



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1

Ingeniería de Software I (2do cuatrimestre de 2008)

Grupo 5

Integrante	LU	Correo electrónico
Gonzalez, Emiliano	426/06	xjesse_jamesx@hotmail.com
Gonzalez, Sergio	481/06	seges.ar@gmail.com
Martínez, Federico	17/06	federicoemartinez@gmail.com
Sainz-Trápaga, Gonzalo	454/06	gonzalo@sainztrapaga.com.ar

Se presenta en este trabajo una especificación completa de la solución propuesta para el proyecto de software de administración de pizzería. En el mismo se presenta un panorama general así como un análisis detallado del problema, y nuestra propuesta para su resolución. En primer lugar se plantea una descripción general de la solución, y a continuación se detallan algunos aspectos importantes haciendo uso de herramientas desarrolladas en clase como diagramas de actividad, máquinas de estado finito y otras.



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice general

1. Introducción	1
1.1. Objetivo del documento	1
1.2. Convenciones de notación	1
1.3. Destinatarios del documento	1
1.4. Descripción del problema	2
1.5. Documentos relacionados	2
1.6. Organización del informe	2
2. Descripción General	3
2.1. Perspectiva del producto	3
2.2. Funciones principales del producto	3
2.3. Características de los usuarios	4
2.4. Restricciones	4
3. Consideraciones globales de la solución	5
3.1. Contexto del sistema	5
3.2. Objetivos de la solución y requerimientos del sistema	7
3.3. Casos de Uso	20
3.3.1. Diagrama de Casos de Uso	20
3.3.2. Descripción de Casos de Uso	21
3.3.3. Matriz de trazabilidad	36
3.4. Conceptos del Sistema	38
3.4.1. Diagrama de modelo conceptual	38
3.4.2. Restricciones al modelo conceptual	40
4. Aspectos particulares de la solución	46

4.1. Ciclo de Pedido	46
4.2. Ingreso de pedidos	49
4.2.1. Pedidos Remotos	51
4.2.2. Pedidos locales	57
4.3. Registro de usuarios	59
4.4. Facturación	59
4.5. Cancelación	60
4.6. Cola de pedidos	60
4.7. Cola de Horno	63
4.8. Consulta de estado de pedidos	66
4.8.1. Estimación de Tiempos	66
4.9. Delivery	67
4.10. Alternativa de Contingencia	68
4.10.1. Contexto del funcionamiento de contingencia	69
4.10.2. Casos de Uso de contingencia	71
4.11. Información estadística	72
4.12. ABM de stock	73
5. Apéndice I: Posibilidades de mejora	75
5.1. Seguimiento de módulos del horno	75
5.2. Seguimiento de tiempos por estado	75
5.3. Preparación adelantada	76
5.4. Carga diferida de pedidos	76
5.5. Seguimiento individual de cada producto	76
6. Apéndice II: Glosario	77
6.1. Terminología	77
7. Apéndice III: Conclusiones	79
7.0.1. Dificultades en la realización del trabajo	79

Parte 1

Introducción

1.1. Objetivo del documento

Este documento tiene varios propósitos. Su contenido abarca esencialmente la descripción del problema que debe resolverse, y una enumeración detallada de los componentes de la solución propuesta para el mismo. Con esta información, se pretende lograr varios objetivos.

En primer lugar, este documento permite validar que el problema a resolver ha sido comprendido correctamente y que no hubieron omisiones en su descripción. Al ser este documento autocontenido, puede asumirse que el conocimiento completo de los analistas sobre el problema ha sido volcado aquí. Toda parte faltante debería ser discutida para su inclusión. Del mismo modo, se espera que el sistema aquí descrito responda correctamente a las necesidades del cliente.

En segunda instancia este documento sirve de base para todo el desarrollo posterior, ya que debe estipular las necesidades a cubrir y las soluciones propuestas de forma de dar una visión general. Es de esperarse que todos los documentos del proyecto involucren en mayor o menor medida a éste, y en ese sentido se trata de la base para todos los documentos más refinados que podrían necesitarse a lo largo del proyecto. Por esta razón, permite también a usuarios, desarrolladores y otras personas ajenas al sistema que deban internalizarse con su funcionamiento conocer los datos esenciales que refieren al mismo, a modo de introducción para su participación en el proyecto.

Por último, este documento es de utilidad para otros agentes asociados al desarrollo del proyecto, ya que permite que el software sea testeado para corroborar que sus funcionalidades concuerdan con lo aquí planteado, y que responsables de desarrollo puedan estimar tiempos y costos del mismo gracias a la descripción de los componentes a construir.

1.2. Convenciones de notación

En este documento se utilizan las siguientes abreviaciones:

- **CU** es la abreviatura de Caso de Uso
- **FSM** es la abreviatura de *Finite State Machine* (máquina de estado finito)
- **DA** es la abreviatura de Diagrama de Actividad

1.3. Destinatarios del documento

El presente documento está dirigido en primer lugar al dueño y los empleados de la pizzería, con el objetivo de que lo vean y examinen la dinámica de uso del sistema para validar que las funcionalidades propuestas son correctas. A su vez, estas personas son idóneas para sugerir mejoras o modificaciones a la operatoria propuesta, ya que son las que en el futuro deberán interactuar fluidamente con el software.

En segunda instancia, también son destinatarios del documento aquellas personas que supervisarán el desarrollo del software por parte del cliente que lo encargó, en particular el área de Calidad de Software designado por éste, que utilizará este documento como referencia de comportamiento del software a los efectos de realizar pruebas de funcionamiento y desempeño.

1.4. Descripción del problema

El problema a abordar es la introducción de mejoras en la administración de una pizzería. En particular, se deberán resolver algunos problemas que con el mecanismo actual de organización son recurrentes y representan una dificultad importante en el día a día del negocio. Por otro lado, el sistema deberá mejorar todo el proceso de gestión y manejo de pedidos, con el objeto de acrecentar su rendimiento mediante el mejor uso de los recursos tanto físicos como humanos. Finalmente, el sistema debe tener en cuenta el proyecto de expansión de la pizzería mediante la generación de nuevas formas de pedido y cobro, así como la incorporación de un servicio de *delivery*. Es de crucial importancia que el sistema utilice su capacidad de registro para realizar estadísticas que permitan conocer el rendimiento y las limitaciones en la producción de la pizzería.

Contamos entonces cuatro objetivos principales:

- Resolver los problemas de operación actuales (evitar que se pierdan pedidos, evitar tomar pedidos que no pueden ser satisfechos)
- Mejorar la administración de la pizzería (gestión del stock, planeamiento eficaz del uso de hornos y cocina, informatización de las comunicaciones entre los participantes del proceso)
- Ampliar las formas de pedido y pago, permitiendo ingresar pedidos por teléfono, Web o SMS
- Realizar análisis estadísticos en base a la información registrada

1.5. Documentos relacionados

El principal documento a tener en cuenta es *Enunciado TP V1*, donde se detallan los requerimientos e intenciones de los clientes a los efectos del sistema. Como no se hicieron reuniones formales para discutir la funcionalidad del sistema, no se dispone de minutas de reuniones ni ninguna otra fuente primaria de información. La gran mayoría de las cuestiones que conciernen a la funcionalidad del sistema fueron explicadas en clase o discutidas por correo electrónico con los docentes de la materia.

1.6. Organización del informe

El presente documento se organiza de la siguiente manera:

1. **Introducción:** se presentan los objetivos del documento, así como una breve descripción del problema a resolver y del alcance de este documento.
2. **Descripción General:** se presenta una descripción general de la solución, su alcance y perspectiva, al tiempo que se presentan las características de los usuarios, limitaciones, supuestos y dependencias que pudieran afectar al software a desarrollar.
3. **Consideraciones globales de la solución:** se describe la solución desde un punto de vista global. En este apartado se encuentran los requerimientos del sistema, los principales casos de uso, conceptos y contexto de funcionamiento del software.
4. **Aspectos particulares de la solución:** se analizan los distintos aspectos de la solución desde una óptica mas específica a cada componente de la solución. En este inciso se detallan cuestiones tales como el ingreso de pedidos, el ciclo de vida de un pedido y la administración de los hornos, entre otros.
5. **Apéndice I - Extensiones:** se detallan posibles extensiones y mejoras al software descrito previamente, que si bien están por fuera de la solución propuesta podrían ser de interés para futuras revisiones del producto.
6. **Apéndice II - Glosario:** se describe la terminología utilizada en todo el documento para referirse a agentes, componentes y otros términos relevantes que aparecen en este documento.

Parte 2

Descripción General

Dado que uno de los aspectos principales del problema es la organización de los pedidos, es una parte fundamental del producto controlar el estado de los mismos, permitir su ingreso controlado y brindar un seguimiento que permita conocer qué pasa con los mismos. Por otro lado, se espera que el software sea capaz de interactuar con diversos tipos de clientes a fin de lograr variadas formas de ingreso de pedidos.

El sistema deberá además ser capaz de brindar estadísticas que permitan estudiar el funcionamiento del negocio, así como también de interactuar con diversos sistemas externos para lograr estos objetivos. En particular, son de interés el sistema de facturación del negocio y un sistema de pagos por tarjeta de crédito.

2.1. Perspectiva del producto

El producto es autocontenido en lo referente a la gestión de los pedidos, así como también en el manejo del stock de productos e insumos. Sin embargo el software no se encarga de la facturación, ya que la misma es responsabilidad del sistema de facturación que está actualmente instalado en la pizzería.

2.2. Funciones principales del producto

El sistema se encarga de monitorear el ciclo de vida de los pedidos que se realizan a la pizzería y el stock de las materias primas. Es responsable también de la optimización del uso del horno, la estimación de lapsos de producción y entrega, y por último de conservar un registro de los eventos de interés con fines estadísticos.

En el caso de los pedidos realizados remotamente (vía Web o SMS), el sistema se encarga también de su ingreso. Para los pedidos realizados de forma directa (ya sea a un mesero o llamando por teléfono) existirá un individuo responsable de su ingreso.

Desde el momento del ingreso de los pedidos, el sistema verifica su factibilidad en función del stock de insumos. Una vez ingresados los pedidos, el sistema actualiza la información de stock inmediatamente (y atómicamente junto con la inserción del pedido). A continuación se inserta el pedido en la cola de pedidos y se presenta a quien ingresó el pedido (usuario o encargado) una estimación de la demora en la preparación del pedido.

Dentro de la cocina, los maestros pizzero y empanadero tienen acceso al sistema en el cual pueden observar los pedidos que deben preparar e indicar su estado de preparación (en espera, preparado, en el horno). Tras la preparación, si el pedido debe ser entregado el sistema continúa monitoreando su estado y la demora en la entrega, tras la cual se recibe una notificación por parte del delivery. El sistema envía los datos de los pedidos que deben ser facturados al sistema de facturación mediante un archivo en un formato especificado que contiene información tal como la cantidad de items, su precio y la forma de pago.

En todo momento el encargado de pedidos puede reordenar la cola de pedidos, o cancelar alguno de ellos, así como los usuarios pueden consultar el estado de los mismos (vía Web). Un segundo encargado responsable de stock (que puede o no ser la misma persona física que el encargado de pedidos) puede actualizar el stock de insumos.

El sistema no es responsable de la logística de la distribución ni de la interacción directa con los clientes que no utilizan los medios Web o SMS. Tampoco existen operaciones automatizadas que afecten al negocio: todo lo que el sistema hace es coordinar a los actores que participan. El proceso de preparación y cocción de los alimentos es totalmente manual, así como lo es la reposición del stock y el armado de los pedidos para su posterior entrega.

2.3. Características de los usuarios

Dadas las características mencionadas anteriormente sobre el producto, resulta imprescindible que los usuarios que interactúan con el sistema tengan conocimiento de como utilizarlo.

Es el caso de los empleados del local, los mozos deben saber como utilizar las computadoras de mano que le serán dadas para ingresar nuevos pedidos al sistema. En el caso de los maestros dentro de la cocina (ya sea el maestro empanadero, o el maestro pizzero) es necesario que sepan como pedirle al sistema el próximo pedido a preparar, y además como realizar los diferentes cambios de estado por los que pasa un pedido (desde que se comienza a preparar, hasta que ya se tiene el pedido listo para entregar) utilizando la pantalla de tipo touchscreen. Por último, el encargado de pedidos, que es el encargado de ingresar todos los pedidos que no sean remotos, cancelarlos, administrar las asignaciones a los hornos, y con la posibilidad de realizar consultas estadísticas, debe estar capacitado para poder desempeñarse sin tener inconvenientes.

Si bien no se requieren empleados con conocimientos técnicos, si se espera que estos sepan interactuar con el sistema de manera fluida. Por esta razón podría ser necesario un periodo de capacitación para que los distintos actores aprendan como utilizar correctamente el sistema.

Con respecto a los clientes, aquellos que quieran realizar pedidos via WEB deberán saber como desempeñarse en internet, es decir tener mínimos conocimientos del uso de un navegador de internet y como ingresar a las distintas paginas. Los usuarios que quieran utilizar el servicio, haciendo un pedido mediante mensajes de tipo SMS deberán saber como utilizar un teléfono celular para enviar los mensajes, además saber cual es el número al cual deben enviarlos para que el sistema los reciba. Aquellos usuarios que utilicen el teléfono, deberán saber como realizar una llamada y cual es el número telefónico del local.

2.4. Restricciones

Dado que a nivel presupuestario no se estipularon restricciones, en principio la solución propuesta no cuenta con limitaciones a nivel de hardware. Sin embargo, no hay razón para suponer que el sistema requiera de un poder de cómputo extraordinario, y por lo tanto es de esperarse que sea suficiente contar con computadoras actuales de potencia media. Los mozos deberán contar con una PDA cada uno, y los maestros deberán interactuar con el sistema gracias a una terminal de pantalla táctil, puesto que el uso de un teclado en la cocina puede traer problemas sanitarios. Por último, el encargado de pedidos deberían contar con una computadora convencional pero sería bueno que disponga de un monitor de buen tamaño ya que debe poder monitorear mucha información.

Respecto al lenguaje o plataforma de desarrollo, no existen restricciones tampoco más que el sistema utilizado sea capaz de utilizar el hardware previamente descrito.

Parte 3

Consideraciones globales de la solución

El objetivo de esta sección es evaluar el problema en su totalidad y describir la solución propuesta desde un punto de vista holístico. En la sección siguiente se detallarán de forma más precisa varios aspectos técnicamente importantes de lo aquí descrito. Sin embargo, optamos por realizar esta descripción de lo general a lo particular para ofrecer una mayor claridad en la explicación sin resignar detalle en los puntos importantes.

3.1. Contexto del sistema

A continuación presentamos el contexto y las interacciones que realiza el sistema con el mundo exterior mediante el diagrama de contexto de nuestra solución.

El objetivo de este diagrama es modelar la interfaz del sistema con sus usuarios y con los demás agentes involucrados en su funcionamiento. Las flechas están coloreadas del mismo color que los agentes responsables de controlar los eventos representados por esas flechas, para mayor claridad en la lectura del diagrama.

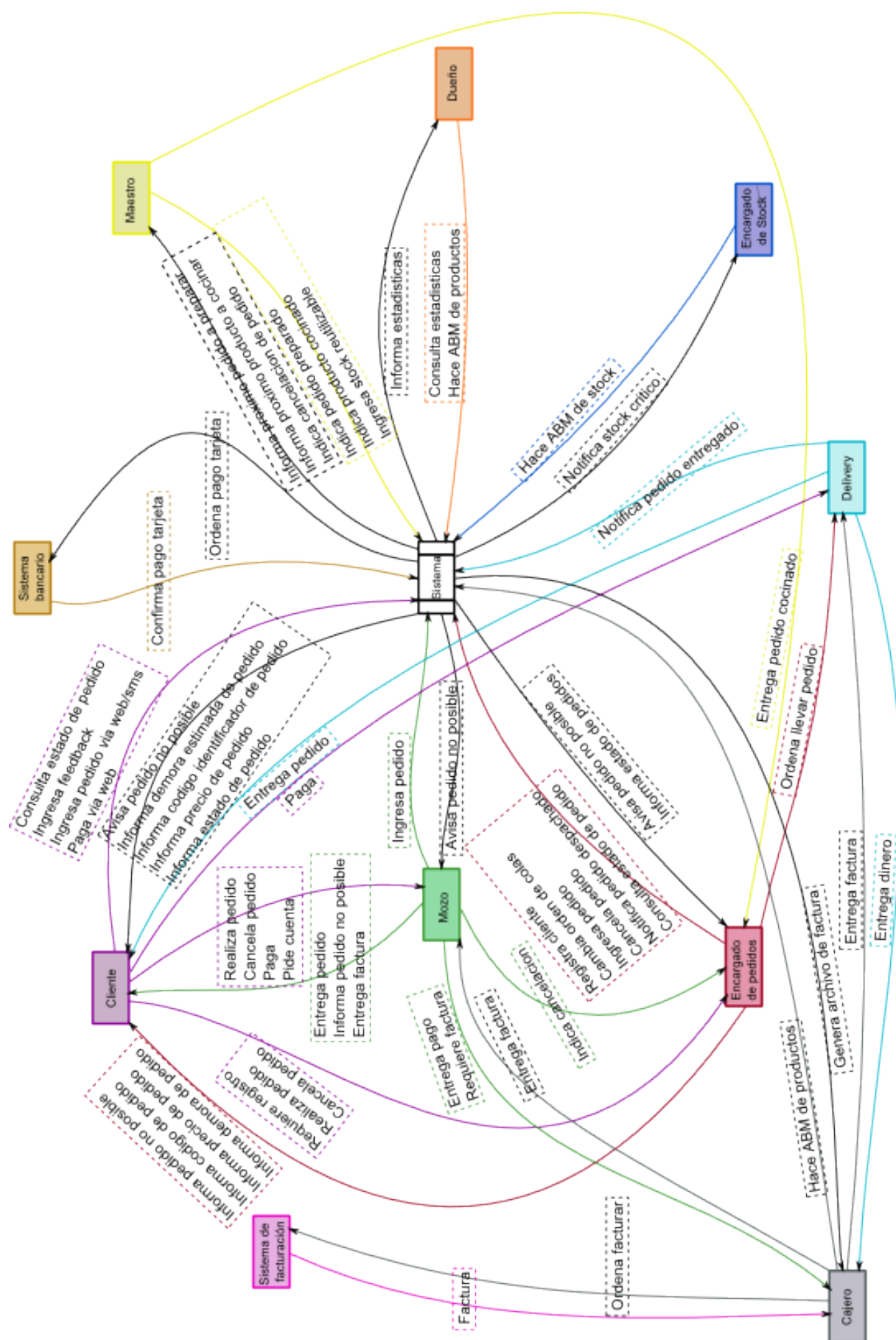
Rápidamente puede verse que los agentes que más trascendencia tienen son el encargado de pedidos y el cliente. Desde el punto de vista del sistema, estos dos agentes tienen una gran participación. El encargado de pedidos se encarga de toda la coordinación de la producción y de la priorización de pedidos antes de su entrada a la cocina. Los clientes, por su parte, en razón de que pueden autogestionarse ingresando pedidos a través de la Web o de SMS, necesitan también una interfaz rica en funcionalidades.

Otros agentes de importancia son el Mozo, el Cajero, el Delivery y los Maestros, que también tienen interacción directa con el sistema y participan activamente en toda la dinámica de la pizzería, incluyendo también muchas actividades que no pasan por el sistema pero que son esenciales para su funcionamiento.

Por último tenemos agentes que son de menor jerarquía por la cantidad de operaciones que realizan. Podemos contar entre estos al encargado de stock, al dueño y a los sistemas bancario y de facturación, que tienen roles más tangenciales y solo interactúan con el sistema de forma más esporádica.

En razón de estas observaciones, será razonable destinar un tiempo acorde a la importancia de cada agente a pulir las interfases que lo conciernen. Esto es particularmente crítico para el caso de los clientes, en que disponer de una buena interfaz amigable y práctica en lugar de una mala puede ser la diferencia entre realizar o no una venta.

Por último cabe aclarar que este diagrama representa el modo de operación normal. Existe un modo de operatoria de contingencia donde las interacciones se ven modificadas. Esta alternativa de contingencia se discute en la sección 4.10, donde se incluye un segundo diagrama de contexto representativo de ese escenario.



3.2. Objetivos de la solución y requerimientos del sistema

A continuación se presenta el diagrama de objetivos de la solución, junto con un detalle de los requerimientos del sistema que corresponden a los objetivos de bajo nivel. Los requerimientos del sistema están marcados en el diagrama en color verde, y sus agentes asociados aparecen en color naranja para facilitar su diferenciación.

A raíz de su tamaño, el grafo de objetivos se presenta particionado en 5 fragmentos, cada uno seguido de los requerimientos que aparecen en él. En primer lugar se muestra la parte superior del diagrama, donde se observa únicamente el objetivo de primer nivel: mejorar el rendimiento de la pizzería para aumentar las ganancias. Éste se bifurca en 5 objetivos de segundo nivel según lo elicitado a partir del documento presentado por el cliente:

- Mantener el control del stock
- Generar promociones efectivas
- Lograr información estadística de funcionamiento
- Lograr un buen rendimiento de la cocina
- Lograr mejor atención al cliente

Cada uno de estos objetivos se refinó progresivamente hasta lograr discernir en el diagrama los requerimientos sobre el sistema a construir. Al momento de refinar, se excluyeron explícitamente refinamientos que no resultaban determinantes para el sistema, así como muchas expectativas sobre agentes que se considera que no aportan a la especificación y complican la legibilidad. Se puso especial atención a que el diagrama enfatice los problema específicos planteados por los clientes, como ser la toma de pedidos que no pueden ser satisfechos o evitar quedarse sin stock de algún insumo.

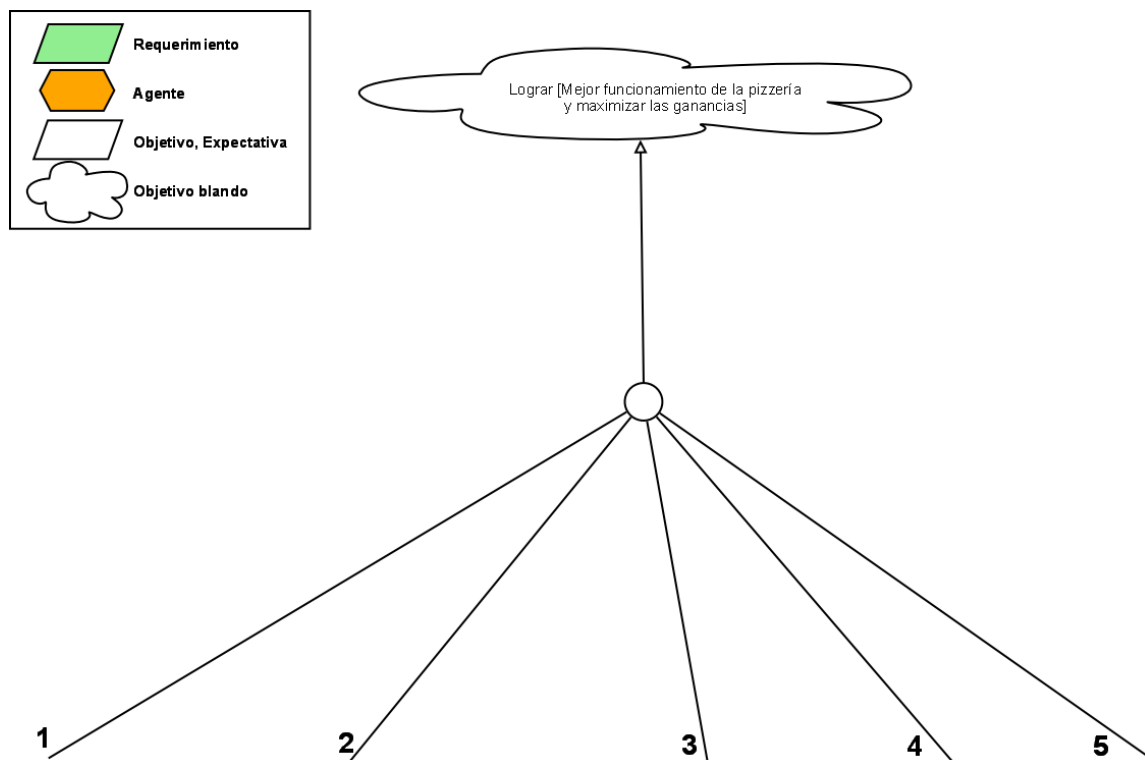
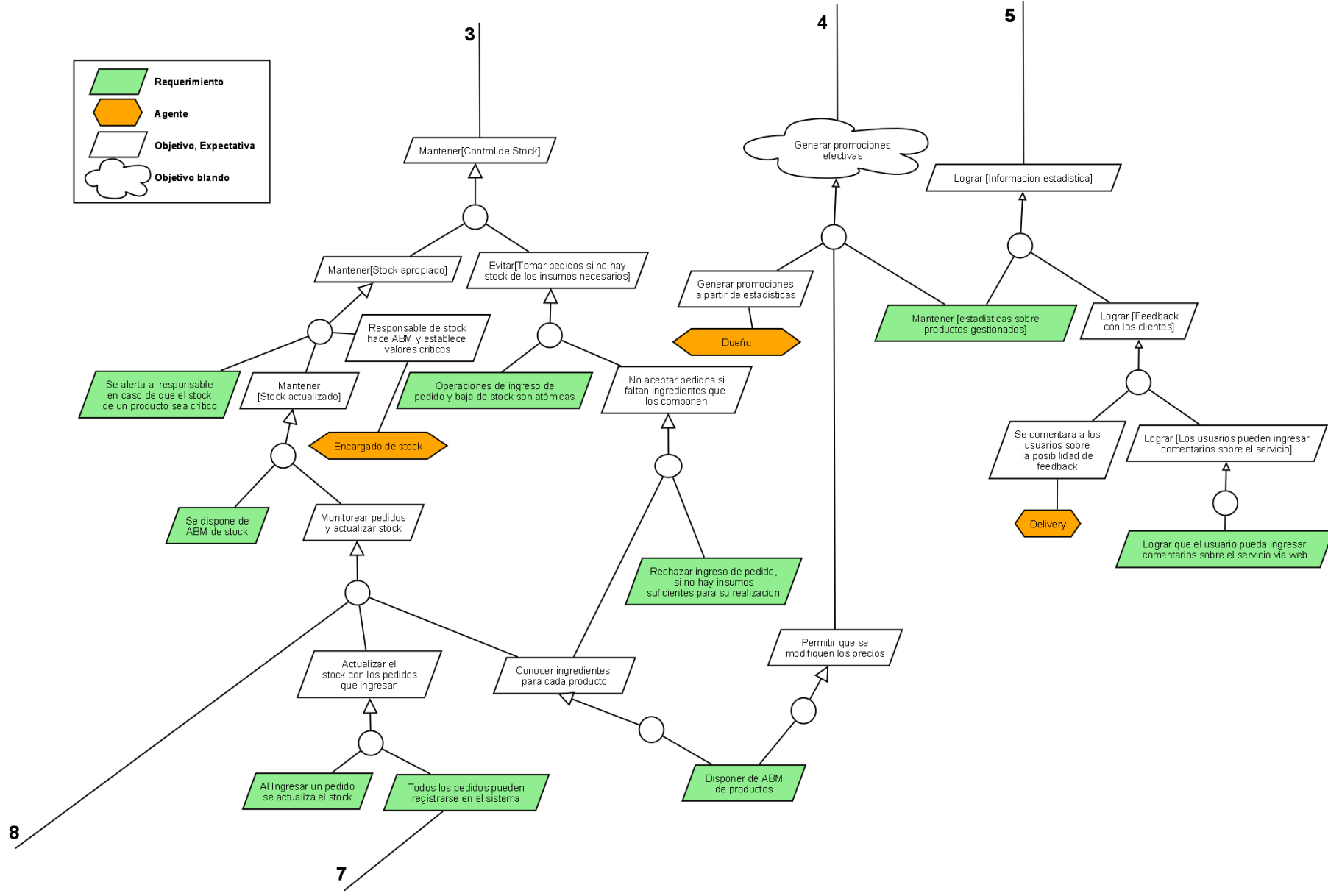


Figura 3.1: Raíz del árbol de objetivos del sistema



Requerimiento:	Se alerta al responsable en caso de que el stock de un producto sea crítico
Numero:	1
Tipo:	Funcional
Importancia:	Esencial
Descripción:	Cuando el stock de un producto queda por debajo de un límite preestablecido, se debe generar un mensaje dirigido al responsable de stock a fin de notificar esta situación.
Motivo:	Es una funcionalidad pedida explícitamente, que ayuda a mantener un stock apropiado de los distintos productos y evitar así la falta de insumos para realizar pedidos.

Requerimiento:	Se dispone de ABM de stock
Numero:	2
Tipo:	Funcional
Importancia:	Importante
Descripción:	El sistema debe permitir que las altas, bajas y modificaciones de productos se registren en el sistema, para de esta manera controlar el stock de las materias primas y bebidas.
Motivo:	Mantener el stock de insumos actualizados, lo que contribuye a tener un control de stock apropiado.

Requerimiento:	Al ingresar pedido se actualiza el stock
Numero:	3
Tipo:	Funcional
Importancia:	Importante
Descripción:	El sistema decrementará el stock de materias primas o bebidas necesarias para un pedido al momento en que este ingresa al sistema.
Motivo:	Permite que el ingreso de pedidos modifique el stock consistentemente y de esta manera tener un stock actualizado.

Requerimiento:	Todos los pedidos pueden registrarse en el sistema
Numero:	4
Tipo:	Funcional
Importancia:	Esencial
Descripción:	Los pedidos realizados pueden ingresarse al sistema y registrarse.
Motivo:	Este requerimiento ayuda controlar el stock de insumos y conocer el estado de los pedidos.

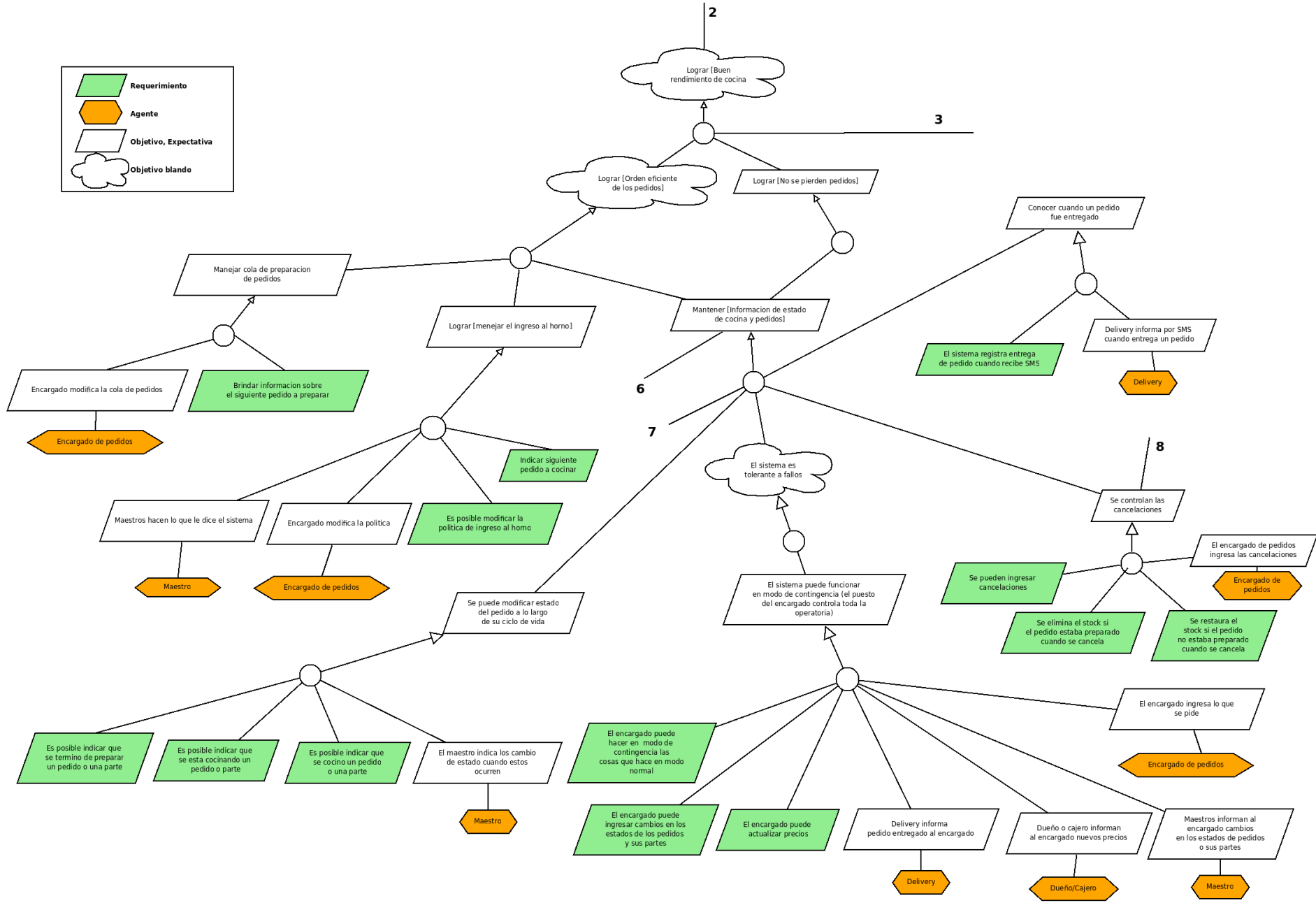
Requerimiento:	Operaciones de ingreso de pedido y baja de stock son atómicas
Numero:	5
Tipo:	No funcional
Importancia:	Importante
Descripción:	El ingreso de un pedido y la disminución del stock de sus materias primas son indisociables. Esto garantiza la consistencia del sistema de stock asegurando así que los recursos necesarios para la realización de un pedido estén efectivamente disponibles al momento de su ingreso.
Motivo:	Evitar tomar pedidos si no se dispone de los recursos para satisfacerlos.

Requerimiento:	Rechazar ingreso de pedido si no hay insumos suficientes para su realización
Numero:	6
Tipo:	Funcional
Importancia:	Esencial
Descripción:	Si cuando se realiza un pedido no se dispone de los recursos necesarios para completarlo, el mismo no debe ingresar al sistema y se debe notificar a quien esté intentando realizar el pedido de dicha situación
Motivo:	Evitar tomar pedidos si no se dispone de los recursos para satisfacerlos.

Requerimiento:	Disponer de ABM de productos
Numero:	7
Tipo:	Funcional
Importancia:	Importante
Descripción:	Se debe proveer de una interfaz para registrar nuevos productos al menú (por ejemplo, nuevas variedades de pizza) así como también modificar precios de productos ya registrados o sacarlos del menú.
Motivo:	Permitir modificaciones al menú y a la lista de precios.

Requerimiento:	Mantener estadísticas sobre productos gestionados
Numero:	8
Tipo:	Funcional
Importancia:	Esencial
Descripción:	El sistema debe ser capaz de retener la información sobre los productos que se gestionan para poder realizar estadísticas sobre los pedidos, y de esta forma conocer más el funcionamiento de la pizzería.
Motivo:	Se pide de forma explícita tener información estadística sobre el funcionamiento de la pizzería, los pedidos, las cancelaciones, y productos más vendidos para hacer promociones.

Requerimiento:	Lograr que el usuario pueda ingresar comentarios sobre el servicio vía web
Numero:	9
Tipo:	Funcional
Importancia:	Deseable
Descripción:	Como aporte original proponemos que el usuario pueda ingresar comentarios sobre el servicio luego de recibir un pedido. El sistema debe ser capaz de registrar esos comentarios vía Web.
Motivo:	Este requerimiento ayuda a generar información estadística, con el objetivo de mejorar el servicio.



Requerimiento:	Es posible indicar que se terminó de preparar un pedido o una parte
Numero:	10
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando se prepara un pedido (o una parte de él), debe ser posible indicar al sistema la tarea que se ha realizado.
Motivo:	Permite mantener actualizado el estado del pedido, además de contribuir a la funcionalidad de poder cambiar el estado del mismo a lo largo de su ciclo de vida.

Requerimiento:	Es posible indicar que se está cocinando un pedido o una parte
Numero:	11
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando un pedido está preparado, el maestro puede comenzar a cocinar el pedido completo o una parte de él. En cualquiera de los dos casos se debe poder indicar al sistema que ese pedido o parte del mismo se comenzó a cocinar.
Motivo:	Ayuda a poder modificar el estado del pedido a lo largo de su ciclo de vida, además de controlar el estado del mismo.

Requerimiento:	Es posible indicar que se cocinó un pedido o una parte
Numero:	12
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando el maestro termina de cocinar un pedido o una parte de él, debe poder indicarlo al sistema.
Motivo:	Ayuda a poder modificar el estado del pedido a lo largo de su ciclo de vida y controlar el estado del mismo.

Requerimiento:	Brindar información sobre el siguiente pedido a preparar
Numero:	13
Tipo:	Funcional
Importancia:	Importante
Descripción:	El sistema administra los pedidos que se encuentran esperando a ser preparados en una cola de preparación de los mismos, de esta se forma indica que pedido es el que sigue y por ende, cual es el que se tiene que preparar.
Motivo:	Permite manejar la cola de pedidos que se encuentran esperando para ser preparados, y además contribuye a tener un orden eficiente de los pedidos que deben ingresar a la cocina.

Requerimiento:	Es posible modificar la política de ingreso al horno
Numero:	14
Tipo:	Funcional
Importancia:	Importante
Descripción:	El encargado de pedidos puede realizar una modificación de la política que administra los pedidos que requieren ingresar al horno para ser cocinados.
Motivo:	Permite tener un mejor control de los pedidos que ingresarán al horno, y contribuir así a tener un buen rendimiento en la cocina.

Requerimiento:	Indicar siguiente pedido a cocinar
Numero:	15
Tipo:	Funcional
Importancia:	Importante
Descripción:	El sistema le indica al maestro que es lo que se debe cocinar a continuación cuando éste tenga la oportunidad de hacerlo.
Motivo:	Este requerimiento contribuye al manejo del ingreso de los pedidos al horno, con el fin de tener un buen rendimiento de la cocina.

Requerimiento:	El encargado puede hacer en contingencia las cosas que hace en modo normal
Numero:	16
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando el sistema se encuentra en modo de contingencia, el encargado de pedidos va a poder continuar haciendo las funciones que tiene asignadas en modo normal.
Motivo:	Este requerimiento aporta al funcionamiento del sistema en modo de contingencia permitiendo que el mismo sea tolerante a fallas.

Requerimiento:	El encargado puede ingresar cambios en los estados de los pedidos y sus partes
Numero:	17
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando el sistema se encuentra en modo de contingencia, es el encargado de pedidos el responsable de realizar las funcionalidades no disponibles para los actores que normalmente son responsables. En este caso, el encargado de pedidos deberá poder realizar los cambios de estado de los diferentes pedidos y sus respectivas partes.
Motivo:	Este requerimiento ayuda a que no se pierda la funcionalidad del sistema en modo de contingencia, ayudando a que el mismo sea tolerante a fallas.

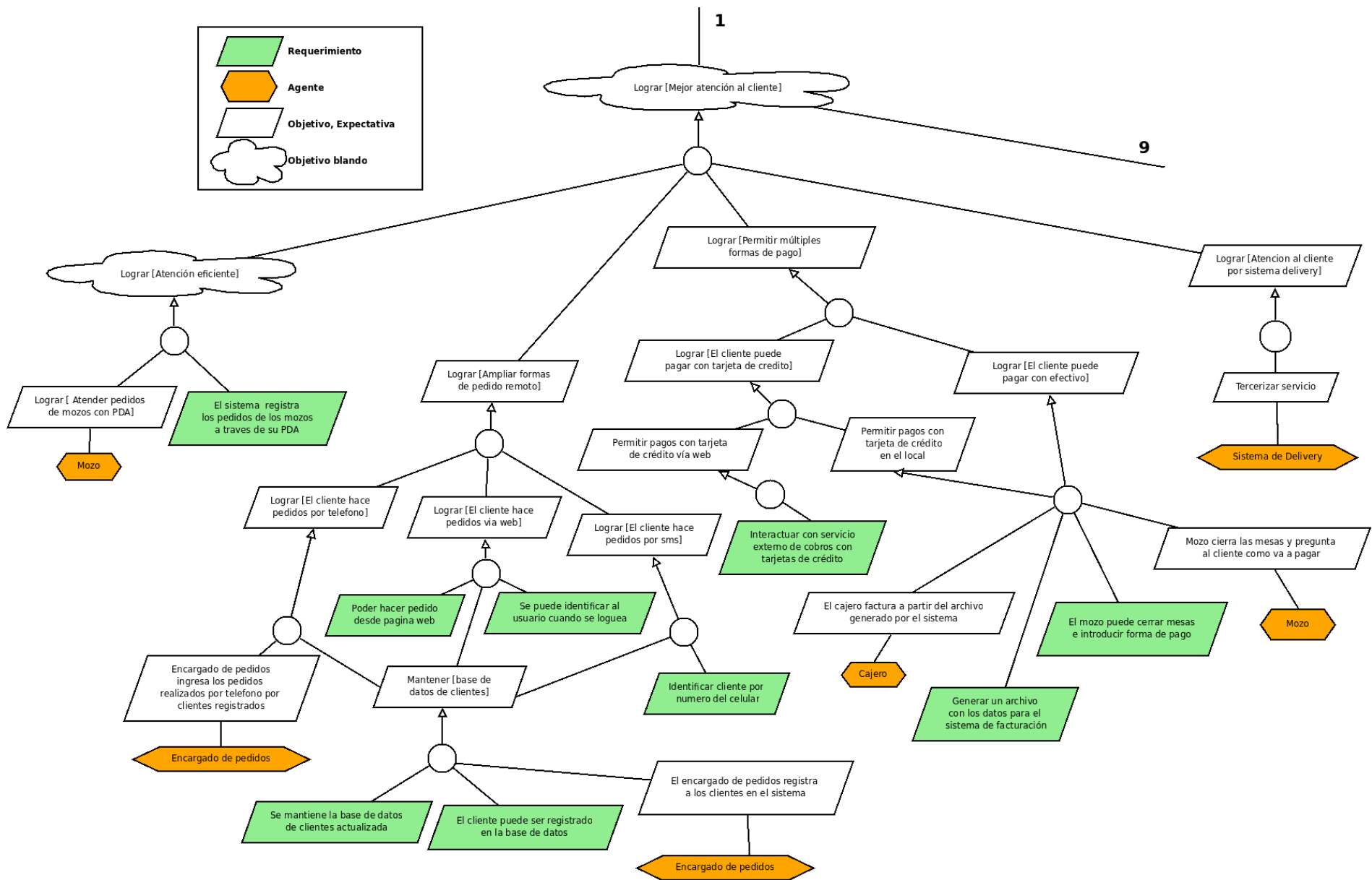
Requerimiento:	El encargado puede actualizar precios
Numero:	18
Tipo:	Funcional
Importancia:	Importante
Descripción:	El encargado puede realizar cambios en los precios de los productos cuando el sistema se encuentra en modo de contingencia.
Motivo:	Ayuda a que la funcionalidad del sistema no se pierda en modo de contingencia, lo que contribuye a que éste sea tolerante a fallas.

Requerimiento:	Se pueden ingresar cancelaciones
Numero:	19
Tipo:	Funcional
Importancia:	Importante
Descripción:	El sistema debe permitir que un pedido pueda ser cancelado, sin importar el estado en el que se encuentre. La excepción es el estado finalizado, que caracteriza al mismo cuando el mozo llevó el pedido a la mesa, el cliente retiró el pedido del mostrador o cuando el delivery logró entregar el pedido.
Motivo:	Este requerimiento contribuye al control de las cancelaciones, que permite mantener información sobre el estado de los pedidos.

Requerimiento:	Se elimina el stock si el pedido estaba preparado cuando se cancela
Numero:	20
Tipo:	Funcional
Importancia:	Importante
Descripción:	Los productos ya preparados o cocinados de un pedido no se pueden reutilizar, por lo que deben ser descartados. Este requerimiento está relacionado con que este descarte sea automatico, es decir que no se tengan que ingresar las bajas de stock correspondientes de forma manual.
Motivo:	Provee un método para controlar las cancelaciones de pedidos y seguir el stock con facilidad. Como el descarte es automático, ayuda también a llevar el estado del stock.

Requerimiento:	Se restaura el stock si el pedido no estaba preparado cuando se cancela
Numero:	21
Tipo:	Funcional
Importancia:	Importante
Descripción:	Si se cancela un pedido antes de prepararlo, las materias primas que no se usaron deben volver al stock a fin de que puedan ser usadas para preparar un nuevo pedido. Dicha funcionalidad es automática.
Motivo:	Permite controlar las cancelaciones, mejorar la eficiencia de la cocina, y contribuye a llevar el estado del stock.

Requerimiento:	El sistema registra entrega de pedido cuando recibe SMS
Numero:	22
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando el delivery manda un mensaje SMS al sistema con el pedido que se entregó, éste registra al pedido como finalizado.
Motivo:	Este requerimiento contribuye a conocer cuando un pedido fue entregado, para mantener información de estado de los pedidos.



Requerimiento:	El sistema registra los pedidos de los mozos a través de sus PDA
Numero:	23
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando un mozo toma un pedido desde una mesa, este puede ingresarlo al sistema utilizando una computadora de mano. Estos pedidos quedan registrados en el sistema.
Motivo:	Contribuye a lograr una atención eficiente de los clientes locales.

Requerimiento:	Poder hacer pedido desde página web
Numero:	24
Tipo:	Funcional
Importancia:	Importante
Descripción:	El usuario tiene que poder ingresar un pedido desde la interfaz que brinda la página web de la pizzería.
Motivo:	Para ampliar las formas de venta, el cliente debe poder hacer pedidos vía web.

Requerimiento:	Se puede identificar al usuario cuando se loguea
Numero:	25
Tipo:	Funcional
Importancia:	Importante
Descripción:	Es necesario poder identificar al usuario que quiere utilizar algún servicio de la pizzería vía web. Para ello tendrá asignado un usuario y una contraseña para ingresar a su cuenta.
Motivo:	Cada usuario tendrá una cuenta, que utilizará para realizar pedidos y revisar el estado de los mismos. Esto se requiere para poder proveer una mejor atención a los clientes.

Requerimiento:	Se mantiene la base de datos de clientes actualizada
Numero:	26
Tipo:	Funcional
Importancia:	Importante
Descripción:	Una base de datos actualizada es necesaria para poder tener información de los usuarios y permitirles realizar pedidos remotos.
Motivo:	Mantener una base de datos de clientes actualizada permite identificar de forma correcta a cada uno de ellos, y de esta manera registrar correctamente sus pedidos, permitiendo así conocer información estadística sobre la distribución de los pedidos. Esta información puede servir para crear promociones efectivas y mejorar la atención al cliente.

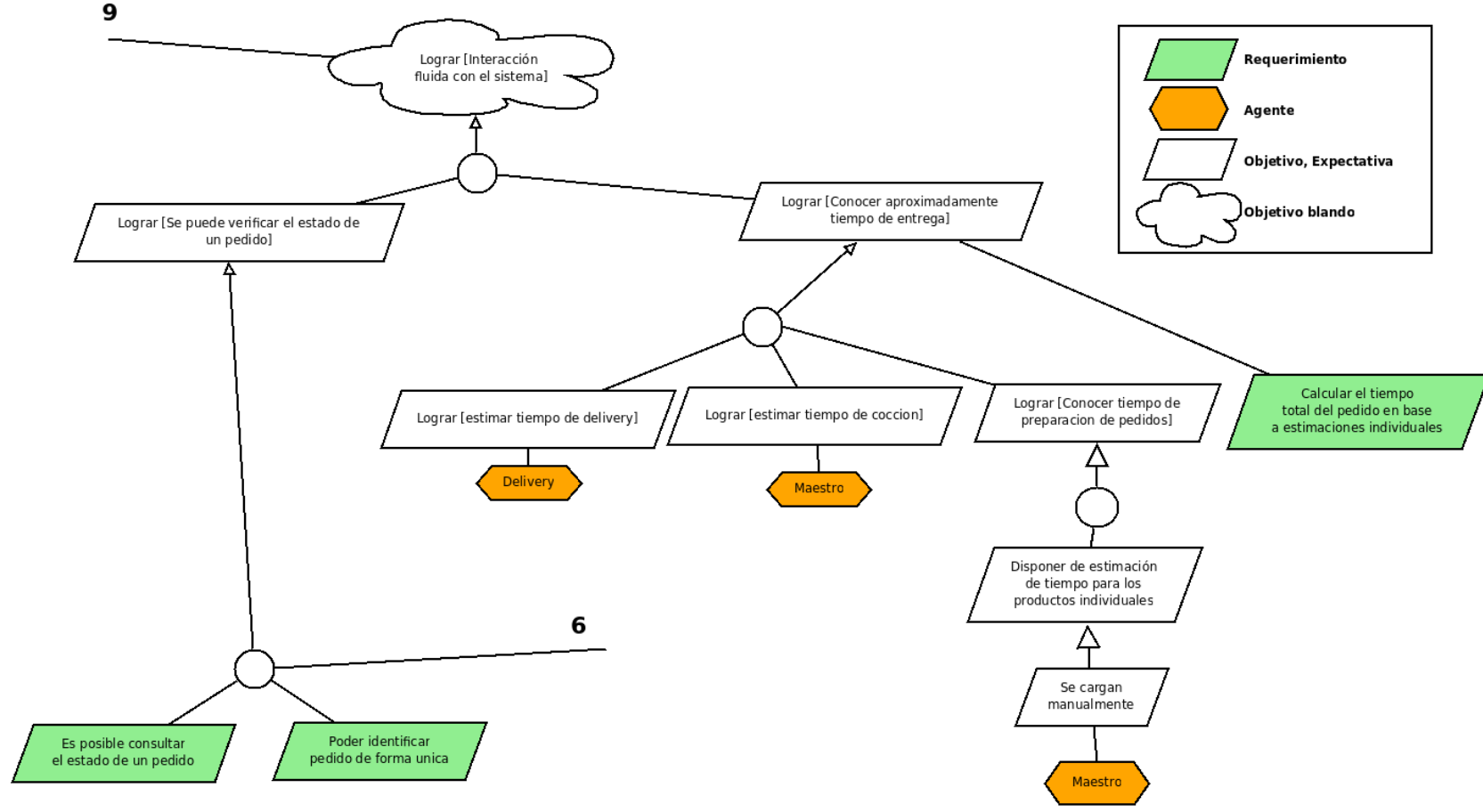
Requerimiento:	El cliente puede ser registrado en la base de datos
Numero:	27
Tipo:	Funcional
Importancia:	Importante
Descripción:	Deberá ser posible que el encargado de pedidos registre nuevos usuarios que se lo piden ya sea personal o telefónicamente. De esta forma accederán a la posibilidad de realizar pedidos remotos.
Motivo:	Ayuda a mantener la base de datos de los clientes actualizada, permitiendo recibir pedidos remotos.

Requerimiento:	Identificar al cliente por su número de celular
Numero:	28
Tipo:	Funcional
Importancia:	Importante
Descripción:	Análogamente al caso Web, para poder hacer un pedido vía SMS es necesario poder identificar el número del celular que envía el mensaje, y que éste corresponda con algún cliente que se encuentre registrado en el sistema.
Motivo:	Contribuye a brindar el servicio de pedidos vía SMS, ampliando las formas de pedido, lo que hace que se mejore la atención a los clientes.

Requerimiento:	Interactuar con servicio externo de cobros con tarjeta de crédito
Numero:	29
Tipo:	Funcional
Importancia:	Importante
Descripción:	El sistema debe ser capaz de interactuar con algun servicio externo tipo PayPal para permitir que un usuario realice el pago de su pedido utilizando una tarjeta de crédito.
Motivo:	Contribuye a tener una mayor variedad formas de pago, dejando que el cliente opte por la opción que más le convenga.

Requerimiento:	Generar un archivo con los datos para el sistema de facturación
Numero:	30
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando se completa un pedido que deberá ser entregado por el delivery (cuando se encuentra en estado Listo, cocinado y con las bebidas), el sistema genera un archivo con los datos necesarios para realizar la factura cuando el cajero lo crea adecuado.
Motivo:	Contribuye a brindar la funcionalidad de pagos en efectivo o con tarjeta de crédito por parte de los clientes, interactuando con el sistema de facturación existente.

Requerimiento:	El mozo puede cerrar mesas e introducir forma de pago
Numero:	31
Tipo:	Funcional
Importancia:	Importante
Descripción:	Cuando se cierra una mesa, el mozo que la atiende debe preguntarle al cliente que forma de pago utilizará e indicarselo al sistema.
Motivo:	Este requerimiento contribuye a permitir diferentes formas de pago.



Requerimiento:	Es posible consultar el estado de un pedido
Numero:	32
Tipo:	Funcional
Importancia:	Importante
Descripción:	Tanto un cliente como el encargado deben tener la posibilidad de consultar el estado de un pedido en cualquier momento. Para el caso de un usuario, lo podrá hacer directamente a través de la Web.
Motivo:	Contribuye a permitir la consulta del estado de un pedido, y a lograr una interacción fluida del sistema con el usuario.

Requerimiento:	Poder identificar un pedido de forma única
Numero:	33
Tipo:	Funcional
Importancia:	Importante
Descripción:	Debe ser posible referenciar a un pedido ingresado al sistema de forma unívoca para fines diversos (consultas de estado, modificaciones, etc)
Motivo:	Teniendo una única forma de identificar un pedido, se puede brindar la opción al cliente de verificar el estado de su pedido de forma remota, lo que mejora la interacción del sistema con el usuario y brinda una mejor atención.

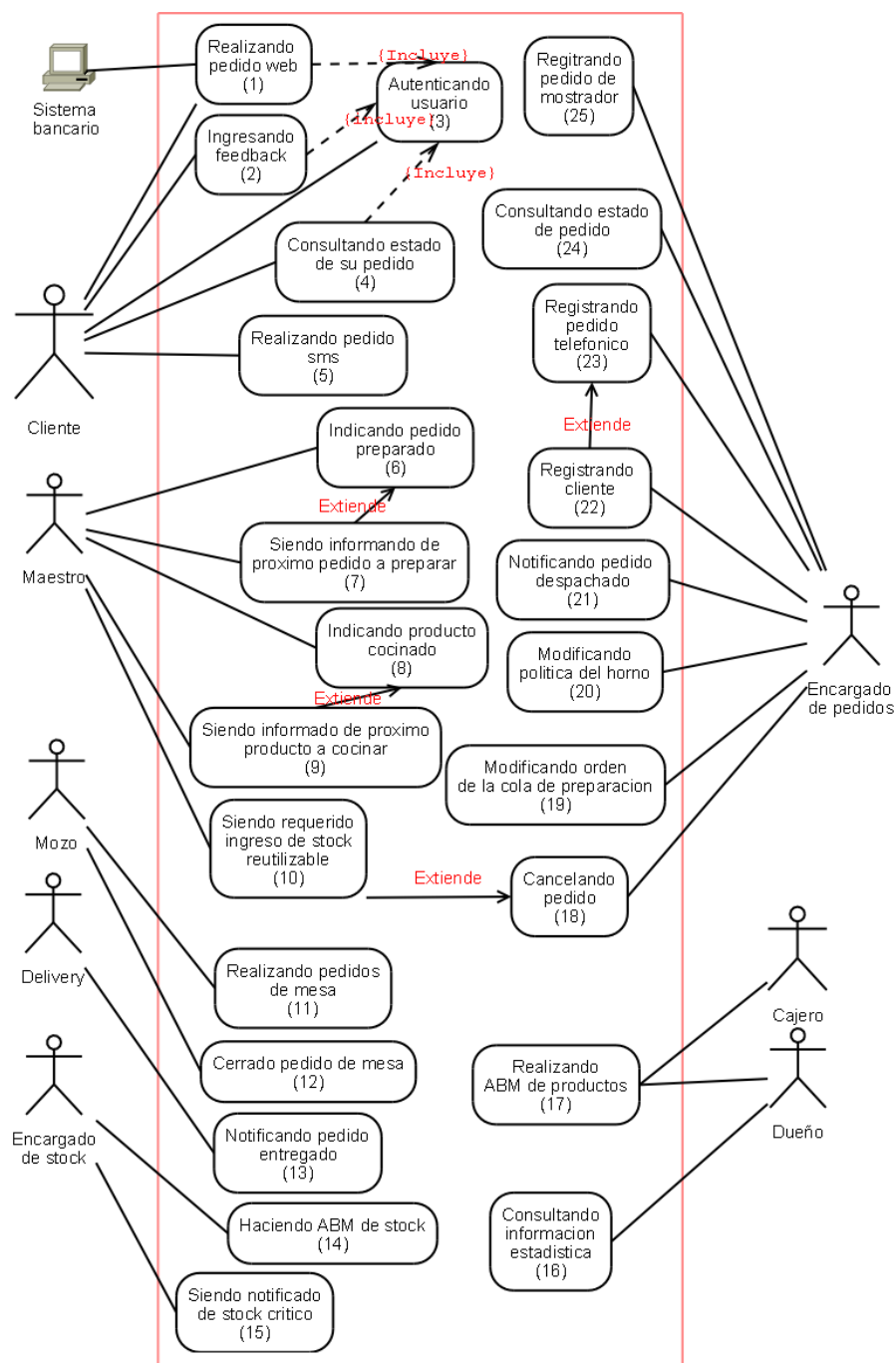
Requerimiento:	Calcular tiempo total del pedido en base a estimaciones individuales
Numero:	34
Tipo:	Funcional
Importancia:	Importante
Descripción:	Los tiempos individuales para los productos deben usarse para poder estimar el tiempo de un pedido teniendo en cuenta la cola de producción, la política de asignación, etc.
Motivo:	Se desea estimar el tiempo de preparación de los pedidos, para contribuir a una mejor atención de los clientes.

3.3. Casos de Uso

A continuación se describen mediante la herramienta de Casos de Uso todas las posibles interacciones de agentes con el sistema. El objeto de esta descripción es permitir observar todos los parámetros en juego en las interfaces del sistema, así como servir de guía para la posterior implementación del sistema.

Se presenta en primer lugar el diagrama de Casos de Uso, que contiene todas las posibles interacciones así como sus vínculos con los agentes involucrados. A continuación se presenta el detalle de cada caso de uso con las operaciones que involucra, y vínculos de trazabilidad con los requerimientos antes descritos.

3.3.1. Diagrama de Casos de Uso



3.3.2. Descripción de Casos de Uso

Caso de Uso: Realizando pedido web	
Número 1	
Actores intervinientes: Cliente, Sistema bancario	
Requerimientos relacionados: 3, 4, 5, 6, 24, 29, 30, 33	
Precondición: TRUE	
Poscondición: Pedido registrado	
Descripción:	El cliente ingresa un pedido al sistema desde la pagina web (ver diagrama de actividades figura 4.7 relacionado para más detalle)
Curso normal	Curso alternativo
1. INCLUYE autenticando usuario	
2. Cliente selecciona de una lista los productos que desea y su cantidad	
3. El sistema brinda al cliente una estimación de la demora del pedido y el costo total	
4. El sistema pregunta al cliente si desea pagar con tarjeta de crédito o pagar en efectivo al delivery	
5. El cliente elige forma de pago	
6. Si el cliente eligió pagar en efectivo, ir al paso 14	
7. Si el cliente eligió pagar con tarjeta y la tarjeta no se encuentra registrada, el sistema le pide al cliente que ingrese su número de tarjeta, sino ir al paso 12	
8. El cliente ingresa la tarjeta al sistema	
9. El sistema pide al sistema bancario verificar si la tarjeta existe	
10. Si la tarjeta existe, el sistema pide al sistema bancario verificar el monto	10. Sino el sistema muestra un mensaje de error al cliente. Ir al paso 4
11. El sistema registra la tarjeta	
12. El sistema pide al sistema bancario verificar si la tarjeta tiene fondos	
13. Si el pago con tarjeta es posible, el sistema espera que el cliente confirme el pedido	13. Sino Ir al paso 2
14. El cliente presiona el boton de enviar pedido	
15. El sistema verifica si el pedido es viable (ver figura 4.4 FSM Chequeo de stock para más detalle)	
16. Si el pedido es posible, ir al siguiente paso	16. Si el pedido no es posible, se muestra mensaje de error al cliente indicando que productos no estan disponibles. Ir al paso 2
17. Si el cliente eligió pagar con tarjeta, el sistema indica al sistema bancario que haga el pago	
18 El sistema registra el pedido como ingresado	
19. El sistema actualiza el stock	
20. El sistema brinda al cliente un id de pedido para que luego consulte el estado del pedido si lo necesita	
21. Fin CU	

Caso de Uso: Ingresando feedback	
Número 2	
Actores intervinientes: Cliente	
Requerimientos relacionados: 8, 9, 33	
Precondición: TRUE	
Poscondición: Comentario ingresado	
Descripción:	Permite que el usuario envíe al sistema su opinión sobre un pedido recibido por delivery
Curso normal	Curso alternativo
1. El cliente ingresa a la página web de feedback	
2. Se incluye CU autenticando usuario	
3. Ingresa al apartado de feedback	
4. Ingresa el número de pedido	
5. El sistema verifica que el número de pedido corresponda con un pedido enviado por delivery a ese cliente, y abre el cual el mismo no halla ingresado feedback	
6. Si el número era válido, se muestra un formulario al usuario para que envíe su opinión	5. Si el número no coincide se envía un mensaje de error al usuario
7. Fin CU	

Caso de Uso: Autenticando usuario	
Número 3	
Actores intervinientes: Cliente	
Requerimientos relacionados: 4, 9, 24, 25, 32	
Precondición: TRUE	
Poscondición: Cliente logueado	
Descripción:	Autenticación de un cliente web
Curso normal	Curso alternativo
1. El cliente ingresa su nombre de usuario en el campo correspondiente	
2. El cliente ingresa su contraseña en el campo correspondiente	
3. El cliente presiona el botón de ingresar	
4. El sistema valida los datos	
5. Si los datos ingresados son válidos, cliente autenticado	5. Los datos no son válidos, se muestra al cliente mensaje informativo y se le permite re ingresar los datos
6. Fin CU	

Caso de Uso: Consultando estado de su pedido	
Número 4	
Actores intervinientes: Cliente	
Requerimientos relacionados: 32, 33, 34	
Precondición: TRUE	
Poscondición: Se muestra el estado del pedido	
Descripción:	Permite al cliente ver en que estado se encuentran sus pedidos en un momento dado
Curso normal	Curso alternativo
1. INCLUYE autenticando usuario	
2. El cliente selecciona consultar estado de pedidos	
3. El sistema muestra al cliente la lista de los pedidos del usuario pendientes de entrega cada uno con su estado correspondiente y demora estimada	
4. Fin CU	

Caso de Uso: Realizando pedido sms	
Número 5	
Actores intervinientes: Cliente	
Requerimientos relacionados: 3, 4, 5, 6, 28, 30, 33	
Precondición: TRUE	
Poscondición: Pedido ingresado al sistema	
Descripción:	El cliente hace un pedido via sms
Curso normal	Curso alternativo
1. El cliente consulta el catálogo de códigos de producto y combos, publicados en la web del sistema junto con el formato que debe tener el mensaje	
2. El cliente envía un mensaje al sistema con el pedido a solicitar	
3. El sistema verifica si el número desde donde proviene el mensaje está registrado con algún cliente	
4. Si el número no se encuentra registrado, el sistema responde al cliente sugiriéndole registrarse por teléfono o personalmente y el pedido es descartado. Ir a Fin CU	3. Si el número está registrado pero el mensaje no corresponde con ningún código se notifica al cliente que el código no es válido
5. El sistema verifica si es posible satisfacer el pedido (ver figura 4.4 FSM Chequeo de stock para más detalle)	
6. Si es posible se ingresa el pedido al sistema y se envía al usuario un mensaje, notificándole que su pedido está ingresado, el importe total, el tiempo de demora aproximado y el código de pedido para realizar consultas sobre el mismo	6. Sino el sistema le envía un mensaje al cliente avisándole que el pedido no podrá ser satisfecho, informando además qué producto/s no están disponibles
7. El sistema actualiza el stock	
8. Fin CU	

Caso de Uso: Indicando pedido preparado	
Número 6	
Actores intervinientes: Maestro	
Requerimientos relacionados: 8, 10, 13, 33	
Precondición: El maestro tiene un pedido en preparación	
Poscondición: El pedido se registra como preparado	
Descripción:	El maestro, luego de preparar un pedido, indica al sistema que el mismo está preparado (ver figura 4.13 para más detalle)
Curso normal	Curso alternativo
1. El maestro indica al sistema que finalizó la preparación del último pedido que le pidió	
2. El sistema registra el último pedido que le ordenó preparar al maestro como preparado	
3. Si hay pedido para preparar el sistema le informa que debe preparar a continuación. EXTIENDE caso de uso Siendo informado de proximo pedido a preparar	
4. Fin CU	

Caso de Uso: Siendo informado de proximo pedido a preparar	
Número 7	
Actores intervinientes: Maestro	
Requerimientos relacionados: 8, 10, 13, 33	
Precondición: La cola de pedidos a preparar no está vacía	
Poscondición: La parte de pedido comienza a prepararse	
Descripción:	El sistema le ordena al maestro que parte de pedido preparar
Curso normal	Curso alternativo
1. El sistema indica al maestro preparar el primer pedido en la cola de pedidos a preparar y espera que el maestro confirme	
2. El maestro indica al sistema que empezará a preparar la parte pedida	
3. El pedido cambia su estado a “en preparacion” y sale de la cola de pedidos	
4. Si hay más pedidos a preparar, ir a paso 1	
5. Fin CU	

Caso de Uso: Indicando producto cocinado	
Número 8	
Actores intervinientes: Maestro	
Requerimientos relacionados: 8, 11, 12, 15, 33	
Precondición: True	
Poscondición: La parte de pedido se registra como cocinada	
Descripción:	El maestro, luego de cocinar una parte de un pedido, indica al sistema que la misma está cocinada (ver figura 4.14 para más información)
Curso normal	Curso alternativo
1. El maestro indica al sistema que finalizó la cocción de ciertas partes de un pedido	
2. El sistema muestra al maestro un menú de partes que debían cocinarse y espera que el maestro indique la parte	
3. El maestro indica al sistema qué parte finalizó la cocción	
4. El sistema registra la parte como cocinada	
5. El sistema verifica si la última parte cocinada completa el pedido	
6. Si es así, el sistema registra al pedido como listo	
7. Si hay productos para cocinar el sistema le informa al maestro que debe poner a continuación. EX-TIENDE caso de uso Siendo informado de próximo producto a cocinar	
8. Fin CU	

Caso de Uso: Siendo informado de próximo producto a cocinar	
Número 9	
Actores intervinientes: Maestro	
Requerimientos relacionados: 8, 11, 12, 15, 33	
Precondición: La cola del horno no está vacía	
Poscondición: La parte comienza a cocinarse	
Descripción:	El sistema le ordena al maestro que parte de pedido debe cocinar (ver figura 4.14 para más información)
Curso normal	Curso alternativo
1. El maestro pide al sistema que pedido debe cocinar a continuación	
2. El sistema indica al maestro una parte a cocinar	
3. Si es la primera parte de un pedido, el sistema cambia el estado del mismo a “en horno”	
4. Fin CU	

Caso de Uso: Siendo requerido ingreso de stock reutilizable	
Número 10	
Actores intervinientes: Maestro	
Requerimientos relacionados: 21, 33	
Precondición: Hay cancelación de un pedido que está en preparación	
Poscondición: Los insumos reutilizables son reingresados al stock	
Descripción:	Cuando el maestro esta preparando un pedido que es cancelado, este debe reingresar al sistema los insumos que no utilizó en la preparación
Curso normal	Curso alternativo
1. El sistema indica al maestro que el pedido en preparación fue cancelado	
2. El sistema pide al maestro que indique que parte de los insumos se puede reutilizar para reingresar al stock	
3. El maestro ingresa las cantidades y tipo de los insumos reutilizables	
4. El maestro confirma reingresar insumos	
5. El sistema lo registra y actualiza el stock	
6. Fin CU	

Caso de Uso: Realizando pedidos mesa	
Número 11	
Actores intervinientes: Mozo	
Requerimientos relacionados: 3, 4, 5, 6, 23, 30, 33	
Precondición: TRUE	
Poscondición: El pedido de la mesa queda registrado	
Descripción:	El cliente del local hace un pedido a un mozo, quien ingresa el pedido (ver figura 4.9 para más información)
Curso normal	Curso alternativo
1. El cliente informa al mozo lo que quiere pedir	
2. El mozo marca en su PDA los productos que el cliente pide con sus respectivas cantidades	
3. Cuando el cliente informó todo el pedido, el mozo ingresa el pedido	
4. El sistema verifica si el pedido es posible (ver figura 4.4 FSM Chequeo de stock para más detalle)	
5. Si el pedido se puede tomar, el sistema genera el id del pedido, calcula el costo total, y si no era de solo bebidas, estima el tiempo del mismo, y muestra estos datos al mozo	5. Sino el sistema indica mediante un mensaje de error que producto/s no estaba/n disponible/s, y el mozo le comunica al cliente sobre la situación. Ir a paso 1
6. El sistema actualiza el stock	
7. Fin CU	

Caso de Uso: Cerrando pedido de mesa	
Número 12	
Actores intervinientes: Mozo	
Requerimientos relacionados: 23, 30, 31, 33	
Precondición: TRUE	
Poscondición: Se registra la forma de pago del pedido y se cierra la mesa	
Descripción:	Permite que el cliente le informe al mozo su forma de pago, que el mozo la registre y el pedido este listo para ser facturado
Curso normal	Curso alternativo
1. El cliente llama al mozo	
2. El cliente dice al mozo que va a pagar	
3. El mozo selecciona el/los pedidos de la mesa, y pide al cliente forma de pago	
4. El mozo ingresa la forma de pago que el cliente comunica y cierra la mesa	
5. Fin CU	

Caso de Uso: Notificando pedido entregado	
Número 13	
Actores intervinientes: Delivery	
Requerimientos relacionados: 8, 22, 33	
Precondición: True	
Poscondición: El pedido se registra como entregado	
Descripción:	Permite que el delivery mediante SMS informe inmediatamente la entrega o no entrega de un pedido
Curso normal	Curso alternativo
1. El delivery entrega el pedido al cliente	1. Si es imposible entregar el pedido, el delivery envía un mensaje para que quede cancelado
2. El delivery envía un mensaje con el código del pedido	
3. El sistema recibe el mensaje y el estado del pedido cambia a entregado	
4. Fin CU	

Caso de Uso: Haciendo ABM de stock	
Número 14	
Actores intervinientes: Encargado de stock	
Requerimientos relacionados: 2	
Precondición: TRUE	
Poscondición: El stock queda actualizado con las modificaciones	
Descripción:	Se puede hacer alta, baja y modificación de insumos y kits comprados o caducados (los diagramas de máquina de estado explican un poco más en la sección ABM de stock la relación de este caso de uso con el caso de uso Siendo notificado de stock crítico)
Curso normal	Curso alternativo
1. El encargado de stock elige que operación realizar	
2. El sistema le pide al encargado de stock que ingrese el nombre o número del insumo	
3. El encargado de stock ingresa el nombre o el número identificador	
4.1. Si se trata de un alta ir a paso 6	
4.2. Si se trata de una modificación ir a paso 9	
4.3. Si se trata de una baja ir a paso 9.	
5. El sistema permite ingresar la cantidad y otros datos del insumo	
6. El encargado de stock ingresa cantidad en stock del nuevo insumo y datos	
7. El sistema agrega el insumo a la lista con su cantidad en stock y datos	
8. Ir a Fin CU	
9. El sistema verifica que el insumo exista en la lista de insumos	
10. Si es así, se sigue al siguiente paso	Si el insumo no existe, el sistema produce un mensaje de error y se procede a Fin CU
11. Si se trata de se una baja de un insumo, el sistema elimina el insumo de la lista, sino se procede a paso 13	
12. Ir a Fin CU	
13. El sistema le pide al encargado de stock que ingrese la cantidad a aumentar o a quitar	
14. El encargado de stock ingresa la cantidad en stock del insumo	
15. El sistema modifica la cantidad en stock del insumo	
16. Fin CU	

Caso de Uso: Siendo notificado de stock crítico	
Número 15	
Actores intervinientes: Encargado de stock	
Requerimientos relacionados: 1	
Precondición: Algún kit o insumo está por debajo del nivel crítico	
Poscondición: Surge una advertencia de stock crítico	
Descripción:	En caso de que el stock de algún producto quede por debajo de un nivel crítico, el sistema avisará al encargado de stock de esta situación (los diagramas de máquina de estado explican un poco más en la sección ABM de stock la relación de este caso de uso con el caso de uso Siendo notificado de stock crítico)
Curso normal	Curso alternativo
1. El sistema detecta que la cantidad de un insumo es por debajo del nivel crítico	
2. El sistema advierte al encargado de stock qué insumo es crítico hasta que este confirme que el mensaje le ha llegado	
3. El encargado de stock confirma al sistema que el mensaje le llegó	
4. Fin CU	

Caso de Uso: Consultando información estadística	
Número 16	
Actores intervinientes: Encargado/Dueño	
Requerimientos relacionados: 8, 34	
Precondición: TRUE	
Poscondición: Las estadísticas son mostradas en pantalla	
Descripción:	El dueño puede consultar la información estadística sobre los productos y pedidos gestionados
Curso normal	Curso alternativo
1. Si el dueño desea observar información estadística sobre un producto ir a paso 2, sino ir a paso 6	
2. El dueño le indica al sistema que requiere datos estadísticos sobre un producto o pedido	
3. El dueño ingresa el número de producto o pedido sobre el cual necesita averiguar estadísticas	
4. El sistema verifica que el producto o pedido exista	
5. Si es así, el sistema imprime en pantalla los datos estadísticos	Sino se produce un mensaje de error y se procede a Fin CU
6. El dueño elige que estadísticas observar entre las disponibles (ver detalle en la sección información estadística)	
7. Cuando el dueño observe las estadísticas que deseaba, ir a fin CU	
8. Fin CU	

Caso de Uso: Realizando ABM de productos	
Número 17	
Actores intervinientes: Cajero, Dueño	
Requerimientos relacionados: 7	
Precondición: TRUE	
Poscondición: Se realizan las altas, bajas o modificaciones pedidas por el Dueño o cajero	
Descripción:	Permite ingresar nuevos productos a la carta de la pizzería, indicar tiempos estimados de cocción y preparación, modificar precios y dar de baja productos
Curso normal	Curso alternativo
1. El dueño o cajero eligen si desean hacer un alta, baja o modificación	
2. Si quieren hacer un alta, ingresan el nombre del producto, precio, ingredientes y tiempo de cocción y preparación (que deben ser dichos al dueño o cajero por el maestro). Si terminaron ir a Fin CU, sino ir a 1	
3. Si quieren hacer una baja, eligen hacer baja y el producto que quieren dar de baja. Si terminaron ir a Fin CU, sino ir a 1	
4. Si lo que querían es hacer una modificación, pueden escoger el producto y modificar los datos del mismo. Si terminaron ir a Fin CU, sino ir a 1	
5. Fin CU	

Caso de Uso: Cancelando pedido	
Número 18	
Actores intervinientes: Encargado de pedido	
Requerimientos relacionados: 19, 20, 21	
Precondición: Pedido no entregado	
Poscondición: Pedido cancelado	
Descripción:	Un cliente solicita cancelar un pedido, por teléfono o personalmente, y el encargado de pedidos se encarga de esta cancelación.
Curso normal	Curso alternativo
1. El cliente informa al encargado de pedido su nombre o nombre de usuario y el id de pedido, o el cliente informa al mozo que desea cancelar un pedido y el mozo informa al encargado de pedidos el deseo de la mesa de cancelar el pedido	
2. El encargado ingresa al sistema el nombre proporcionado por el cliente y el id de pedido	
3. Se verifica que el pedido no haya sido entregado	3. Si ya fue entregado, no se puede cancelar. Ir a fin CU
4. El sistema verifica que el pedido exista y que el cliente haya hecho el pedido	
5. Si el pedido existe y está en estado preparando, se deben reingresar los insumos reutilizables. EX-TIENDE caso de uso Agregando insumos reutilizables	
6. Fin CU	

Caso de Uso: Modificando orden de cola de preparación	
Número 19	
Actores intervinientes: Encargado de pedidos	
Requerimientos relacionados: 14, 33	
Precondición: True	
Poscondición: Orden de preparación modificado	
Descripción:	Permite que el encargado de pedidos modifique la cola de pedidos
Curso normal	Curso alternativo
1. El encargado de pedido selecciona modificar cola de pedidos	
2. Selecciona de la lista asignando a cada maestro un pedido y lo mueve hacia la posición deseada	
3. Ingresa aceptar	
4. Fin CU	

Caso de Uso: Modificando política del horno	
Número 20	
Actores intervinientes: Encargado de pedidos	
Requerimientos relacionados: 8	
Precondición: No hay pedidos hechos	
Poscondición: Política de cola modificada	
Descripción:	El encargado de pedidos modifica la política de cola del horno. Este caso de uso se lleva a cabo cuando inicia el sistema
Curso normal	Curso alternativo
1. El encargado de pedidos selecciona modificar política de horno	
2. El sistema muestra las políticas posibles y espera que el encargado elija una	
3. El encargado elige una política	
4. Si el encargado eligió la política de cola normal, el sistema registra que la política de horno a usar será cola. Ir a Fin CU	
5. Si el encargado eligió la política de colas con sector ágil, el sistema le pide que ingrese cuales son los módulos ágiles	
6. El encargado ingresa los módulos ágiles	
7. El sistema registra que la política de horno a usar será cola con sector ágil	
8. Fin CU	

Caso de Uso: Notificando pedido despachado	
Número 21	
Actores intervinientes: Encargado de pedidos	
Requerimientos relacionados: 30, 33	
Precondición: El pedido debe estar listo	
Poscondición: Pedido marcado como despachado (o entregado en el caso de que sea local)	
Descripción:	Permite indicar que un pedido salió con el delivery o que fue entregado al cliente en mostrador o por el mozo en la mesa
Curso normal	Curso alternativo
1. El encargado de pedidos selecciona al pedido que se va a despachar de entre los pedidos listos	
2. Si el pedido era para delivery, el pedido pasa a estado enviado y se despachará a cargo del delivery	
3. Si el pedido era para el local, el pedido pasa a estado finalizado y se despachará a cargo del mozo para que le entregue el pedido al cliente	
4. Si el pedido era de mostrador, el pedido pasa a estado finalizado	
3. Fin CU	

Caso de Uso: Registrando cliente	
Número 22	
Actores intervinientes: Encargado de pedidos	
Requerimientos relacionados: 3, 4, 5, 6, 25, 26, 27 ,28	
Precondición: El cliente no esta registrado en el sistema	
Poscondición: El cliente esta registrado en el sistema	
Descripción:	El cliente desea registrarse en la pizzería y el encargado de pedidos lo atiende y registra
Curso normal	Curso alternativo
1. Si el cliente quiere actualizar sus datos ir a paso 7	
2. El cliente solicita al encargado de pedidos registrarse (por via telefónica o personalmente)	
3. El encargado pide al cliente los datos: <ul style="list-style-type: none"> ▪ Nombre ▪ Apellido ▪ Dirección ▪ Teléfono ▪ Usuario, password (opcional, solo necesario para hacer pedidos via Web) ▪ E-Mail (opcional) ▪ Célular (opcional, solo necesario para poder hacer pedidos vis SMS) 	
4. El cliente informa al encargado de pedidos sus datos	
5. El encargado de pedidos registra estos datos en el sistema	
6. El sistema registra el usuario en la lista de usuarios, ir a fin CU	
7. El encargado de pedidos verifica los datos del usuario	
8. El usuario indica que datos modificar, y con que valores	
9. El encargado ingresa dichos datos	
10. El sistema actualiza los datos	
11. Fin CU	

Caso de Uso: Registrando pedido telefónico	
Número 23	
Actores intervinientes: Encargado de pedidos	
Requerimientos relacionados: 33	
Precondición: True	
Poscondición: Pedido ingresado al sistema	
Descripción:	El cliente quiere hacer un pedido por vía telefónica, el encargado de pedidos lo atiende y luego de verificar los datos ingresa el pedido al sistema (ver figura 4.5 para más información)
Curso normal	Curso alternativo
1. El cliente llama al número telefónico de la pizzería y lo atiende el encargado de pedidos	
2. El encargado le pide al cliente sus datos de identificación	
3. El cliente informa al encargado de pedidos sus datos	
4. El encargado indica al sistema que desea ingresar un pedido y luego ingresa los datos del cliente al sistema	
5. El sistema verifica si el cliente está registrado	
6. Si el cliente está registrado, el sistema muestra al encargado el perfil del cliente	6. Sino el encargado de pedidos le sugiere al cliente la posibilidad de registrarse. Si el cliente no lo desea entonces ir a fin CU, sino se procede a registrarlo (EXTIENDE Registrando cliente e ir a paso 9)
7. El encargado de pedidos verifica el teléfono del cliente registrado con el teléfono desde el cual cliente está llamando	
8. Si el teléfono registrado no coincide con el teléfono desde el cual está llamando pero el cliente tiene usuario y clave, el encargado le pide al cliente que este indique usuario y clave y verifica	8. Si el teléfono registrado no coincide con el teléfono desde el cual el cliente está llamando y el cliente no tiene usuario y clave informar que no es posible tomar el pedido e ir a Fin CU
9. El encargado de pedidos indica al cliente que le dicte el pedido	
10. Una vez finalizado, el encargado de pedidos intenta ingresar el pedido al sistema	
11. El sistema verifica si es posible satisfacer el pedido (ver figura 4.4 FSM Chequeo de stock para más detalle)	
12. Si el pedido es posible y no es de solo bebida, el sistema estima el tiempo del mismo, calcula el costo total y el encargado se lo comunica al cliente, sino el sistema indica mediante un mensaje de error que producto/s no estaba/n disponible/s, para que el encargado se lo comunique al cliente. Ir a paso 9	
13. Si era de solo bebida, se registra el pedido como listo y se calcula el costo total, se lo comunica al cliente	
14. El sistema actualiza el stock	
15. Si el pedido es mixto, el sistema pide al encargado de pedidos que asigne un horno	
16. El sistema genera el identificador de pedido y notifica al encargado de pedidos sobre estos datos, para que el encargado se lo notifique al cliente	
17. Fin CU	

Caso de Uso: Consultando estado de pedido	
Número 24	
Actores intervinientes: Encargado de pedidos	
Requerimientos relacionados: 32, 33, 34	
Precondición: TRUE	
Poscondición: Se muestra la información del pedido consultado	
Descripción:	El encargado le pide información al sistema sobre un pedido en particular o sobre el cliente que haya hecho el pedido
Curso normal	Curso alternativo
1. El encargado de pedidos indica al sistema que desea consultar el estado de un pedido	
2. El sistema pide al encargado que ingrese el identificador o cliente del pedido quiere consultar	
3. El encargado de pedidos indica al sistema que pedido desea consultar	
4. El verifica si el pedido existe	
5. Si es así, seguir al siguiente paso	Sino, ir a Fin CU
6. Si el encargado pidió consultar un pedido identificando al cliente, el sistema mostrará toda la información de los pedidos pendientes de entrega del cliente	
7. Si el encargado pidió consultar un pedido identificando al mismo mediante el identificador, el sistema mostrará toda la información correspondiente al pedido que tiene tal identificador	
8. Fin CU	

Caso de Uso: Registrando pedido mostrador	
Número 25	
Actores intervinientes: Encargado de pedidos	
Requerimientos relacionados: 3, 4, 5, 6, 30, 33	
Precondición:	
Poscondición: Pedido ingresado al sistema	
Descripción:	El cliente quiere hacer un pedido en el local para llevar
Curso normal	Curso alternativo
1. El cliente comunica al encargado de pedidos que desea hacer un pedido	
2. El encargado pregunta si el cliente esta registrado	
3. Si es así el cliente da sus datos al encargado, si no el pedido se registrara sin cliente	
4. El encargado indica al sistema que desea ingresar un pedido de mostrador	
5. El encargado de pedidos indica al cliente que le dicte el pedido	
6. Una vez finalizado, el encargado de pedidos intenta ingresar el pedido al sistema	
7. El sistema verifica si es posible satisfacer el pedido (ver figura 4.4 FSM Chequeo de stock para más detalle)	
8. Si el pedido es posible y tiene alguna empanada o pizza, el sistema estima el tiempo del mismo, calcula el costo total y el encargado se lo comunica al cliente, sino el sistema indica mediante un mensaje de error que producto/s no estaba/n disponible/s, para que el encargado se lo comunique al cliente. Ir a paso 5	
9. Si solo tenia bebidas, el pedido se ingresa como listo y se calcula el total	
10. El sistema actualiza el stock	
11. Si el pedido es mixto, el sistema pide al encargado de pedidos que asigne un horno	
12. El sistema genera el identificador de pedido y notifica al encargado de pedidos sobre estos datos, para que el encargado se lo notifique al cliente	
13. El encargado pregunta forma de pago al cliente, e ingresa esa forma de pago en el sistema, a fin de que luego se pueda generar el archivo para el sistema de facturación	
14. Fin CU	

3.3.3. Matriz de trazabilidad

A continuación presentamos una matriz donde se pueden observar las relaciones entre requerimientos y casos de uso. Dicha tabla contiene en sus filas a los requerimientos y en sus columnas a los casos de uso. Se indica con una cruz cuando un caso de uso contribuye a satisfacer un cierto requerimiento. Esto permite visualizar rápidamente la completitud del sistema en lo que a requerimientos refiere.

Es de notar que hay tres filas en la tabla que no tienen ningún caso de uso asociado. Se trata de los requerimientos 16, 17 y 18, y la razón de esta anomalía es que los requerimientos afectan al modo de operación de contingencia. Los casos de uso correspondientes a este modo no se encuentran listados en esta sección, sino que se discuten más adelante en la sección 4.10, y por tanto no aparecen en esta tabla.

Requerimiento\Caso de uso	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1															X										
2														X											
3	X				X						X												X		X
4	X		X		X						X												X		X
5	X				X						X												X		X
6	X				X						X												X		X
7																	X								
8		X				X	X	X	X				X			X				X					
9		X	X																						
10						X	X																		
11								X	X																
12								X	X																
13						X	X																		
14																			X						
15								X	X																
16																									
17																									
18																									
19																		X							
20																		X							
21										X								X							
22													X												
23											X	X													
24	X		X																						
25			X																			X			
26																						X			
27																						X			
28					X																	X			
29	X																								
30	X				X						X	X									X				X
31												X													
32			X	X																				X	
33	X	X		X	X	X	X	X	X	X	X	X	X						X		X		X	X	X
34				X												X								X	
Requerimiento\Caso de uso	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

3.4. Conceptos del Sistema

A continuación presentaremos un diagrama de conceptos que pretende modelar los principales conceptos visibles en nuestra solución. El mismo intenta brindar una visión completa de lo que el sistema puede observar.

3.4.1. Diagrama de modelo conceptual

Aclaraciones sobre el diagrama:

- Todas las herencias son de tipo **Disjoint Complete**, se omitió la etiqueta por simplicidad.
- El tipo **Date** debe almacenar no solo la fecha sino también la hora (técnicamente es un **DateTime**).

El diagrama ilustra en primer lugar que el concepto central del que depende el resto es el **Pedido**. Un pedido consta de una fecha de ingreso (el momento en que fue registrado en el sistema), un identificador único y un estado. A su vez, un pedido tiene asociados **Productos** que lo constituyen, y estos se cuantifican mediante la clase de asociación **Items** (que registra cuantas unidades de cada producto hay en el pedido, así como el precio unitario del producto en el momento de la venta). Si bien el precio está registrado en cada **Producto**, la distinción del precio de dicha clase y de la clase **Item** está dada porque los productos pueden cambiar de precio en cualquier momento y por lo tanto es necesario registrar el valor correspondiente al momento de ingreso del pedido. Notar que el producto hace entonces referencia por ejemplos a “pizza de muzzarella” y no a una “pizza de muzzarella” en particular.

En función del estado en que se encuentra un **Pedido**, puede tener asociadas varias clases cuya relación es condicional. Muchas de estas clases tienen una fecha que registra el momento en que el pedido ingresó al estado. Esta fecha puede ser necesaria por fines estadísticos (caso de **FechaFinalización** o **FechaCancelación**) o para identificación de los pedidos (**FechaComienzo** en la clase **Preparación**). A su vez, estas clases asociadas pueden registrar información adicional referente a un estado. Por ejemplo, un pedido cancelado tiene una razón de cancelación, mientras que un pedido finalizado podría eventualmente tener un *feedback* asociado, con el comentario del cliente sobre su pedido.

Un pedido puede ser **Remoto** o **Local** (la distinción precisa está disponible en el Glosario). Un pedido remoto debe estar asociado necesariamente a un cliente, que será aquel al que se le haga la entrega del pedido. Esta relación es opcional para los pedidos hechos en el local. La **Forma de Pago** se asocia de forma diferenciada a un pedido según si este es remoto o local puesto que en el caso de pedidos remotos siempre se conoce al momento de ingresar el pedido el medio de pago que utilizará el cliente, mientras que para los pedidos locales esta información no estará disponible hasta tanto el mozo no la averigüe.

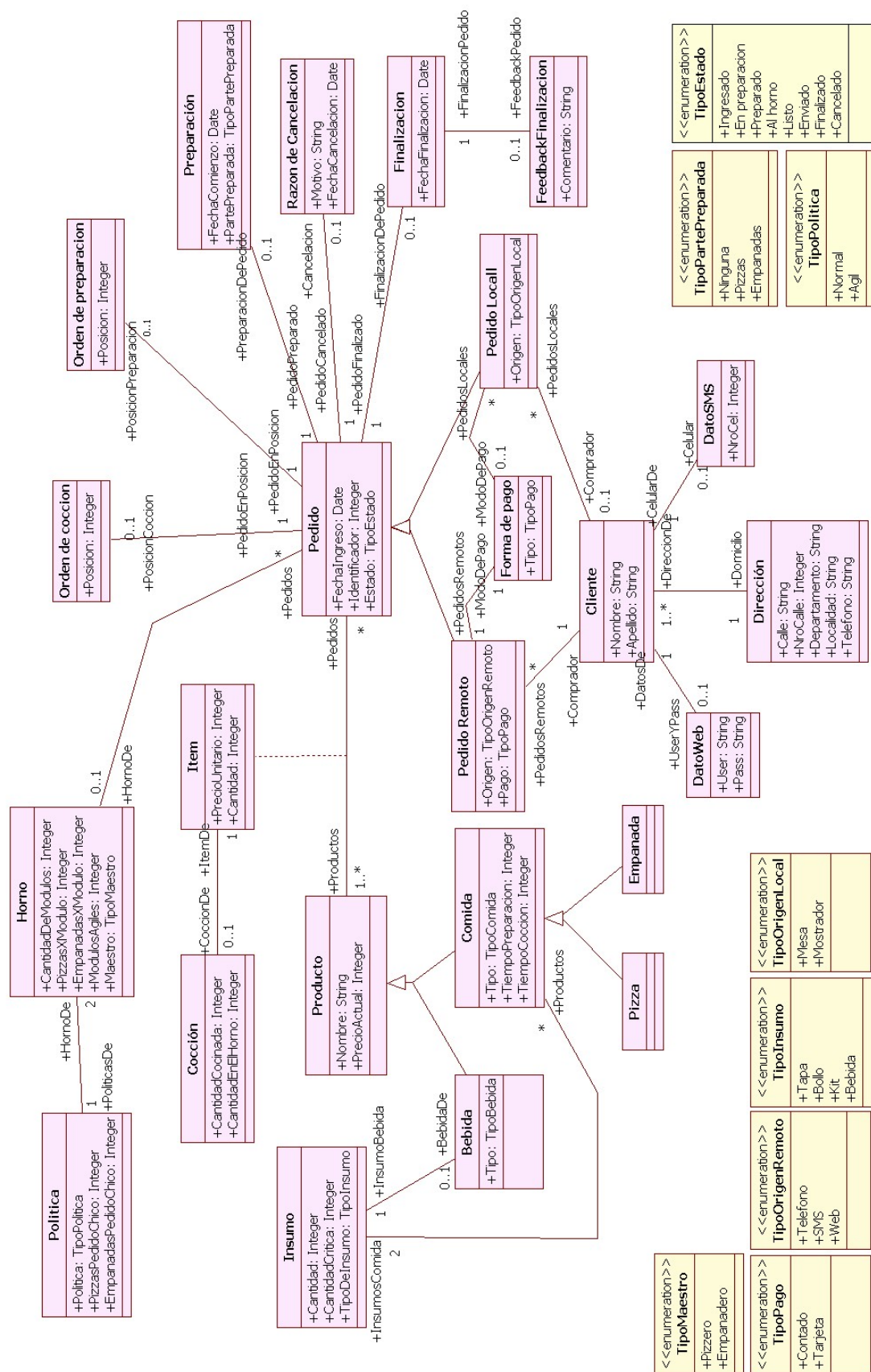
Las clases **Orden de cocción** y **Orden de preparación** identifican el orden asignado a los pedidos en las colas de pedidos y de horno respectivamente. Almacenan un número que sirve únicamente como criterio de comparación entre pedidos para establecer un orden relativo entre ellos. La clase **Preparación** tiene una función parecida, y permite identificar qué parte de un pedido está preparada en el caso de pedidos mixtos (para pedidos simples un pedido no puede estar parcialmente preparado a los ojos del sistema). Todas estas son clases *ad hoc* que permiten registrar detalles propios de cada estado.

Un **Producto** puede ser una **Comida** o una **Bebida**. Las comidas constan de dos insumos para su preparación: un bollo de pizza o tapa de empanada dependiendo del caso, y un kit de ingredientes que se combinan con el bollo o tapa al momento de preparar el pedido. Las bebidas tienen un único insumo (la propia botella o lata de la bebida). La clase **Insumo** tiene por objetivo contabilizar la cantidad remanente en *stock* que hay de cada uno de los insumos involucrados.

Un **Cliente** puede o no tener asociadas las clases **Datos SMS** y **Datos Web**. Estas clases registran información adicional que es necesaria para que el cliente puede hacer pedidos a través de dichos medios. En este sentido, todo pedido cuyo origen sea Web deberá estar asociado a un cliente que tenga esta información asociada. En cualquier caso, la información básica (nombre, apellido y una dirección para hacer entregas) es común a todos los clientes registrados.

Por último, en la parte superior izquierda del diagrama está registrada información referente a la cocción de los pedidos. En particular, se asocia un pedido a un **Horno** (o a ninguno en caso de que el pedido estuviera compuesto únicamente de bebidas). Dicho horno tiene una **Política** donde se almacenan datos de interés para el cálculo de operaciones e invariantes. La clase **Cocción** es la que almacena información sobre cocciones parciales, y debe estar asociada a un pedido siempre que este esté en estado *Al Horno*. Esta clase indica qué cantidad de qué productos ya fueron cocinados o se encuentran en el horno.

Es de destacar que el sistema no guarda registros históricos de la información que es propia de cada estado. Los únicos estados definitivos que puede adquirir un pedido y que se espera persistan una vez que se finaliza el pedido son Cancelado



y Finalizado, que registran algunas informaciones importantes como la razón de cancelación y la hora de finalización del pedido que es de suma utilidad para fines estadísticos. El registro de las informaciones que se descartan son objeto de una propuesta de mejora en la sección correspondiente.

3.4.2. Restricciones al modelo conceptual

A continuación se detallan una serie de restricciones adicionales al modelo conceptual que no son aparentes sobre el diagrama. Para especificarlas se utiliza OCL, y completan el modelo conceptual de sistema ajustando algunos invariantes que deben preservarse en él.

Finalización

1. Un pedido tiene finalización si y solo si su estado es finalizado:

Context Finalización

Inv: $self.PedidoFinalizado.estado = Finalizado \text{ and } (Pedido.allInstances() \rightarrow select(p \mid p.estado = finalizado) = Pedido.allInstances() \rightarrow select(p \mid p.estado = finalizado \text{ and } p.FinalizacionDePedido \rightarrow notEmpty()))$

2. Solo los pedidos remotos tienen feedback:

Context Finalización

Inv: $self.FeedbackPedido \rightarrow notEmpty() \text{ implies } self.PedidoFinalizado.isKindOf(Pedido Remoto)$

Cancelación

3. Todos los pedidos cancelados y solo los cancelados tienen razón de cancelación:

Context Cancelación

Inv: $self.PedidoCancelado.estado = Cancelado \text{ and } (Pedido.allInstances() \rightarrow select(p \mid p.estado = cancelado) = Pedido.allInstances() \rightarrow select(p \mid p.estado = cancelado \text{ and } p.Cancelacion \rightarrow notEmpty()))$

Cliente

4. Solo puede tener pedidos remotos del tipo Web un cliente registrado con Datos Web:

Context Cliente

Inv: $self.PedidosRemotos \rightarrow select(p \mid p.Origen = Web) \rightarrow notEmpty() \text{ implies } self.UserYPass \rightarrow notEmpty()$

5. Solo puede tener pedidos remotos del tipo SMS un cliente con celular registrado:

Context Cliente

Inv: $self.PedidosRemotos \rightarrow select(p \mid p.Origen = SMS) \rightarrow notEmpty() \text{ implies } self.DatosSMS \rightarrow notEmpty()$

DatoSMS

6. Todos los números de celular son diferentes:

Context DatoSMS

Inv: $DatoSMS.allInstances() \rightarrow forall(d \mid self \neq d \text{ implies } self.NroCel \neq d.NroCel)$

DatoWeb

7. Todos los usuarios del servicio web tienen usuarios diferentes:

Context DatoWeb

Inv: $DatoWeb.allInstances() \rightarrow \text{forall}(d \mid \text{self} \leq d \text{ implies } \text{self}.User \leq d.User)$

Dirección

8. Los números de telefono son diferentes para todas las direcciones:

Context Dirección

Inv: $Dirección.allInstances() \rightarrow \text{forall}(d \mid \text{self} \leq d \text{ implies } \text{self}.Telefono \leq d.Telefono)$

Insumo

9. La cantidad y la cantidad crítica son cantidades, por lo cual deben ser valores no negativos:

Context Insumo

Inv: $\text{self.cantidad} \geq 0 \text{ and } \text{self.cantidadCritica} \geq 0$

10. Uno de los insumos es del tipo masa y otro es del tipo tapa:

Context Insumo

Inv: $\text{Insumo.allInstances()} \rightarrow \text{select}(i \mid i.tipoDeInsumo = \text{Bollo}) \rightarrow \text{size}() = 1 \text{ and } \text{Insumo.allInstances()} \rightarrow \text{select}(i \mid i.tipoDeInsumo = \text{Tapa}) \rightarrow \text{size}() = 1$

11. Los kits y los insumos de bebida están vinculados con un unico producto:

Context Insumo

Inv: $(\text{self.tipoDeInsumo} = \text{Kit} \text{ implies } \text{self.productos} \rightarrow \text{size}() = 1) \text{ and } (\text{self.tipoDeInsumo} = \text{Bebida} \text{ implies } \text{self.BebidaDe} \rightarrow \text{size}() = 1)$

Bebida

12. Las bebidas tienen un insumo del tipo Bebida:

Context Bebida

Inv: $\text{self.InsumoBebida.tipoDeInsumo} = \text{Bebida}$

Pizza

13. Una pizza tiene dos insumos: kit y bollo de pizza:

Context Pizza

Inv: $\text{self.InsumosComida.TipoDeInsumo} \rightarrow \text{Includes}(\text{Bollo}) \text{ and } \text{self.InsumosComida.TipoDeInsumo} \rightarrow \text{Includes}(\text{Kit})$

Empanada

14. Una empanada tiene dos insumos: kit y tapa de empanada:

Context Empanada

Inv: $\text{self.InsumosComida.TipoDeInsumo} \rightarrow \text{Includes}(\text{Tapa}) \text{ and } \text{self.InsumosComida.TipoDeInsumo} \rightarrow \text{Includes}(\text{Kit})$

Horno

15. El horno tiene por lo menos un módulo:

Context Horno

Inv: *self.CantidadDeModulos* ≥ 1

16. En cada módulo entra por lo menos una pizza y una empanada:

Context Horno

Inv: *self.PizzasXModulo* ≥ 1 and *self.EmpanadasXModulo* ≥ 1

17. La cantidad de módulos ágiles de un horno es una cantidad (es no negativa) y es a lo sumo igual al total de módulos del horno:

Context Horno

Inv: *self.ModulosAgiles* ≥ 0 and *self.ModulosAgiles* \leq *self.CantidadDeModulos*

18. En la pizzería hay solo 2 hornos:

Context Horno

Inv: *Horno.allInstances()* \rightarrow *size()* = 2

19. Todos los hornos tienen la misma política:

Context Horno

Inv: *Horno.allInstances()* \rightarrow forall(*h* | *h.politicasDe* = *self.politicasDe*)

20. Cada horno tiene un maestro distinto:

Context Horno

Inv: *Horno.allInstances()* \rightarrow forall(*h* | *h* \neq *self* implies *h.Maestro* \neq *self.Maestro*)

Producto

21. Los precios son valores positivos:

Context Producto

Inv: *self.precioActual* > 0

22. Los nombres de los productos son únicos:

Context Producto

Inv: *Producto.allInstances()* \rightarrow forall(*p* | *p* \neq *self* implies *p.nombre* \neq *self.nombre*)

Comida

23. Las comidas tardan un tiempo no nulo en cocinarse y en prepararse:

Context Producto

Inv: *self.tiempoPreparacion* > 0 and *self.tiempoCoccion* > 0

Preparacion

24. Todos los pedidos en estado en preparación y solo esos tienen Preparación:

Context Preparacion

Inv: *self.PedidoPreparado.estado* = *En Preracion* and (*Pedido.allInstances()* \rightarrow select(*p* | *p.estado* = *En Preparacion*) =

Pedido.allInstances() \rightarrow *select*(*p* | *p.estado* = *En Preparacion* and *p.PreparacionDePedido* \rightarrow *notEmpty()*)

25. Si la parte preparada de un pedido no es ninguna entonces el pedido es mixto (si fuera por ejemplo de solo pizzas y las pizzas ya están preparadas, el pedido ya estaría preparado totalmente):

Context Preparacion

Inv: *self.PartePreparada* \neq Ninguna implies (*self.PedidoPreparado.Productos* \rightarrow *exists*(*p* | *p.isKindOf*(*Pizza*)) and *self.PedidoPreparado.Productos* \rightarrow *exists*(*p* | *p.isKindOf*(*Empanada*)))

Item

26. El precio unitario es positivo:

Context Item

Inv: *self.precioActual* > 0

27. La cantidad es mayor que 0 (si fuera 0 el pedido no está llevando nada de ese producto, por lo cual el item no debería existir):

Context Item

Inv: *self.cantidad* > 0

28. Un item (de comida) tiene una cocción si y solo si su pedido está en estado Al Horno:

Context Item

Inv: if *self.CoccionDe* \rightarrow *notEmpty()* then *self.Pedidos.estado* = *Al Horno* else (*self.Pedidos.estado* \neq *Al Horno* or *self.Productos.isKindOf*(*Bebida*))

Cocción

29. La cantidad cocinada y la cantidad en el horno son no negativas:

Context Coccion

Inv: *self.cantidadEnElHorno* ≥ 0 and *self.cantidadCocinada* ≥ 0

30. En toda cocción la cantidad en el horno junto con la ya cocinada es menor o igual que la cantidad de total que había que cocinar para el pedido:

Context Coccion

Inv: *self.cantidadEnElHorno* + *self.cantidadCocinada* \leq *self.itemDe.cantidad*

Orden de preparación

31. Los pedidos que tienen un orden de preparación son todos los que están en estado ingresado:

Context Orden de preparación

Inv: *self.PedidoEnPosición.estado* = *Ingresado* and not (*Pedido.allInstances()* \rightarrow *exists*(*p* | *p.estado* = *ingresado* and *p.PosiciónPreparación* \rightarrow *isEmpty()*))

32. Las posiciones en el orden son todas diferentes y están en el rango de la cantidad de pedidos ingresados:

Context Orden de preparación

Inv: *self.Posicion* \leq *Orden de preparacion.allInstances()* \rightarrow *size()* and *self.Posicion* ≥ 1 and *Orden de preparacion.allInstances()* \rightarrow *forall*(*o* | *o* \neq *self* implies *o.Posicion* \neq *self.Posicion*)

Orden de cocción

33. Los pedidos que tienen un orden de preparación son todos los que están en estado preparado:

Context Orden de cocción

Inv: $\text{self.PedidoEnPosición.estado} = \text{Preparado and not } (\text{Pedido.allInstances()} \rightarrow \text{exists}(p \mid p.\text{estado} = \text{preparado and } p.\text{PosiciónCoccion} \rightarrow \text{isEmpty()}))$

34. Si tomamos todos los pedidos que esperan por un horno, vale que las posiciones son únicas y además están en el rango de la cantidad de pedidos que esperan por ese horno:

Context Orden de cocción

Inv: $\text{self.posición} \geq 1 \text{ and } \text{self.posición} \leq \text{Pedido.allInstances()} \rightarrow \text{select}(p \mid p.\text{hornoDe} = \text{self.pedidoEnPosición.hornoDe}) \rightarrow \text{size()} \text{ and } \text{Orden de coccion.allInstances()} \rightarrow \text{forall}(o \mid (o \leq \text{self and } o.\text{pedidoEnPosición.hornoDe} = \text{self.pedidoEnPosición.hornoDe}) \text{ implies } \text{self.posición} \leq o.\text{posición})$

Pedido

35. Los identificadores de pedido son únicos:

Context Pedido

Inv: $\text{Pedido.AllInstances()} \rightarrow \text{forall}(p \mid p \leq \text{self implies } p.\text{identificador} \leq \text{self.identificador})$

36. Los pedidos que tienen solo bebidas no están en los estados ingresados, en preparación, preparado o al horno :

Context Pedido

Inv: $\text{self.productos} \rightarrow \text{forall}(p \mid p.\text{isKindOf}(\text{Bebida})) \text{ implies } (\text{self.estado} \leq \text{ingresado and } \text{self.estado} \leq \text{en preparacion and } \text{self.estado} \leq \text{ingresado and } \text{self.preparado} \leq \text{al horno})$

37. Solo los pedidos que contienen únicamente bebidas no tienen horno asignado:

Context Pedido

Inv: $\text{if } \text{self.productos} \rightarrow \text{forall}(p \mid p.\text{isKindOf}(\text{Bebida})) \text{ then } \text{self.hornoDe} \rightarrow \text{isEmpty()} \text{ else } \text{self.hornoDe} \rightarrow \text{notEmpty()}$

38. Los pedidos de solo pizzas (sin empanadas) van al horno del maestro pizzero:

Context Pedido

Inv: $(\text{self.productos} \rightarrow \text{exists}(p \mid p.\text{isKindOf}(\text{Pizza})) \text{ and } \text{not } \text{self.productos} \rightarrow \text{exists}(p \mid p.\text{isKindOf}(\text{Empanada}))) \text{ implies } \text{self.hornoDe} \rightarrow \text{forall}(h \mid h.\text{maestro} = \text{pizzero})$

39. Los pedidos de solo empanadas van al horno de solo empanadas:

Context Pedido

Inv: $(\text{not } \text{self.productos} \rightarrow \text{exists}(p \mid p.\text{isKindOf}(\text{Pizza})) \text{ and } \text{self.productos} \rightarrow \text{exists}(p \mid p.\text{isKindOf}(\text{Empanada}))) \text{ implies } \text{self.hornoDe} \rightarrow \text{forall}(h \mid h.\text{maestro} = \text{Empanadero})$

40. Un pedido al horno tiene por lo menos algo cocinado o en el horno:

Context Pedido

Inv: $\text{self.estado} = \text{Al Horno implies } \text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Comida})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrdered-Set()} \rightarrow \text{first()}) \rightarrow \text{exists}(c \mid c.\text{cantidadAlHorno} + c.\text{cantidadCocinada} > 0)$

41. La cantidad de productos cocinados no excede la capacidad del horno:

Context Horno

Inv: $\text{Coccion.allInstances()} \rightarrow \text{select}(c \mid c.\text{itemDe.producto.isKindOf}(\text{comida}) \text{ and } c.\text{itemDe.pedido.hornoDe.includes}(\text{self})) \rightarrow \text{collect}(c \mid \text{if } c.\text{itemDe.producto.isKindOf}(\text{empanada}) \text{ then } c.\text{cantidadEnElHorno} / \text{self.empanadasXModulo} \text{ else } c.\text{cantidadEnElHorno} / \text{self.pizzasXModulo}).\text{sum()} \leq \text{self.cantidadDeModulos}$

42. La cantidad de empanadas de un pedido que se está cocinando es múltiplo del tamaño del módulo del horno salvo que no queden más empanadas por cocinar (por ejemplo si al pedido solo le quedaban 3 empanadas y entran 4 por módulo):

Context Pedido

Inv: $\text{pedido.estado} = \text{Al Horno} \text{ implies } (\text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Empanada})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{collect}(\text{cantidadEnElHorno}).\text{sum}().\text{mod}(\text{self.horno} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()).\text{empanadasXModulo}) = 0 \text{ or } \text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Empanada})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{forall}(c \mid c.\text{itemDe.cantidad} - c.\text{cantidadEnElHorno} - c.\text{cantidadCocinada} = 0))$

43. La cantidad de pizzas de un pedido que se están cocinando es múltiplo del tamaño del módulo del horno salvo que no queden más pizzas por cocinar:

Context Pedido

Inv: $\text{pedido.estado} = \text{Al Horno} \text{ implies } (\text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Pizza})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{collect}(\text{cantidadEnElHorno}).\text{sum}().\text{mod}(\text{self.horno} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()).\text{pizzasXModulo}) = 0 \text{ or } \text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Pizza})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{forall}(c \mid c.\text{itemDe.cantidad} - c.\text{cantidadEnElHorno} - c.\text{cantidadCocinada} = 0))$

44. La cantidad de empanadas cocinadas de un pedido es múltiplo del tamaño del módulo del horno, salvo que se cocinaran todas las del pedido:

Context Pedido

Inv: $\text{pedido.estado} = \text{Al Horno} \text{ implies } (\text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Empanada})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{collect}(\text{cantidadCocinada}).\text{sum}().\text{mod}(\text{self.horno} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()).\text{empanadasXModulo}) = 0 \text{ or } \text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Empanada})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{forall}(c \mid c.\text{item.cantidad} - c.\text{cantidadCocinada} = 0))$

45. La cantidad de pizzas cocinadas de un pedido es múltiplo del tamaño del módulo del horno, salvo que se cocinaran todas las pizzas del mismo:

Context Pedido

Inv: $\text{pedido.estado} = \text{Al Horno} \text{ implies } (\text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Pizza})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{collect}(\text{cantidadCocinada}).\text{sum}().\text{mod}(\text{self.horno} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()).\text{PizzasXModulo}) = 0 \text{ or } \text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Pizza})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{forall}(c \mid c.\text{itemDe.cantidad} - c.\text{cantidadCocinada} = 0))$

46. Si hay política normal, en cada horno hay a lo sumo un pedido en cocción parcial, es decir con cosas cocinadas y cosas sin cocinar:

Context Pedido

Inv: $(\text{self.estado} = \text{Al horno} \text{ and } \text{self.hornoDe} \rightarrow \text{asOrderedSet}().\text{first}().\text{politicadDe} = \text{Normal}) \text{ implies } (\text{self.item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Comida})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()) \rightarrow \text{exists}(c \mid c.\text{itemDe.cantidad} < c.\text{cantidadCocinada} + c.\text{cantidadAlHorno}) \text{ implies } \text{Pedidos.allInstances}() \rightarrow \text{forall}(p \mid (p <> \text{self} \text{ and } p.\text{estado} = \text{self.estado} \text{ and } p.\text{hornoDe} = \text{self.hornoDe}) \text{ implies } (p.\text{item} \rightarrow \text{select}(i \mid i.\text{producto.isKindOf}(\text{Comida})) \rightarrow \text{collect}(\text{coccionDe} \rightarrow \text{asOrderedSet}() \rightarrow \text{first}()).\text{forall}(c \mid c.\text{itemDe.cantidad} = c.\text{cantidadCocinada} + c.\text{cantidadAlHorno}))$

Hay una restricción similar que vale para la política ágil, en cuyo caso hay a lo sumo 2 pedidos al horno a medio cocinar y uno de ellos es chico. Dicho invariante es muy largo y engorroso de escribir en OCL por lo que decidimos omitir su especificación.

47. Un pedido local sin forma de pago es de tipo mesa:

Context Pedido Local

Inv: $\text{Self.formaDePago} \rightarrow \text{isEmpty}() \text{ implies } \text{self.origen} = \text{Mesa}$

Por ultimo, elegimos string para los numeros de telefono porque podrian empezar con 0s. Ocl provee una operacion para pasar un string a Integer pero desconocemos que ocurre si la conversion no se puede realizar, por lo que decidimos explicar este invariante en castellano.

Parte 4

Aspectos particulares de la solución

4.1. Ciclo de Pedido

Desde que un pedido es ingresado al sistema, recorre una sucesión de estados hasta que llega a manos del cliente o eventualmente es cancelado. Esta sucesión de estados es lo que denominamos **ciclo de vida de pedido**. A continuación detallamos los estados que componen este ciclo y como es la dinámica de las transiciones entre estados.

Los estados en los que puede encontrarse un pedido una vez que ingresó al sistema son:

- **Ingresado:** El pedido pasó los chequeos necesarios, se determinó que es posible llevarlo a cabo con éxito y se ingresó al sistema, yendo a parar a una cola donde espera a que la cocina esté lista para prepararlo.
- **En preparación:** El pedido ya fue recibido por la cocina y está siendo preparado por un cocinero.
- **Preparado:** El pedido ya fue preparado en su totalidad y está listo para ingresar al horno cuando haya lugar disponible.
- **En el horno:** El pedido (o al menos una de sus partes) está en el horno.
- **Listo:** El pedido está listo para ser entregado al cliente.
- **Enviado:** El pedido fue entregado al delivery para su entrega a domicilio.
- **Finalizado:** El pedido fue entregado con éxito (ya sea a la mesa o a domicilio).
- **Cancelado:** El pedido fue cancelado antes de su entrega.

En breve, un proceso ingresa al sistema cuando es requerido por un cliente por algún medio de comunicación. Una vez que la cocina está disponible para prepararlo, se indica a los maestros la tarea que debe ser realizada. Una vez completada la preparación de la comida, los responsables de cocina lo indican al sistema, que planifica de forma independiente la asignación de lugares en los hornos. Finalmente, después de que el pedido es horneado, se entrega a la mesa o al delivery según corresponda. Por último, cuando existe confirmación de la entrega se marca el pedido como finalizado y se almacena con fines de registro. El pedido puede ser cancelado en cualquier punto (salvo después de ser finalizado), con implicaciones varias en el sistema de control de stock.

Los estados presentados tienen utilidad para el sistema puesto que existe la necesidad de indicar al cliente sobre el estado de su pedido. Así, se podrá indicar si un pedido está siendo preparado o ya fue enviado al destinatario. Si bien para estos fines alcanza con 3 estados (preparando, enviando, finalizado), se agregan los demás con fines de control para el sistema. Además, la granularidad más fina permite al sistema hacer predicciones y organizar de formas más eficientes el uso de los recursos. Por ejemplo, el controlar los tiempos de preparación por separado de los de horneado, resulta más sencillo, en caso de un cuello de botella en la cocina, determinar cual es el factor limitante.

En la figura 4.1 se observan los posibles estados de un pedido (no compuesto por solo bebidas) y las transiciones que pueden realizarse entre dos de ellos. Si bien este modelo carece de formalidad, es útil como una primera aproximación a la dinámica del sistema.

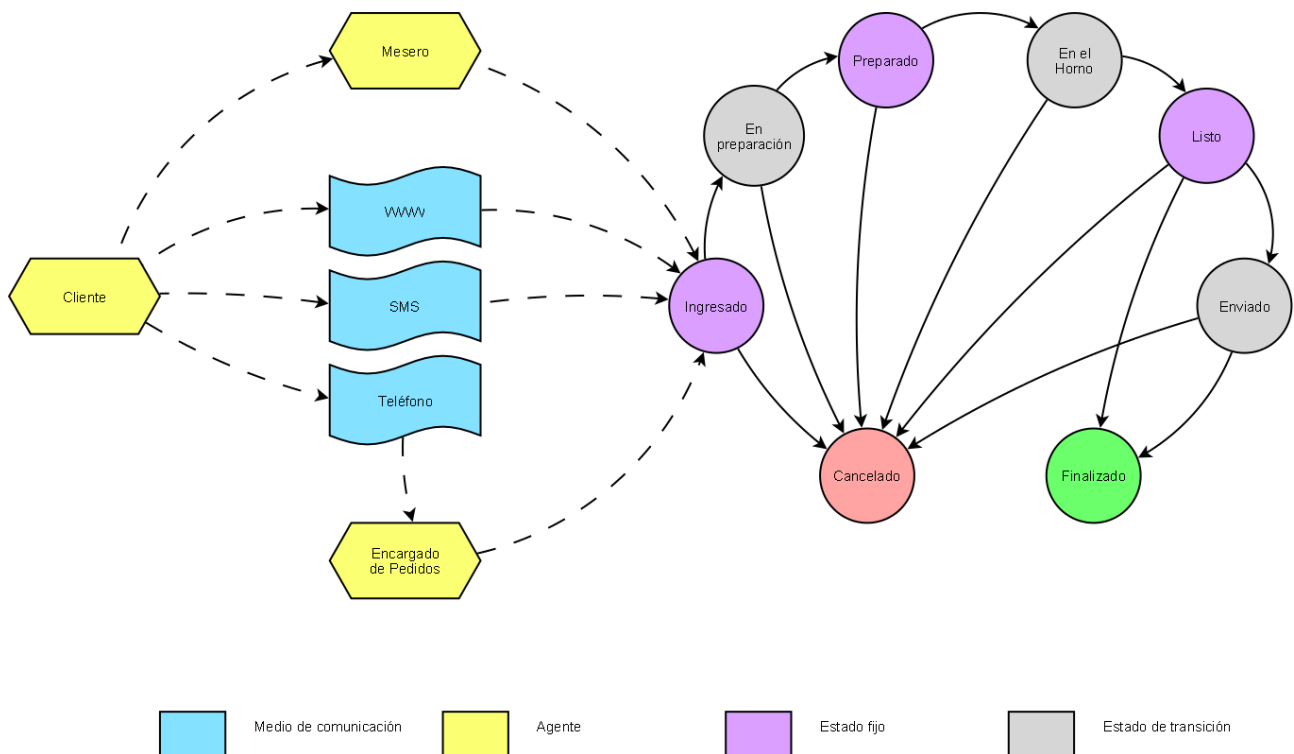


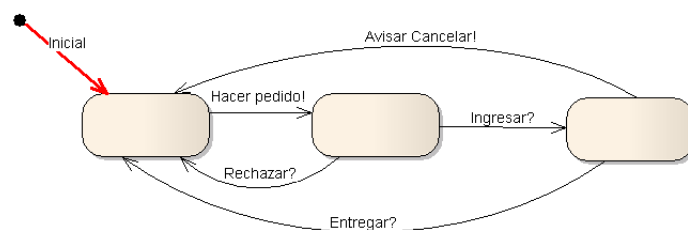
Figura 4.1: Ciclo de Vida de un pedido

Resulta de interés considerar un caso simple donde un cliente ingresa un pedido con el objetivo de seguir al mismo hasta que éste finaliza. A continuación modelaremos mediante FSM el ciclo de vida de un pedido, sin entrar en grandes detalles, pero aún así pudiendo observar cuestiones interesantes.

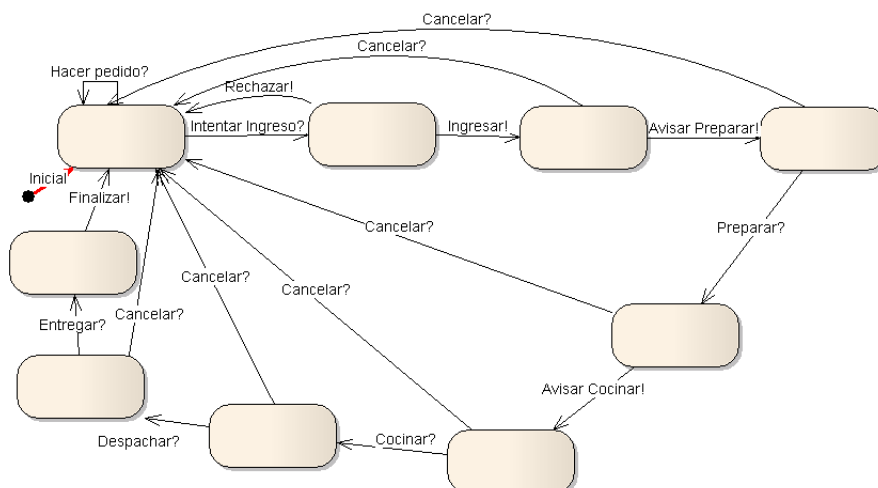
El escenario que vamos a modelar consiste en un cliente que hace pedidos de a uno (pudiendo cancelarlos), el encargado de pedidos que realiza el ingreso de los mismos y despacha los pedidos, el maestro que prepara y cocina el pedido, el sistema que monitorea el estado de los pedidos y el delivery que se ocupa de la entrega de los mismos. Si bien este modelo no tiene en cuenta la existencia de otros pedidos o clientes, nos sirve para analizar gran parte de la funcionalidad involucrada en el tratamiento de los pedidos.

En estas FSM podemos ver en qué partes del ciclo de vida tienen relevancia los distintos agentes. De este modo observamos como el sistema es el que permite o no el ingreso de un pedido, rechazándolo en el caso de que el stock sea insuficiente, así como también se encarga de avisar al maestro que prepare o cocine un determinado pedido. Además vemos como el sistema permite que el pedido avance hasta llegar a estar listo para despachar y monitorea casi en su totalidad las transiciones de estado del pedido.

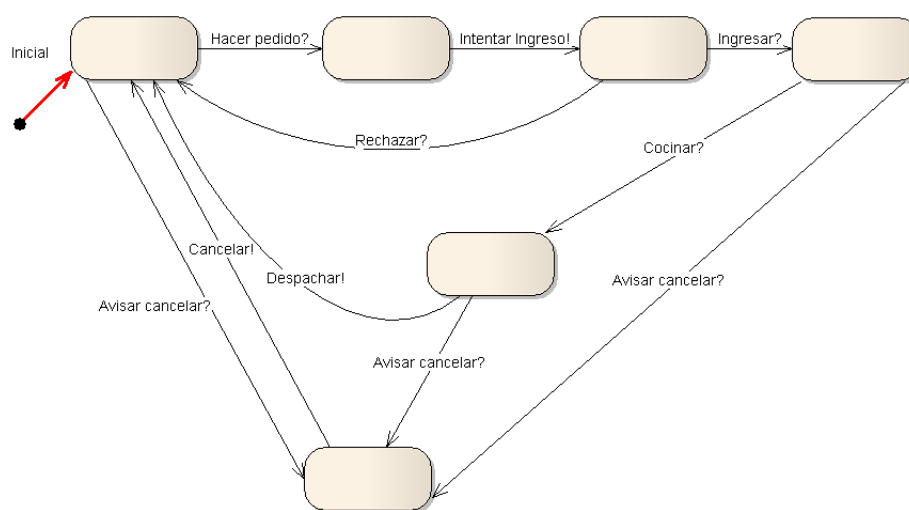
Es interesante notar también que desde que ingresa hasta que es entregado un pedido, el cliente puede en todo momento cancelarlo. La dinámica de cancelación de un pedido es objeto de mayor detalle en la sección 4.5.



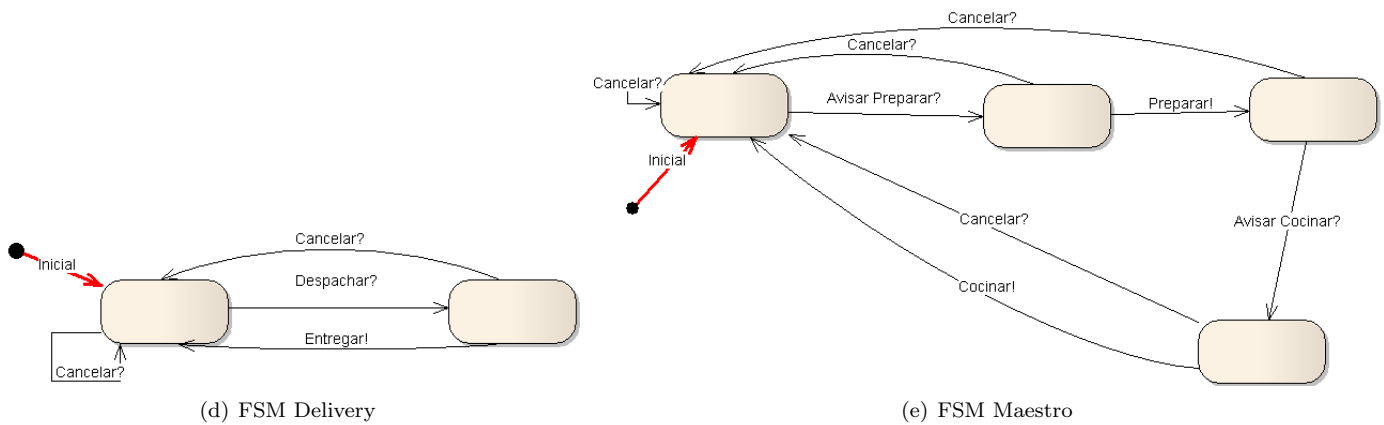
(a) FSM Cliente



(b) FSM Sistema



(c) FSM Encargado de Pedidos



Por último, la FSM que resulta de la composición de las anteriores nos describe de forma global el ciclo de vida de un pedido, de forma análoga al primer diagrama presentado.

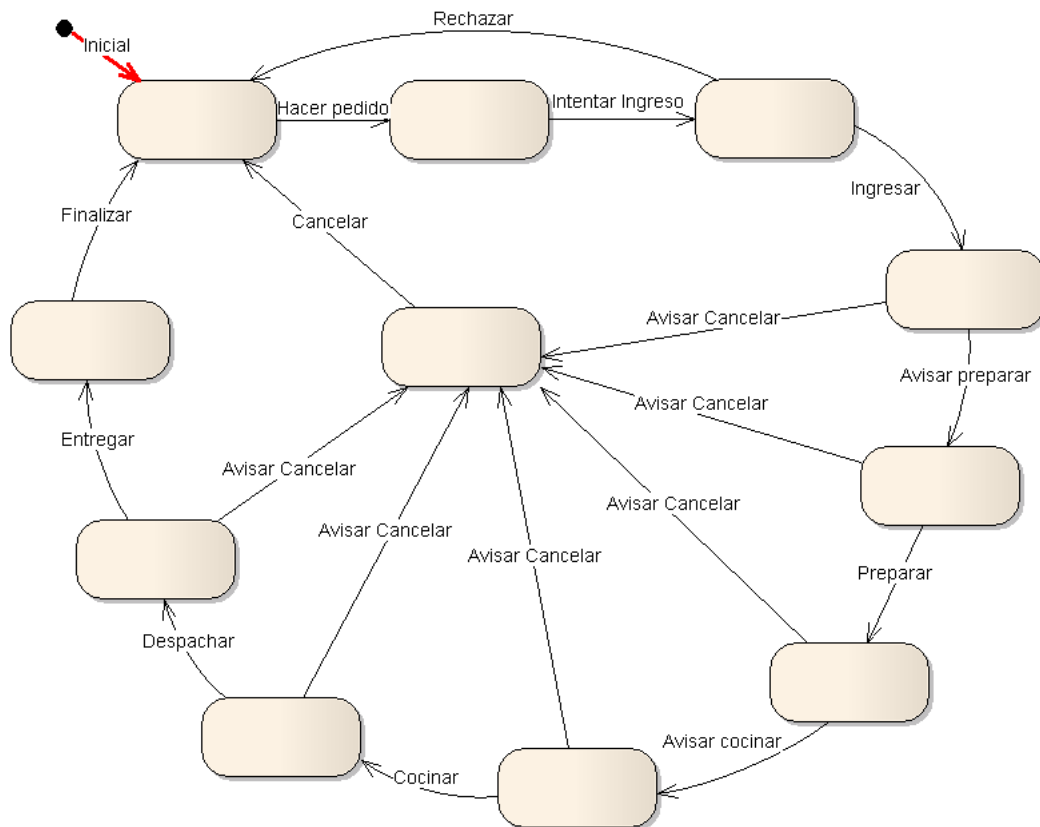


Figura 4.2: FSM Pedido = FSM Cliente || FSM Encargado de Pedidos || FSM Delivery || FSM Maestro || FSM Sistema

4.2. Ingreso de pedidos

Los pedidos se distinguen en dos categorías esenciales:

- **Local**, si fue realizado en la pizzería por un cliente de forma personal
- **Remoto**, si fue ingresado al sistema por un cliente desde su hogar u otro lugar, por vía telefónica, Web o SMS.

Todos los pedidos tienen un único horno asignado, a excepción de los pedidos que no incluyen comidas y por lo tanto no deben ser cocinados. Este horno se determina automáticamente, a excepción de los pedidos mixtos en que el encargado de pedidos decide a qué horno serán asignados.

Cada vez que se ingresa un pedido, se realiza un chequeo de la disponibilidad del stock, a fin de evitar tomar pedidos que no se pueden satisfacer. Los pedidos insatisfacibles son uno de los problemas mayores en la operación de la pizzería, y es un requerimiento importante atacar este problema.

La siguiente FSM presenta un modelo del control de stock para el ingreso de pedidos.

Supongamos que en la pizzeria hay hasta k pedidos, hay e tipos de empanadas y i tipos de pizzas. Consideremos las siguientes variables, todas ellas comenzando con un valor $STOCK$ constante.

stock pizza 1 : $[0...MAX_{STOCK}]$, ... stock pizza i : $[0...MAX_{STOCK}]$, stock empanada 1 : $[0...MAX_{STOCK}]$, ... stock empanada e : $[0...MAX_{STOCK}]$, stock bollo : $[0...MAX_{STOCK}]$, stock tapas : $[0...MAX_{STOCK}]$,

Análogamente, controlamos también el stock de los *kits* necesarios para la preparación de pizzas y empanadas. $pizza\ n\ m$ indica cuantos kits de pizza del tipo n son necesarios para el pedido m . Análogamente tenemos la misma situación para los kits de empanadas:

pizza 1 1 : $[0...MAX]$... pizza 1 k : $[0...MAX]$ pizza 2 1 : $[0...MAX]$... pizza i k : $[0...MAX]$

empanada 1 1 : $[0...MAX]$... empanada 1 k : $[0...MAX]$ empanada 2 1 : $[0...MAX]$... empanada e k : $[0...MAX]$

Por último, registramos también la cantidad de pizzas y empanadas que requiere cada pedido:

pizzas 1 : $[0...MAX]$... pizzas k : $[0...MAX]$

empanadas 1 : $[0...MAX]$... empanadas k : $[0...MAX]$

Entonces, la FSM de la figura 4.3 modela el ingreso de pedidos para cada pedido $j = 0...k$.

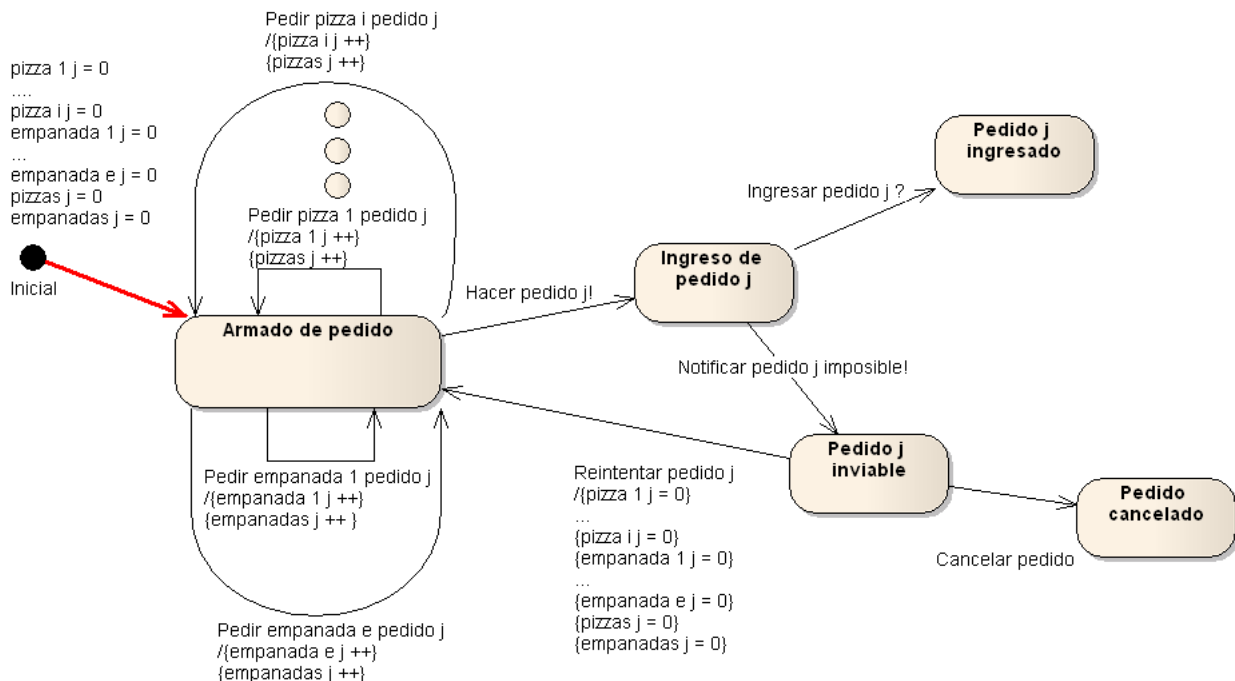


Figura 4.3: FSM ingreso de pedido j

Se observa en esta FSM que durante el estado de armado de un pedido se seleccionan distintos productos con una cantidad dada de cada uno de ellos. Cuando el pedido está armado, se intenta hacer el pedido. En esa situación, el mismo puede ser ingresado o puede ser cancelado, pudiendo intentar hacer un pedido diferente o desistir.

En la figura 4.4 vemos la FSM que se encarga de controlar el stock para evitar el ingreso de pedidos no satisfacibles. Esta FSM toma esta decisión en base a la cantidad de stock disponible y de los productos necesarios para preparar el pedido en cuestión. Las guardas largas en esta FSM lo que hacen es controlar que el stock alcance para que el pedido que se intenta ingresar pueda ser satisfecho.

La FSM que regula el ingreso de pedidos en base al stock se obtiene como: FSM Ingreso Controlado de Pedidos = FSM ingreso de pedidos 1 || ... || FSM ingreso de pedidos k || FSM chequeo de stock

Analizando las trazas de la composición, podemos notar que cuando se hace un pedido, la decisión de si se puede ingresar o no, así como el decremento del stock componen un evento atómico. Esto implica que es imposible que un pedido esté a punto de ingresar y antes de que se decremente el stock un segundo pedido es ingresado. Dicha situación constituye una condición de competencia y podría hacer que se tomen pedidos que luego no se puedan satisfacer.

A continuación detallaremos los distintos tipos de pedidos, así como también describiremos las actividades propias de su ingreso.

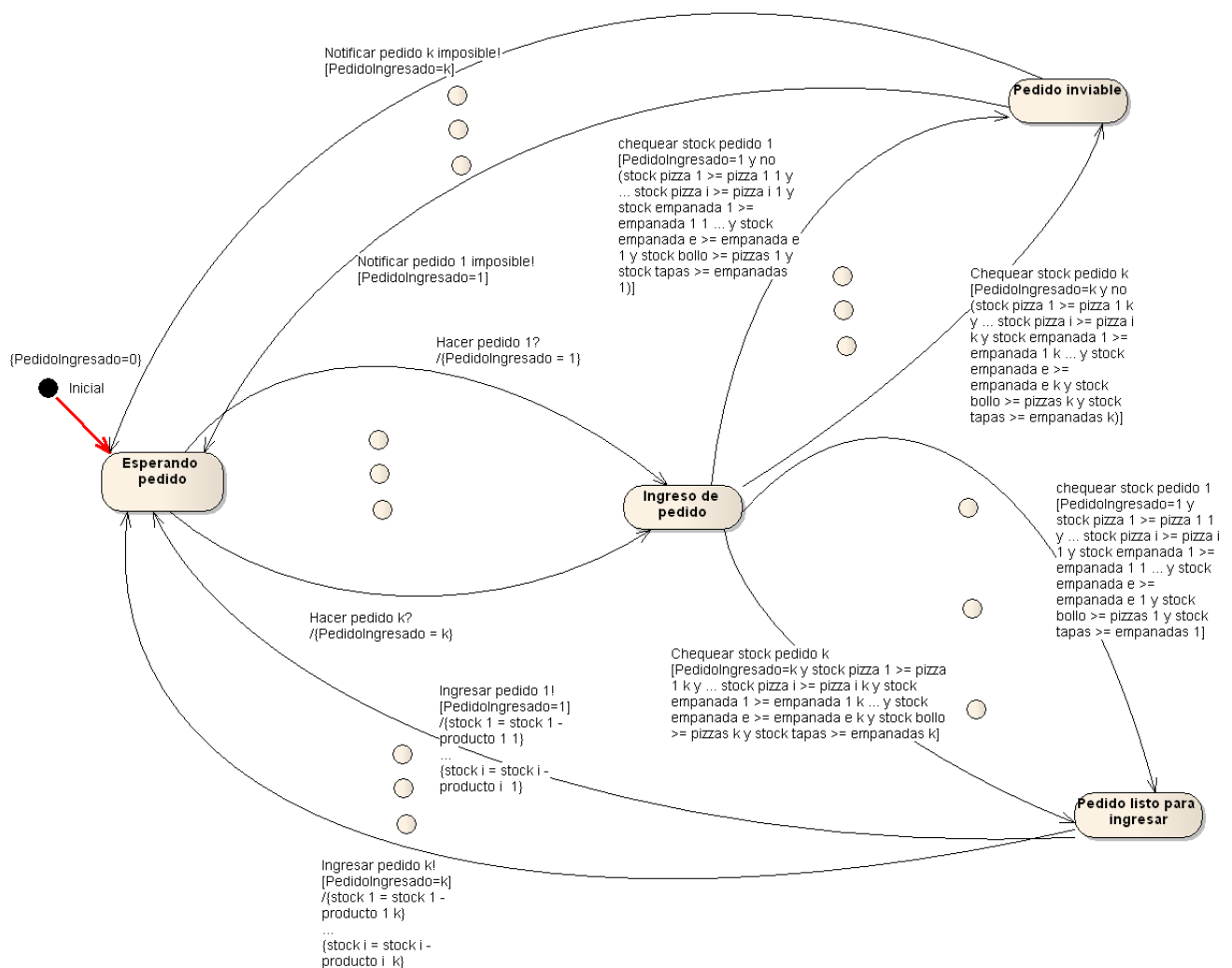


Figura 4.4: FSM chequeo de stock

4.2.1. Pedidos Remotos

Los pedidos remotos requieren que los usuarios estén registrados en el sistema. Esto se hace por cuestiones de seguridad, a fin de evitar la realización de pedidos por parte de usuarios anónimos que podrían cancelar constantemente o dar direcciones falsas. El registro de usuarios se realiza personal o telefónicamente. Fue una decisión de diseño no permitir el registro de

usuarios vía Web, ya que consideramos que el método no era del todo fiable (una vez más, es fácil crear usuarios ficticios y hacer pedidos en broma).

Si un usuario desea poder hacer pedidos vía Web, deberá elegir un nombre de usuario y una contraseña al momento de registrarse, así como proveer una dirección de correo electrónico. Si se desean hacer pedidos SMS, el usuario deberá brindar un número de celular, al que se “autoriza” para efectuar pedidos a nombre de ese usuario. Para todo tipo de pedidos que involucren *delivery*, el usuario deberá registrar una dirección. Como no es necesario registrarse para comer en el local, esto redundante en que todos los usuarios registrados lo hacen inicialmente para recibir *delivery* y por lo tanto todos los usuarios registrados deben tener esta información.

A continuación describiremos los procedimientos para la realización de pedidos remotos.

Pedidos Telefónicos

El cliente que desee realizar un pedido telefónico debe llamar al número que la pizzería ha dispuesto para tal fin. Allí es atendido por el encargado de pedidos, al que deberá brindar información de identificación. Esta información debe ser suficiente para individualizarlo entre los usuarios registrados (nombre, dirección o nombre de usuario). El encargado corrobora entonces el número desde el que está llamando el usuario para verificar su identidad (eventualmente, si el cliente llama desde otro número puede identificarse con usuario y password si lo tuviera).

Si el usuario no estaba registrado, el encargado debe sugerirle la posibilidad de registrarse. Si el usuario no está registrado y no desea registrarse, el encargado no tomará el pedido.

Una vez autenticado el usuario, el mismo procede a dictar su pedido al encargado. Hecho esto, el encargado de pedidos intenta ingresar el pedido al sistema, el cual verifica si es posible satisfacer dicho pedido. En caso de no ser posible, el sistema indicará mediante un mensaje de error que el pedido no es posible de realizar en ese momento e indicará que producto(s) no estaba(n) disponible(s). El cliente puede entonces optar por un pedido diferente.

Si el pedido ingresa con éxito, el sistema lo registra en el estado Ingresado. Se genera además un identificador único de pedido, se estima el tiempo de entrega del mismo y se calcula el costo total. Además si el pedido es mixto, se brinda al encargado la posibilidad de seleccionar el horno donde se cocinará el pedido.

El siguiente diagrama de actividad permite ilustrar esta situación:

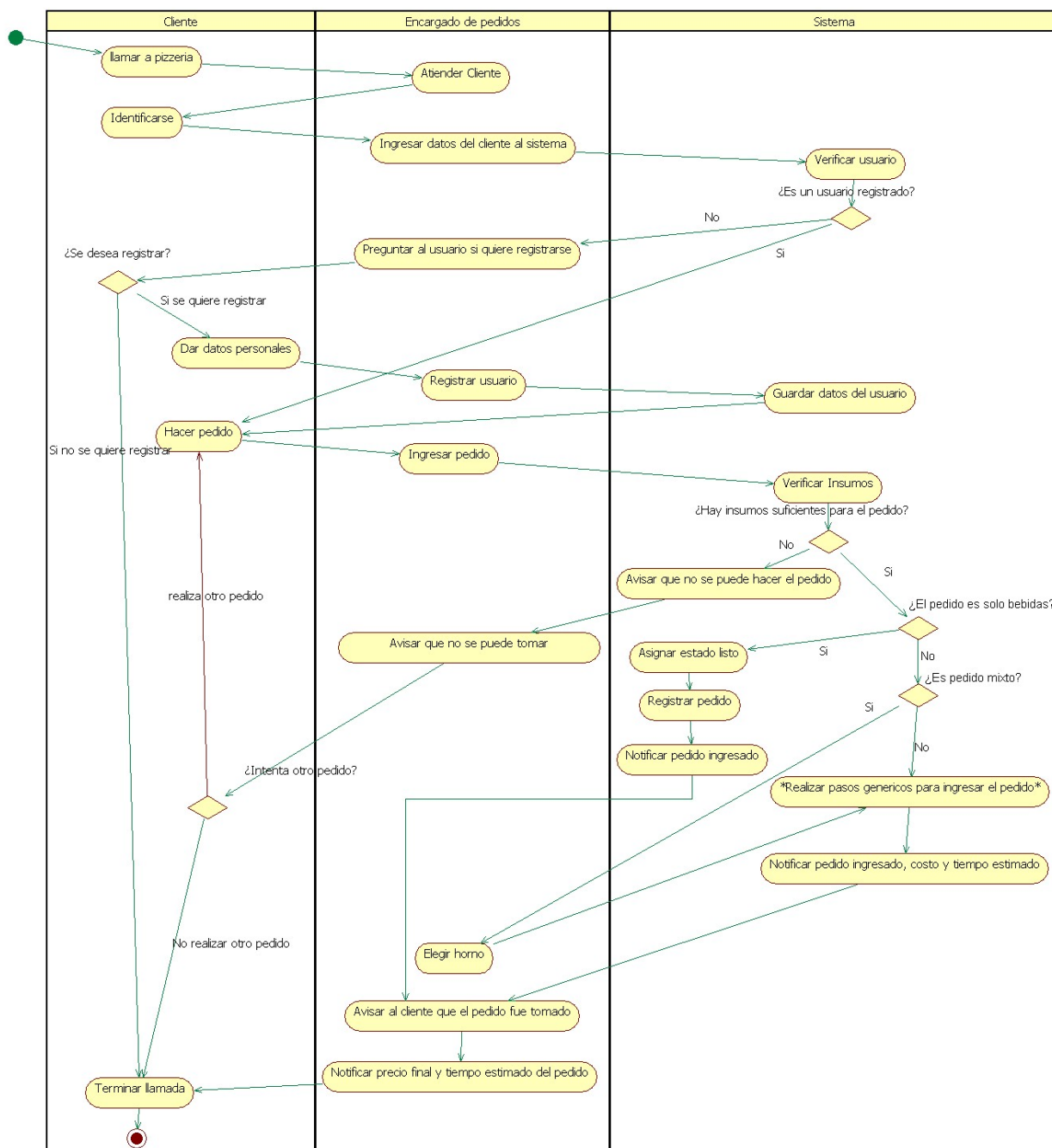


Figura 4.5: Diagrama de actividad del ingreso de un pedido telefónico

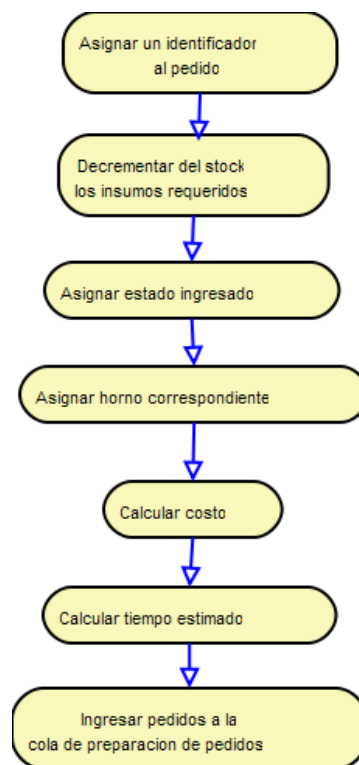


Figura 4.6: Partes comunes en el ingreso de pedido

Una cosa que queremos hacer notar del diagrama es que permite observar la relación existente entre el caso de uso de ingreso de pedido telefónico y el registro de un nuevo cliente. En el diagrama de casos de uso podíamos ver esta relación mediante un extiende, en el diagrama podemos ver claramente la misma relación: Si el cliente no estaba registrado, puede optar por registrarse y hacer el pedido o no hacer el pedido.

Pedidos Web

Para que los clientes puedan realizar pedidos vía Web, la pizzería contará con una sitio en línea. El cliente que desee hacer un pedido por esta vía debe ingresar a la página y autenticarse con su nombre de usuario y contraseña, que previamente habrá obtenido como se describió en 4.2.1.

Una vez autenticado, el cliente tendrá la posibilidad de ingresar un pedido. Entonces, el cliente podrá elegir qué productos desea y en qué cantidad a partir de una lista en línea. Dicha lista solo contiene productos que están disponibles (los que ya están fuera de stock no se incluyen). Cuando marcó lo que deseaba puede finalizar el proceso de elección de productos y proceder a ingresar el pedido. El sistema verifica entonces la viabilidad del pedido, y en caso de no ser posible realizarlo notifica al cliente mediante un mensaje, indicando qué producto no se encuentra disponible. El cliente puede volver a ingresar un pedido diferente si así lo desea.

En caso de que sea posible registrar el pedido, se estima el tiempo del mismo y se calcula el costo total. Esta información se muestra al usuario por pantalla. A continuación se le permite al usuario elegir si prefiere pagar contra entrega al *delivery* o con tarjeta de crédito. En este último caso, la operación se hace íntegramente a través de Internet, gracias a la interacción del sistema con un servicio adecuado. Si la operación con tarjeta falla, se muestra un mensaje de error y se permite al usuario reingresar los datos de la misma o cambiar de forma de pago. Hecho esto, se confirma todo el pedido. Si entre la elección de los productos y la confirmación del pedido se acaba el stock de materias primas, se muestra un error y se vuelve a la pantalla de selección de productos (esto debería ser muy raro).

Una vez que el pedido fue ingresado, si se pagaba con tarjeta, se hace efectivo el cobro. El pedido se encola entonces automáticamente en la cola de pedidos donde es atendido regularmente.

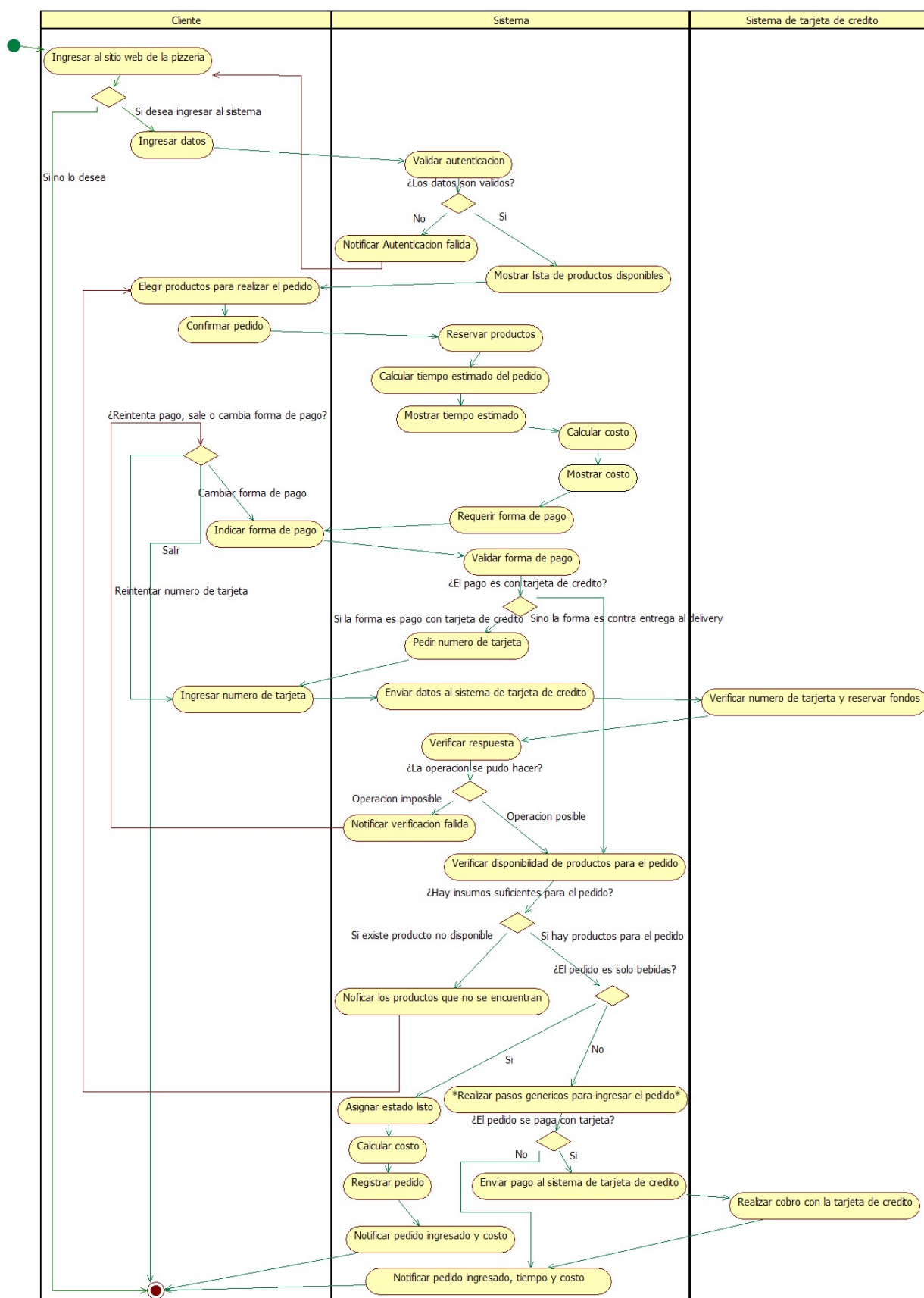


Figura 4.7: Diagrama de actividad del ingreso de un pedido Web

En este diagrama podemos ver que muchos de los pasos son comunes al ingreso de pedidos telefónicos, sin embargo hay dos grandes diferencias que queremos marcar. En primer lugar en al hacer un pedido web, este puede pagarse en el momento mediante la interacción con el sistema de tarjetas de credito. Y en segundo lugar, vemos que en este caso a diferencia de en el anterior, la asignación del horno a un pedido mixto es automática, no es hecha por el encargado de pedidos.

El horno que se asigna es el que tenga menos tiempo de espera estimado, donde el tiempo de espera estimado se obtiene como la suma de los tiempos de cocción estimado de los productos de los pedidos que esperan por un horno. Describimos a continuación la operación (como Query sin contexto) que permite obtener el horno que corresponde.

```
TiempoEsperaHorno(h:Horno):Integer = h.pedidos → select( p | p.estado = preparado or p.estado = Ingresado or
p.estado = en preparacion) → collect(p | p.productos → select(pr | pr.isKindOf(Comida)) → collect( pr | pr.asOclType(Comida).
tiempoDeCoccion) → sum() ) → sum()
```

```
HornoAIngresar():Horno = Horno.allInstances() → select( h | Horno.allInstances()-¿forall(h1 | TiempoEsperaHorno(h)
<= TiempoEsperaHorno(h1))) → ¿asOrderedSet() → ¿first()
```

Pedidos SMS

El cliente que desee realizar un pedido vía SMS debe enviar un mensaje al servidor SMS de la pizzería, junto con el pedido que quiere hacer. Para esto, su número de celular debe estar registrado en la base de datos del sistema. Si el celular no estaba registrado, el sistema descarta el pedido y le envía al cliente un mensaje indicándole la necesidad de registrar su número de celular, y el número para comunicarse telefónicamente y proceder a dicho registro.

Para identificar los productos que desea ordenar, el cliente cuenta con un catálogo con códigos de producto ya sea en papel o consultable vía Internet.

Análogamente al pedido vía Web, el sistema verifica si es posible satisfacer el pedido. En caso de no ser posible, el sistema le envía un mensaje al cliente para avisar que el pedido no podrá ser satisfecho e indicando qué producto no está disponible. Si el pedido sí podía ser tomado, el sistema lo registra como ingresado, genera un identificador de pedido, estima el tiempo del mismo, calcula el costo total y luego de todo esto procede a enviarle un mensaje de respuesta de SMS con esos datos al cliente.

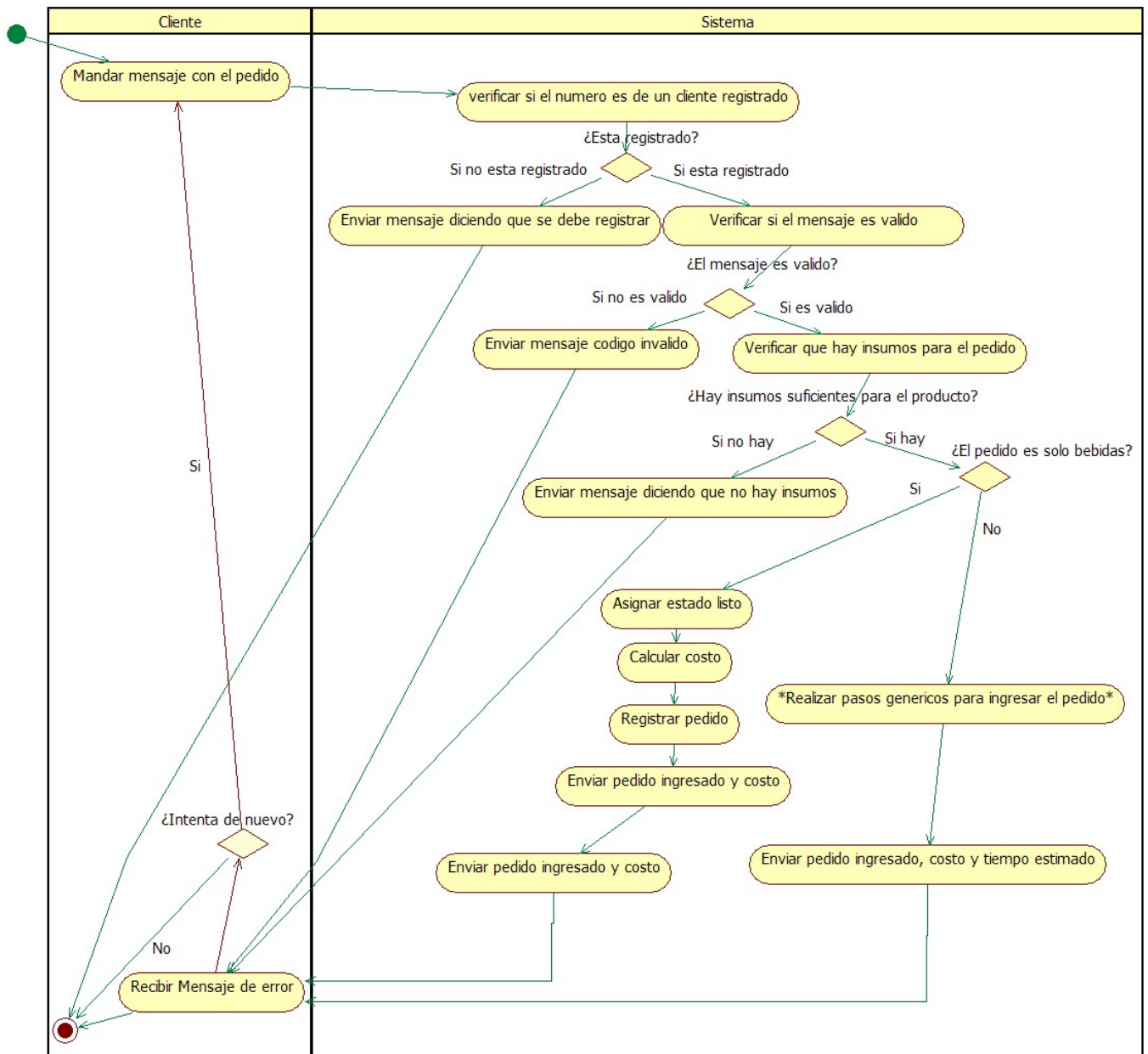


Figura 4.8: Diagrama de actividad del ingreso de un pedido SMS

4.2.2. Pedidos locales

Los pedidos locales pueden realizarse a los mozos, quienes cuentan con un PDA donde ingresan los datos de los mismos. Un cliente dicta su pedido a algún mozo, quien marca los productos y sus cantidades en su PDA. Hecho esto, el mozo ingresa el pedido, lo cual se realiza de forma inalámbrica dentro del local. En ese momento el sistema verifica la factibilidad del mismo, y si no es posible realizar el pedido alerta al mozo para que informe de la situación al cliente.

Si es posible ingresar el pedido el sistema genera un identificador del pedido, estima el tiempo del mismo y calcula el costo total. Análogamente a las otras formas de ingreso de pedido, el sistema elige automáticamente el horno asociado al pedido si correspondiera, o en caso de pedidos mixtos permite al encargado realizar esa elección.

Por otro lado, un pedido local (de mostrador) se solicita al encargado de pedidos, quién registra los productos del pedido de manera similar a como registra los pedidos telefónicos, con la excepción de que en este caso podría no tener los datos del

usuario.

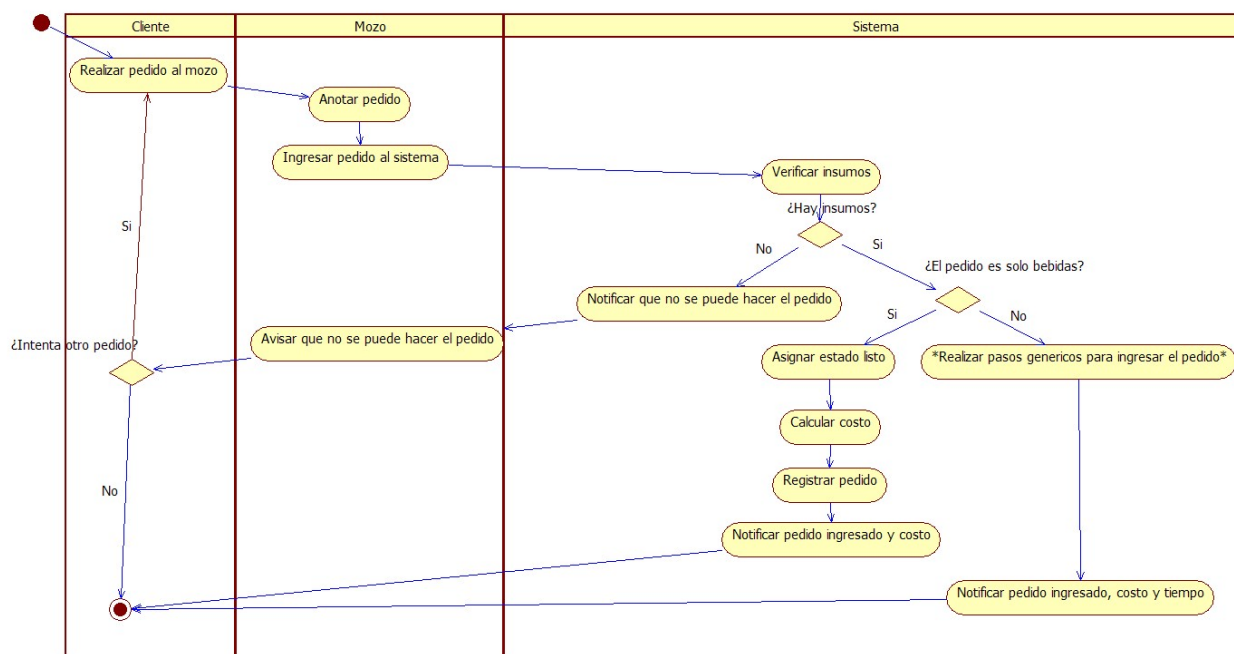


Figura 4.9: Diagrama de actividad de toma de un pedido por parte del mozo



Figura 4.10: Interfaz del mozo: Puede ingresar o cerrar pedidos

4.3. Registro de usuarios

El registro de usuarios es responsabilidad del encargado de pedidos, y se realiza por vía telefónica así como también personalmente.

Para registrar a un usuario el encargado de pedidos debe tomar los siguientes datos:

- Nombre
- Apellido
- Dirección
- Teléfono
- Usuario y Password (opcional, solo necesario para hacer pedidos vía Web)
- Celular (opcional, solo necesario para poder hacer pedidos vía SMS)

Una vez registrado, el usuario puede en otro momento solicitar modificar algún dato, o agregar algún dato opcional.

Veamos operacionalmente como funciona el registro de un usuario. La siguiente operación permite registrar un usuario nuevo sin datos Web ni SMS:

Operation: RegistroUsuario (*nombreC : String, apellidoC : String, CalleC : String, NumeroC : Integer, LocalidadC : String, DepartamentoC : String, TelefonoC : String*):

```
pre { Direccion.allInstances() → forall(d | d.telefono <> TelefonoC }
post { Cliente.allInstances() → size() = Cliente@pre.allInstances() → size() + 1 }
post { (Cliente.allInstances() - Cliente@pre.allInstances()) → forall(c | c.ocIsNew() and c.Nombre =
nombreC and c.apellido = apellidoC) }
post { Direccion.allInstances() → size() = Direccion@pre.allInstances() → size() + 1 }
post { (Direccion.allInstances() - Direccion@pre.allInstances()) → forall(d | d.ocIsNew() and d.calle =
CalleC and d.numero = NumeroC and d.localidad = LocalidadC and d.departamento = DepartamentoC
and c.telefono = TelefonoC) }
```

De forma análoga se pueden definir operaciones que agreguen datos para realizar pedidos por SMS o vía Web. A modo de ejemplo, veamos la operación para el agregado de datos Web:

Operation: AgregarDatosWeb (*c : Cliente, nick : String, clave : String*):

```
pre { not (DatoWeb@pre.allinstances().pass → includes(nick)) }
pre { Clientes.allInstances() -¿includes(c) }
pre { c.userYPass() → isEmpty() }
post { DatoWeb.allinstances() → size() = DatoWeb@pre.allinstances() → size()+1 }
post { (DatoWeb.allinstances() - DatoWeb@pre.allinstances()) → forall(d — d.usr = nick and d.pass =
clave and d.datosDe = c) }
```

4.4. Facturación

La facturación de los pedidos de la pizzería está a cargo del software de facturación que posee el establecimiento y por lo tanto no es responsabilidad del sistema aquí especificado. El sistema solo provee la facilidad de generar un archivo con los datos necesarios para realizar la factura en el momento adecuado.

En el caso de que el pedido sea remoto o haya sido hecho en el mostrador, dicho archivo se genera cuando el pedido está listo para salir a entrega. En el caso de pedidos realizados por clientes que están sentados en el local, la facturación se hace cuando el cliente indica al mozo que desea pagar ya que es necesario determinar la forma de pago.

Si el cliente está registrado, los datos del mismo son parte de este archivo. Para clientes que acuden en persona al local y no estén registrados, los campos de identificación simplemente son dejados en blanco.

4.5. Cancelación

La cancelación de un pedido se realiza únicamente por vía telefónica o personalmente. Esta decisión se tomó de forma análoga a la de solo permitir el registro de usuarios por estos medios: permitir hacer cancelaciones por Internet puede traer problemas a raíz del ciber vandalismo. Una cancelación se puede realizar siempre y cuando no se haya entregado el pedido, es decir que el pedido no se encuentre en estado finalizado.

Los productos ya preparados se descartan al cancelarse el pedido. En cambio, los insumos no utilizados regresan al stock. Dado que no se informa el estado de preparación de cada ítem del pedido por separado, el sistema no puede discernir frente a una cancelación exactamente cuantos insumos deben volver al stock. Por esta razón, el sistema muestra al maestro un mensaje informándole que el pedido se canceló y pidiéndole que indique qué parte de los insumos se puede reutilizar. Para pedidos mixtos, el tratamiento de esta situación es individual para cada cocinero.

Para cancelar un pedido vía telefónica el usuario debe identificarse, y luego debe indicar qué pedido quiere cancelar. El procedimiento de verificación de identidad es análogo al caso del ingreso de pedidos por esta vía. En el caso de un pedido hecho en el local, el cliente debe avisar al mozo, que notifica al encargado de pedidos, quien cancela el pedido, o bien puede avisar directamente al encargado de pedidos.

Por último, también se considera cancelado un pedido que no puede ser entregado por el delivery o que debería abortarse por cualquier otra razón. El sistema permite al encargado de pedidos indicar la razón de esta cancelación para fines estadísticos.

A continuación presentamos un diagrama de actividad donde se ponen de manifiesto las actividades que se llevan a cabo cuando el cliente desea cancelar un pedido.

4.6. Cola de pedidos

Se entiende por “cola de pedidos” a la cola que determina el orden en el cual se prepararán los pedidos ingresados al sistema. Esta cola es la que determina qué pedido ingresado debe comenzar a prepararse a continuación. Dicha cola se gestiona de manera automática, pero su orden puede ser modificado manualmente por el encargado de pedidos. Ni bien un producto comienza a prepararse (en su totalidad o alguna de sus partes), el mismo deja de pertenecer a esta cola. Cuando todas las partes de un pedido fueron preparadas, el pedido deja de estar en preparación para pasar a estar en estado preparado.

A continuación exhibimos un prototipo de pantalla de la interfaz que tendrá el encargado de pedidos para controlar la cola de pedidos. Usando esta interfaz, el encargado puede reubicar un pedido para aumentar o disminuir su prioridad. En el ejemplo, el encargado ha dado una prioridad mayor al pedido 4 que fue ingresado recientemente.

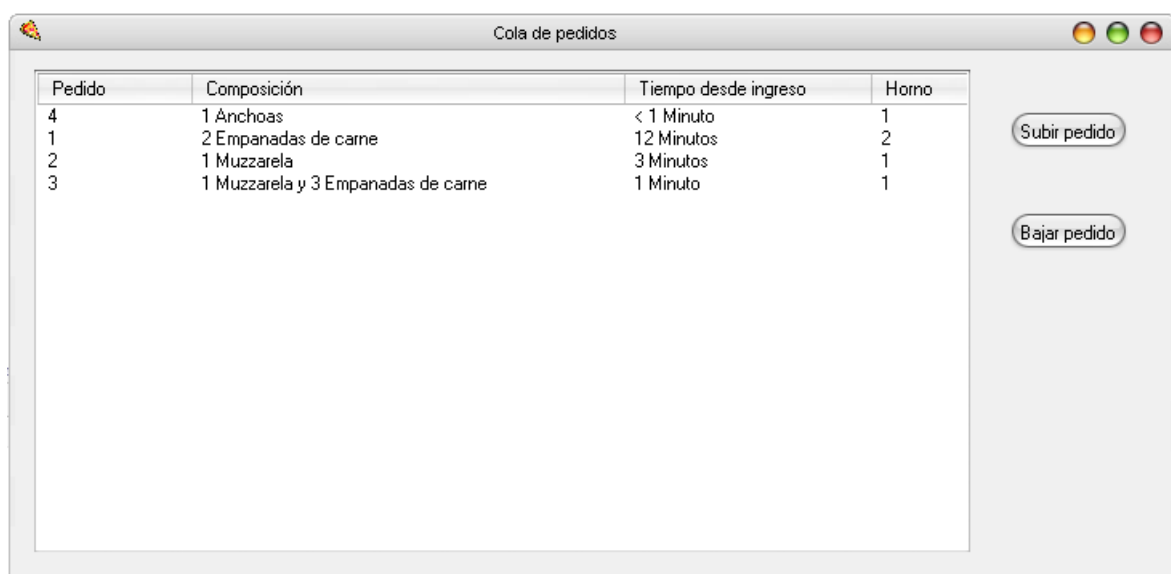


Figura 4.12: Prototipo de la vista de la cola de pedidos del encargado

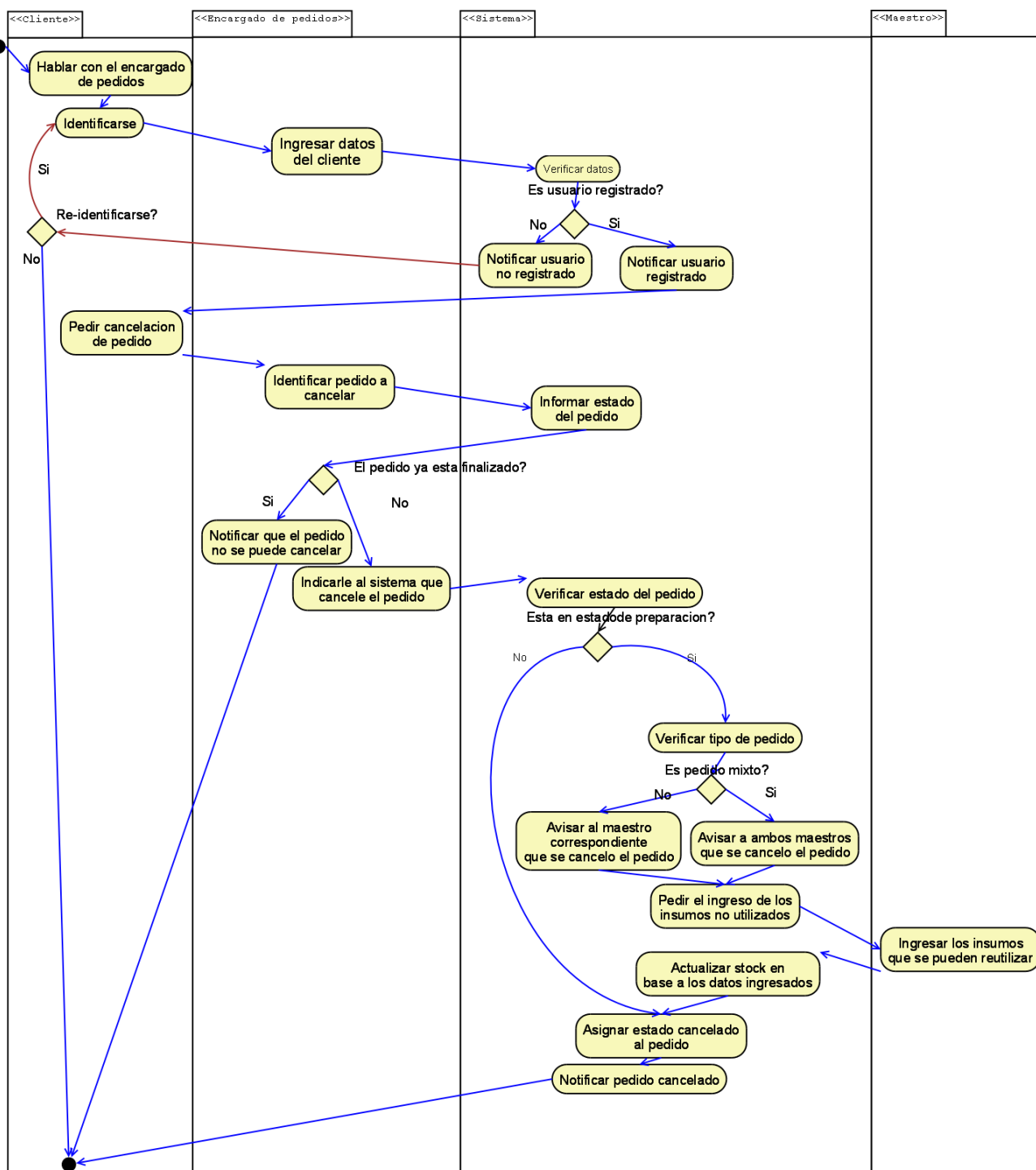


Figura 4.11: Diagrama de actividad de cancelación (personal y telefónica)

Modelaremos la forma en la que se obtiene el próximo pedido a preparar luego de que el maestro informa que terminó de preparar un pedido mediante operaciones. Como es necesario que la operación en cuestión devuelva el próximo pedido, utilizamos *queries* libres de contexto. Supongamos que el maestro pizzero terminó su labor e indica al sistema de esta situación. Al mismo tiempo, hay un pedido listo para comenzar a ser preparado. La idea es buscar primero entre los pedidos mixtos que ya empezó a preparar el empanadero, y elegir de entre estos el que entró primero, ya que decidimos que una vez que entran a la cocina los pedidos se manejan según una política *first in first out* (FIFO). Si no hay ningún pedido en preparación por parte del otro maestro, se busca en la cola de ingreso. En caso de que no hubiera pedidos en esta última cola tampoco, el sistema no indica al maestro que prepare nada. Suponemos por simplicidad que existe un pedido en alguna de las dos colas.

ProximoPedidoPizzero():Pedido = if PedidosEnPreparacionEmpanadera() → notEmpty() then PrimerPedidoPizzeroEnPreparacion() else PrimerPedidoPizzeroCola()

PrimerPedidoPizzeroEnPreparacion():Pedido = PedidosEnPreparacionEmpanadera() → select(p | PedidosEnPreparacionEmpanadera() → forall(p2 | p2.PreparacionDePedido → asOrderedSet() → first().fechaComienzo >= p.PreparacionDePedido → asOrderedSet() → first().fechaComienzo)) → asOrderedSet() → first()

esMixto(p:pedido):Boolean = p.productos → exists(pr | pr.isKindOf(Empanada)) and p.productos → exists(pr | pr.isKindOf(Pizza))

TienePizzasPorPreparar(p: pedido): Boolean = p.PreparacionDePedido → asOrderedSet() → first().PartePreparada <> Pizzas

PedidosEnPreparacionEmpanadera():Set(Pedido) = Pedidos.allInstances() → select(p | p.estado = En Preparacion and esMixto(p) and TienePizzasPorPreparar(p))

PedidosPizzerosCola():Set(Pedido) = Pedidos.allInstances() → select(p | p.estado = ingresado and p.productos → exists(pr | pr.isKindOf(Pizza)))

PrimerPedidoPizzeroCola():Pedido = PedidosPizzerosCola() → Select(p | PedidosPizzerosCola() → forall(p2 | p2.posicionPreparacion → asOrderedSet() → first().posicion >= p.posicionPreparacion → asOrderedSet() → first().posicion)) → asOrderedSet() → first

Cuando uno de los maestros indica que terminó de preparar su parte, se dispara la búsqueda del próximo pedido a fin de notificarlo inmediatamente. Si no hay pedidos que puedan ser preparados, cuando llegue uno que pueda ser atendido por el maestro, el sistema se lo indicará de inmediato. El diagrama 4.13 nos permite ver las actividades que realiza el sistema cuando el maestro informa que terminó de preparar un pedido. Notemos que la búsqueda que realiza el sistema coincide con lo descripto por la operación antes definida.

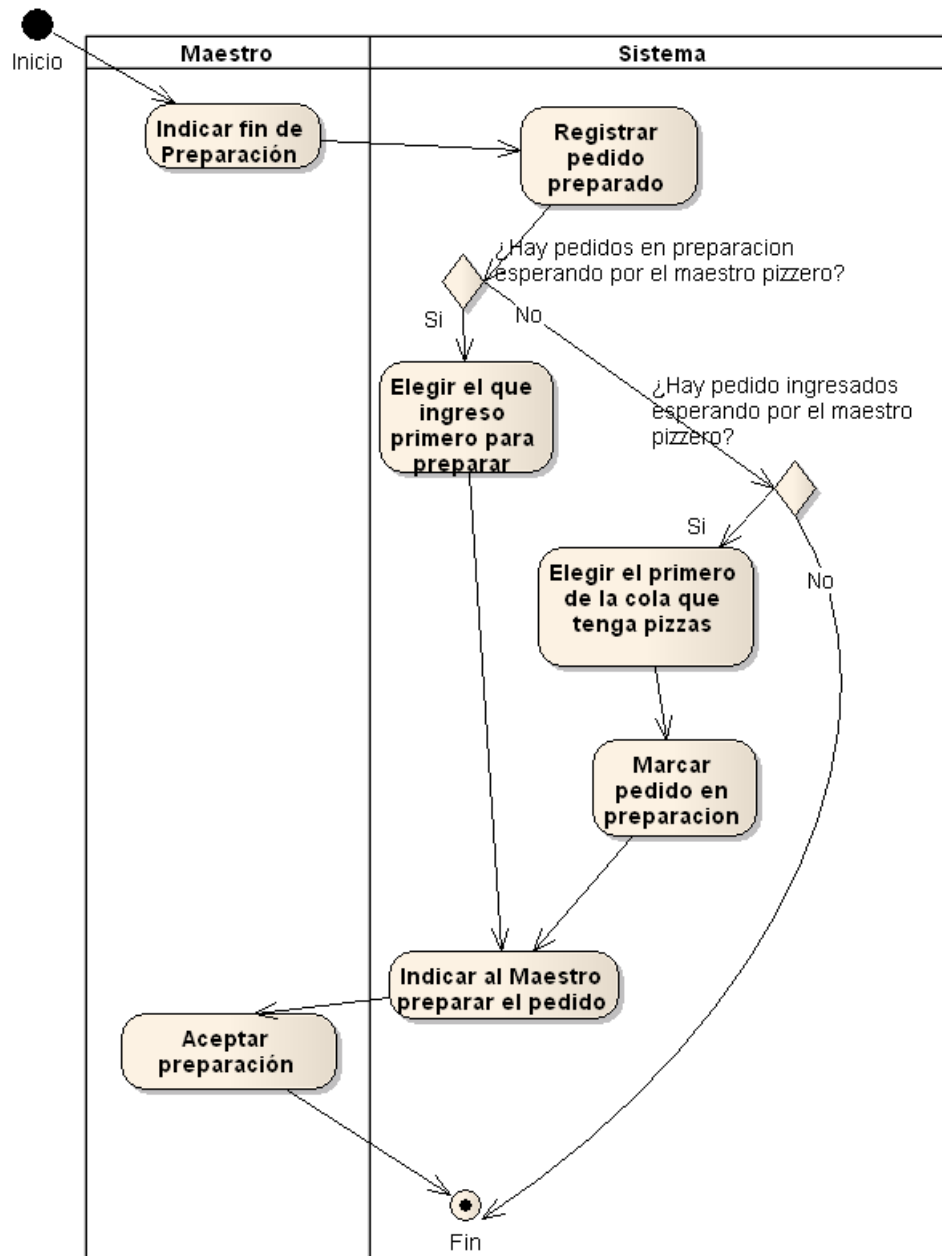


Figura 4.13: Diagrama de actividad: Indicar pedido preparado

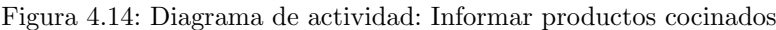
4.7. Cola de Horno

La cola del horno es una segunda cola que establece qué pedido (y en particular qué parte del mismo) se cocinará a continuación. Esta cola se gestiona también de manera automática. El maestro indica cuando retira del horno una parte de algún pedido. El sistema informa entonces qué productos deben ser colocados en el horno. A partir del momento en que alguna parte del pedido es marcada para entrar al horno, el pedido pasa al estado al horno. El sistema identifica qué parte está realmente en el horno, qué parte aún no ingresó y qué parte ya fue cocinada. Cuando todas las partes del pedido fueron cocinadas, se considera que el pedido fue cocinado y está en estado listo. Como dijimos, el sistema muestra qué debe de colocarse en el horno. Cuando el maestro saca algo del horno, debe indicárselo al sistema a fin de que este pueda establecer qué pedido el maestro debe insertar a continuación.

La cola de ingreso al horno admite varias políticas, y una de ellas es la de sectores ágiles. En este caso, algunos módulos funcionan como módulos ágiles, lo que en este contexto significa que dan prioridad a los pedidos más chicos, tomando la idea de la política de *scheduling* llamada *Shortest First* (más corto primero), que se caracteriza por minimizar el tiempo de

espera promedio. Bajo esta política, cuando se libera un modulo ágil y el maestro pregunta qué producto debe ingresar, el sistema tendrá en cuenta la existencia de pedidos más chicos que se puedan ingresar por completo al horno. El cambio de política es llevado a cabo por el encargado de pedidos, quien al cambiar la política a sectores ágiles debe ingresar cuales son los módulos ágiles. Las políticas de horno no se modifican a lo largo del día (se elige al iniciar el sistema por la mañana). Además, las políticas son globales (afectan del mismo modo a ambos hornos). El tamaño y cantidad de los módulos del horno es configurable, sin embargo no se considera necesario un caso de uso para hacer esto. Lo mismo ocurre con la definición de pedido “chico”, puesto que no se espera que estos parámetros cambien con frecuencia.

A continuación veremos en un diagrama de actividad las acciones que se producen cuando uno de los maestros indica que cocinó ciertos productos de un pedido. En el diagrama se nombra como pedido actual a aquel al cual pertenecen los productos que el maestro acaba de informar que terminó de cocinar. Además, se considera que hay pedidos en la cola, de modo que el sistema debe informarle al maestro qué poner a continuación (esto no ocurre siempre, ya que si cuando el maestro informa que cocinó algo no hay nada en la cola del horno, el sistema no le informa que poner sino hasta que exista algun pedido que deba ingresar al horno).



Observamos que en el caso de política ágil el sistema debe ser informado de cuantos módulos ágiles fueron liberados. Esto podría evitarse si el sistema tuviera noción de qué productos están cocinándose en qué módulo (ver posibilidades de mejora en 5). Cuando se está bajo política ágil el sistema intenta ocupar los módulos ágiles en la medida que se liberan utilizando pedidos chicos. Podría ocurrir entonces que existan dos pedidos con productos afuera del horno y dentro del horno, a diferencia de la política normal donde a lo sumo un pedido puede estar parcialmente cocinado. Por esta razón el diagrama diferencia pedido actual y pedido chico actual: el pedido chico actual es un pedido chico que entró a cocinarse porque se

liberó algún módulo ágil, pero quedó con alguno de sus productos sin cocinar.

4.8. Consulta de estado de pedidos

Tanto el cliente como el encargado de pedidos pueden consultar el estado de un pedido determinado. El cliente puede hacerlo vía Web o mediante una llamada telefónica al encargado de pedidos que lo puede consultar personalmente en su terminal del sistema. Para averiguar el estado de un pedido vía Web, el usuario tiene que autenticarse primero. Una vez autenticado podrá acceder a la opción de consultar estado de sus pedidos. El sistema lista entonces los pedidos del usuario pendientes de entrega junto con su estado. Análogamente, el encargado de pedidos puede ingresar un número de pedido para revisar su estado o ver los pedidos asociados a un cliente.

4.8.1. Estimación de Tiempos

A continuación detallaremos las operaciones que permiten estimar el tiempo de finalización de un pedido. Siendo que se trata de una estimación, la siguiente predicción no intenta ser exacta, sino más bien servir de referencia tanto para el cliente como para el encargado de pedidos que fiscaliza el funcionamiento de la pizzería. La estimación es aproximada a tal punto que, dado que no es controlable, no se tiene en cuenta el tiempo que le insume al *delivery* entregar el pedido, y que en general no es despreciable.

Para la estimación nos valemos de los tiempos de preparación y cocción aproximados dados por el maestro al momento de ingresar un producto al sistema. Analizaremos el caso de un pedido recién ingresado, ya que para los otros casos el cálculo es similar con la salvedad de que no se contabiliza el tiempo de las etapas que ya fueron superadas.

En primer lugar, se estima el tiempo de preparación y cocción. Enseguida se estima el tiempo de preparación y cocción de los productos que están delante en la cola. A continuación se repite el proceso para los pedidos que están en preparación, y por último para los pedidos que están encolados para entrar a horno o que están siendo cocinados en este momento.

Presentamos las operaciones en forma de *queries* sin contexto que permiten estimar estos tiempos.

Esta operación nos permite obtener los pedidos que están adelante del pedido p en la cola de preparación:

pedidosAdelantePreparacion (p1:Pedido):set(Pedido)

`Pedido.allInstances() → select (p | p.estado = ingresado and p.posicionPreparación → asOrderedSet() → first() < p.posicionPreparación → asOrderedSet() → first())`

Esta operación calcula el tiempo de preparación de un pedido en base a los tiempos de sus productos:

TiempoPreparacion(p:Pedido):Integer

`p.productos → select(pr | pr.isKindOf(Comida)) → collect(pr | pr.item → select(i | i.pedidos = p) → asOrderedSet() → first().cantidad * pr.asOclType(comida).tiempoPreparacion) → sum()`

Esta operación permite calcular el tiempo de cocción de un pedido que tiene partes ya cocinadas:

TiempoDeCocciónAlHorno(p:Pedido): Integer

`p.item → select(i | i.productos.isKindOf(Comida)) → collect(coccionDe → asOrderedSet() → first()) → collect(c | c.itemDe.productos.asOclType(Comida).tiempoCoccion * (c.itemDe.cantidad - c.cantidadCocinada)) → sum()`

Esta operación nos permite obtener el tiempo de cocción de un pedido en base a los de sus productos:

TiempoDeCoccion(p:Pedido): Integer

`p.productos → select(pr | pr.isKindOf(Comida)) → collect(pr | pr.item → select(i | i.pedidos = p) → asOrderedSet() → first().cantidad * pr.asOclType(Comida).tiempoCocion) → sum()`

Por último estas 3 operaciones nos permiten obtener los pedidos que están en preparación, preparados (y esperan para el mismo horno que p tiene asignado) y aquellos que se están cocinando en el horno asignado a p :

PedidosEnPreparacion() : Set(Pedido)

`Pedido.allInstances() → select(p | p.estado = En Preparacion)`

PedidosColaHorno(p:Pedido) : Set(Pedido)

Pedido.allInstances() \rightarrow select(p1 | p1.estado = Preparado and p.hornoDe = p1.hornoDe)

PedidosHorno(p:pedido) : Set(Pedido)

Pedido.allInstances() \rightarrow select(p1 | p1.estado = Al horno and p.hornoDe = p1.hornoDe)

Con estas operaciones auxiliares, estimamos entonces el tiempo hasta que p esté listo:

EstimarTiempo(p: Pedido): Integer

pedidosAdelantePreparacion \rightarrow collect(p1 | TiempoDeCoccion(p1) + TiempoPreparacion(p1)) \rightarrow Sum() + pedidosEnPreparacion \rightarrow collect(p1 | TiempoDeCoccion(p1) + TiempoPreparacion(p1)) \rightarrow Sum() + PedidosColaHorno(p) \rightarrow collect(p1 | TiempoDeCoccion(p1)) \rightarrow sum() + PedidosHorno(p) \rightarrow collect(p1 | TiempoDeCoccionAlHorno(p1)) \rightarrow sum() + TiempoPreparacion(p) + TiempoCoccion(p)

4.9. Delivery

La operatoria de *delivery* es sencilla. El servicio de distribución se terceriza y la empresa en cuestión provee a la pizzería de personal responsable de las entregas, y los equipa con un teléfono celular.

El *delivery* posee una única interacción con el sistema: cuando el responsable entrega un pedido, debe enviar un mensaje SMS al sistema notificando la entrega del mismo. Con esta información, el sistema marca el pedido como finalizado y termina su ciclo de vida. Si el pedido no puede ser entregado, el *delivery* notificará de la situación al encargado de pedidos que lo marcará como cancelado.

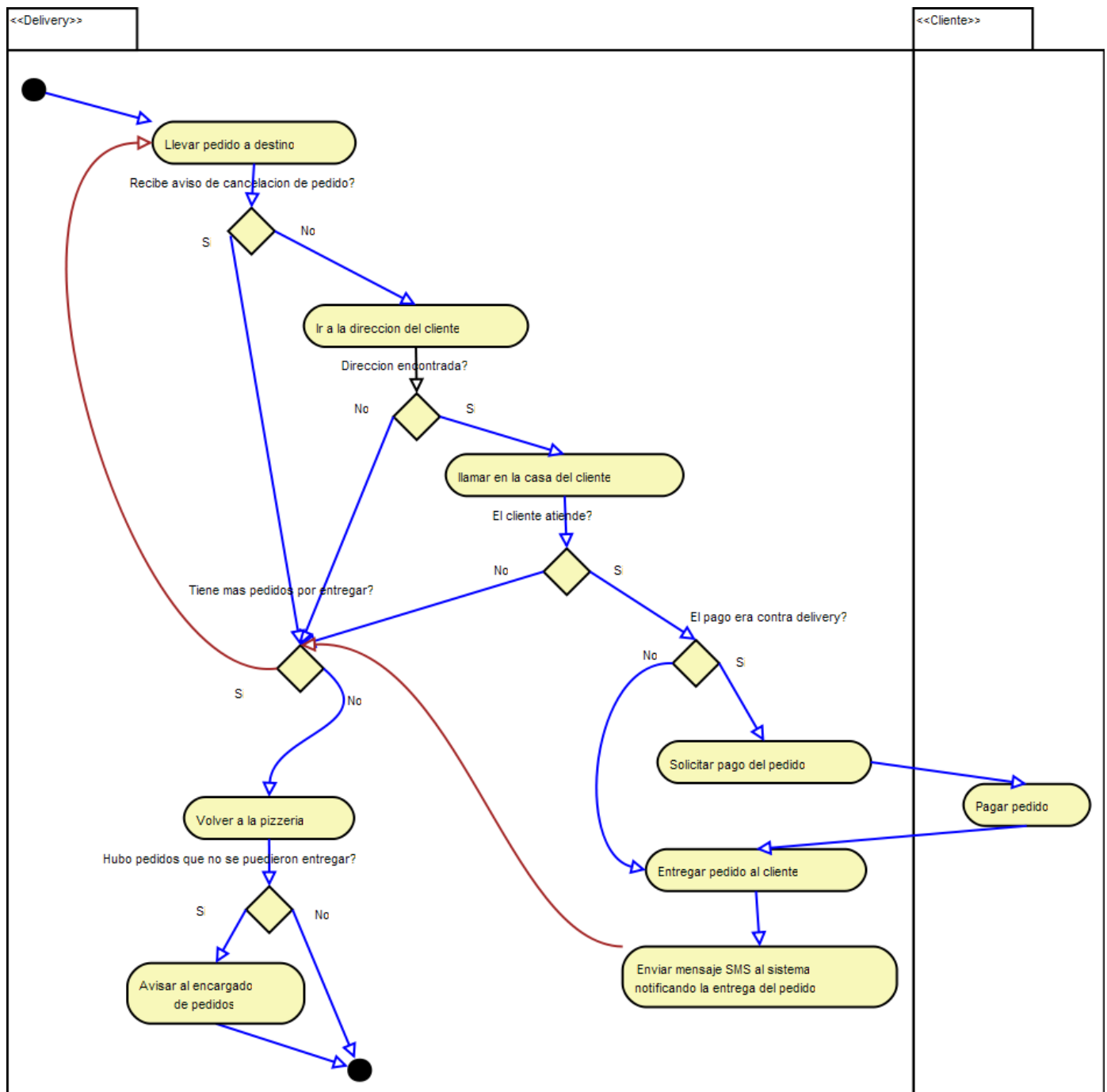


Figura 4.15: Diagrama de actividad de la entrega de un pedido por delivery

4.10. Alternativa de Contingencia

El sistema posee la capacidad de funcionar en lo que denominamos modo de contingencia. Este modo está pensado para permitir que el sistema siga siendo funcional aun cuando solo este disponible el puesto del encargado de pedidos. El sistema permite que toda la operatoria relacionada con la gestión y seguimiento de pedidos (recepción, asignación de horno, armado, ingreso y salida de horno, despacho, control de entrega, actualización de lista de precios) se lleve adelante en el puesto del encargado de pedidos.

En el modo de contingencia se considera que el software de facturación no está disponible, por lo cual la facturación se

realiza manualmente. Queda además fuera de este modo la consulta de estadísticas, así como también la interacción directa del sistema con mozos, clientes y delivery.

De esta manera, el encargado de pedidos es el único agente que utiliza el sistema cuando se encuentra en este modo. Los clientes pueden hacer pedidos solo telefónicamente o en local a un mozo, mientras que los mozos deben comunicar verbalmente el pedido al encargado para que éste lo ingrese. Análogamente, los maestros deberán comunicarse con el encargado de pedidos para que éste actúe como interfaz entre ellos y el sistema (lo cual incluye preguntarle que pedidos deben preparar o cocinar, así como indicarle cada una de las tareas que han finalizado y normalmente ingresarían en el sistema). El *delivery*, que en general utiliza SMS para informar al sistema la entrega de un pedido, ahora deberá hacerlo comunicándose con el encargado de pedidos. Finalmente, el dueño deberá informar al encargado de pedidos los precios que pretende modificar.

4.10.1. Contexto del funcionamiento de contingencia

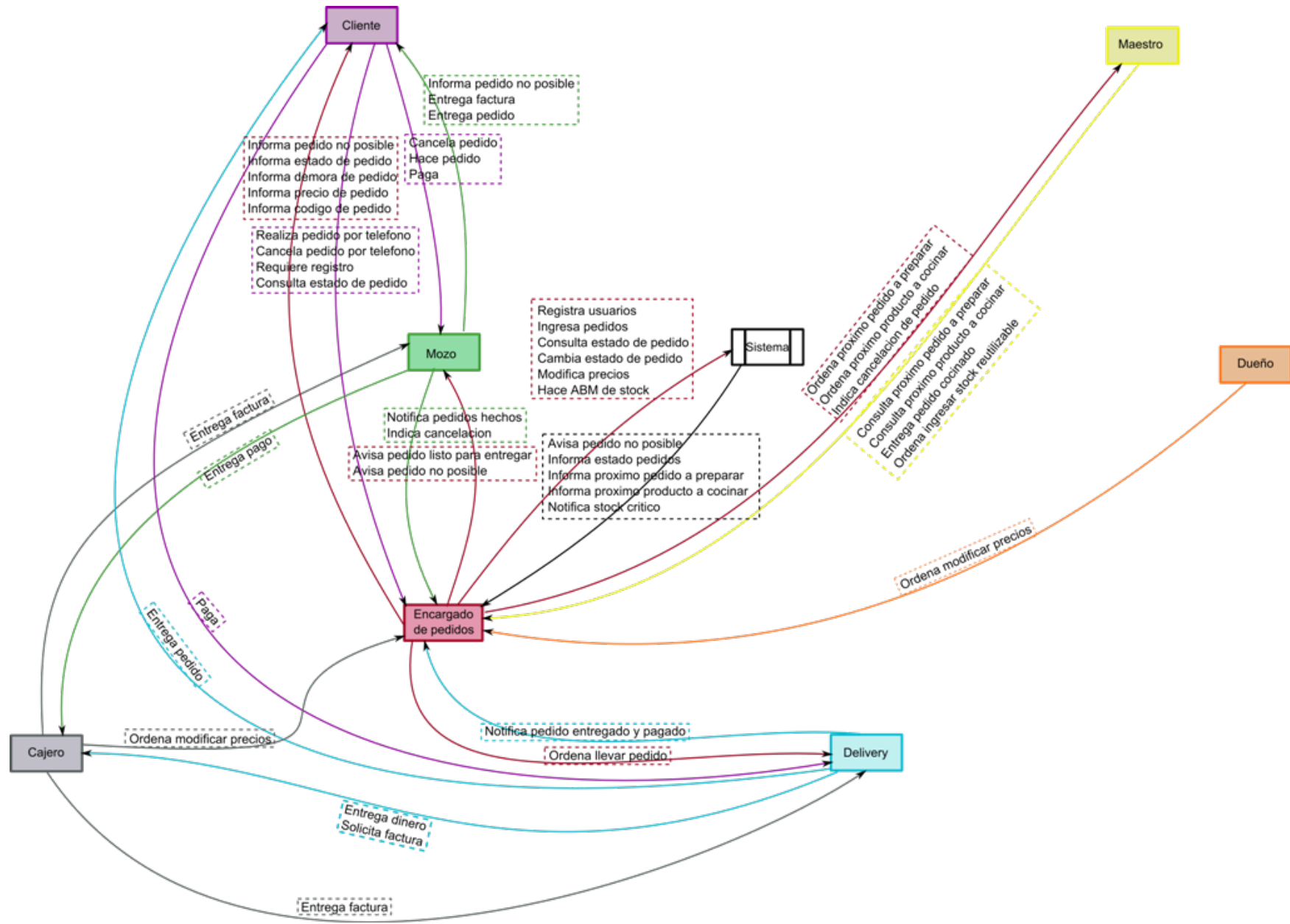
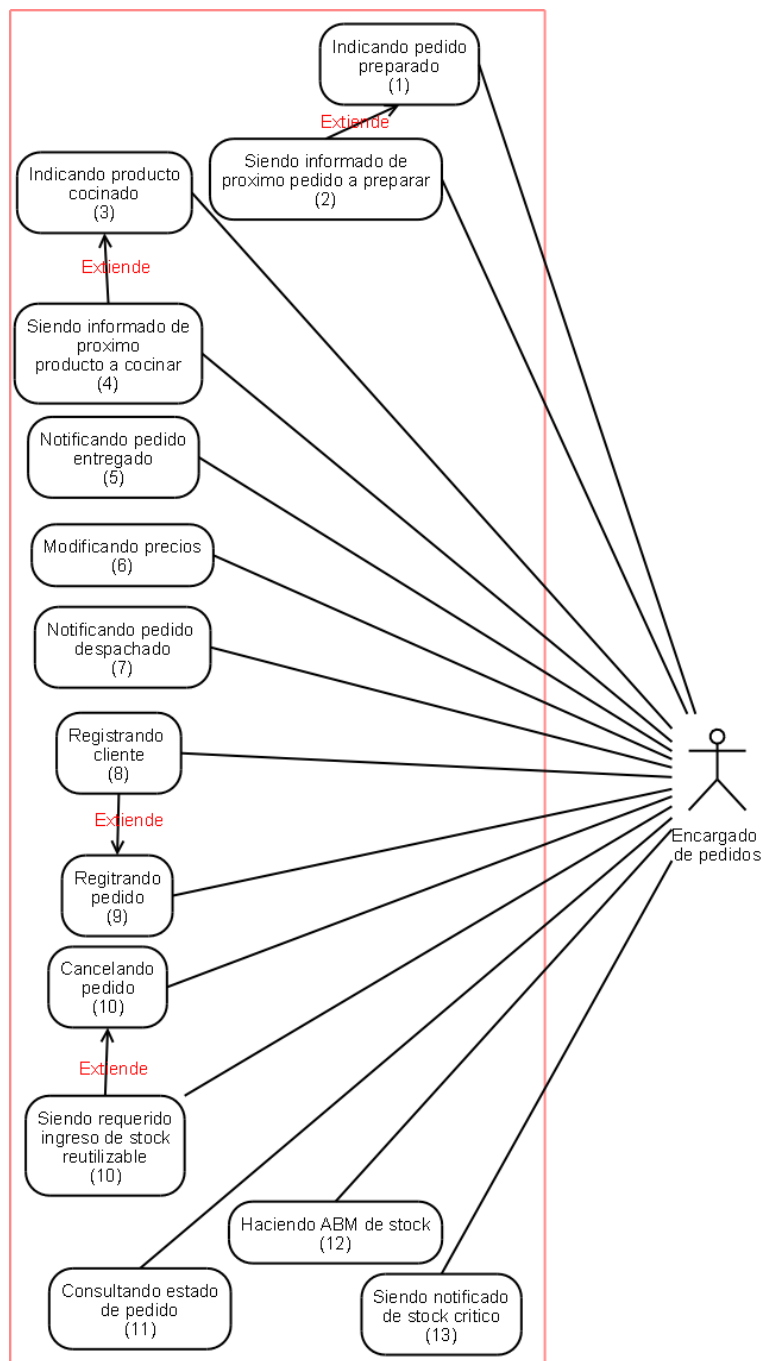


Figura 4.16: Diagrama de contexto de contingencia

Del diagrama se desprende que las funciones principales en las que antes interactuaban otros agentes con el sistema, ahora son responsabilidad del encargado de pedidos, el cual suma a sus responsabilidades previas otras previamente ajenas. Es muy importante notar la sobrecarga de trabajo del encargado de pedidos. Todas las responsabilidades que tenían los demás actores y que eran llevadas a cabo desde otras terminales del sistema ahora son tareas que los mismos deberán llevar a cabo a través del encargado de pedidos, ya que solo su terminal se mantiene en funcionamiento. La cantidad de tareas que debe desarrollar lo convierten en un cuello de botella para el funcionamiento de la pizzería. Por otra parte esta ineficiencia es tolerable dado que la operación en este modo debería ser infrecuente.

4.10.2. Casos de Uso de contingencia

A continuación, se presentan brevemente los casos de uso presentes cuando el sistema está en estado de contingencia. Si bien la gran mayoría son idénticos al modo de operación normal, en algunos aparecen pequeñas diferencias que consideramos útil detallar por separado.



- Indicando pedido preparado y Indicando producto cocinado: el maestro le avisa al encargado de pedidos que terminó la preparación/cocción del pedido o producto y luego el encargado de pedidos procede a registrar tal evento en el sistema.
- Siendo informado de proximo pedido a preparar y Siendo informado de proximo producto a cocinar: el encargado de pedidos solicita al sistema el proximo pedido/producto a preparar/cocinar y luego se lo informa al maestro.
- Notificando pedido entregado: el delivery, al no poder enviar un mensaje al sistema para que registre la entrega (o cancelación) del pedido, deberá avisarle al encargado de pedidos para que este cambie el estado del mismo.
- Modificando precios: el cajero o dueño desea cambiar el precio de los productos, para esto debe informarle al encargado de pedidos los datos necesarios para realizar la operación. Este caso de uso reemplaza a Realizando ABM de productos, ya que hacer baja o alta de un producto no es una operacion crítica durante contingencia.
- Notificando pedido despachado, Registrando cliente, Consultando estado de pedido y Cancelando pedido: estos casos de uso quedan igual a los casos en modo normal.
- Registrando pedido: este caso de uso reemplaza a Registrando pedido telefónico y Registrando pedido de mostrador. Su funcionamiento es igual a los anteriores.
- Haciendo ABM de stock y Siendo notificado de stock crítico: ahora el encargado de pedidos asume las responsabilidades del encargado de stock. Estos son los mismos casos de uso que antes, aunque el actor es el encargado de pedidos.
- Siendo requerido ingreso de stock reutilizable: luego de cancelar un pedido en preparacion, el sistema pide al encargado de pedidos que ingrese el stock reutilizable. El encargado pregunta al maestro cual es el stock reutilizable y luego el encargado de pedidos lo ingresa, similar al caso de uso normal.

Notese que la generación de archivo para facturación en modo contingencia no está a cargo del sistema, por lo cual se realiza manualmente y está a cargo del cajero, como se observa en el diagrama de contexto de contingencia. También queda fuera de este modo la consulta de estadísticas, los pedidos por SMS y el alta y baja de productos, ya que no son necesidades críticas durante la operación de contingencia.

Los casos de uso de la alternativa de contingencia permiten satisfacer los requerimientos 16, 17 y 18, puesto que eran éstos los que establecían que el sistema debería ser capaz de funcionar con solo el puesto del encargado en estado funcional.

4.11. Información estadística

Con el fin de comprender más profundamente el comportamiento del negocio, desarrollamos para la pizzería un conjunto de indicadores que permitan analizar su rendimiento y predecir algunos de los parámetros que afectan su comportamiento, así como optimizar algunas estrategias de venta y producción.

Los indicadores de rendimiento tienen por objetivo caracterizar el funcionamiento de la pizzería en un período dado. Mediante el monitoreo del ingreso y salida de los pedidos, es posible realizar un seguimiento de algunos parámetros interesantes:

- **Tasa de producción de la cocina:** Permite evaluar el la capacidad de la cocina de atender pedidos, con el objeto de determinar sus posibilidades de producción y evaluar si en un momento dado la cocina funciona por debajo o por encima de su nivel habitual.
- **Tasa de ingreso de pedidos:** Permite conocer la demanda que hubo en un momento dado y con esa información inferir la demanda de momentos futuros. Puede medirse de forma global o discriminada por origen de pedido, permitiendo observar cual es la forma más utilizada para ingresarlos.
- **Tiempo medio de espera por un pedido:** Permite conocer el tiempo que debe esperar un cliente por su pedido en función del momento en que realizó su pedido. Puede medirse de forma discriminada para clientes locales o remotos.
- **“Combos” más populares:** Permite conocer algunos conjuntos de productos que suelen ser ordenados juntos por los clientes.
- **“Perfiles” de cliente:** Permite evaluar los tipos de compras que realiza cada cliente gracias a la información personalizada que se tiene sobre cada uno.
- **Productos más populares:** Permite conocer cuales son los productos que circulan con mayor frecuencia y por tanto deberían ser considerados más importantes para cuestiones tales como la ubicación de los mismos en la carta.

En particular, el conocimiento sobre la demanda permite mantener un menor stock de productos, disminuyendo así el costo financiero y manteniendo una mayor eficiencia en la administración de los recursos. A su vez, determinar los productos y combinaciones más populares puede conducir a promociones más efectivas, así como a disminuciones de costos por un mejor

conocimiento de los insumos necesarios para su producción. Por último, los perfiles de cliente permiten ejecutar estrategias de fidelización: al conocer en detalle y de forma individual las compras que realiza cada cliente se puede determinar cuales son los clientes más importantes y en función de eso ofrecerles algún tipo de beneficio.

4.12. ABM de stock

El ABM de stock permite realizar altas, bajas y modificaciones de los kits y demás insumos para la elaboración de productos. Además, permite modificar el valor crítico de cada insumo, de modo que el sistema pueda utilizar dicho valor para alertar al encargado de stock cuando la cantidad disponible de un producto descienda por debajo de dicho valor. La notificación se realiza con un mensaje en pantalla o mediante correo electrónico si esto se considerara conveniente.

Para ver como funciona el control por stock crítico, proponemos las siguientes FSM en las que se modela un ingreso de pedidos que consumen cierta cantidad de un stock determinado (figura 4.17). Por simplicidad, asumimos que solo consume cierta cantidad de un único insumo, pero se puede extender relativamente fácil el modelo para considerar el caso general, aunque a costa de claridad en la lectura.

El control de stock esta modelado en la FSM de la figura 4.18. Como vemos, el sistema permite que se ingresen pedidos, pero cuando la cantidad de stock es menor al valor crítico, se produce un aviso y luego se permite igualmente seguir ingresando pedidos. Cuando el encargado de stock (4.19) observa el aviso puede incrementar los valores asociados a cada insumo, aumentando así el stock. Entonces, el sistema vuelve a monitorear el valor del stock para alertar si queda nuevamente bajo el valor crítico. En particula, si el encargado deja el stock en un valor inferior al crítico, el sistema volverá a mostrar el mensaje de stock crítico.

En estas FSM utilizamos la variable *stock* : $[0..MAX]$ que comienza inicializada en *STOCK* (un valor constante). *critico*, *k* y *n* son valores constantes.

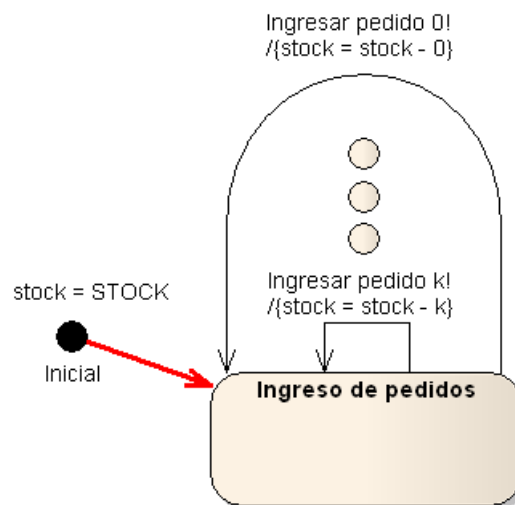


Figura 4.17: FSM Ingreso de pedido

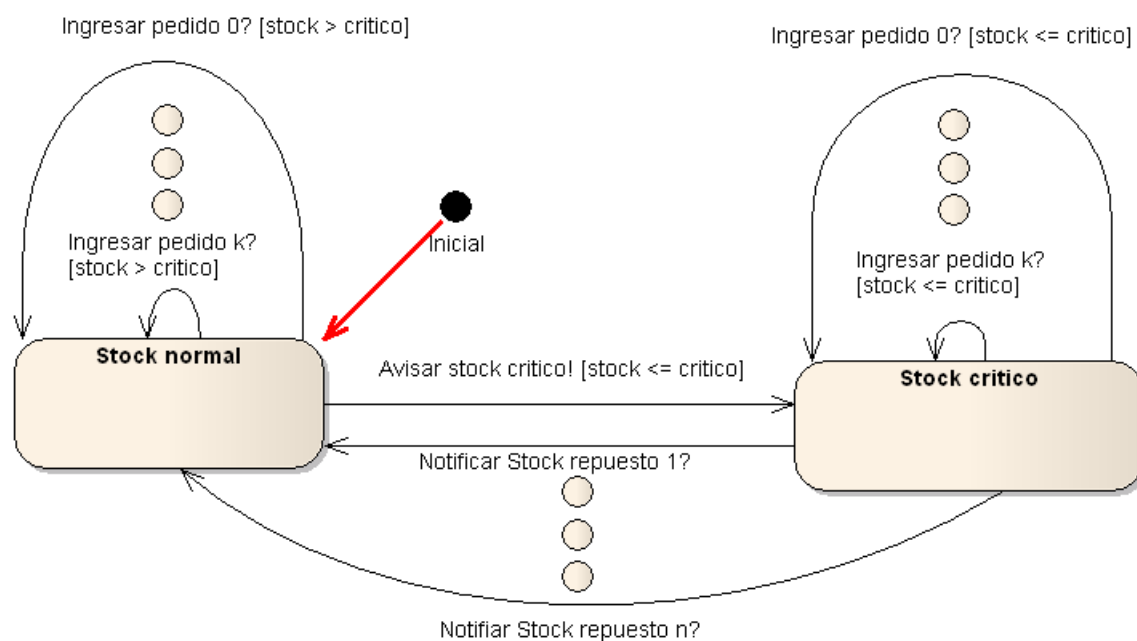


Figura 4.18: FSM Aviso de stock crítico

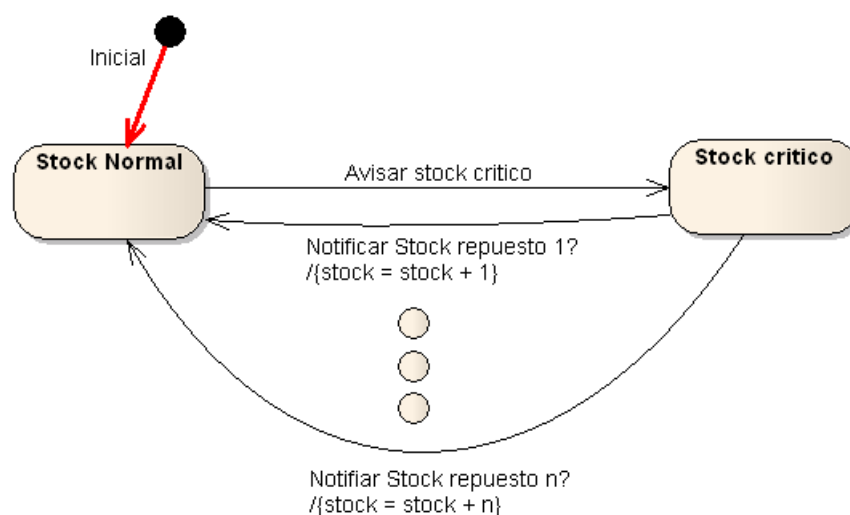


Figura 4.19: FSM responsable de stock

Finalmente FSM Control de stock crítico = FSM ingresar pedidos || FSM aviso stock crítico || FSM responsable de stock

Parte 5

Apéndice I: Posibilidades de mejora

Se detallan a continuación algunas propuestas de mejora que consideramos pueden resultar de sumo interés para el funcionamiento de la pizzería. Estas mejoras se consideraron para su inclusión en el sistema especificado pero fueron dejadas de lado dadas las limitaciones de tiempo para el desarrollo.

La implementación de estas funcionalidades podrá ser objeto de mejoras posteriores al sistema propuesto.

5.1. Seguimiento de módulos del horno

Una mejora sencilla de implementar es el seguimiento individual de cada módulo del horno así como de qué pedido lo ocupa en un momento dado. En la implementación requerida, el sistema no puede determinar en qué lugar del horno se encuentra un pedido.

Esto no representa una limitación seria en contextos donde los tamaños y cualidades de los módulos son idénticas en todos ellos. El seguimiento individual de cada módulo permitiría tener divisiones irregulares del horno, lo cual puede ser necesario para algunos tipos de horno.

En cualquier caso, el seguimiento individual del módulo del horno permite ofrecer una mejora sustancial desde el punto de vista de usabilidad, ya que el individualizar cada módulo permite a un maestro indicar al sistema qué pedido se ha terminado de cocinar sencillamente eligiendo cuál es el módulo del horno que acaba de vaciar. Si los módulos son indistinguibles, el maestro deberá identificar el pedido dentro de una lista de pedidos en proceso de cocción, tarea que podría ser no trivial para pedidos similares.

5.2. Seguimiento de tiempos por estado

El hecho de seguir individualmente los tiempos que un pedido insume en cada estado ofrece una mejora importante desde un punto de vista estadístico ya que permite individualizar los tiempos de cada parte del proceso de atención de un pedido.

Si se dispone de esta información, se pueden hacer predicciones mucho más precisas del tiempo estimado para la salida de un pedido, o de la tasa de producción de la cocina, facilitando así la indentificación de cuellos de botella que limiten la capacidad de producción.

La implementación de esta funcionalidad se ve extendida por la característica de seguimiento individual de productos detallada más abajo

5.3. Preparación adelantada

Al disponer de estadísticas precisas sobre la demanda que recibe el local es posible, en función de las predicciones hechas por el sistema, optimizar el uso de los recursos de producción del local abaratando así costos.

Dado que el tamaño del staff de la pizzería depende principalmente de la cantidad de pedidos que se desea poder procesar en un lapso dado, es de esperarse que la pizzería emplee trabajadores suficientes para funcionar durante picos de demanda. Esto redundaría en la disponibilidad de capacidad ociosa en los momentos de menor demanda.

Si se pueden predecir adecuadamente los picos de demanda, el sistema puede indicar a los cocineros preparar pizzas que se sabe que se venderán durante un pico de demanda. Al hacer esto durante momentos de capacidad ociosa, se puede maximizar la ocupación de los cocineros manteniendo así pequeño el tamaño del staff.

Si se asume que el cuello de botella del proceso está en los empleados, esto reduce los costos operativos del negocio, con lo cual se trata de una característica muy deseable. Además, el sistema puede controlar que los pedidos preparados no expiren siguiéndolos individualmente.

5.4. Carga diferida de pedidos

El sistema contempla la necesidad de ser tolerante a fallos. Sin embargo, esta consideración se limita a permitir la operación de toda la pizzería a través de la terminal del encargado de pedidos. Si bien esta funcionalidad es deseable, resulta importante también considerar qué hacer en caso de una caída total. En la eventualidad de que esto ocurra, el sistema no podrá hacer ningún procesamiento relativo al funcionamiento de la pizzería.

Si bien no se puede hacer nada mientras la situación que cause el problema se prolongue (falla de hardware, pérdida de energía), tiene sentido permitir el ingreso diferido de pedidos que se procesaron manualmente durante la falla del sistema. La idea es registrar cuando el sistema vuelve a funcionar todos los eventos que se produjeron mientras éste no era capaz de registrarlos.

Para esto es necesario agregar una interfaz especial del tipo ABM para el registro de dichos eventos. Esta interfaz deberá, además de permitir ingresar los pedidos y sus datos, dejar que el usuario ingrese datos tales como los tiempos y duraciones de los eventos, y otras informaciones que durante la operatoria normal el sistema infiere por sí solo.

Esto permite realizar un seguimiento apropiado del stock y ingresar datos sobre pedidos que de otro modo no serían registrados, con el fin de tener estadísticas válidas del período de falla.

5.5. Seguimiento individual de cada producto

Otra mejora interesante es ampliar el alcance del sistema de control de stock para permitir una visión más granular de los productos involucrados en el proceso de producción.

Esto abarca desde los lotes de stock hasta cada pizza producida, así como eventualmente la relación entre ambos (qué lote de insumos fue utilizado para producir qué pizza). Esto nos da algunas posibilidades interesantes, aunque produce un costo adicional importante de recursos ya que aumenta notoriamente la cantidad de datos que se almacenan.

Para empezar, es posible controlar los vencimientos de los insumos adquiridos además de su stock, así como la fecha de su compra y el costo de cada operación. Esto permite no solo evitar problemas sanitarios, sino también observar la influencia de parámetros económicos como la inflación en los costos de operación de la pizzería.

En segunda instancia, controlar individualmente cada uno de los productos que salen del negocio permite mejorar sustancialmente la precisión de las estadísticas y predicciones que puede hacer el sistema.

Por último, si se lleva registro de la vinculación entre lotes de insumos y productos vendidos resulta más sencillo solucionar (o en su defecto, desvincularse de) responsabilidades en caso de tener algún problema sanitario. Si un cliente se enferma, el conocer exactamente cual es la partida de ingredientes que fueron utilizados en la preparación de ese alimento permite aislarla para determinar si tenía algún problema o sencillamente evitar seguirla utilizando.

Parte 6

Apéndice II: Glosario

6.1. Terminología

- **Pedido:** Se denomina “pedido” a todo pedido realizado por un cliente (a través de alguno de los medios de contacto disponibles) y que ya ha sido procesado por el sistema. Un pedido no es tal, y no se registra información sobre él, hasta el momento en que se da de alta en el sistema.
- **Pedido chico:** Se denomina pedido “chico” a un pedido que está por debajo de un tamaño prefijado (como por ejemplo, un pedido que puede cocinarse completamente en un solo módulo del horno). Esta definición es de interés para la política ágil de cola de horno, y debe ser configurable en el sistema (aunque no es necesario que pueda cambiar durante la ejecución).
- **Encargado de pedidos:** El encargado de pedidos es un empleado cuya labor es la de administrar los pedidos y el funcionamiento general controla la facturación y supervisa la distribución de los pedidos. Además tiene privilegios para situaciones especiales (reorganización de la cola de pedidos, cambios de política de pedidos, cancelaciones y acceso al ABM de usuarios). En caso de contingencia (sistema caído o con funcionalidad limitada), este agente deberá controlar el funcionamiento manual del sistema y posteriormente ingresar los datos de los pedidos recibidos para su registro en el sistema.
- **Encargado de stock:** El encargado de stock es la persona responsable de recibir las notificaciones relacionadas a los niveles de stock e ingresar los datos respectivos a las altas y bajas del stock. En la mecánica actual de la pizzería no hay un encargado de stock definido. Si bien su papel será desempeñado por alguno de los otros agentes (dueño o encargado de pedidos), decidimos separar su rol puesto que se trata de una tarea independiente de las demás.
- **Cajero:** El cajero se encarga únicamente de modificar los precios de los productos. El resto de sus labores (cobrar a los clientes, imprimir facturas, etc) no involucran al sistema.
- **Dueño:** El dueño es responsable de la gestión de la carta de la pizzería, sus precios y es el interesado en acceder a las estadísticas sobre el funcionamiento del negocio.
- **Cocinero:** El cocinero es cualquier persona que trabaja en la cocina. En principio es ya sea el maestro pizzero o el maestro empanadero, pero podría ser un ayudante de cocina en general.
- **Estadísticas:** Las estadísticas son registros históricos de los indicadores de rendimiento, que permiten evaluar la progresión del negocio a lo largo del tiempo.
- **Indicadores de rendimiento:** Son valores numéricos que describen el rendimiento de la pizzería en un período determinado. Pueden depender de factores diversos. Algunos ejemplos de indicadores de rendimiento son la tasa de producción de la cocina, la tasa de ocupación del horno, el tiempo medio de espera por un pedido o la cantidad de valoraciones negativas obtenidas por el sistema de feedback. Algunos de ellos se le presentan al encargado de pedidos en tiempo real.
- **Maestro Pizzero:** Es el cocinero encargado de las pizzas, que se ocupa de determinar los ingredientes que componen a una pizza dada, prepararla y supervisar su cocción si se hace en su horno. En el contexto del sistema, es responsable de los cambios de estado de los pedidos en la sección que concierne a la cocina.
- **Maestro Empanadero:** Análogo al maestro pizzero pero para la preparación de empanadas.
- **Mesero o Mozo:** Es el responsable de atender a los clientes en las mesas y registrar sus pedidos en el PDA que luego los transfiere al sistema. Si el sistema está degradado y no dispone del PDA, registrar los pedidos mentalmente o en papel y luego se los entrega al encargado de pedidos.

- **Cliente:** Es cualquier individuo que esté interesado en adquirir productos de la pizzería, a través de cualquiera de los medios de contacto y pedido.
- **Cliente local:** Cliente que concurre personalmente al local y es atendido por un mesero en su mesas.
- **Cliente remoto:** Cliente que hace su pedido desde su casa u otro lugar y requiere que le sea entregado por el servicio de delivery.
- **Cliente telefónico:** Cliente remoto que realiza su pedido por teléfono (es atendido por el encargado de pedidos que a su vez ingresa el pedido en el sistema manualmente).
- **Cliente Web:** Cliente remoto que realiza su pedido a través de la página web de la pizzería. El sistema registra su pedido sin ninguna intervención humana.
- **Cliente SMS:** Cliente remoto que realiza su pedido a través de mensajes de texto. El sistema registra su pedido sin ninguna intervención humana.
- **Servicio de delivery:** Servicio de entrega a domicilio de los pedidos. Este servicio se subcontrata a un tercero que provee toda la logística de entregas, debiendo la pizzería únicamente indicar las entregas que deben realizarse.
- **ABM de stock:** Sistema que permite ingresar cambios a la base de datos de stock. Se considera stock a todas las compras de lotes de materias primas que se usan a continuación en la cocina. Se contemplan también datos como fecha de ingreso y vencimiento del lote, para una mayor granularidad en el control.
- **ABM de productos:** Sistema que permite ingresar productos que luego podrán ser vendidos en la pizzería. Por ejemplo, un producto es una nueva gaseosa o un nuevo sabor de empanada. Se incluyen además de su nombre el precio y los ingredientes necesarios para la preparación. A través de esta interfaz se hacen los cambios de precios.
- **Sistema de facturación:** Es un sistema informático externo que mediante una interfaz predefinida se encarga de la facturación y emisión de comprobantes de todas las ventas que se producen en la pizzería.
- **Sistema de SMS:** Es un sistema informático externo que provee un servicio de interfaz con las empresas de telefonía celular, permitiendo que el sistema procese mensajes que los clientes envían o realizar el envío de mensajes sobre el sistema SMS.
- **Pedido diferido:** Los pedidos diferidos son pedidos que se realizaron por fuera del sistema (por diversas razones, una de las cuales puede ser una caída del sistema). Se permite al encargado registrar dichos pedidos después de que fueron procesados manualmente para que sean tenidos en cuenta en las estadísticas.
- **Cola de pedidos:** Los pedidos que ingresan al sistema pasan a una cola de pedidos donde pueden ser reordenados por el encargado antes de que la cocina esté lista para procesarlos. Una vez que les llega su turno, los pedidos son preparados y pasan a la cola de horno donde esperan ser preparados. El orden de la cola de pedidos es en principio FIFO con modificaciones manuales, pero podría cambiarse de ser necesario.
- **Cola de horno:** Los pedidos que ya fueron preparados en la cocina pasan a la cola de horno en forma de fragmentos que pueden cocinarse simultáneamente en un único módulo de horno. Allí se ordenan y procesan según la política de horno que tiene al menos dos variantes (cola convencional y cola ágil)
- **Combo:** Un “combo” es un conjunto de productos que suelen ser pedidos juntos (por ejemplo, una pizza napolitana y una cerveza grande). La idea es que el sistema sea capaz de observar de forma automática este tipo de conjuntos para que luego sirvan de base a promociones.

Parte 7

Apéndice III: Conclusiones

El trabajo nos permitió comprender la dificultad y la enorme cantidad de trabajo que involucra la realización de una especificación completa para un sistema informático usando el modelo de *Software Requirements Specification*. Tuvimos la oportunidad de ejercitar la gran mayoría de las técnicas propuestas en clase que nos fueron de utilidad para ilustrar nuestra especificación y facilitar su comprensión. Esto nos permitió comprender mejor las aptitudes y limitaciones de cada técnica, así como en qué contextos es conveniente o no el uso de cada una.

Por el alto grado de acoplamiento que existe entre los elementos de un documento de este tipo, el trabajo representó un desafío también desde el punto de vista de la coordinación de los esfuerzos y de los miembros del equipo, para lograr ser productivos sin producir inconsistencias en el documento resultante.

7.0.1. Dificultades en la realización del trabajo

Durante la realización del trabajo enfrentamos varias dificultades extraordinarias que estaban fuera de nuestras manos. Queríamos dejar constancia de las mismas en esta conclusión porque consideramos que afectaron seriamente la calidad del trabajo final.

En primer lugar, es destacable el enorme salto de complejidad entre el trabajo práctico y los ejercicios resueltos en las clases. Si bien entendemos que el trabajo debe ser de una dificultad mayor, hubo casos paradigmáticos donde los contenidos fueron vistos de forma muy superficial o directamente no fueron presentados en clase, poniéndonos en la necesidad de improvisar. En particular, el modelo de operaciones no fue visto en clase por fuera de su breve introducción en la teórica, y los ejercicios presentados en la práctica respecto de modelo conceptual eran de una dificultad significativamente menor a la del trabajo práctico.

En segundo lugar, muchas veces nos encontramos bloqueados a la espera de que se resolvieran cuestiones sobre la operatoria que no fueron definidas correctamente. A dos semanas de comenzado el TP, todavía se estaban resolviendo en clase. Fue el caso, por ejemplo, del funcionamiento de la cola de pedidos, que no estaba definido claramente en el documento de especificación y fue definido bastante más adelante. Por esta razón pasamos muchos días, a veces incluso fines de semana enteros sin poder avanzar. En función de esto, las primeras semanas del trabajo fueron muy poco productivas debiendo apurarnos sobre el final, una vez que teníamos todas las herramientas, para lograr terminar en el plazo especificado.

La utilidad del TP0 se vio seriamente comprometida por problemas de aparente desinformación entre los docentes. Cuando consultamos sobre el grado de refinamiento (o el criterio para hacerlo de forma selectiva) para el modelo de objetivos, obtuvimos respuestas diversas, predominantemente del tipo “está a su criterio, omitan lo que no importa”. Cuando nos corrigieron el trabajo, nos indicaron que había refinamientos que no eran completos desde un punto de vista formal. Finalmente, cuando fue entregada la plantilla para el documento del TP1, se detallaba en ella un criterio preciso de qué cosas se deben refinar y qué cosas no, obligándonos a rever lo producido y duplicar esfuerzos.

Por último, nunca fue definida en las clases prácticas la idea general del trabajo práctico. Dicho entendimiento se dificultó aún más en la medida de que las consultas a diversos ayudantes generaban respuestas diametralmente opuestas. No fue hasta que en la teórica Sebastián Uchitel dijo que el documento que debíamos preparar se llamaba SRS que pudimos investigar por nuestra cuenta cual era el objetivo de un documento de este tipo.