



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

## Trabajo Práctico Nro. 1 (R)

01/07/2010

Ingeniería de Software I

Integrante	LU	Correo electrónico
Dinota, Matías	076/07	matiasgd@gmail.com
Frid, Igal Pablo	231/07	ipfrid@gmail.com
Huel, Federico Ariel	329/07	federico.huel@gmail.com
Leveroni, Luciano	360/07	lucianolev@gmail.com
Mosteiro, Agustín	125/07	agustinmosteiro@gmail.com



**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>

## Introducción

El objetivo del siguiente trabajo es modelar un sistema para una cadena de pizzerías, denominada Pizza Hack. Con este fin, se utilizarán técnicas de ingeniería de requerimientos, a saber: Diagrama de Contexto y de Objetivos. A su vez, se incluyen en el presente trabajo las explicaciones y detalles de cada uno de los diagramas para clarificar lo expuesto de forma gráfica en dichos diagramas.

El sistema a modelar pretende resolver las necesidades y problemas de una cadena de pizzerías típica, en la que cada uno de los locales atiende al público a través de un mostrador, es decir, no ofrece la posibilidad de consumir las pizzas en el mismo local. Además, cabe aclarar que la cadena de pizzerías desea mantener el mismo menú en todos sus locales, por lo que un cambio en el menú de un local deberá verse reflejado en la cartilla de los restantes. También se debe mencionar que en caso de que algún pedido no se pueda preparar en el local donde se encuentra el cliente, se ofrecerá derivar su pedido a otro local de la cadena. Por último, hay que tener en cuenta que el sistema no debe tener un servidor centralizado para la coordinación entre locales.

Por medio de la aplicación de las técnicas de ingeniería de requerimientos se busca explorar distintas opciones para modelar el sistema deseado por la cadena de pizzas. Finalmente, se estudiarán las ventajas y desventajas que conlleva elegir cada una de estas alternativas en los aspectos que consideramos relevantes para el funcionamiento del sistema y el beneficio de la cadena de pizzas.

## Presunciones

Para la confección del trabajo, se hicieron las siguientes presunciones:

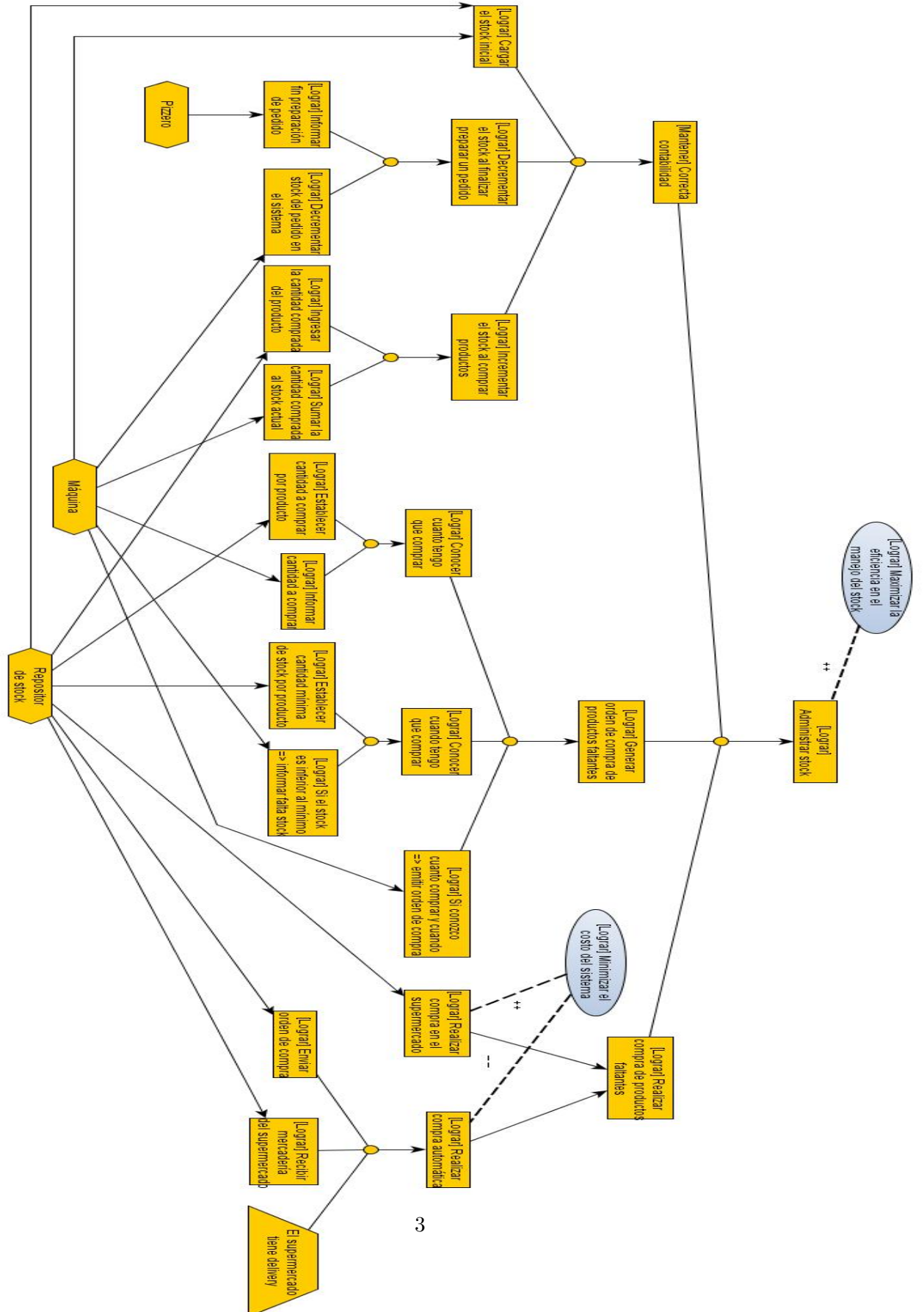
- Pizza Hack no brinda servicio de delivery ni tampoco brinda la posibilidad de comer en el local.
- La máquina no contempla manejo de dinero.
- Los cambios sólo podrán proponerse al finalizar el día, es decir cuando todos los locales ya se encuentren cerrados, para asegurarse de que dicho cambio no genere imprevistos en el menú cuando el local está operando.
- Cuando un local propone un cambio, debe hacerlo durante su turno, es decir hay un sistema de turnos para proponer cambios entre locales.
- El supermercado al que se realizan las compras de los productos para preparar los pedidos posee un sistema de delivery.

## Vistas

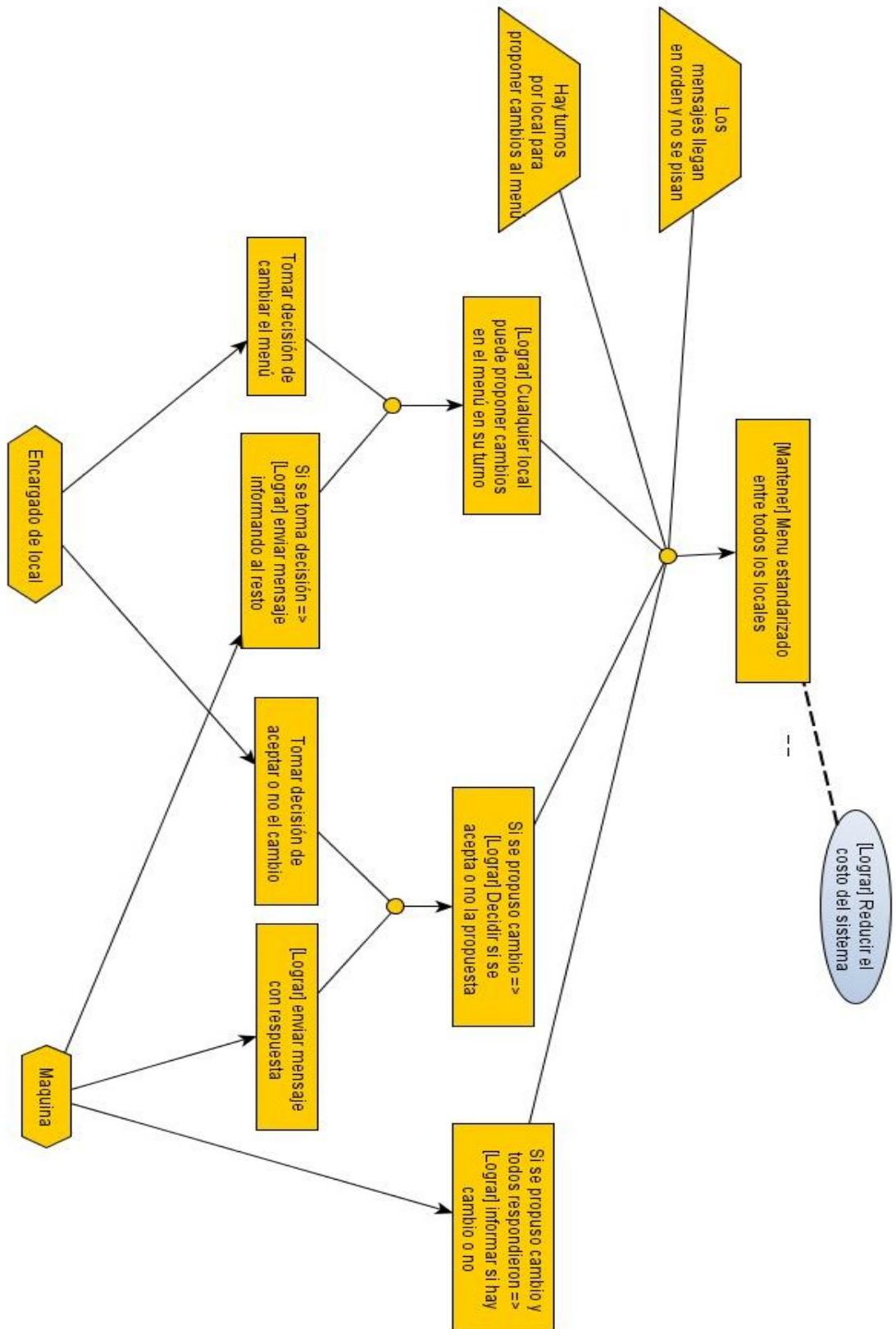
### Diagramas de objetivos

En la siguiente sección se presentarán los gráficos correspondientes a los diagramas de objetivos referentes al sistema:

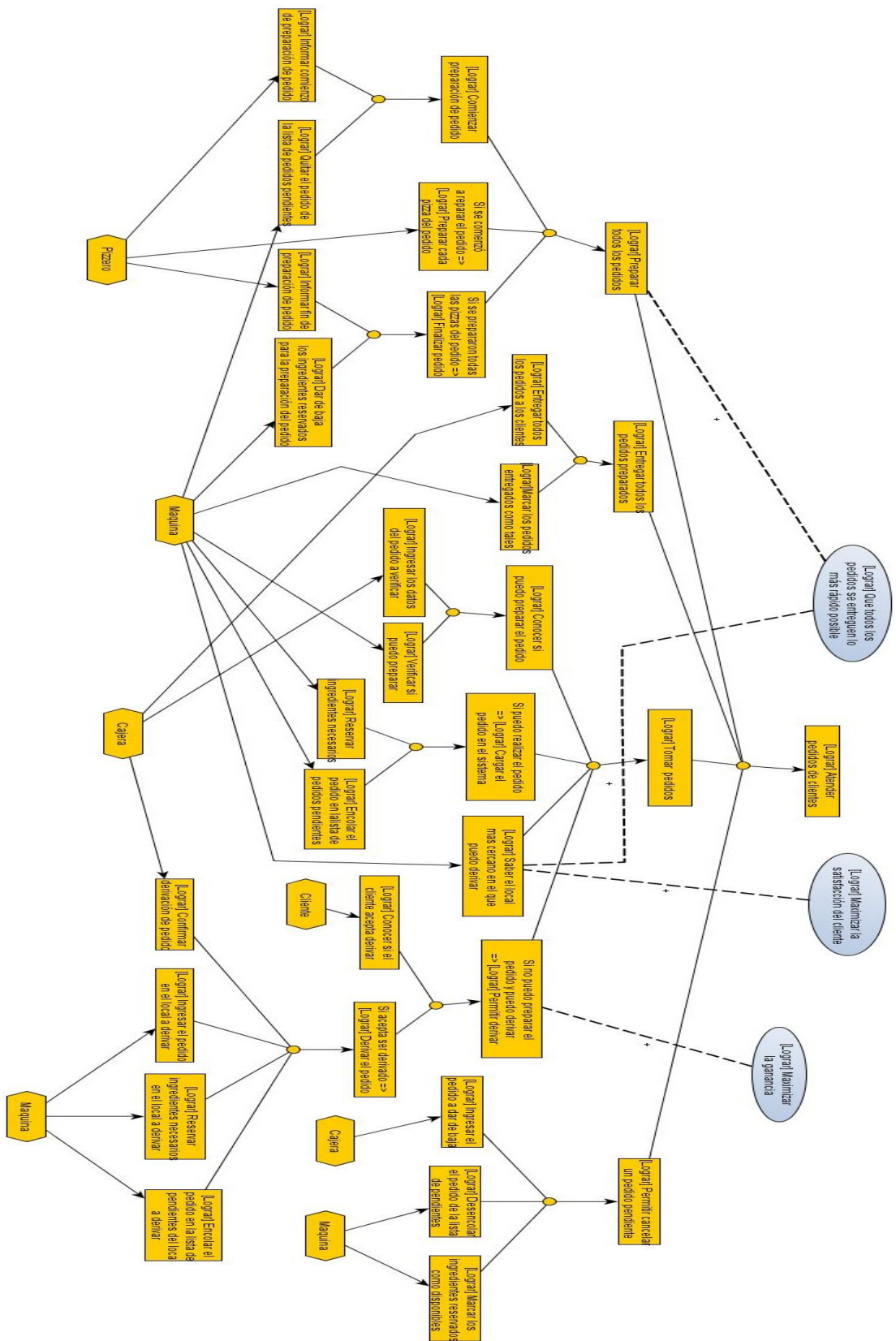
## Administrar Stock



## Menú estandarizado



## Tomar pedidos



## Explicaciones

### Diagrama: Administrar stock

- **Objetivo:** Mantener correcta contabilidad.  
**Definición:** Significa que en todo momento se desea conocer el stock disponible.
- **Objetivo:** Lograr cargar stock inicial.  
**Definición:** Cuando se abre un local, es el repositor de stock quien se encarga de realizar una carga inicial de stock.
- **Objetivo:** Lograr decrementar el stock al finalizar de preparar un pedido.  
**Definición:** El stock se descuenta una vez que el pedido se termina de preparar. Esto se realiza luego de que el pizzero informa que terminó de preparar un pedido.
- **Objetivo:** Lograr incrementar el stock al comprar productos.  
**Definición:** Una vez que el repositor de stock compró los productos necesarios, tiene que cargar en el sistema la cantidad comprada de cada producto para poder registrar el nuevo stock.
- **Objetivo:** Lograr conocer cuánto tengo que comprar.  
**Definición:** Este objetivo permite automatizar la realización de una orden de compra de productos. Simplemente se busca en el sistema cuál es la cantidad predeterminada a comprar de cierto producto.
- **Objetivo:** Lograr establecer cantidad a comprar por producto.  
**Definición:** El repositor de stock es el encargado de establecer una cantidad predeterminada a comprar de un producto. Este valor puede ir cambiando de según las necesidades del local.
- **Objetivo:** Lograr conocer cuándo tengo que comprar.  
**Definición:** Una vez que el stock de un producto atravieza la cota mínima de stock del producto, el sistema dispara una alarma para informar que se debe realizar una compra sobre ese producto ya que está próximo a agotarse.
- **Objetivo:** Lograr establecer cantidad mínima de stock por producto.  
**Definición:** El repositor de stock es el encargado de establecer la cota mínima de stock aceptable antes de tener que ir a reponer. Este valor puede ir cambiando de según las necesidades del local.
- **Objetivo:** Si conozco cuánto comprar y cuándo comprar =¿Lograr emitir orden de compra.  
**Definición:** Una vez que el sistema dispara la alarma por falta de stock (cuándo comprar) y obteniendo la cantidad predeterminada a comprar de un producto (cuánto comprar), el sistema emitirá una orden de compra para ese producto con la cantidad predeterminada especificada.

- **Objetivo:** Lograr realizar compra en el supermercado.  
**Definición:** El repositor de stock se dirige el mismo a realizar las compras de productos faltantes con la órden de compra emitida por el sistema.
- **Objetivo:** Lograr realizar compra automática.  
**Definición:** El repositor de stock envía la órden de compra emitida por el sistema al supermercado para que el mismo se encargue de enviar la mercadería.
- **Objetivo:** Lograr recibir mercadería del supermercado.  
**Definición:** El repositor de stock es quién debe aguardar por el pedido realizado al supermercado.

### **Diagrama: Menú estandarizado entre todos los locales**

- **Objetivo:** Mantener Menú estandarizado entre todos los locales  
**Definición:** Los encargados de los locales deberán poder proponer una modificación, agregado o remoción de cualquier producto del menú actual. Este cambio deberá ser aceptado por los demás locales para lograr que el mismo se haga efectivo. Con este propósito, el sistema deberá ser capaz de informar el cambio que desea hacerse a los demás locales para que estos puedan decidir su aceptación o rechazo. Una vez que todos los locales responden, se toma una decisión en base a dichas respuestas. El cambio sólo es aceptado si la decisión de todos los locales es afirmativa.
- **Objetivo:** Lograr Cualquier local puede proponer cambios en el menú en su turno  
**Definición:** Una vez que un local posea el turno para realizar una propuesta de cambio en el menú, el sistema debe brindar la posibilidad de efectuar un cambio, ya sea un agregar, quitar o modificar algún producto de la cadena de pizzas.
- **Objetivo:** Si se toma decisión => Lograr enviar mensaje al resto  
**Definición:** Cuando el encargado de un local decide proponer un cambio en el menú, el sistema deberá ser capaz de enviar un mensaje a los demás locales notificando el cambio que se propone.
- **Objetivo:** Si se propuso cambio => Decidir si se acepta o no la propuesta  
**Definición:** Si un local propuso un cambio, se recibirá un mensaje pidiendo la confirmación o el rechazo del mismo. El encargado debe tomar esta decisión y el sistema deberá ser capaz de enviar su respuesta al local que propuso dicha modificación del menú.
- **Objetivo:** Lograr enviar mensaje con respuesta  
**Definición:** Se debe lograr enviar un mensaje con la respuesta al cambio de menú.
- **Objetivo:** Si se propuso cambio y todos respondieron => lograr informar si hay cambio o no  
**Definición:** Si se recibe la respuesta de todos los locales al cambio propuesto, se debe lograr enviar un mensaje de confirmación o rechazo del cambio, a través de un correo electrónico, a todos los locales para que se pueda actualizar el menú en cada uno de los

locales. Si un local no llegara a responder al cambio de menú propuesto en el lapso de tres horas se considera que la respuesta de dicho local fue afirmativa.

### Diagrama: Tomar pedidos

- **Objetivo:** Lograr preparar todos los pedidos  
**Definición:** Significa que deben prepararse todos los pedidos que se hayan efectuado en ese local sin dejar ninguno incompleto.
- **Objetivo:** Comenzar preparación de pedido  
**Definición:** Una vez que el pizzero le informa al sistema que comenzará con la preparación de un pedido pendiente, dicho pedido deja de estar pendiente y pasa a estar en estado de *En proceso*.
- **Objetivo:** Si se prepararon todas las pizzas del pedido => lograr finalizar pedido  
**Definición:** El pizzero debe preparar cada una de las pizzas del pedido para recién ahí poder marcar el pedido como finalizado. Una vez que se terminen de preparar todas las pizzas, el pizzero le informa al sistema que terminó de preparar el pedido. Una vez hecho esto el sistema le cambia el estado al pedido a *Finalizado* y se decrementa el stock reservado para el pedido que ya fue utilizado en la preparación del mismo.
- **Objetivo:** Lograr entregar todos los pedidos a los clientes  
**Definición:** Una vez que el pedido se encuentra en estado *Finalizado*, el mismo debe ser entregado al cliente y luego ponerlo en estado Entregado.
- **Objetivo:** Lograr conocer si puedo preparar el pedido.  
**Definición:** Para saber si puedo preparar un pedido o no dentro de un local se debe primero ingresar el pedido y luego el sistema chequea si tiene stock disponible o no para su preparación e informa a la cajera.
- **Objetivo:** Si puedo preparar el pedido => lograr cargar el pedido en el sistema  
**Definición:** Si el local posee stock para preparar un pedido, debe poder cargarse el mismo en el sistema. Una vez que el pedido se carga, se reservan automáticamente todos los ingredientes necesarios para su reparación para garantizar su entrega.
- **Objetivo:** Lograr saber el local mas cercano en el que puedo derivar.  
**Definición:** Para cumplir con este objetivo, el sistema realiza una comunicación con sus locales más cercanos donde les informa el pedido a preparar. Los locales al recibir la información evalúan si tienen stock suficiente para satisfacer el pedido, de ser así le informan al local que pueden y si no informan que no. Una vez que el local tiene los datos de sus locales vecinos, tiene la información de cuál es el local más cercano en el cuál se puede derivar un pedido.
- **Objetivo:** Si no puedo preparar el pedido y puedo derivar => lograr permitir derivar el pedido.  
**Definición:** Una vez que conozco el local más cercano a derivar, se le consulta al cliente si desea derivar el pedido a otro local y retirarlo ahí. Si el cliente afirma se realiza el



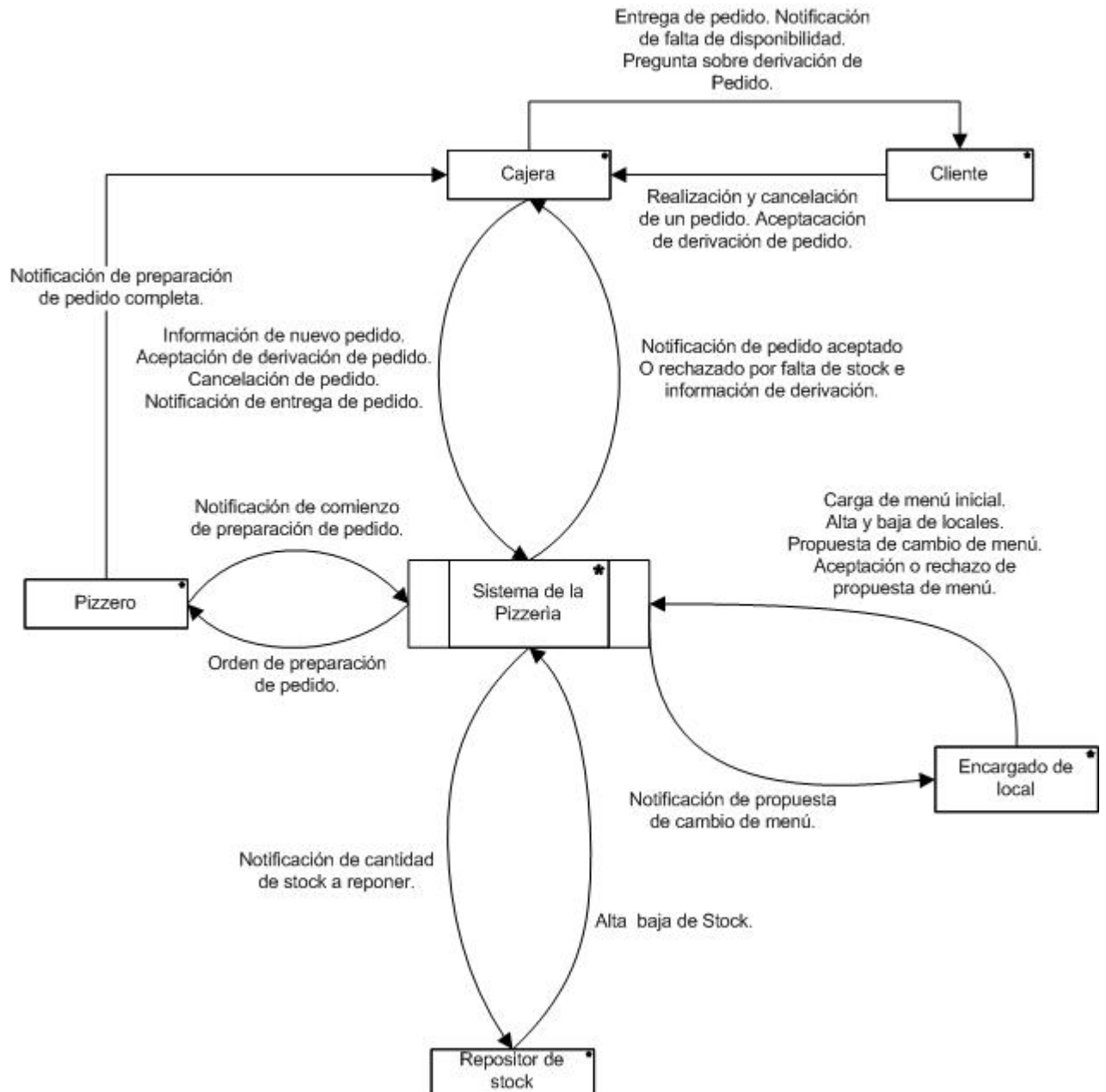
pedido en el otro local, lo cual implica la reserva de ingredientes en el otro local para garantizar su preparación y se ingresa el pedido en la lista de pedidos pendientes del nuevo local.

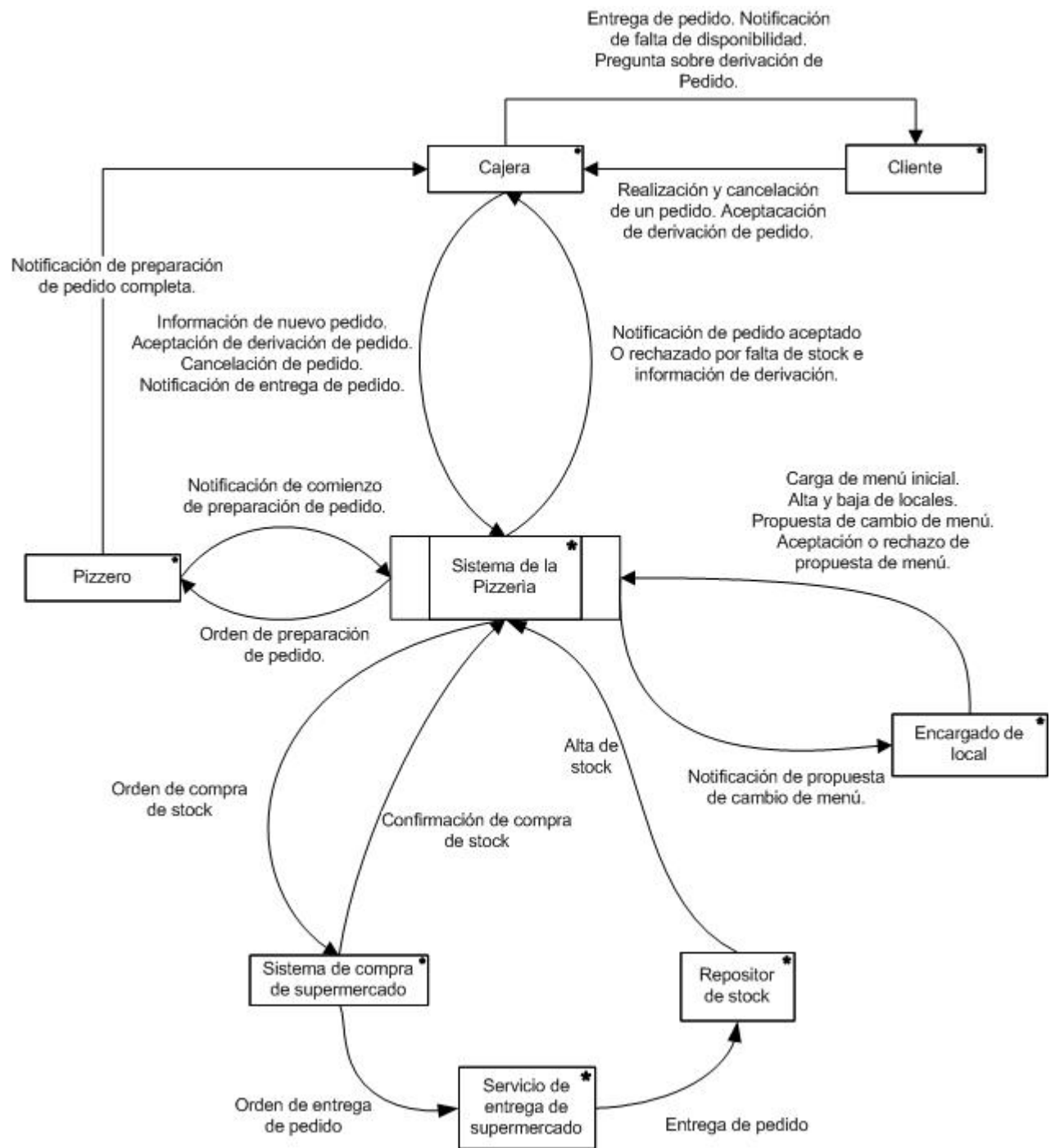
- **Objetivo:** Lograr cancepar un pedido pendiente

**Definición:** Sólo se pueden cancelar los pedidos que se encuentran pendientes de preparación, es decir, los pedidos que no se comenzaron a preparar. Esto es para que puedan reutilizarse los ingredientes reservados en caso de una cancelación.

## **Diagramas de contexto**

En la siguiente sección se presentarán los gráficos correspondientes a los diagramas de contexto referentes al sistema. Se mostrarán dos diagramas que representan las dos alternativas planteadas en los o-refinamientos del diagrama de objetivos de administración de stock:





## Escenarios

A continuación se mostrarán algunos de los posibles escenarios que se podrían llegar a presentarse en la cadena de pizzas descripta.

- Un cliente realiza un pedido a la cajera de un local. La cajera ingresa los datos del pedido a verificar y la máquina le informa si puede preparar o no el pedido. El mismo puede ser preparado. La cajera confirma el pedido con el cliente y lo carga en el sistema. En este momento, éste se almacena junto con los demás pedidos pendientes y se reservan los ingredientes necesarios para su preparación. Mientras tanto, el pizzero se encuentra preparando distintos pedidos cuando, en un determinado momento, el sistema le informa que el siguiente es el del cliente anteriormente mencionado. Una vez listo el pedido, se le entrega al cliente y éste se retira.
- Un cliente realiza un pedido a la cajera de un local. La cajera ingresa los datos del pedido a verificar y la máquina le informa si puede preparar o no el pedido. El mismo no puede ser preparado. El sistema le informa a la cajera el local más cercano donde se puede preparar dicho pedido. Se le ofrece al cliente la opción de derivar su pedido a ese local y éste acepta. La cajera deriva el pedido reservándose los ingredientes necesarios en el otro local donde el cliente lo retira posteriormente.
- Al pizzero se le cae una de las pizzas al suelo. En ese instante, informa al encargado lo ocurrido y el mismo da de baja el stock correspondiente. A continuación, el pizzero vuelve a realizar la pizza.
- Luego de realizar el pedido, el cliente se arrepiente y decide cancelarlo. En este caso, la cajera ingresa dicha cancelación en el sistema retirando el pedido de la lista de pedidos pendientes. Si la pizza no se comenzó a preparar, los ingredientes reservados para dicho pedido están disponibles para ser utilizados en algún otro. Si la pizza ya fue preparada o se encuentra en preparación, los ingredientes se pierden.
- Luego de finalizar la preparación de un pedido, el sistema detecta que el stock de un ingrediente se encuentra por debajo de la cota mínima aceptable, por lo cual emite una alarma. Una vez emitida la alarma el sistema procede a obtener la cantidad a comprar de dicho ingrediente para su reposición. Una vez que el sistema dispone de todos los datos de productos faltantes, emite una orden de compra. El repositor de stock recoge dicha orden de compra y se dirige al supermercado a comprar los productos necesarios. Al retornar a la pizzería con los nuevos ingredientes, ingresa al sistema y carga los nuevos valores de stock para los ingredientes comprados recientemente.
- Un local es poseedor del turno para realizar modificaciones al menú de todos los locales. El encargado de dicho local realiza los cambios y se los informa al resto de los encargados. Una vez que todos ven los cambios e informan su conformidad con el mismo, el encargado emite finalmente los cambios al menú, informándole a todos los locales.

## Discusión

En primer lugar, se procedió a la construcción de un diagrama de objetivos para comenzar a pensar sobre los aspectos importantes del sistema. Por cuestiones de claridad, analizaremos las discusiones referentes a la construcción de cada parte de este diagrama por separado.

En lo referente a los asuntos del manejo de stock, como primera idea se pensó en la solución clásica contar con un agente (Repositor de stock) encargado de solucionar cada inconveniente del tema. Luego, se pensó en hacer uso de la máquina para poder contar con un registro del

stock disponible de manera tal que el repositor solamente monitoree periódicamente el equipo para realizar las compras necesarias. En este ámbito se presentó la posibilidad típica en la que el repositor realiza las compras por su cuenta en cualquier mercado y además surgió la idea de hacer dichas compras a través del sistema de delivery del supermercado. Este último caso aumenta la complejidad del sistema, que tendría que ser capaz de comunicarse con el sistema de entregas del supermercado, por lo que aumentaría también el costo del sistema en consecuencia.

Otro aspecto interesante del sistema consistió en permitir la modificación del menú de la cadena por parte de los locales. A partir de este evento, surgió un tema que resultó ser central en el diseño del sistema. Para mantener la consistencia del mismo, al realizar cambios en el menú, era necesario informar al resto de los locales de tal cambio ya que el sistema a realizar será de tipo distribuido. Esto obligó a pensar fuertemente sobre este aspecto. Como primera opción, se pensó en asignar un local central encargado de tomar las decisiones con respecto a dichos cambios. Sin embargo, esto chocaba con parte de lo requerido para el sistema ya que no todos los locales podrían proponer cambios al menú. Finalmente, se decidió por realizar un sistema basado en una comunicación descentralizada entre los locales. Cada local dispone de un turno o intervalo de tiempo en el que puede proponer los cambios que considera necesarios en el menú. Es decir, si un local decide hacer un cambio y es su turno, el sistema enviará mensajes a los demás locales para que estos confirmen o rechacen dicha modificación.

Uno de los aspectos más importantes a analizar fue la atención del cliente. En primer lugar, se trató el tema relacionado con la derivación de clientes a otros locales. Esta característica del sistema resultó interesante ya que brinda una posibilidad extra a los clientes de obtener su pedido, maximizando así su satisfacción. Otra manera de lograr esto último fue tratar de disminuir el tiempo de espera del cliente en el caso de que el pedido pueda ser preparado en el local. Para cumplir con este requerimiento, se utilizó un algoritmo que optimice el orden de preparación de cada pedido, de manera de poder disminuir el tiempo global de realización de los pedidos.

A pesar de todas estas consideraciones, como en todo sistema, existen casos para los cuales el sistema no presenta una solución. Entre ellos, se encuentra el caso en que, durante la preparación de un pedido, el pizzero comete un error que arruina la preparación, desperdiciando los ingredientes correspondientes. El problema aparece cuando el local no cuenta con reservas de stock para preparar la pizza nuevamente. Otro problema relacionado con la pérdida de stock ocurre cuando, en caso de contar con el sistema automático de compra de stock, el proveedor en cuestión se equivoca al traer los ingredientes.

## Conclusiones

A la hora de encarar el trabajo práctico se presentaron dos caminos posibles por los cuales comenzar con el diseño del sistema. Pudimos haber comenzado realizando el diagrama de contexto, sin embargo se optó por realizar primero el diagrama de objetivos. Esta decisión se tomó ya que nos pareció más intuitivo pensar en objetivos generales del sistema que en acciones puntuales del mismo. Se nos ocurrió que a partir de conocer bien cuáles eran los distintos objetivos a cumplir se podrían deducir con mayor facilidad los eventos que ocurren en el dominio manteniendo así una trazabilidad entre ambos esquemas.

A pesar de lo mencionado anteriormente, fue durante el desarrollo del diagrama de objetivos donde se presentaron la mayor parte de las dificultades.

El primer problema con el que nos encontramos fue que una vez terminado el diagrama notamos ciertas inconsistencias y surgieron nuevos objetivos que no habíamos contemplado en primera instancia. Esto produjo una importante pérdida de tiempo ya que tuvimos que refactorizar el esquema numerosas veces.

También se presentaron inconvenientes a la hora de distinguir entre objetivos duros y blandos. En la primer etapa del armado del diagrama, por no tener en cuenta esta distinción pensamos a todos los objetivos como duros. Rápidamente nos dimos cuenta que varios de ellos en realidad debían ser considerados como blandos ya que cuadraban más con este tipo de objetivo.

Como conclusión final del trabajo pudimos observar que si bien en una primera instancia el sistema a modelar parecía sencillo, luego de analizarlo con mayor profundidad notamos que su complejidad era mayor a la esperada. De no haber hecho este análisis, los distintos problemas que se presentaron durante el modelaje hubieran representado un costo mucho mayor en etapas posteriores. Es por esto que nos pareció realmente importante realizar una ingeniería de requerimientos a la hora de encarar el desarrollo de un sistema.