

Índice

Introducción	2
Aclaraciones Generales	2
Vistas	3
Modelo Conceptual	3
Diagrama de Casos de Uso	6
Diagrama	6
Detalles de Casos de Uso	7
Diagramas de Actividad	13
Ciclo de vida de una orden	13
Ejemplo de manejo de stock	14
Actualización del menú	15
Máquinas de Estados Finitas	16
Sincronización del menú	16
Pedido a distancia	21
Discusión	23
Diagrama de Contexto	23
Diagramas de Objetivos	24
Pedido a distancia	24
Mantener menú estandar	24
Nivel adecuado de stock	24
Brinda servicios del menú	24
Conclusiones	25

Introducción

El objetivo de este trabajo es proporcionar la documentación necesaria para establecer un marco a los objetivos proporcionados por la cátedra. Dichos objetivos fueron refinados por los directivos de la cadena Pizza Hack y se presentan en un diagrama, el mismo es considerado un modelo final, es decir que no se presentaban opciones.

La documentación a presentar modela comportamientos presentados en el trabajo práctico número 1. Para el cumplimiento de este objetivo se tuvo en cuenta la trazabilidad de los modelos proporcionados con los diagramas de objetivos y de contexto recibidos y también resultó importante mantener la trazabilidad entre los distintos modelos realizados a efecto de reflejar de manera eficiente y clara los distintos comportamientos. Con este fin, se utilizaron diferentes técnicas de especificación. A continuación se establece una introducción a la idea final de cada técnica:

- Casos de Uso: Este modelo se utilizó para poder explicar los comportamientos que involucran al mundo y a la máquina, es decir, las que involucran a los humanos, la interacción con la impresora y al servidor de internet para sentar algunos ejemplos.
- Modelo Conceptual: Este modelo se utilizó para poder mostrar los diferentes componentes que participan para cumplir todos los objetivos finales.
- Diagrama de Actividad: Este modelo se utilizó para aclarar el comportamiento de algunas secuencias de acciones entre varios actores diferentes.
- Máquina de estados finita: Este modelo se utilizó para poder describir de manera más eficiente el comportamiento del software en algunas actividades que cobran mayor importancia por dentro del mismo.

A medida que se vayan realizando los diferentes modelos, se adjuntarán sus explicaciones en base a las decisiones tomadas, así como también se mostrará como los diferentes modelos encajan entre sí.

Por último, se presentarán algunas discusiones y conclusiones finales sobre lo realizado.

Aclaraciones Generales

En este trabajo práctico solo se realizó una presunción importante. Al leer los diagramas de objetivos, el diagrama de contexto y los escenarios presentados se pudieron observar algunas inconsistencias, por ejemplo en el diagrama de contexto no existe relación entre el mozo y el cliente y, sin embargo, los escenarios no indican lo mismo. En estos casos, se decidió seguir la línea de los diagramas y no de los escenarios, aunque cada decisión se encuentra detallada en el cuerpo del trabajo.

Vistas

En esta sección presentaremos las diferentes especificaciones realizadas durante el presente trabajo práctico. Se explicará como se abordó cada parte del trabajo, explicando para que momentos fue utilizada cada técnica de especificación, y el porque de esta desición.

En un aspecto general, se dividieron las técnicas de especificación según los siguientes criterios:

- **Modelo Conceptual:** Se utilizará para un entendimiento global del funcionamiento total de la pizzería, entendiendo las entidades que interactúan dentro y con el mismo.
- **Diagrama de Casos de Uso:** Se utilizará para mostrar todas las interacciones que la máquina tiene con los diversos actores.
- **Diagramas de Actividad:** Se utilizarán para mostrar secuencias de acciones, usualmente agruparán diversos casos de uso que posean un hilo conductor.
- **Máquinas de Estado Finitas:** Se utilizarán para mostrar las acciones que ocurren principalmente dentro de la máquina para de esta manera, junto a los demás esquemas, poder dar un panorama completo del comportamiento del software.

Modelo Conceptual

El modelo conceptual es una primera aproximación a todo lo que se quiere describir detalladamente en el transcurso de este trabajo. Explica de manera temprana la relación que existe entre los actores externos y las diferentes partes que componen a la pizzería, para lograr un buen entendimiento de las interconexiones presentes sin indicar ningún tipo de comportamiento por el momento.

A continuación se presenta el diagrama realizado para este modelo, seguido de las pertinentes explicaciones.

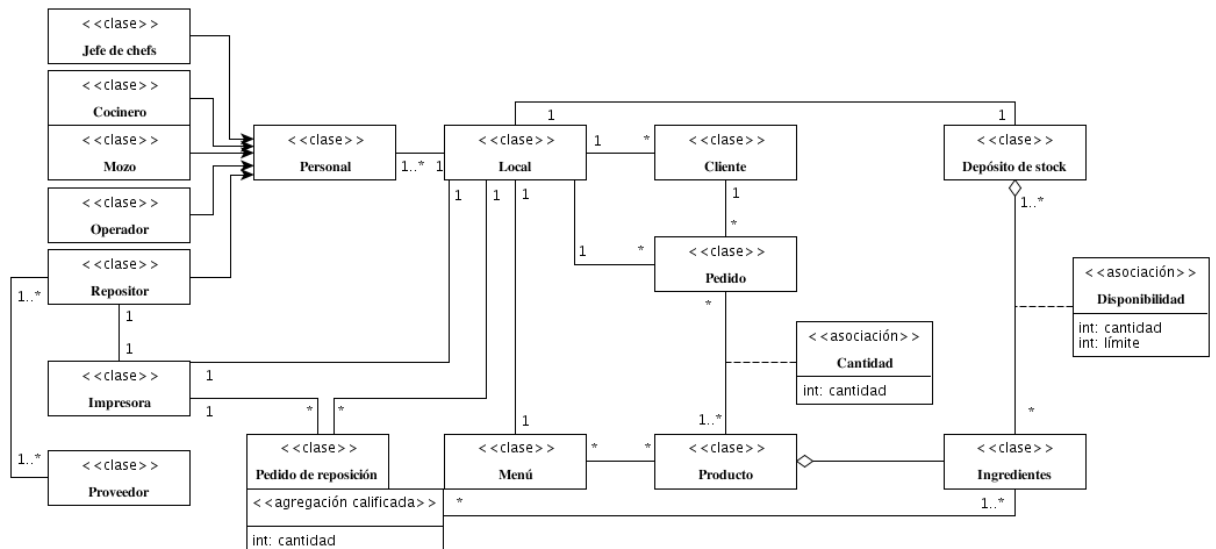


Figura 1: Modelo Conceptual

Del diagrama anterior se explayan los siguientes puntos para un mejor entendimiento:

- Dado que hay diferentes cargos dentro del personal se decidió por utilizar el recurso de herencia para identificarlos. Si bien cada subclase no tiene ningún atributo muy distintivo, el repositor tiene algunas relaciones particulares. Este fue el factor determinante para elegir usar herencia y no poner el tipo de personal como un atributo enumerado, ya que entonces habría que poner las relaciones del repositor a personal y luego restringir por OCL que el relacionado sea de tipo repositor.
- Por la clase menú se entiende el menú que la pizzería ofrece más allá del stock que se encuentre disponible.
- Por la clase producto se entiende cualquier producto que la pizzería alguna vez ofreció. Esto se realizó de esta manera dado que pensar los productos solamente como los que se encuentran en el menú podría traer inconsistencias. Un cliente puede realizar un pedido que contenga un producto que está en el menú (y haya stock para realizarlo), pero una vez hecho este pedido, el producto podría ser removido del menú en una actualización, y aún así el pedido con el producto desaparecido del menú podría perdurar en el sistema dado que no se realizó todavía. Es por esto que pueden existir productos por fuera del menú.
- El cliente esta pensado como una persona o un grupo de personas dentro un local, por eso la multiplicidad cliente-local es uno. Si esa persona o grupo de personas se movilizacen hacia otro local no hay razón para no pensarlos como un nuevo cliente dentro del otro local.
- Un pedido de reposición esta conformado por ingredientes y sus cantidades pedidas, por lo que se optó por utilizar una agregación calificada para modelar esto.

A continuación se detallarán las restricciones sobre el modelo que no se desprenden del diagrama mediante el lenguaje OCL, indicando en cada caso que fue lo que se quiso restringir en idioma castellano.

Se explicitan las correspondencias de cada local con cada tipo de empleado (clase personal y sus subclases).

context Local

```
self.personal→select(p|p.isKindOf(Jefe de Chefs)).size()= 1
self.personal→select(p|p.isKindOf(Cocinero)).size()> 0
self.personal→select(p|p.isKindOf(Mozo)).size()> 0
self.personal→select(p|p.isKindOf(Operador)).size()= 1
self.personal→select(p|p.isKindOf(Repositor)).size()= 1
```

El menú es el mismo para todos los locales.

context Local

```
Local.allInstances().menú→asSet()→size()= 1
```

Para todo depósito, cada ingrediente cuyo nivel de stock se encuentre por debajo de mínimo tolerable tiene que estar en un único pedido de reposición relacionado con el local correspondiente al depósito en cuestión.

context Depósito de stock

```
self.allInstances()→select(d|d.cantidad<d.límite)→forAll(d|d.ingredientes.pedidodereposición.size()=
1 and d.ingredientes.pedidodereposición→forAll(p|p.local.depósitodestock=d.depósitodesotck)
```

Se pide, para un pedido de reposición, una cantidad de stock mayor o igual al límite tolerable de los ingredientes que se encuentran en el pedido de reposición. Esto es así para que una vez que se realiza el pedido, seguro se cuente con el mínimo necesario.

context Ingredientes

`self.pedidodereposición→.forall(p—p.cantidad≥self.disponibilidad.límite)`

La impresora utilizada en un pedido de reposición es la impresora del local que emitió dicho pedido.

context Pedido de reposición

`self.Impresora=self.Local.Impresora`

Para todo pedido, el local en donde está ese pedido debe ser el local donde el cliente realizó ese pedido.

context Pedido

`pedido.local = pedido.cliente.local`

Diagrama de Casos de Uso

Esta técnica sirve para mostrar como son las interacciones entre el mundo y la máquina, es decir que se abstraen varias de las relaciones presentes obviando las relaciones propias entre actores que se encuentran por fuera de la máquina, así como las interacciones que son intrínsecas de la máquina.

Diagrama

A continuación se presenta el diagrama de casos de uso para toda la máquina, es decir, se engloban todas las interacciones en un mismo diagrama y luego se detallará en particular cada caso de uso.

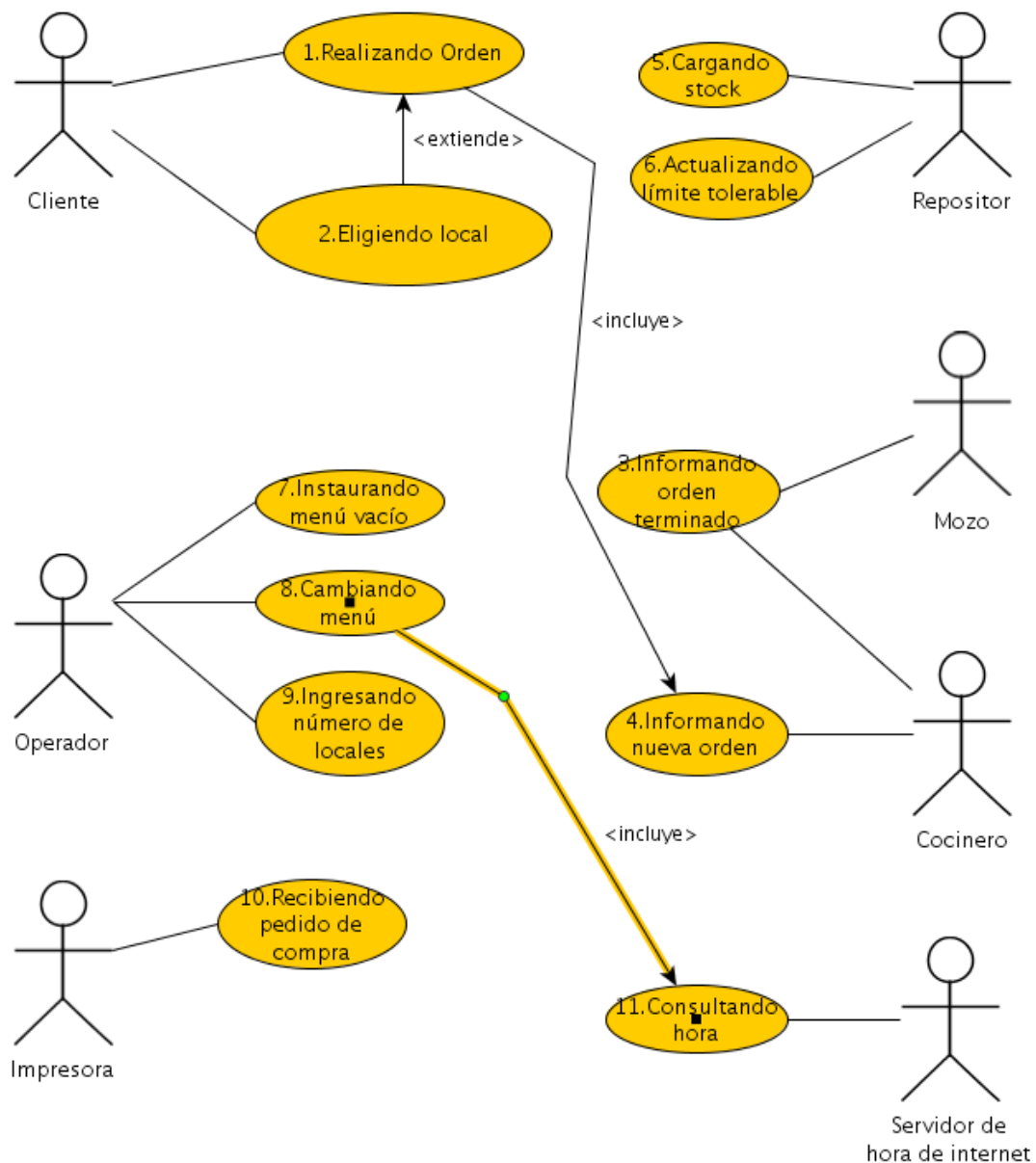


Figura 2: Diagrama de Casos de Uso

Detalles Casos de Uso

A continuación se detallará cada caso de uso, dando la descripción de la sucesión de hechos, además de los actores y la pre y postcondición.

Para seguir un hilo conductor, los casos de uso se encuentran numerados en forma secuencial agrupados por el o los actores participantes.

Se seguirá este mismo secuenciamiento para realizar la descripción.

En primer lugar se encuentra uno de los casos de uso más importante en todo el funcionamiento de la pizzería. La realización de un nuevo pedido por parte del cliente.

Nombre Casos de uso: 1.Realizando orden	
Actores: Cliente	
Precondición: True	
Postcondición: Se logra hacer el pedido	
Curso Normal	Curso Alternativo
1.1 El cliente realiza una orden.	
2.1 El sistema verifica que se puede realizar la orden (stock). Si hay stock ir a 3.1, si no hay stock ir a 4.1	
3.1 El sistema toma el pedido y decrementa el stock debidamente.	
3.1.1 Se incluye el caso de uso “Informando nueva orden”. Ir a 5.1	
4.1 El sistema notifica que no se puede realizar el pedido. Busca otros locales con stock	Si no hay otros locales con stock ir a 5.1
4.1.1 Se extiende al caso de uso “Elegiendo local”.	
5.1 Fin de caso de uso	

Nombre Casos de uso: 2.Elegiendo Local	
Actores: Cliente	
Precondición: El cliente realizó un pedido que no se puede efectuar en el propio local	
Postcondición: El cliente elige un nuevo local	
Curso Normal	Curso Alternativo
1.1 El cliente recibe una lista de locales donde se puede satisfacer su pedido	
2.1 El cliente elige un local de la lista recibida	2.2 El cliente no eligen ningún local. Ir a 4.1.
3.1 El software debe informar la nueva orden en el local elegido y decrementa el stock del local elegido. Se incluye Caso de Uso Informando Nueva Orden	
4.1 Fin Caso de Uso	

Este caso de uso, en contraste a la mayoría de los demás, contiene una precondición. Este se incluye para imponer una restricción que no se puede imponer desde el diagrama mismo dado a que no existe expresividad para esto.

La precondición del caso de uso dice *el cliente realizó un pedido que no se puede efectuar en el propio local*. Esta precondición se incluye para mostrar que este caso de uso no tiene sentido por sí solo sin una precedencia directa del caso de uso *Realizando Nueva Orden*. Esto no se puede expresar en el diagrama mismo dado que en la relación entre los dos casos de uso aparece la etiqueta *extiende*. Esto es así dado que no siempre que se realiza una nueva orden se elige otro local, por lo que no sería correcta la inclusión de una etiqueta *incluye*. Luego, se usa esta

precondición para indicar esta restricción y para poder hacer la descripción del caso de uso ya asumiendo que el cliente realizó una orden que no pudo ser satisfecha en el propio local.

Nombre Casos de uso: 3.Informando Orden Terminada	
Actores: Mozo, Cocinero	
Precondición: Existe una orden en proceso	
Postcondición: Se informa que se finalizó una orden	
Curso Normal	Curso Alternativo
1.1 El cocinero cocina una pizza que se encontraba en una orden en proceso	
2.1 El cocinero ingresa en el sistema la finalización de la orden pertinente	
3.1 El software actualiza el sistema de ordenes, tildando como realizada la orden que el cocinero notificó	
4.1 El software avisa al mozo correspondiente que la orden se encuentra finalizada para retirar	
5.1 El mozo se entera que se terminó una orden y queda dispuesto a ir a buscar la orden para su entrega	
6.1 Fin Caso de uso	

Nombre Casos de uso: 4.Informando nueva orden	
Actores: Cocinero	
Precondición: Un cliente realizó un pedido	
Postcondición: El cocinero es informado de la nueva orden a realizar	
Curso Normal	Curso Alternativo
1.1 El software, luego de procesar el pedido del cliente, informa al cocinero sobre la nueva orden	
2.1 El cocinero es notificado de la nueva orden y se dispone a su realización	
3.1 Fin Caso de Uso	

Se puede notar que los casos de uso descriptos hasta el momento se encuentran bastante relacionados, dado que todos juntos conforman una secuencia común de una cena completa de un cliente en un determinado local. Si bien esta relación es evidente, en el diagrama de casos de uso no se encuentran en un hilo secuencial dado que esto no es correcto. Desde un punto de vista de casos de usos atómicos no sería correcto relacionar todos los casos de uso anteriormente descriptos ya que por ejemplo, entre la realización de una nueva orden y la información de una orden finalizada pueden pasar horas en el medio y muchos otros sucesos, por lo que no sería correcto unirlos en el diagrama.

Sin embargo, como la relación secuencial es notoria, estos casos de uso serán posteriormente agrupados en un diagrama de actividad que muestre un hilo de ejecución completo.

Nombre Casos de uso: 5.Cargando Stock	
Actores: Repositor	
Precondición: True	
Postcondición: Se carga el stock en el sistema	
Curso Normal	Curso Alternativo
1.1 El repositor recibe nuevo stock de algún proveedor	
2.1 El repositor ingresa al sistema el stock a actualizar	
3.1 El software actualiza el stock de los productos que fueron sumados al deposito de ingredientes	
4.1 Fin Caso de Uso	

Nombre Casos de uso: 6.Actualizando límite tolerable	
Actores: Repositor	
Precondición: True	
Postcondición: Se actualiza el límite tolerable de algún producto	
Curso Normal	Curso Alternativo
1.1 El repositor decide modificar el límite tolerable de algún ingrediente en particular	
2.1 El repositor ingresa al sistema el ingrediente que quiere actualizar y su correspondiente límite actualizado	
3.1 El software actualiza el límite tolerable para el ingrediente que fue señalado	
4.1 Fin Caso de Uso	

Al igual que sucedió con los primeros casos de uso, los casos de uso 5 y 6 también se encuentran correlacionados aunque atómicamente no lo estén. A su vez, también es claro que en una secuencia de acciones del mundo real, estos casos de uso podrían estar relacionados con los casos de uso 1(realizando orden) y 10(recibiendo pedido de compra). Estas relaciones secuenciales se verán modeladas posteriormente en el diagrama de actividad referente a todo lo que sucede con respecto al stock cuando se realiza una orden.

Nombre Casos de uso: 7.Instaurando Menú vacío	
Actores: Operador	
Precondición: True	
Postcondición: Se carga el menú vacío en el sistema	
Curso Normal	Curso Alternativo
1.1 El Operador carga un menú vacío al sistema	
2.1 El sistema se actualiza para trabajar con este nuevo menú vacío	
3.1 Fin Caso de Uso	

Nombre Casos de uso: 8.Cambiando Menú	
Actores: Operador	
Precondición: True	
Postcondición: Se actualiza el menú	
Curso Normal	Curso Alternativo
1.1 El operador posee una nueva actualización para el menú	
2.1 El operador carga en el sistema la actualización	
3.1 El software mira el número de local al que pertenece el operador	
4.1 El software consulta la hora. Se incluye Casos de uso 11	
5.1 El software calcula si el operador se encuentra apto para modificar el menú	El operador no se encuentra apto. Ir a 7.1
6.1 El software actualiza el menú	
7.1 Fin Caso de Uso	

Nombre Casos de uso: 9.Ingresando número de locales	
Actores: Operador	
Precondición: True	
Postcondición: Se ingresan los números de los locales	
Curso Normal	Curso Alternativo
1.1 El Operador carga el número de locales	
2.1 El sistema actualiza el número de locales para realizar cálculos de round robin	
3.1 Fin Caso de uso	

Los 3 casos de uso anteriormente descriptos son los que más están relacionados con el hecho de la actualización del menú. Esto, como ya se explicó con anterioridad, se mostrará de una manera más visual en un diagrama de actividad que muestre lo que sucede al actualizar un menú en el mundo y en la interfaz. Mientras que luego, con una máquina de estados, se modelará lo que sucede dentro de la máquina con esta acción.

Nombre Casos de uso: 10.Recibiendo pedido de compra	
Actores: Impresora	
Precondición: True	
Postcondición: La impresora recibe un nuevo pedido de compra	
Curso Normal	Curso Alternativo
1.1 El software posee uno o muchos ingredientes que deben ser repuestos	
2.1 El software genera el pedido de compra para satisfacer la necesidades calculadas	
3.1 El software manda a la impresora la correspondiente orden de compra	
4.1 Fin Caso de Uso	

Nombre Casos de uso: 11.Consultando Hora	
Actores: Servidor de hora de internet	
Precondición: Un operador esta intentando actualizar el menú	
PostCondición: Se obtiene una actualización de la hora	
Curso Normal	Curso Alternativo
1.1 El software requiere la hora para calculos de round robin	
2.1 El Software le pide al servidor una actualización de la hora	
3.1 El Servidor devuelve lo solicitado al software	
4.1 Fin Caso de Uso	

Diagramas de Actividad

Los diagramas de actividad son una representación para mostrar el secuenciamiento de varias acciones que se encuentran relacionadas por algún hilo conductor. Los diagramas de actividad se encuentran enriquecidos por el hecho de poder soportar varios actores en un mismo hilo mediante la separación por andariveles de las interacciones de cada actor, es por esto que son muy adecuados para modelar las acciones que suceden entre la máquina y los actores externos e incluso las acciones que suceden entre los actores externos sin interacción con la máquina.

Si bien las interacciones entre el mundo y la máquina ya fueron descriptas en el diagrama de casos de uso, se realizó de forma diferente ya que la técnica de casos de uso se basa en centrarse atómicamente en cada caso de uso y solo se explicita una relación entre dos casos de uso cuando la relación es inmediata.

En contraposición a los casos de uso, se utilizarán los diagramas de actividad para modelar las secuencias de acciones que sucederían en la pizzería, sin centrarse en cada paso de la secuencia particularmente (dado que ya fueron explyados en los casos de uso).

Se decidió que la técnica sería utilizada para los siguientes casos particulares:

- El ciclo de vida completo de una orden.
- Ejemplo de manejo de stock.
- Actualización del menú.

Ciclo de vida de una orden

Este diagrama de actividad sirve para agrupar las acciones sucedidas en el ciclo de vida de una orden, es decir desde el momento que un cliente pide una orden hasta el momento que recibe su pizza(o bien no la recibe porque no se puede satisfacer). Cabe destacar que en el diagrama de actividad se abstraen varias situaciones para centrarse solamente en la orden en sí. Es decir, si bien usar los ingredientes para realizar una orden podría desencadenar un pedido al proveedor, no será modelado en este diagrama, sino en el siguiente.

En este diagrama se destaca la diferencia entre que el pedido se cocine en el local donde el cliente se encuentra o en otro, por lo que se diferencian entre el cocinero y el mozo del local propio y el cocinero y el mozo del local externo donde el cliente va a ir a buscar la pizza.

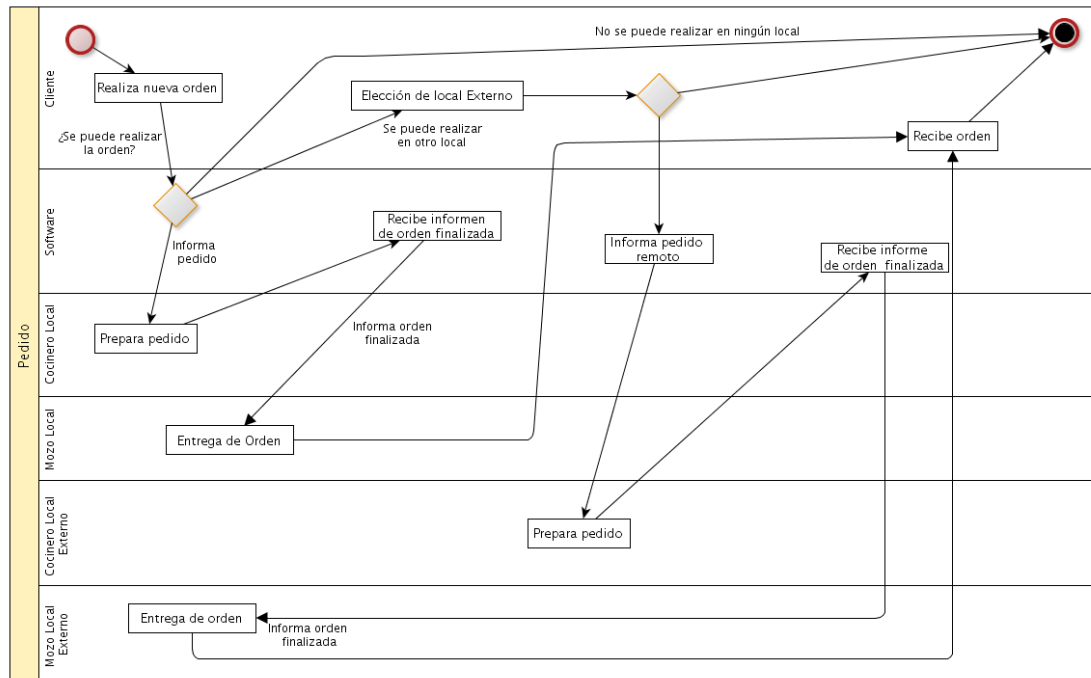


Figura 3: Ciclo de vida de una orden

Ejemplo de manejo de stock

Este diagrama de actividad se utilizará para modelar las acciones pertinentes al manejo de stock cuando se realiza una orden. Dado que solo se quiere modelar lo relacionado con el stock, ya que los otros aspectos respectivos a la realización de una orden fueron modelados anteriormente, el diagrama presenta una secuencia de acciones donde un cliente realiza una orden que se puede satisfacer por dicho local para poder mostrar concretamente que sucede con el stock sin pérdida de generalidad.

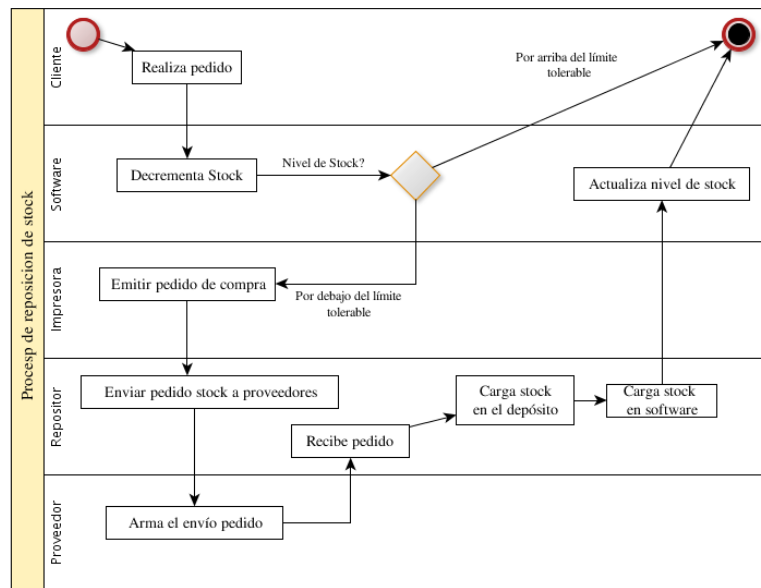


Figura 4: Ejemplo de manejo de stock

Actualización del menú

Este diagrama de actividad sirve para modelar el hilo de ejecución de las acciones referidas a la actualización de un menú en cuanto a la interacción de los actores externos participantes con la máquina. Lo que sucede dentro de la máquina para mantener una consistencia entre los locales es inherente a la máquina misma por lo que será modelado posteriormente con la técnica de máquinas de estados.

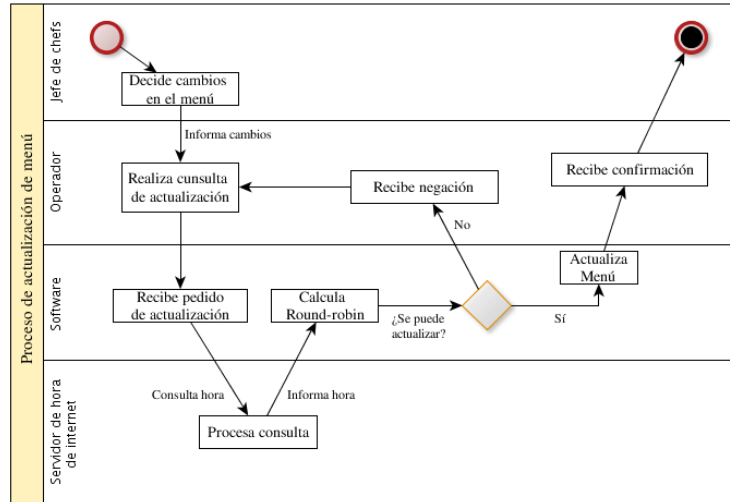


Figura 5: Actualización del menú

Máquinas de Estados Finitas

Las máquinas de estados finitas son importantes en este trabajo práctico para poder modelar las acciones que suceden totalmente dentro del software. En este trabajo en particular hay muchas interacciones entre el software y los agentes externos y no existen tantas acciones dentro del mismo que necesiten de una explicación con máquinas de estado, por lo que este modelo se utilizará en pocos casos.

En particular en este trabajo, esta técnica se utilizó para modelar dos aspectos que son relevantes al software por dentro, siendo dignos de una explicación más hexaustiva. Por un lado se explicará el proceso de actualización de menú, viendo como se realiza la sincronización entre los diferentes locales. Por otro lado, se utilizará la técnica para explicar como el software puede revisar el stock de otros locales para derivar el pedido de un cliente.

Sincronización del menú

En esta máquina de estados se explicarán los cambios internos del software a respuesta de los estímulos externos. Para modelar los estímulos externos se realizarán máquinas de estados por cada agente externo que interactue con el sistema.

A continuación se irán explicando cada agente, para llegar a la explicación final del accionamiento del software. El índice i en cada máquina de estados, va desde 1 hasta la cantidad de locales y hace referencia al número del local.

Jefe de chefs:

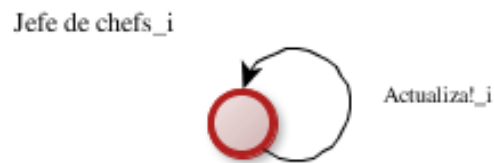


Figura 6: Jefe de Chefs

El accionar del jefe de chefs es muy simple, cuando tenga ganas va a querer actualizar y esto es lo único significativo que hace. Esta acción tendrá que ser *escuchada* por el operador de su propio local.

Servidor de hora de internet:

Servidor



Figura 7: Servidor de hora

El accionar del servidor de hora también es simple, solamente esta dispuesto a escuchar una acción que es que le pidan la hora, cuando esta sucede, setea el tiempo en la hora actual del día. Es el mismo servidor por todos los locales por eso no tiene diferenciación de índice ni en el nombre del servidor, pero si tiene índice la transición para no forzar que todas las máquinas pidan hora al mismo tiempo.

Operador:

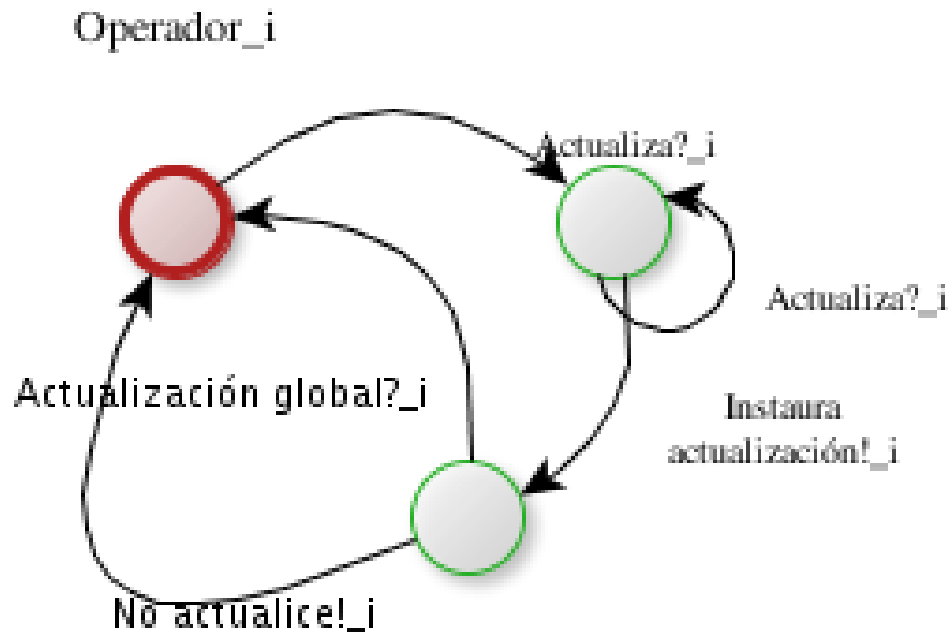


Figura 8: Operador

El Operador ya tiene algunas acciones que merecen una explicación más detallada. Por un lado el operador debe escuchar la acción de actualizar generada por su propio jefe de chefs, luego una vez que posee una actualización del jefe de chefs, busca instaurar esta actualización en el software del local, generando la acción *Instaura actualización*. Aquí el Operador, pasa a un estado intermedio donde espera a que el software le diga si se pudo instaurar globalmente la actualización o si no se pudo. Durante este estado, el Operador no escucha actualizaciones de su jefe de chefs. Además cabe destacar que si el software no le da permiso al operador para instaurar la actualización esta actualización se pierde y el jefe de chefs debe volver a generarla. Estas fueron decisiones tomadas por el grupo, ya que no se especifica como debe ser esto ni en los objetivos ni en el diagrama de contexto, por lo que quedó a criterio del grupo.

SoftwareLocal:

El software comienza en un estado inicial donde lo que tiene permitido hacer es escuchar las actualizaciones globales de los otros locales o, por otro lado, esperar un pedido de instauración de actualización de su propio operador. Una vez que su operador le pide instaurar una actualización, el software pasa a un estado donde va a querer hacer una actualización. En este estado el software pide la hora al servidor para saber si es su momento para actualizar o no. Si es su momento, después de un timeout el software *saldrá* por la transición superior, generando una actualización global que será escuchada por los demás locales. En caso de no ser su momento, no realizará su actualización volviendo al estado inicial.

Pedido a distancia

En un pedido a distancia solo se presentarán dos diferentes máquinas de estado, una por el cliente de un local y otra por el software de cada local. Para simplicidad del modelo se asume que un solo cliente por local está involucrado por vez en este proceso.

A continuación se presenta la máquina de estados del cliente.

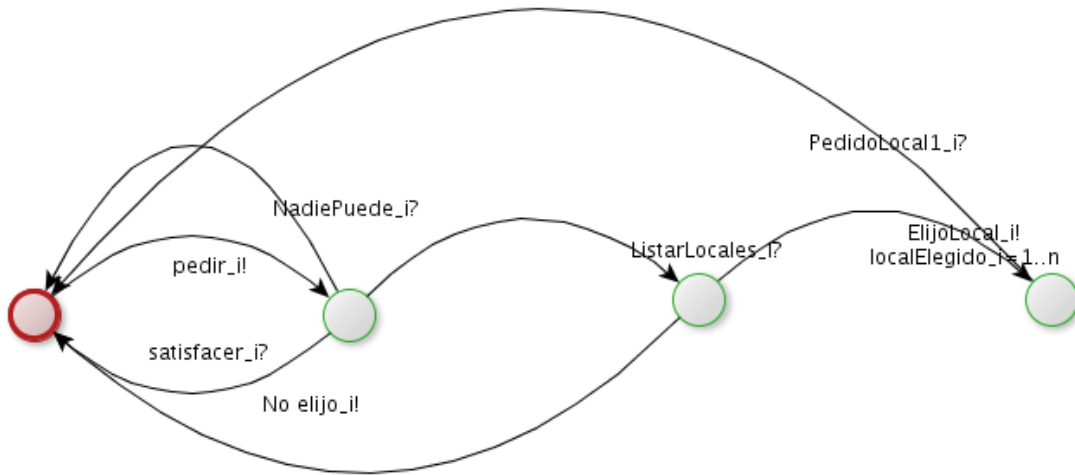


Figura 10: Cliente

El cliente empieza en un estado inicial en donde lo que puede hacer es pedir una pizza, si la pizza puede ser hecha por el mismo local o si ningún local puede satisfacerlo, el cliente vuelve al estado inicial ya sea satisfecho o no, pero sin nada más que hacer en esta secuencia. Por otro lado, el software puede presentarle una lista de locales para elegir y el cliente pasa a un estado de elección. Si no elige ningún local, termina el proceso. Si, por el contrario, elige un local, pasará a un estado donde se generará el pedido al otro local terminando el proceso.

Luego, se presenta la máquina de estados del software de cada local. Antes que nada cabe destacar que en todo nodo la fsm está *escuchando* el pedido StockTodos de las demás máquinas así como también los pedidoslocal de las demás máquinas que les mandan pedidos, lo cual se omite para una mayor claridad del dibujo ya que no aporta mayor información más que la idea intuitiva que un local siempre tiene que estar dispuesto a recibir un pedido de otra máquina o a que le pidan su stock.

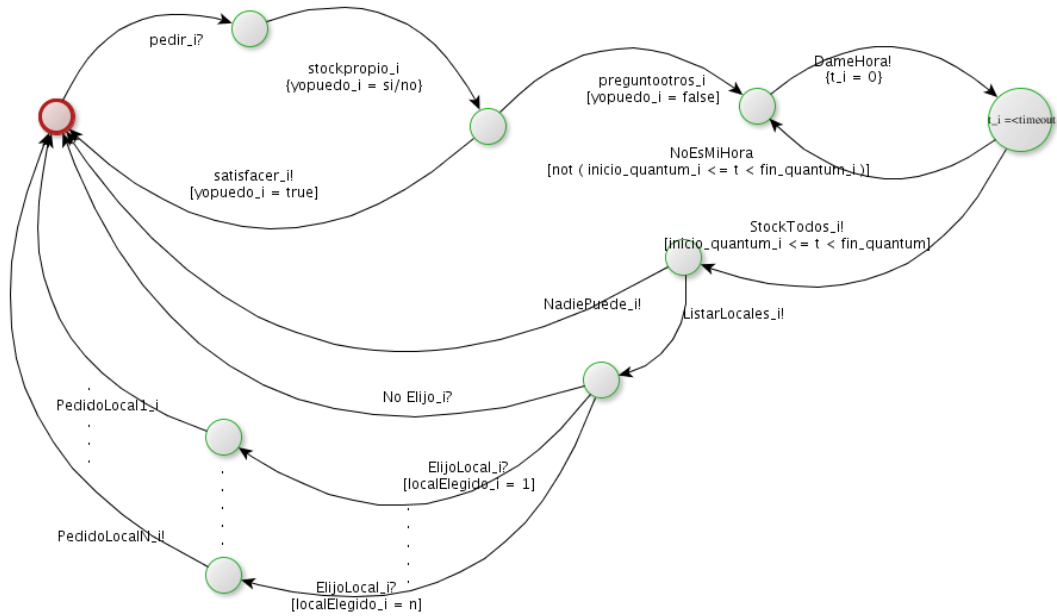


Figura 11: Software

La máquina comienza en un estado inicial donde lo único que hace es escuchar el pedido de un cliente. Luego de que un cliente hace un pedido, el software va a revisar su stock propio y se va a setear la variable `yopuedo_i` en base a si puede satisfacer el pedido o no. Si puede satisfacerlo, lo satisface y vuelve al estado inicial esperando por otro cliente. Si no puede, pasara a un estado para preguntar por los otros. Aquí el software pide la hora para saber si esta en su quantum, con una solución análoga a la propuesta en la máquina de estados para actualizar el menú. Una vez que esta en su hora, genera una transición denominada, `StockTodos_i!` que sera escuchada por todas las maquinas, que indicarán si pueden realizar el pedido en cuestión. Si nadie puede satisfacer el pedido, el software vuelve al estado inicial a esperar por otro cliente. Si existen otros locales que puedan hacer el pedido, el software se los lista al cliente.

Aquí pueden pasar dos cosas. En primer lugar, el cliente podría no elegir ningún local a distancia por lo que se vuelve al estado inicial. Por otro lado, el cliente puede elegir un local y setear la variable `localelegido_i`, de esta manera el software saldra por la unica transición que tenga la condición true, llegando a un estado donde lo que va a hacer es generar un pedido externo al local correspondiente.

De esta manera, se genera todo el ciclo posible en un pedido a distancia, mostrando la funcionalidad del software en función de los requisitos del cliente.

Discusión

Durante el desarrollo del presente trabajo práctico, se fue explicando en cada momento el porque de la elección de las diferentes técnicas de modelado en cada uso en particular. Dicha decisión es en parte arbitraria y en parte no, es decir hubo decisiones que estuvieron influenciadas por la naturaleza de las técnicas en sí mismas mientras que hubo otras decisiones que fueron más allá de las técnicas y que tuvieron que ver con la expresividad que se le quiso dar a cada parte del trabajo en particular. Con lo anteriormente dicho, lo que se quiere notar es que hubo partes de la solución donde no hubo decisiones personales, por ejemplo la técnica de casos de uso sirve para modelar las interacciones entre el mundo y la máquina y eso no fue una decisión del grupo, mientras que la cantidad y la especificidad de los diagramas de actividad y las máquinas de estado, si fueron una decisión tomada por el grupo.

En lo siguiente, se verá porque las decisiones se tomaron de esta manera, explicando como estas decisiones logran que la conjunción de los modelos prediquen sobre todos los aspectos necesarios expuestos en los diagramas de objetivo y en el diagrama de contexto.

Diagrama de Contexto

Se puede ver que el diagrama de contexto se encuentra explicado en su totalidad por los diferentes modelos que se explotaron en este trabajo práctico.

Por un lado, el diagrama de Casos de Uso modela todas las interacciones que existen entre la máquina y el mundo. Es decir, cada caso de uso explica atómicamente cada una de las relaciones presentes entre un actor externo y el software, como por ejemplo la interacción que existe entre el cocinero y el software, cuando el primero le avisa al segundo que la orden fue finalizada.

Dada la naturaleza de la técnica, quedan por fuera de la misma, las interacciones presentes entre los actores externos en los que la máquina no tiene participación.

Para resolver esto, se utilizaron los diferentes diagramas de actividad. Basicamente, se puede notar, que cada una de las relaciones del diagrama de contexto se puede categorizar en una o más de las 3 siguientes categorías:

- Las relaciones que tienen inferencia en una orden de un producto.
- Las relaciones que tienen inferencia sobre el manejo adecuado del stock de la pizzería.
- Las relaciones que tienen inferencia sobre la actualización y el correcto mantenimiento del menú.

Es por esto, que se decidió realizar tres diferentes diagramas de actividad, para englobar las diferentes acciones y mostrar así como estan relacionadas en el mundo real, aspecto que se escapaba de los casos de uso, debido a su atomicidad.

Por otro lado, el modelo conceptual también aporta a la explicación del diagrama de contexto, aunque como ya se mencionó, su aporte es un poco más transversal a todos los objetivos y al diagrama de contexto, dando una visión más general.

Por último, cabe destacar que las máquinas de estado no realizaron un mayor aporte a la explicación de este diagrama, dado que estan más bien centradas en lo que sucede dentro del software.

Diagramas de Obejetivos

Pedido a distancia

Este objetivo se centra en el hecho de poder pedir a distancia, pero si vemos las refinaciones vemos que este diagrama se centra en el hecho de las acciones internas que deben ocurrir para esto, ya que toda hoja esta asignada al software. Es por esto, que este objetivo se explica con una máquina de estado, ya que muestra que pasa con el software por dentro para lograr un pedido a distancia.

Mantener menú estandar

El objetivo de mantener un menú estandar posee un refinamiento en tres subobjetivos. Por un lado se encuentra el simple objetivo referente al menú vacio, este objetivo no tiene complejidad y se encuentra descrito en un caso de uso que hace referencia a esta acción atómica.

Por otro lado, los otros dos subobjetivos si presentan una complejidad mayor y tratan sobre el aspecto más importante, que es como lograr una correcta sincronización. Los comportamientos del sistema referentes a la satisfabilidad de estos subobjetivos se encuentran explicados de dos maneras diferentes: Por un lado existe un diagrama de actividad que explica lo que sucede por fuera de la máquina, es decir explica lo que sucede entre todos los actores que tienen que ver en esta acción, y por el otro se encuentra una máquina de estados que muestra lo que sucede dentro del software mismo, al momento de querer lograr estos subobjetivos.

Nivel adecuado de stock

Este objetivo se encuentra abordado de diferentes maneras también. Se encuentra explicado mediante los casos de uso que hacen referencia a la interacción entre la máquina y los diferentes actores que tiene influencia en la variabilidad del stock como el repositor y el cliente, y también por los actores que son influenciados por la variabilidad del stock, como podría ser la impresora.

Además se puede encontrar un diagrama de actividad que modela toda una secuencia que podría suceder en el mundo real, mostrando como esta secuencia se relaciona con el stock.

Brindar servicios del menú

Este objetivo se encuentra refinado en dos subobjetivos que predican sobre poder realizar una orden, o impedir que se realice si no se puede. Estos subobjetivos se encuentra explicados por diferentes partes del trabajo práctico. Por una lado, todo lo referente a stock esta bastante ligado al objetivo anterior y, por ende, se encuentra explicado en el diagrama de actividad mencionado anteriormente, y en los casos de uso correspondientes. Mientras que todo lo referente al pedido de una orden en sí y lo que sucede con ella se encuentra también explicado de dos formas diferentes. Por un lado se tienen los casos de uso atómicos inherentes a cada parte de la acción total de pedir una pizza y recibirla, ejemplos de esto podrían ser los casos de uso Informando nueva orden y eligiendo local. Por otro lado, se tiene un diagrama de actividad específico para el ciclo de vida de una orden que muestra como interactuan todas las partes y las diferentes alternativas que se pueden presentar.

Conclusiones

Luego de finalizar con los modelos presentados en este trabajo, se llegó a varias conclusiones.

En primer lugar se encontró un gran contraste entre el primer trabajo práctico y el segundo. En el primer trabajo el enunciado era bastante vago y ambiguo debido a que se describía una situación coloquial. Esto trajo aparejado el hecho de que la principal acción del grupo fue tomar una gran cantidad de decisiones y fundamentar las diferentes alternativas posibles, en los cuales los modelos a utilizar no eran muy restrictivos. Por otro lado, en este segundo trabajo práctico el grupo se encontró con una aproximación más formal, por lo que la mayoría de las decisiones fueron sobre cual modelo utilizar para explicar cada parte de lo entregado por la cátedra, pero una vez realizado esto el trabajo tuvo más que ver con la correcta utilización de las técnicas (sus sintaxis, sus restricciones) y no con presentar alternativas, ya que estas ya habia sido podadas entre el primer trabajo y los supuestos directivos de la pizzería.

En segundo lugar, se concluye en que las etapas al encarar el trabajo fueron satisfactorias aunque se podrían considerar otras maneras. A saber, la primer etapa del trabajo consistió en definir el alcance de cada modelo por sobre lo que se quería explayar. Una vez hecho esto, se comenzo con el modelo conceptual dado que provee una idea temprana de como se relacionan todas las partes involucradas en el manejo de la pizzería. Al tener la idea dada por el modelo conceptual, se prosiguió con la realización de los casos de uso, dado que estos pueden explicar bien de manera atómica como es la relación entre la máquina y los diferentes actores. Una vez tenidas las relaciones de forma individual, se realizaron los diagramas de actividad para poder agrupar estas relaciones individuales en secuencias de acciones más representativas de posibles escenarios del mundo real. Por último, se abordaron las máquinas de estado para dar una mayor explicación a las acciones internas del software. modelo conceptual primero para entender todas las partes

Por otro lado, una conclusión importante a la que se llegó tiene que ver con la idea de saber si se esta subespecificando o sobreespecificando. Si bien en el trabajo se presenta el resultado final, el proceso de elegir que modelos utilizar para describir cada parte no fue menor. En todos los momentos de este proceso surgió la duda de si estan quedando cosas sin explicar o si las explicaciones son redundantes. Para dar un ejemplo concreto se citará un aspectos ligados a estos problemas.

Un ejemplo asociado a estos problemas podría ser el subobjetivo presente en la rama central del objetivo Mantener[Nivel Adecuado de Stock en cada local]. En este subobjetivo podemos ver que hay varias acciones ligadas al software. Esto trajo la duda si esta parte debería ser modelada con una máquina de estados.

Por un lado, si se realizaba esta máquina, se podría pensar que hay demasiada redundancia en cuanto a la explicación de manejo de stock, ya que se encontraría abordado por los cuatro modelos, y además seguramente esta máquina no aportaría mucho más que otro modelo como el diagrama de actividad.

Por otro lado, si no se realizaba esta máquina, se podría pensar que lo que sucede dentro del software para, por ejemplo, hacer las cuentas sobre los ingredientes podría quedar subespecificado.

En este caso en particular, se decidió que la máquina sería redundante ya que la cuenta sobre los ingredientes es bastante trivial y la acción importante es la relación con la impresora, acción que ya se encuentra explicada en los casos de uso y en el diagrama de actividad.

Este caso particular, ejemplifica como esto va sucediendo a medida que se toman las decisiones, que usualmente fueron decididas en base al criterio de los ingredientes sobre cuanto se quiere ahondar en cada cuestión.

Por último, al igual que en el primer trabajo práctico, se concluye en que siempre sería una

herramienta muy buena poder contrastar las ideas grupales y los diagramas con algún experto de dominio con el fin de limar ambigüedades y posibles inconsistencias.