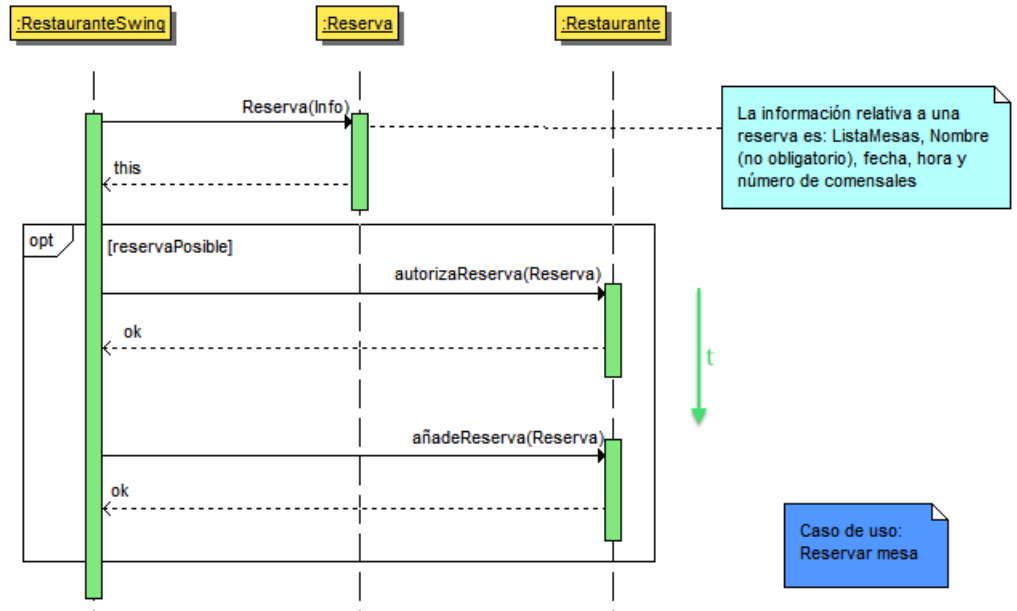


Figura 24: Diagrama de secuencia de reservar mesa del restaurante

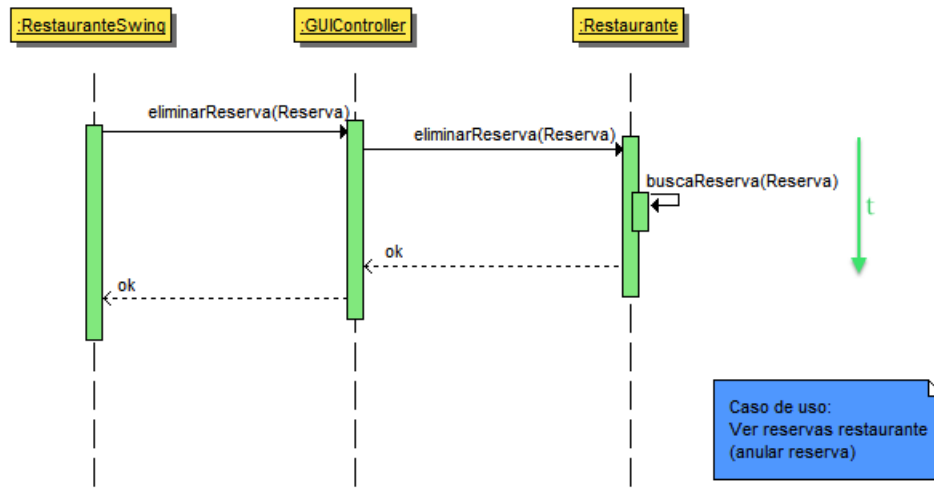


Figura 25: Diagrama de secuencia de anular reservas del restaurante

### 1.3. Comandas del restaurante

Para tomar una comanda, RestauranteSwing se lo solicita al GUIController, pasándole las consumiciones y la mesa a la que le corresponden. Este se lo pasa al restaurante, que llama al constructor de comanda con la lista de consumiciones. Finalmente se añade la comanda a la mesa antes mencionada.

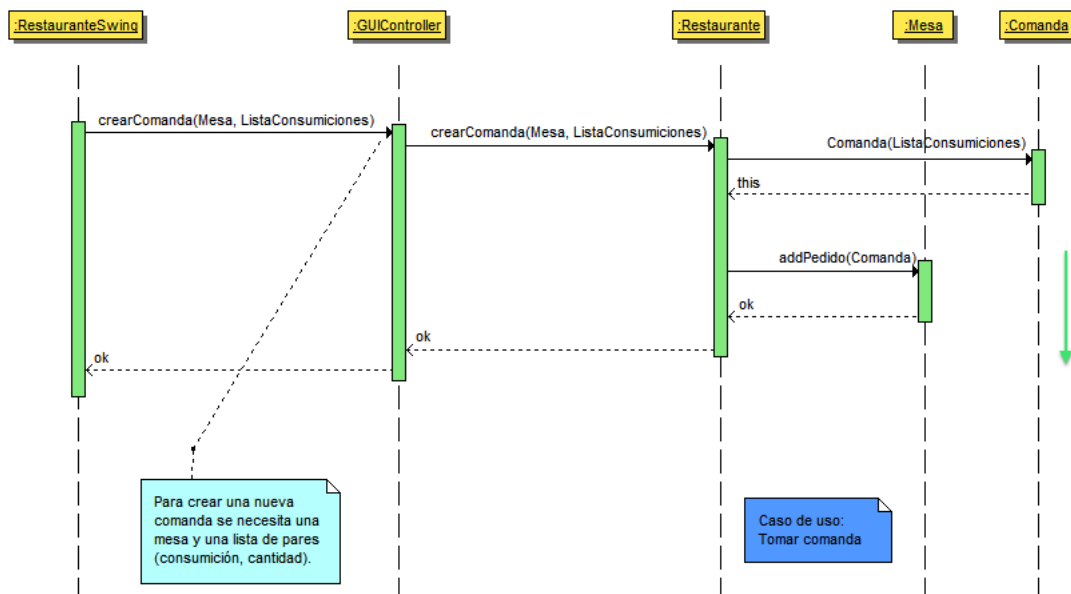


Figura 26: Diagrama de secuencia de tomar comanda

El proceso es el mismo, pero en vez de llamar al constructor de la comanda, se elimina

de la mesa correspondiente.

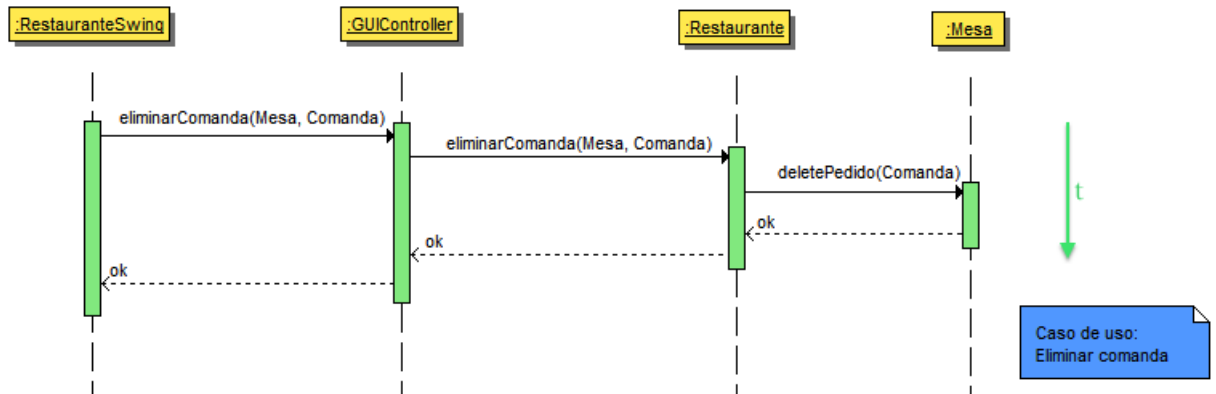


Figura 27: Diagrama de secuencia de eliminar comanda

Esta parte está en fase de pruebas, pero ya tiene su diagrama de secuencia definido. Permite añadir o eliminar consumiciones de la comanda correspondiente a una mesa. Nótese que es indiferente si la consumición existía ya o se trata de algo que ya se había pedido (como por ejemplo una botella de agua) y se incrementa su cantidad.

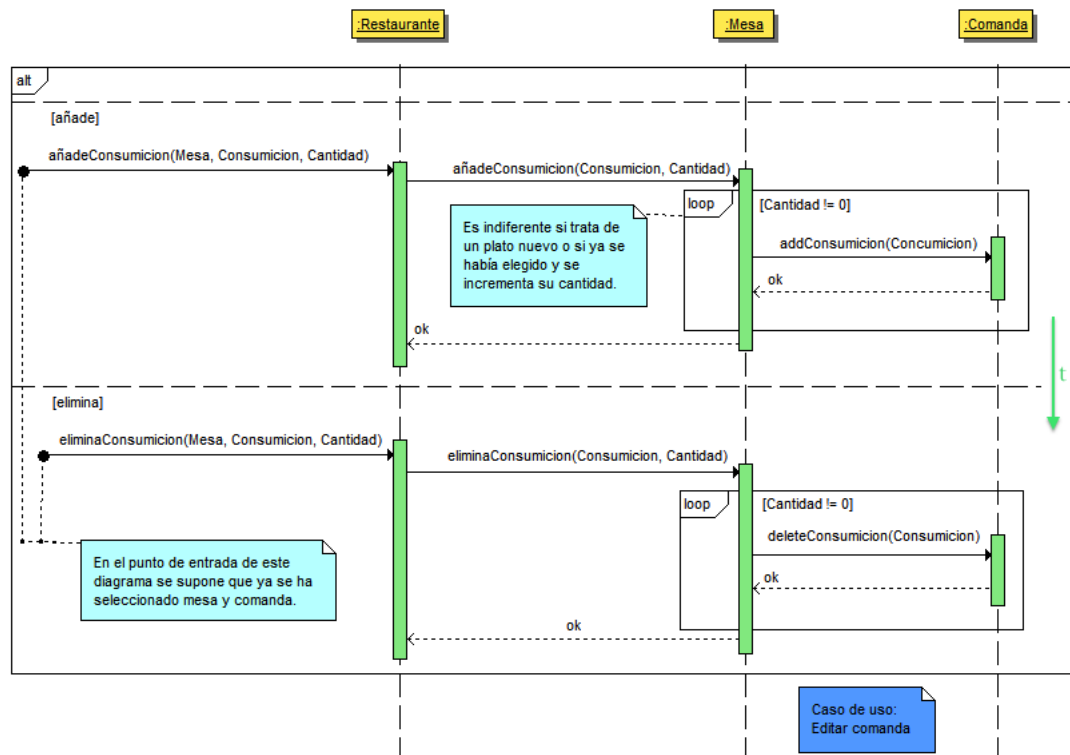


Figura 28: Diagrama de secuencia de modificar comanda

### 1.4. Factura del restaurante

La clase RestauranteGUI llama al controlador, pasándole la mesa de la que se quiere extraer la factura. Éste transmite el mensaje al restaurante, donde se encuentra la mesa en cuestión. Sobre ella, se llama a la clase factura, que ordena la lista de comandas que recibe. Simplemente recorre tipos de platos (primero bebidas, luego primeros...) y para cada uno, recorre las comandas, recogiendo la consumición que toque. Devuelve la cadena con la factura total.

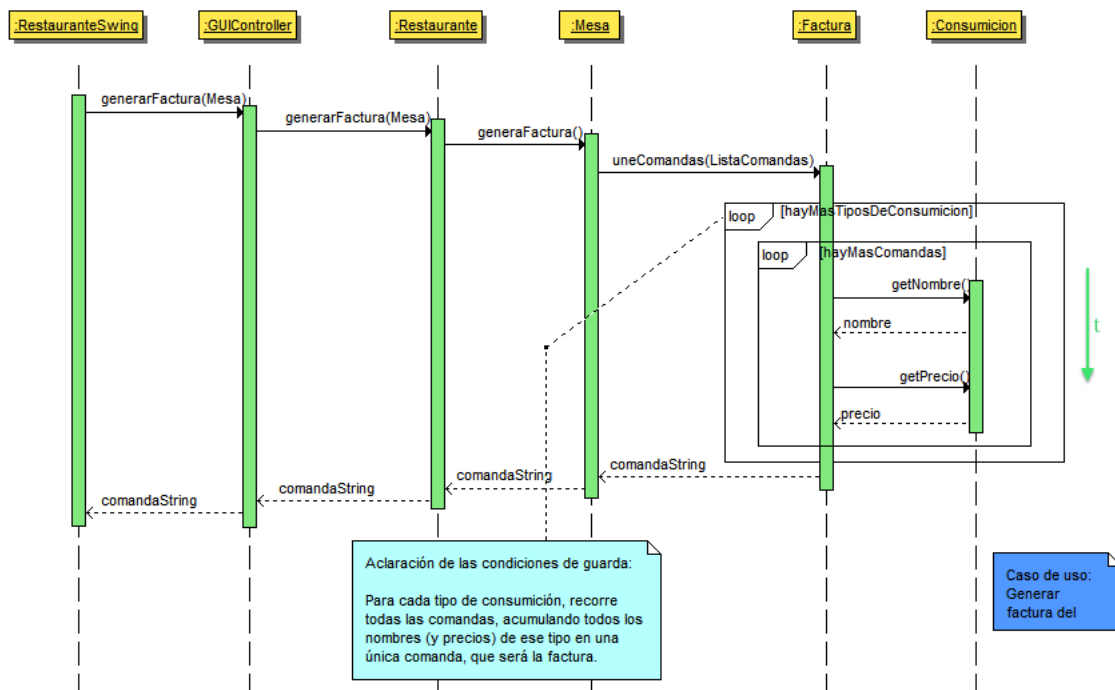


Figura 29: Diagrama de secuencia de generar factura del restaurante

### 1.5. Recuento de existencias

Esta parte no se implementa en esta versión, pero está en fase de desarrollo para presentársela al cliente como complemento. Simplemente habría una lista con las existencias, que tienen una cantidad asociada. En caso de necesitar modificar una de estas cantidades, la interfaz llama al controlador, que llama al inventario (donde está las existencias) y se cambia la cantidad de la que corresponda.

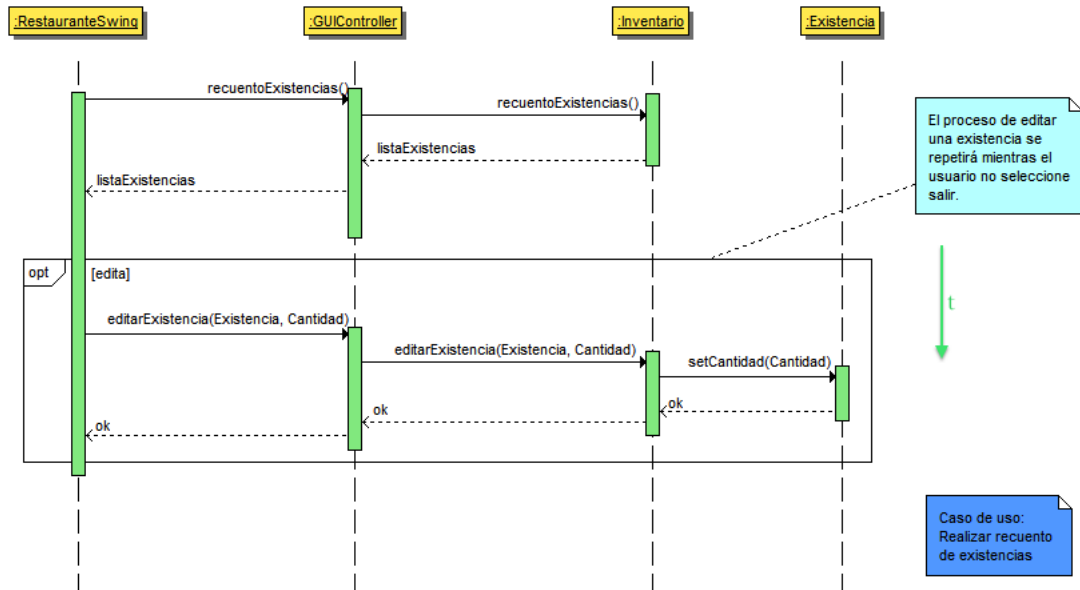


Figura 30: Diagrama de secuencia de realizar recuento de existencias del restaurante

## 2. Diagramas de clases de diseño

### 2.1. Menú del restaurante

Para mayor claridad hemos separado estos diagramas del menú en tres partes. En la figura 30, vemos la lógica del menú: implementa MenuInfo y lleva un array de observers y arrays con las consumiciones. En la siguiente figura, vemos las consumiciones concretas que implementan la interfaz consumición. En la figura 32 aparece la vista del menú.

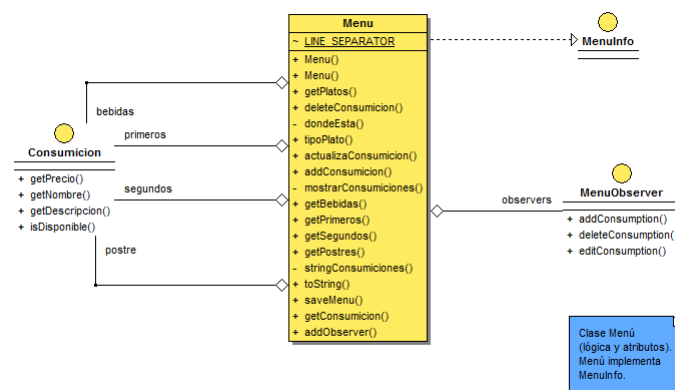


Figura 31: Diagrama de clases de diseño de la lógica del menú

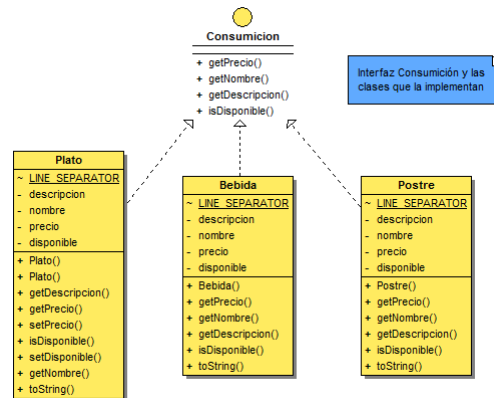


Figura 32: Diagrama de clases de diseño de las consumiciones

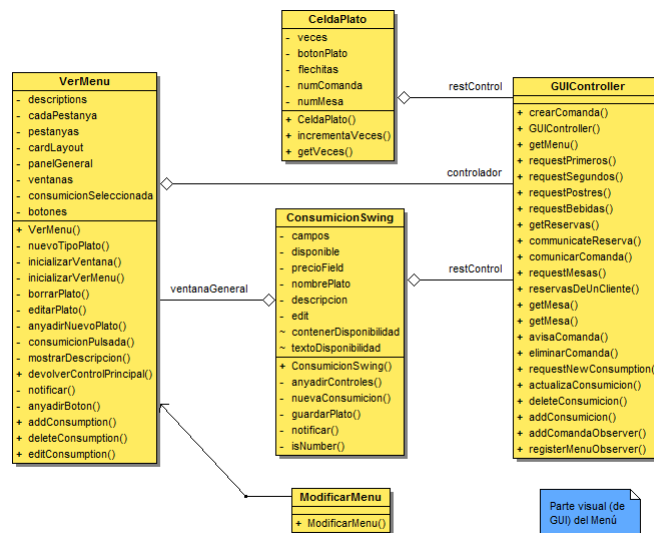


Figura 33: Diagrama de clases de diseño de la interfaz del menú

## 2.2. Reservas del restaurante

De la misma forma, hemos separado lógica de la vista en las reservas. Realizar reserva lleva una reserva que a su vez lleva una fecha e implementa ReservaInfo. La distribución de la vista es similar al caso anterior.

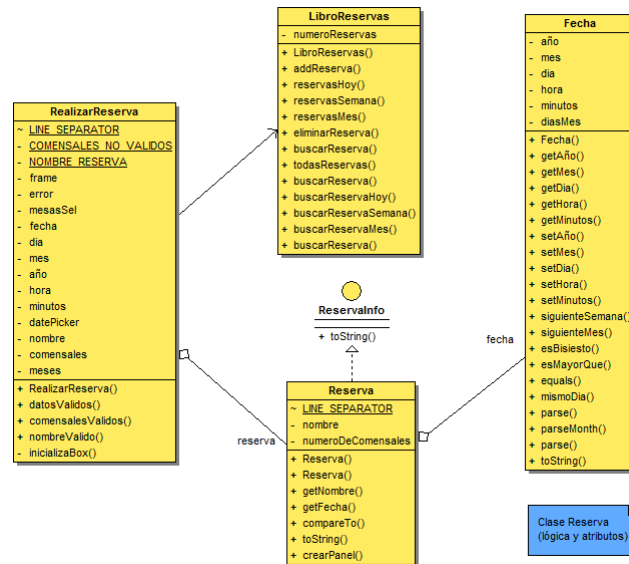


Figura 34: Diagrama de clases de diseño de la lógica de las reservas

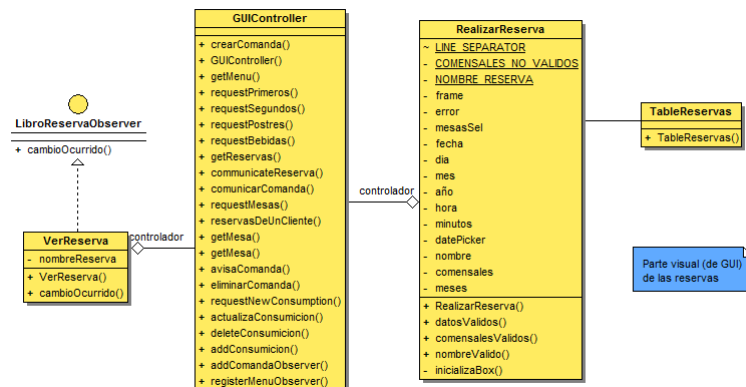


Figura 35: Diagrama de clases de diseño de la interfaz de las reservas

### 2.3. Comandas del restaurante

En cuanto a las comandas, llevan observadores del tipo comanda, que son implementados por RealizarPedido. A su vez, este lleva una ComandaInfo, implementada por comanda. El diagrama de las mesas es totalmente análogo, cambiando Comanda por Mesa.

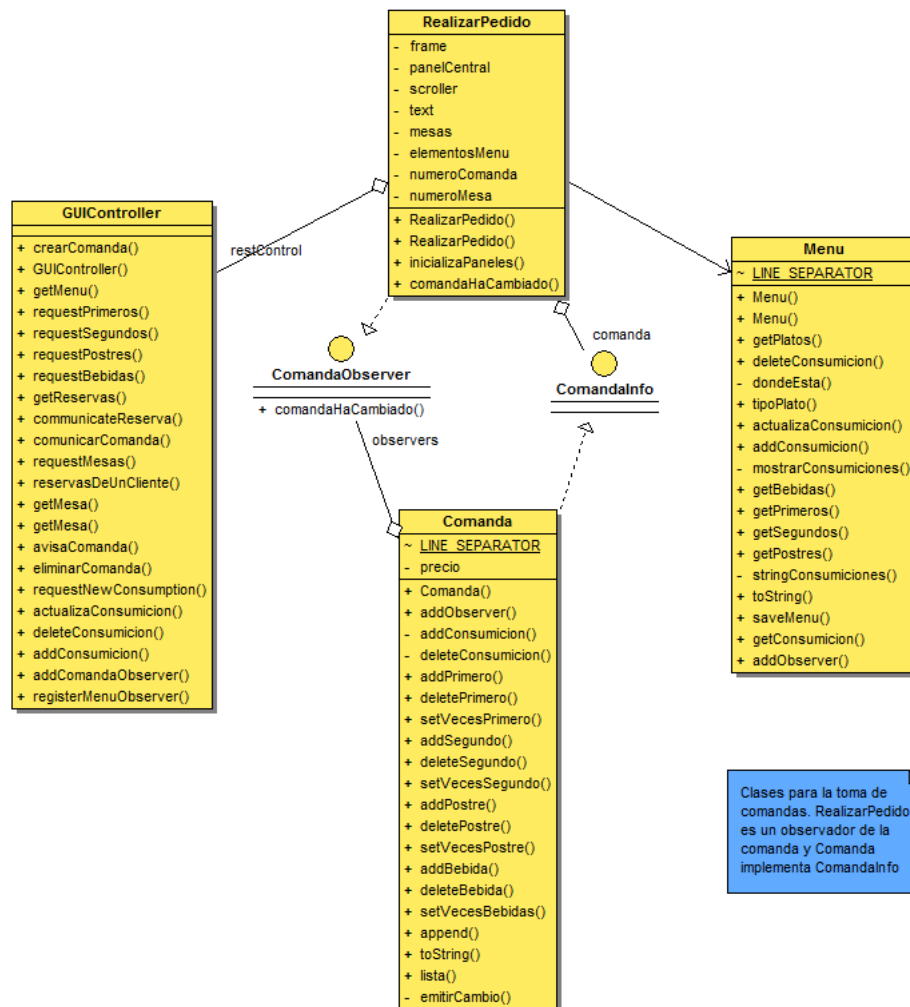


Figura 36: Diagrama de clases de diseño de las comandas



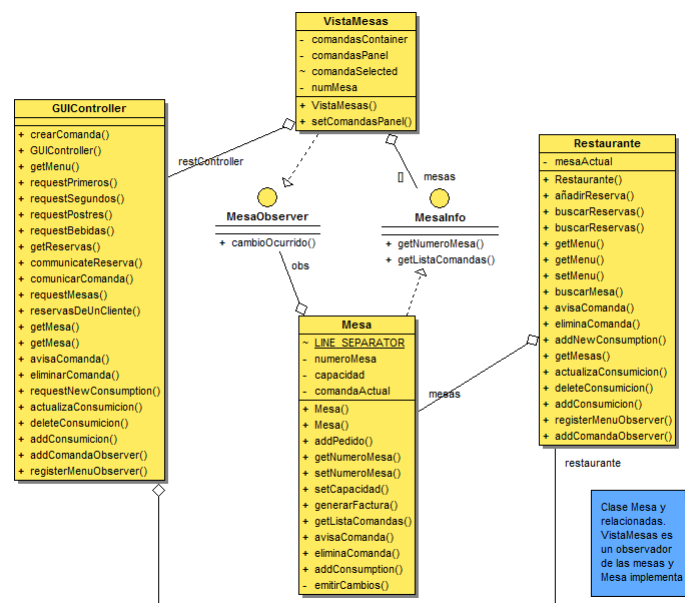


Figura 37: Diagrama de clases de diseño de las mesas

## 2.4. Factura del restaurante

GenerarFactura lleva al controlador y a la mesa de que la que se requiera la factura. Llama a factura pasándole las comandas y se recogen las consumiciones en una única comanda.

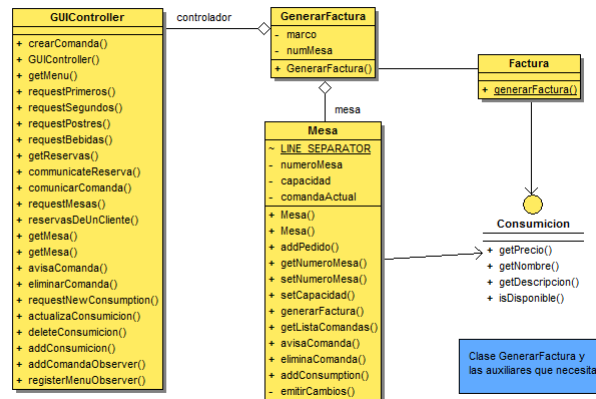


Figura 38: Diagrama de clases de diseño de generar facturas del restaurante

## 2.5. Cargadores de ficheros

Para la gestión de datos, se han escogido ficheros, como ya se dijo. Se tiene una clase padre general de la que heredan los tres cargadores distintos: el de mesas, el del menú y el de reservas.

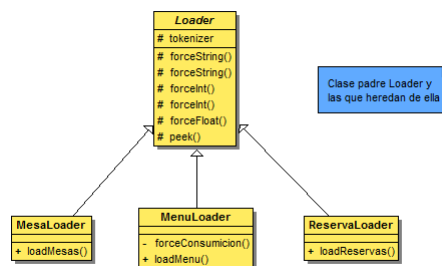


Figura 39: Diagrama de clases de diseño de los cargadores de ficheros

## 3. Diagramas de componentes

No se consideran necesarios.

## 4. Diagramas de despliegue

No se consideran necesarios puesto que la aplicación no tiene la complejidad necesaria.

Parte V

Patrones de diseño

Para implementar la aplicación KIKE HOSTELERIA ® hemos utilizado el patrón MVC. Con él hemos conseguido separar la lógica de la vista, utilizando para la interacción de ambos el Controller.

El Controller se encarga de transmitir las notificaciones recibidas por la Vista, para que el Modelo pueda trabajar con ellas. A su vez, los cambios en el Modelo se notifican a la Vista a través del patrón Observer, es decir, la Vista se registra en el Modelo como observador a través del Controlador, y cuando el Modelo sufre un cambio avisa a los observadores. Para que el modelo no tenga que estar distinguiendo entre qué tipos de observadores tiene, se hace que la Vista implemente una interfaz con los métodos de notificación recibida, mejorando la cohesión.

Para mejorar el bajo acoplamiento se ha utilizado el patrón Indirección. Por ejemplo, cuando una clase de la Vista necesitaba cierta información del Modelo, se delegaba en el Controlador la responsabilidad de transmitir los correspondientes datos. Una buena manera de ver la delegación de actividades es mediante los diagramas de secuencia antes comentados. Por ejemplo, en la figura 29, se ve como se va delegando la generación de la factura desde RestauranteSwing hasta Factura. Así se reduce el acoplamiento.

Por último se han utilizado las variaciones protegidas. Cuando una clase del Modelo requiere ser utilizada por una clase de la Vista, se hace que la clase del Modelo implemente una interfaz y la Vista trabaje con la interfaz. De esta manera conseguimos proteger la vista del modelo.

