



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Ingeniería de Software I

Trabajo práctico II

Ingeniería de Requerimientos - Grupo 2

Resumen

En el presente trabajo se utilizan las técnicas de ingeniería de requerimientos para llevar a cabo un proyecto

Integrante	LU	Correo electrónico
Acosta, Javier Sebastian	338/11	acostajavier.ajs@gmail.com
Gomez, Fernando Nahuel	695/11	fernando.gmz12@gmail.com
Jabalera Gasperi, Fernando	56/09	fgasperijabalera@gmail.com
Russo, Christian Sebastián	679/10	christian.russo@gmail.com
Vuotto, Lucas Gabriel	385/12	lvuotto@dc.uba.ar

Palabras claves:

Requerimientos, Diagrama de Casos de Uso, Diagrama de Clases, OCL, Diagrama de Actividad, FSM

Índice

1. Introducción	3
2. Presunciones	5
3. Vistas	6
3.1. Modelo de operaciones	6
3.1.1. Diagrama de casos de uso	6
3.1.2. Detalle de los casos de uso	8
3.1.3. Trazabilidad	21
3.2. Modelo conceptual	23
3.2.1. Diagrama de clases	23
3.2.2. Restricciones OCL	25
3.2.3. Generación de los reportes de ventas	27
3.2.4. Trazabilidad	27
3.3. Finite State Machine - FSM	29
3.3.1. Diagramas de FSM	29
3.3.2. Trazabilidad	32
3.4. Modelo de comportamiento	33
3.4.1. Diagrama de actividad	33
3.4.2. Trazabilidad	36
3.5. Trazabilidad	37
4. Discusión	39
4.1. Modificación de los pedidos	39
4.2. Rechazo de los pedidos	40
4.3. Pedidos no entregados	40
4.4. Fechas de entrega de los pedidos	40
5. Conclusiones	41

1. Introducción

En este segundo trabajo, el objetivo es brindar otras vistas sobre lo realizado en el trabajo anterior con respecto a la cadena Mes %. Para ello utilizamos las siguientes técnicas:

- **Diagrama de casos de uso:** Este diagrama permite mostrar las interacciones de los actores que participan en alguna operación del sistema. También detallar quién realiza cada operación y observar los estereotipos “incluye” y “extiende” entre casos de uso.
- **Diagrama de clases:** Este diagrama permite observar los conceptos más importantes en un sistema, que los llamaremos clases. También permite observar los atributos de las clases, relaciones de herencia, clases de asociación y relaciones entre clases.
- **Diagrama de estados:** Este diagrama permite observar el comportamiento de los distintos componentes. Solo se modela los componentes que necesitan sincronizarse con otros.
- **Diagramas de actividad:** En este diagrama se pueden observar las operaciones necesarias para realizar alguna tarea en el sistema. Estas tienen asociadas un orden y pueden realizarse en forma paralela. También se pueden observar andariveles, que son los responsables de cumplir dicha operación.

Para realizar todo este análisis, partimos de uno de los diagramas de contexto (*el número 3*) presentados en el trabajo anterior. Dicho diagrama sufrió algunas modificaciones a medida que fuimos realizando este nuevo trabajo ya que notamos ciertas falencias en el mismo. A continuación presentamos el diagrama de contexto elegido, señalando en color rojo las modificaciones realizadas:



2. Presunciones

Primero listaremos las presunciones de dominio que realizamos en la primer instancia del trabajo:

1. El proveedor tiene stock infinito.
2. Los envíos a los locales siempre se realizan en horarios en los que hay empleados para recibirlos.
3. La información relevante que los dueños del sistema quieren se basa en estadísticas obtenidas a partir de las ventas online.
4. Los clientes, al registrarse, proporcionan un nombre de usuario, una contraseña, información de contacto y una foto con su DNI para confirmar su identidad.
5. Los locales también pueden modificar los pedidos que realizan en la página.
6. Todos los locales cuentan con depósitos propios

Estas presunciones nos resultaron suficientes, de modo que no hemos realizado ninguna presunción extra en esta nueva instancia del trabajo.

3. Vistas

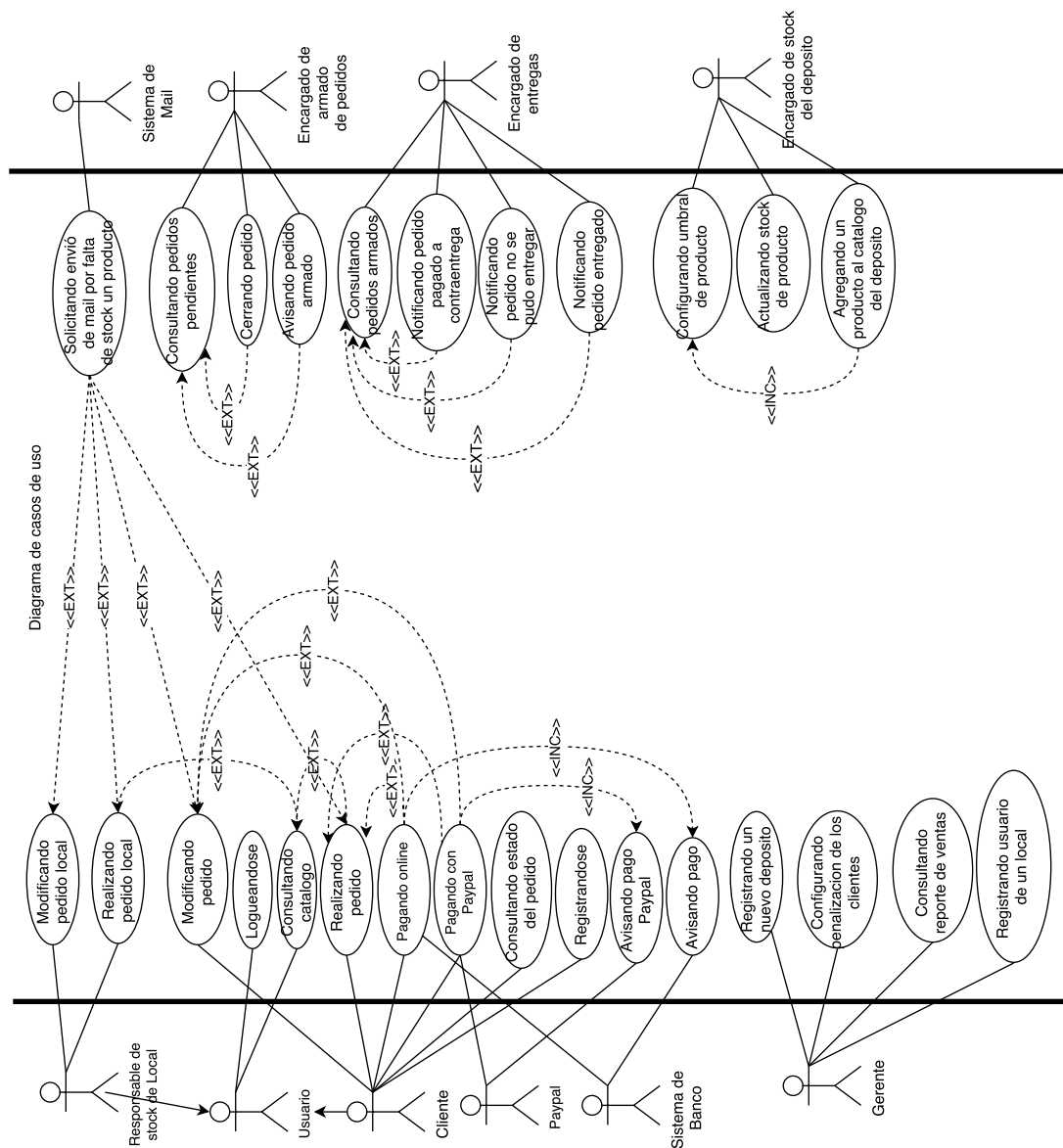
3.1. Modelo de operaciones

3.1.1. Diagrama de casos de uso

En este primer modelo mostraremos todos los escenarios en los que algún agente (encargados, clientes, locales, gerente, etc) participa en alguna operación del sistema. A estos agentes los llamaremos actores y los representamos con la figura de una persona. También mostraremos las relaciones de extensión e inclusión entre las operaciones, en caso de que existan. Luego, detallaremos por completo los pasos que conlleva realizar cada una de las operaciones, es decir, como el actor (o actores) que participan en dicha operación interactúan con nuestro sistema. También se visualizan relaciones de herencia entre actores, donde los casos de uso del actor padre también pertenecen al actor hijo.

Si bien esta técnica detalla los pasos de una operación y muestra los actores que participan en cada una, no permite visualizar un orden a seguir entre operaciones ni tampoco modelar aspectos puros del sistema (comportamiento automático en donde no hay actores). El orden entre operaciones lo mostraremos más adelante, cuando utilicemos diagramas de actividad.

A continuación mostraremos el diagrama de casos de uso resultante y el detalle de las operaciones de cada uno de los casos:



3.1.2. Detalle de los casos de uso

CASO DE USO	Registrándose
ACTORES	Cliente
PRE	-
POST	El cliente se registra en el sitio

Curso Normal	Curso alternativo
1. El cliente ingresa sus datos personales (nombre, apellido, DNI, domicilio y numero teléfono). 2. El cliente ingresa un nombre de usuario. 3. El cliente ingresa una contraseña. 4. El cliente sube al sitio una copia de su DNI. 5. El sistema valida que los datos ingresados no pertenezcan a un usuario existente. 6. El sistema guarda los datos del usuario. 7. FIN CU.	5.1 Si los datos ya existen, volver al paso 1.

CASO DE USO	Logueandose
ACTORES	Usuario
PRE	El usuario debe estar registrado.
POST	El usuario ingresa al sitio con su cuenta.

Curso Normal	Curso alternativo
1. El usuario ingresa el nombre de su cuenta y su contraseña. 2. El sistema valida los datos. 3. FIN CU.	2.1. Si los datos son incorrectos, volver al paso 1.

CASO DE USO	Consultando catálogo.
ACTORES	Usuario
PRE	El usuario está logueado.
POST	El usuario visualiza el catálogo de todos los productos disponibles.

Curso Normal	Curso alternativo
1. El sistema muestra al usuario todos los productos disponibles en alguno de los depósitos del sistema y el stock disponible de cada uno en dicho depósito. 2. Si el usuario quiere realizar un pedido de productos. EXT CU: "Realizando Pedido". 3. FIN CU.	

CASO DE USO	Realizando pedido local.
ACTORES	Responsable de stock del local
PRE	El responsable de stock del local debe estar logueado y debe haber consultado el catálogo de productos previamente.
POST	El responsable de stock del local realizó un pedido con éxito.

Curso Normal	Curso alternativo
<ol style="list-style-type: none"> 1. El sistema verifica que el responsable de stock del local no tenga un pedido pendiente de entrega. 2. El responsable de stock del local selecciona el producto que desea adquirir. 3. El responsable de stock ingresa la cantidad de unidades que desea del producto. 4. El responsable de stock vuelve al paso 2 por cada producto que desea adquirir. 5. El responsable de stock confirma el pedido. 6. El sistema chequea que en el depósito haya suficiente stock disponible para cada producto solicitado en el pedido. 7. El sistema propone al responsable de stock tres posibles fechas al azar, en días hábiles, dentro de los próximos siete días para que le sea entregado su pedido. 8. El responsable de stock selecciona la fecha que desea en que le entreguen el pedido. 9. El sistema almacena los datos del pedido. 10. El sistema reserva en el depósito, por cada producto pedido, la cantidad solicitada de dicho producto. 11. Si el pedido causo que el stock de un producto sea menor que el umbral configurado para dicho producto, EXT CU: "Solicitando envió de mail por falta de stock un producto". 12. El sistema vuelve al paso 11 por cada producto que haya caído debajo de su umbral por causa de los productos pedidos. 13. FIN CU. 	<ol style="list-style-type: none"> 1.1. Si el responsable de stock del local tiene un pedido pendiente, rechazar pedido. FIN CU. 6.1. Si el stock disponible no es suficiente para alguno de los productos, mostrar al usuario un mensaje de error diciendo "Stock insuficiente". FIN CU.

Es importante notar que el sistema solo arregla una fecha de entrega con el encargado de stock del local y no un horario. Es el encargado de entregas quien se encarga de entregar los pedidos a los locales en el horario en que estos operan.

CASO DE USO	Realizando Pedido.
ACTORES	Cliente
PRE	El Cliente está logueado y consultó el catálogo de productos.
POST	El cliente realizó un pedido con éxito.

Curso Normal	Curso alternativo
<p>1. El sistema verifica que el cliente no esté penalizado.</p> <p>2. El sistema verifica que el cliente no tenga un pedido pendiente de entrega.</p> <p>3. El Cliente selecciona el producto que desea adquirir.</p> <p>4. El Cliente ingresa la cantidad de unidades que desea del producto, la cual debe estar entre 1 y el stock disponible en el depósito en ese momento.</p> <p>5. El Cliente vuelve al paso 3 por cada producto que desea adquirir.</p> <p>6. El sitio calcula el importe total del pedido.</p> <p>7. El Cliente confirma el pedido.</p> <p>8. El sistema chequea que en el depósito haya suficiente stock disponible para cada producto solicitado en el pedido.</p> <p>9. El sistema propone al cliente tres posibles fechas al azar, en días hábiles, dentro de los próximos siete días para que le sea entregado su pedido.</p> <p>10. El Cliente selecciona la fecha que desea en que le entreguen el pedido.</p> <p>11. El Cliente selecciona el método de pago que desea entre Paypal, pagar con tarjeta de crédito o débito, o pagar a contraentrega.</p> <p>12. Si el Cliente desea pagar con tarjeta de crédito o débito. EXT CU: "Pagando Online".</p> <p>13. Si el Cliente desea pagar con PayPal. EXT CU: "Pagando con PayPal".</p> <p>14. El sistema almacena los datos del pedido.</p> <p>15. El sistema reserva en el depósito, por cada producto pedido, la cantidad solicitada de dicho producto.</p>	<p>1.1 Si el Cliente está penalizado, rechazar pedido. FIN CU.</p> <p>2.2 Si el Cliente tiene un pedido pendiente, rechazar pedido. FIN CU.</p> <p>8.1. Si el stock disponible no es suficiente para alguno de los productos, mostrar al cliente un mensaje de error diciendo "Stock insuficiente". FIN CU.</p>

16. Si el pedido causo que el stock de un producto sea menor que el umbral configurado para dicho producto, EXT CU: “Solicitando envío de mail por falta de stock un producto”.	
17. El sistema vuelve al paso 16 por cada producto que haya caído debajo de su umbral por causa de los productos pedidos.	
18. FIN CU.	

Es importante notar que la cantidad de unidades de un producto que el cliente puede pedir es igual a la cantidad de unidades disponibles en el deposito en el momento que el cliente **inicia** el pedido. Esta cantidad pudo haber disminuido (*por un pedido de otro usuario*) e impedir que el cliente concrete su pedido. En este caso, el cliente deberá iniciar la operación nuevamente.

Se verifica si el cliente está penalizado comprobando que haya una instancia de “*Penalización*” asociada a la cuenta cuyo intervalo de fechas contenga a la fecha actual. Se puede ver en más detalle el flujo de penalización en el modelo de máquinas de estado.

También existe un diagrama de actividad que muestra el orden completo del flujo de la realización de un pedido (*por parte de un cliente*), desde que se consulta el catálogo hasta que el pedido es entregado.

CASO DE USO	Pagando Online.
ACTORES	Cliente, Sistema de Banco
PRE	El cliente está logueado y debe estar realizando un pedido.
POST	El cliente realiza el pago del pedido vía online.

Curso Normal	Curso alternativo
1. El cliente completa los datos de la tarjeta (tipo, nombre del titular, número, código de seguridad, fecha de vencimiento)	1.1 El usuario dejó algún campo en blanco. Se vuelve al paso 1.
3. El sistema de banco valida los datos de la tarjeta del cliente.	3.1. Si el sistema de banco informa que los datos de la tarjeta no son válidos, volver al paso 1.
4. El sistema envía al sistema de banco los datos de la tarjeta del cliente y el monto a cobrarle.	
4. INC CU: “Avisando pago”.	
5. FIN CU.	

CASO DE USO	Avisando pago.
ACTORES	Sistema de Banco
PRE	El sistema envió al sistema de banco los datos de tarjeta de un cliente y un monto a cobrar.
POST	El sistema de banco confirma que el pago del monto se realizo correctamente.

Curso Normal	Curso alternativo
1. El sistema de banco chequea que el pago se pueda realizar correctamente. 2. El sistema de banco envía al sistema la nota de pago del pedido. 3. El sistema almacena la nota de pago. 4. El sistema marca el pedido como pagado. 5. FIN CU.	1.1. Si el sistema de banco informa que no se puede, el sistema muestra un mensaje de error. FIN CU

CASO DE USO	Pagando con PayPal
ACTORES	Cliente, PayPal
PRE	El cliente está logueado y debe haber realizado un pedido.
POST	El cliente realiza el pago del pedido vía PayPal.

Curso Normal	Curso alternativo
1. El cliente completa los datos de su cuenta de PayPal. 2. El sistema envía a PayPal los datos de usuario y el monto a pagar. 3. PayPal chequea si los datos son correctos. 4. INC CU: "Avisando pago PayPal". 5. FIN CU.	3.1. Si los datos de usuario son incorrectos, volver al paso 1.

CASO DE USO	Avisando pago PayPal
ACTORES	Cliente, PayPal
PRE	El sistema envió a Paypal los datos de usuario de un cliente y el monto a cobrar.
POST	Paypal confirma que el pago del monto se realizo correctamente.

Curso Normal	Curso alternativo
1. Paypal chequea que el pago se pueda realizar correctamente. 2. Paypal envía al sistema la nota de pago del pedido. 3. El sistema almacena la nota de pago. 4. El sistema marca el pedido como pagado. 5. FIN CU.	1.1. Si el Paypal informa que no se puede, el sistema muestra un mensaje de error. FIN CU

CASO DE USO	Consultando estado del pedido.
ACTORES	Usuario
PRE	El usuario está logueado y debe haber realizado un pedido.
POST	El usuario visualiza el estado actual del pedido realizado.

Curso Normal	Curso alternativo
1. El sitio muestra al usuario si su pedido está pagado o no. 2. Si el cliente realizó un pedido y este aún no fue cerrado, muestra el estado "Pendiente". Si el pedido se encuentra cerrado, muestra el estado "Cerrado". Si el pedido ya fue armado, muestra el estado "Armado". Si el pedido fue entregado, muestra el estado "Entregado". Si el pedido no fue entregado, muestra el estado "No entregado" 3. FIN CU.	

CASO DE USO	Modificando pedido.
ACTORES	Cliente
PRE	El cliente está logueado, debe haber realizado un pedido y que su único pedido esté sin cerrar.
POST	El cliente logra modificar el pedido sin cerrar.

Curso Normal	Curso alternativo
1. El Cliente selecciona el producto que desea adquirir. 2. El Cliente ingresa la cantidad de unidades que desea del producto, la cual debe estar entre 1 y el stock disponible en el depósito en ese momento. Si este producto ya estaba en el pedido, la cantidad solicitada se suma a la cantidad pedida previamente. 3. El Cliente vuelve al paso 1 por cada producto que desea adquirir. 4. El sitio calcula el importe de los nuevos productos agregados al pedido y de las unidades adicionales pedidas de productos que ya estaban en el pedido. 5. El Cliente confirma el pedido. 6. El sistema chequea que en el depósito haya suficiente stock disponible para cada nuevo producto solicitado en el pedido y para las unidades extra de productos previamente solicitados. 7. Si cuando el Cliente realizo el pedido original eligió pagar con tarjeta de crédito o débito. EXT CU: "Pagando Online".	6.1. Si el stock disponible no es suficiente para alguno de los productos, mostrar al usuario un mensaje de error diciendo "Stock insuficiente", reiniciar la lista de productos solicitados al estado anterior del pedido y volver al paso 1.

<p>8. Si cuando el Cliente realizo el pedido original eligió pagar con PayPal. EXT CU: “Pagando con PayPal”.</p> <p>9. El sistema almacena los nuevos datos del pedido.</p> <p>10. El sistema reserva en el depósito, por cada producto pedido, la cantidad solicitada de dicho producto (si es un nuevo producto en el pedido) o la diferencia entre la cantidad anterior y la nueva cantidad solicitada (si es un producto que ya estaba en el pedido original).</p> <p>11. Si el pedido causo que el stock de un producto sea menor que el umbral configurado para dicho producto, EXT CU: “Solicitando envío de mail por falta de stock un producto”.</p> <p>12. El sistema vuelve al paso 11 por cada producto que haya caído debajo de su umbral por causa de los nuevos productos pedidos.</p> <p>13. FIN CU.</p>	
--	--

Es importante notar que solo permitimos que los clientes agreguen nuevos productos y/o que aumenten las unidades de los productos que ya pidieron, pero **no** permitimos que quiten cosas de su pedido original. Esto es debido a que estos productos ya pudieron haber sido pagados por el cliente (*si es que eligió pagar online con tarjeta o con Paypal*), y hemos decidido no hacer reembolsos.

CASO DE USO	Modificando pedido local.
ACTORES	Responsable de stock del local.
PRE	El responsable de stock del local debe haber realizado un pedido y que su único pedido esté sin cerrar.
POST	El responsable de stock del local logra modificar el pedido sin cerrar.

Curso Normal	Curso alternativo
<p>1. El responsable de stock del local selecciona el producto que desea adquirir.</p> <p>2. El responsable de stock del local ingresa la cantidad de unidades que desea del producto, la cual debe estar entre 1 y el stock disponible en el depósito en ese momento. Si este producto ya estaba en el pedido, la cantidad solicitada se suma a la cantidad pedida previamente.</p> <p>3. El responsable de stock del local vuelve al paso 1 por cada producto que desea adquirir.</p> <p>4. El responsable de stock del local confirma el pedido.</p> <p>5. El sistema chequea que en el depósito haya suficiente stock disponible para cada nuevo producto solicitado en el pedido y para las unidades extra de productos previamente solicitados.</p> <p>6. El sistema almacena los nuevos datos del pedido.</p> <p>7. El sistema reserva en el depósito, por cada producto pedido, la cantidad solicitada de dicho producto (si es un nuevo producto en el pedido) o la diferencia entre la cantidad anterior y la nueva cantidad solicitada (si es un producto que ya estaba en el pedido original).</p> <p>8. Si el pedido causo que el stock de un producto sea menor que el umbral configurado para dicho producto, EXT CU: "Solicitando envío de mail por falta de stock un producto".</p> <p>9. El sistema vuelve al paso 8 por cada producto que haya caído debajo de su umbral por causa de los nuevos productos pedidos.</p> <p>10. FIN CU.</p>	<p>5.1. Si el stock disponible no es suficiente para alguno de los productos, mostrar al usuario un mensaje de error diciendo "Stock insuficiente", reiniciar la lista de productos solicitados al estado anterior del pedido y volver al paso 1.</p>

A los locales tampoco les permitimos quitar productos de sus pedidos. El porque de esta decisión se explica en detalle en la sección **Discusión**.

CASO DE USO	Registrando un nuevo depósito.
ACTORES	Gerente
PRE	-
POST	El gerente agrega un nuevo depósito correctamente al sistema.

Curso Normal	Curso alternativo
1. El gerente ingresa domicilio y teléfono del nuevo depósito. 2. El sistema valida que el domicilio y el teléfono ingresados no estén asignados a otro deposito y que sean validos. 3. El sistema guarda la información del nuevo deposito. 4. FIN CU.	2.1. Si los datos ingresados ya están en uso o no son validos, volver al paso 1.

CASO DE USO	Configurando penalización de los clientes.
ACTORES	Gerente
PRE	-
POST	El gerente configura el sistema de penalización para los clientes.

Curso Normal	Curso alternativo
1. El gerente ingresa el tiempo que dura la penalización de los clientes. 2. El sistema valida que el tiempo ingresado sea un numero entero mayor a uno. 3. El sistema guarda los datos ingresados. 4. FIN CU.	2.1. Si el valor ingresado no es valido, volver al paso 1.

CASO DE USO	Consultando reporte de ventas.
ACTORES	Gerente
PRE	-
POST	El gerente visualiza los reportes de ventas.

Curso Normal	Curso alternativo
<ol style="list-style-type: none"> 1. El gerente selecciona el tipo de reporte que desea ver. 2. Si selecciono el tipo "Productos mas vendidos", el sistema deberá generar y mostrar un listado de todos los productos del catalogo ordenados en forma decreciente según cuantas unidades de cada producto fueron pedidas en base a todos los pedidos realizados al sistema. 3. Si selecciono el tipo "Depósitos con mas pedidos" el sistema deberá generar y mostrar un listado de todos los depósitos registrados en el sistema ordenado en forma decreciente según cuantos pedidos se haya realizado en cada uno de ellos. 4. Si selecciono el tipo "Métodos de pago" el sistema deberá generar y mostrar una tabla indicando, para cada uno de los métodos de pago ofrecidos (Tarjeta online, Paypal o contraentrega), cuantos pedidos utilizaron dicho método de pago. 5. Si el gerente desea ver otro reporte de ventas, volver al paso 1. 6. FIN CU. 	

CASO DE USO	Registrando usuario de un local.
ACTORES	Gerente
PRE	-
POST	El gerente logra registrar un local correctamente.

Curso Normal	Curso alternativo
<ol style="list-style-type: none"> 1. El gerente ingresa un nombre de usuario 2. El gerente ingresa una contraseña. 3. El gerente ingresa los datos de local (dirección y teléfono). 4. El sistema valida los datos ingresados. 5. El sistema guarda los datos ingresados. 6. FIN CU. 	<ol style="list-style-type: none"> 4.1. Si el nombre de usuario ingresado ya esta en uso, volver al paso 1.

CASO DE USO	Consultando pedidos pendientes.
ACTORES	Encargado de armado de pedidos
PRE	-
POST	El encargado de armar pedidos logra visualizar los pedidos pendientes.

Curso Normal	Curso alternativo
1. El sistema lista los pedidos pendientes al Encargado de armado de pedidos. 2. Si el Encargado de armado de pedidos quiere cerrar un pedido. EXT CU: "Cerrando pedido". 3. Si el Encargado de armado de pedidos quiere notificar que un pedido ya fue armado. EXT CU: "Avisando pedido armado". 4. FIN CU.	

CASO DE USO	Cerrando pedido.
ACTORES	Encargado de armado de pedidos
PRE	El Encargado de armado de pedidos seleccionó un pedido pendiente.
POST	Se marca un pedido como cerrado.

Curso Normal	Curso alternativo
1. El encargado de armado de pedidos marca al pedido como cerrado. 2. El sistema guarda el nuevo estado del pedido. 3. FIN CU.	

CASO DE USO	Avisando pedido armado.
ACTORES	Encargado de armado de pedidos
PRE	El Encargado de armado de pedidos seleccionó un pedido pendiente.
POST	Se marca un pedido como armado.

Curso Normal	Curso alternativo
1. El encargado de armado de pedidos marca al pedido como armado. 2. El sistema guarda el estado del pedido. 3. FIN CU.	

CASO DE USO	Consultando pedidos armados.
ACTORES	Encargado de entregas
PRE	-
POST	Se listan los pedidos que fueron armados.

Curso Normal	Curso alternativo
1. El sistema muestra al Encargado de entregas el listado de pedidos armados. 2. Si el Encargado de entregas quiere notificar que un pedido fue pagado a contraentrega EXT CU "Notificando pedido pagado a contraentrega". 3. Si el Encargado de entregas quiere notificar que un pedido fue entregado EXT CU "Notificando pedido entregado". 4. Si el Encargado de entregas quiere notificar que un pedido no se pudo entregar EXT CU "Notificando pedido no se pudo entregar". 5. FIN CU.	

CASO DE USO	Notificando pedido entregado.
ACTORES	Encargado de entregas
PRE	Se seleccionó un pedido armado.
POST	Se notifica en el sistema que el pedido se entregó.

Curso Normal	Curso alternativo
1. El encargado de entregas marca el pedido como entregado. 2. El sistema guarda el nuevo estado del pedido y la fecha actual. 3. FIN CU.	

CASO DE USO	Notificando pedido pagado a contraentrega.
ACTORES	Encargado de entregas
PRE	Se seleccionó un pedido armado.
POST	Se notifica en el sistema que el pedido se pagó a contraentrega.

Curso Normal	Curso alternativo
1. El Encargado de entregas marca el pedido como pagado a contraentrega. 2. El sistema guarda el nuevo estado del pedido y la fecha actual. 3. FIN CU.	

CASO DE USO	Notificando pedido no se pudo entregar.
ACTORES	Encargado de entregas
PRE	Se seleccionó un pedido armado.
POST	Se notifica en el sistema que el pedido no se pudo entregar.

Curso Normal	Curso alternativo
1. El Encargado de entregas marca un pedido como no entregado. 2. El Encargado de entregas ingresa el motivo de no haber podido entregar el pedido. 3. El sistema guarda el motivo, el nuevo estado del pedido y la fecha actual. 4. FIN CU.	

CASO DE USO	Configurando umbral de producto.
ACTORES	Encargado de stock del depósito
PRE	-
POST	Se configura umbral para el depósito.

Curso Normal	Curso alternativo
1. El Encargado de stock del depósito ingresa un código de producto. 2. El Encargado de stock del depósito ingresa un umbral para el producto. 3. El sistema guarda los datos ingresados. 4. FIN CU.	1.1 Si el código es inválido volver a paso 1.

CASO DE USO	Actualizando stock de producto.
ACTORES	Encargado de stock del depósito
PRE	-
POST	El stock es actualizado.

Curso Normal	Curso alternativo
1. El Encargado de stock del depósito ingresa el código de un producto y el nuevo stock 2. El sistema suma al stock previo del producto el valor ingresado. 3. FIN CU.	1.1. Si el código ingresado no corresponde a un producto en el catalogo del deposito, mostrar un mensaje de error. Fin CU.

CASO DE USO	Agregando un producto al catalogo del depósito.
ACTORES	Encargado de stock del depósito
PRE	-
POST	El stock del producto se actualiza al valor ingresado.

Curso Normal	Curso alternativo
1. El Encargado de stock del depósito ingresa el código de un producto y el nuevo stock 2. El sistema almacena el nuevo stock, sobrescribiendo el valor si ya existía. 3. Incluye CU: Configurando Umbral de Producto 4. FIN CU.	

CASO DE USO	Solicitando envío de mail por falta de stock de un producto.
ACTORES	Sistema de mail
PRE	Se activo la alarma de falta de stock de un producto.
POST	El sistema de mail envía un aviso al encargado de stock del deposito.

Curso Normal	Curso alternativo
1. El sistema le indica al sistema de mail que envíe un mail al encargado de stock del depósito para informarlo de cual producto falta stock y cuantas unidades quedan. 2. El sistema de mail envía el mail solicitado. 3. FIN CU.	

3.1.3. Trazabilidad

A continuación listamos los requerimientos que se modelan con este diagrama.

- **Agregar depósitos al sistema:** Modelado con el CU: Registrando nuevo deposito
- **Los usuarios se pueden autentificar:** Modelado con el CU: Logueandose
- **Que el cliente envíe una copia del DNI mediante el sitio:** Modelado con el CU: Registrándose
- **Cerrar un pedido desde la pagina web:** Modelado con el CU: Cerrando Pedido
- **Tomar productos del pedido =>Notificar pedido armado:** Modelado con el CU: Avisando Pedido Armado
- **Configurar umbrales por producto en el sistema:** Modelado con el CU: Configurando umbral del producto.
- **Incrementar stock cuando recibo reposición del proveedor:** Modelado con el CU: Actualizando stock de producto
- **Coordinar fecha de entrega:** Modelado con el CU: Realizando Pedidos.
- **Se puede consultar los registros de las ventas:** Modelado con el CU: Consultando Reportes de ventas
- **Se puede configurar los parámetros del reporte:** Modelado con el CU: Consultando Reportes de ventas

- **Los locales puedan registrarse:** Modelado con el CU: Registrando usuario de un local
- **Los locales pueden y ver y seleccionar los productos que desea recibir:** Modelado con el CU: Consultando catalogo y Realizando Pedido
- **Los locales pueden modificar los pedidos no cerrados:** Modelado con el CU: Modificando Pedido
- **Los clientes pueden y ver y seleccionar los productos que desean recibir:** Modelado con el CU: Consultando Catalogo
- **Los clientes pueden modificar los pedido no cerrados:** Modelado con el CU: Modificando Pedido
- **Un cliente pueda registrarse:** Modelado con el CU: Registrándose
- **Un cliente pueda seleccionar la fecha de entrega:** Modelado con el CU: Realizando pedido
- **Un cliente pueda ingresar sus datos de contacto:** Modelado con el CU: Registrándose
- **Reservar stock:** Modelado con el CU: Realizar Pedido
- **Los productos en falta son filtrado del listado que ve el cliente:** Modelado con el CU: Consultar Catalogo
- **La cantidad máxima permitida para pedir un producto es igual a la que hay en el deposito:** Modelado con el CU: Realizar Pedido
- **Consultar pedidos armados:** Modelado con el CU: Consultar Pedidos Armados
- **Seleccionar pedido entregado:** Modelado con el CU: Notificar Pedido Entregado
- **El cliente pago en contra-entrega =>se notifica pago a contra-entrega:** Modelado con el CU: Notifica pedido pagado a contra-entrega
- **El cliente pago desde la pagina =>se notifica el pago online:** Modelado con el CU: Avisando Pago
- **Coordinar fecha de entrega con proveedor:** Modelado con el CU: Actualizar Stock de un producto
- **Llevar el conteo en base a los pedidos realizados de cuantos productos quedan:** Modelado con el CU: Realizar Pedido
- **Cliente pague el pedido:** Modelado con el CU: Pagando online y Pagando con PayPal
- **Ofrecer pago con PayPal:** Modelado con el CU: Pagando con PayPal
- **Ofrecer pago con crédito:** Modelado con el CU: Pagando online
- **Ofrecer pago con débito:** Modelado con el CU: Pagando online
- **Ofrecer pago con efectivo:** Modelado con el CU: Notificando pedido pagado a contra-entrega

3.2. Modelo conceptual

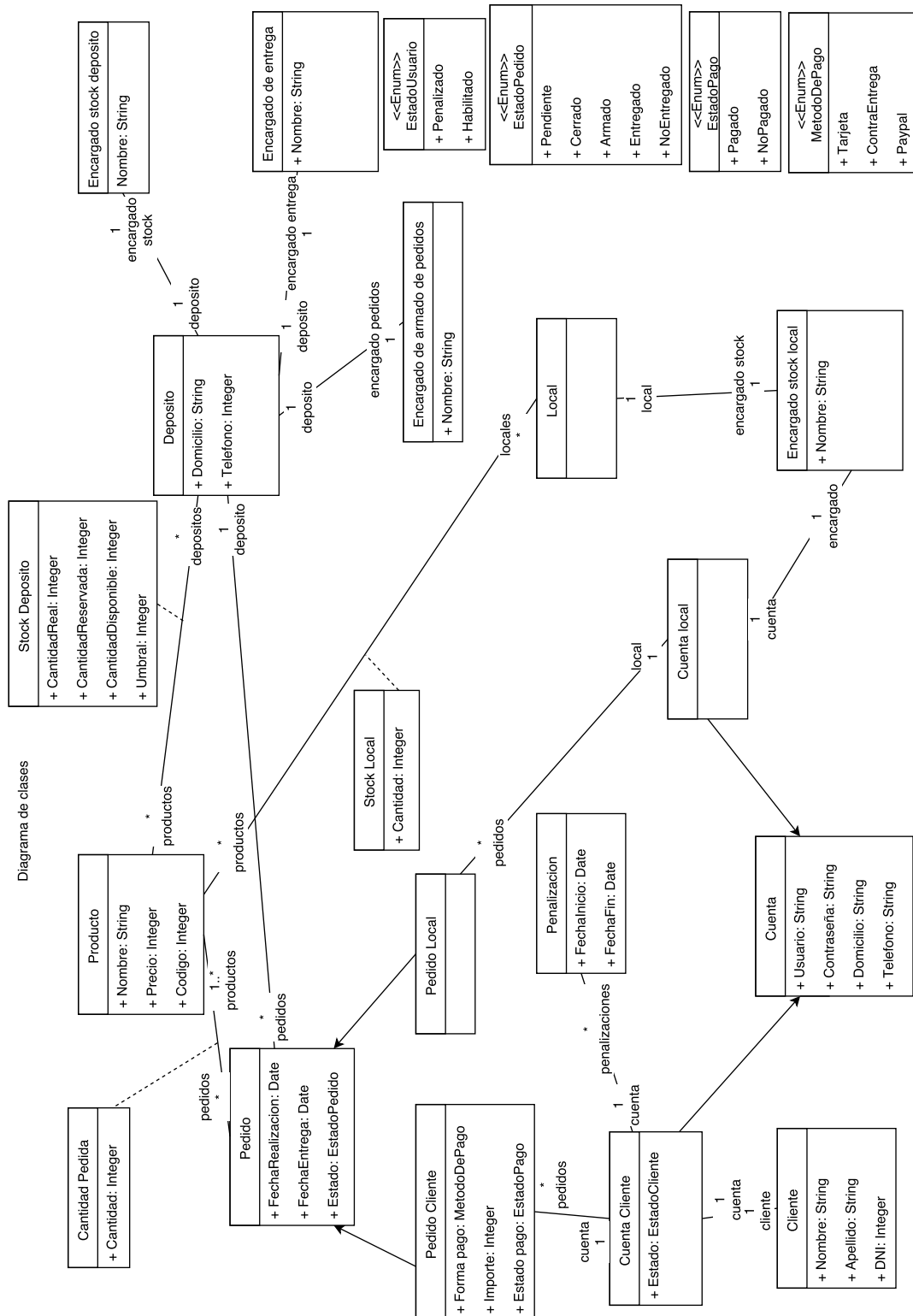
3.2.1. Diagrama de clases

En este modelo se podrá visualizar los conceptos más importantes del sistema. A estos conceptos los llamaremos clases, las cuales tienen atributos que caracterizan a dicho concepto. También mostraremos las relaciones entre estas clases, el tipo de relación (uno a uno, uno a muchos, muchos a muchos, etc), el rol de cada clase en la relación, clases de asociación en la relación y los casos de herencia entre clases. Luego, usaremos OCL para realizar restricciones de instanciación de las clases.

En este diagrama mostramos la información que almacena el sistema:

- Historial de pedidos realizados por un cliente.
- Historial de pedidos realizados por un local.
- Información de cada pedido: estado, fecha de entrega, método de pago, productos pedidos y su cantidad, depósito que atendió el pedido, etc.
- Configuración de los umbrales de cada depósito (por producto).
- Stock reservado y disponible de cada depósito (por producto).
- Historial de penalización de cada usuario.
- Datos personales de cada usuario.
- Datos de cada depósito.

A continuación mostraremos el diagrama de clases y las restricciones OCL realizadas:



3.2.2. Restricciones OCL

1. Los clientes solo pueden tener un pedido activo a la vez.

CONTEXTO	Cuenta Cliente
INVARIANTE	<code>self.pedidos->select(p p.Estado = Pendiente or p.Estado = Cerrado or p.Estado = Armado)->size() <= 1</code>

2. Los locales solo pueden tener un pedido de reposición de stock activo a la vez.

CONTEXTO	Cuenta local
INVARIANTE	<code>self.pedidos->select(p p.Estado = Pendiente or p.Estado = Cerrado or p.Estado = Armado)->size() <= 1</code>

3. Calculo de la cantidad disponible de un producto en el stock de un deposito.

CONTEXTO	Stock Depósito
INVARIANTE	<code>self.CantidadDisponible = self.CantidadReal - self.CantidadReservada</code>

4. La cantidad reservada de un producto en un deposito es igual a la suma de la cantidad pedida de ese producto en los pedidos activos realizados a dicho deposito.

CONTEXTO	Depósito
INVARIANTE	<code>self.Stock Deposito->forAll(sd sd.CantidadReservada = self.pedidos->select(pe pe.productos->exists(sd.Producto) and (pe.Estado = Pendiente or pe.Estado = Cerrado or pe.Estado = Armado))->collect((pe, sd.Producto).Cantidad)->sum()</code>

5. La fecha de inicio de una penalización es anterior a su fecha de finalización.

CONTEXTO	Penalización
INVARIANTE	<code>self.FechaInicio < self.FechaFin</code>

6. No puede haber pedidos por parte de un cliente con fecha dentro del rango de alguna penalización de esa cuenta.

CONTEXTO	Cuenta cliente
INVARIANTE	<code>self.pedidos->forAll(p self.penalizaciones->forAll(pen p.Fecha <pen.FechaInicio or p.Fecha >pen.FechaFin)</code>

7. Los períodos de las penalizaciones de un cliente no pueden solaparse.

CONTEXTO	Cuenta cliente
INVARIANTE	<code>self.penalizaciones->forAll(p1, p2 p1.FechaFin <p2.FechaInicio or p1.FechaInicio >p2.FechaFin)</code>

8. Los productos de los pedidos atendidos por un depósito tiene que estar en el catálogo de dicho depósito.

CONTEXTTO	Pedido
INVARIANTE	<code>self.deposito.productos->includesAll(self.productos)</code>

9. Los pedidos de los locales no pueden estar marcados como no entregados.

CONTEXTTO	Pedido local
INVARIANTE	<code>self.Estado <> NoEntregado</code>

10. Definición de cuando un cliente esta penalizado.

CONTEXTTO	Cuenta cliente
INVARIANTE	<code>if (self.penalizaciones->exists(p p.FechaInicio <= now() <= p.FechaFin) then self.Estado = Penalizado else self.Estado = Habilitado endif)</code>

11. Calculo del importe de un pedido.

CONTEXTTO	Pedido Cliente
INVARIANTE	<code>self.Importe = self.Cantidad Pedida->collect(cp cp.Cantidad * cp.Producto.Precio)->sum()</code>

12. La fecha de entrega de un pedido debe ser igual o posterior a la fecha en la que se realizo el pedido.

CONTEXTTO	Pedido
INVARIANTE	<code>self.FechaRealizacion <= self.FechaEntrega</code>

13. Los pedidos entregados siempre son pagados.

CONTEXTTO	Pedido Cliente
INVARIANTE	<code>self.Estado == "Entregado" implies self.Estado pago == "Pagado"</code>

14. Los pedidos que se pagan online con tarjeta o por Paypal siempre se marcan como pagados.

CONTEXTTO	Pedido Cliente
INVARIANTE	<code>(self.Forma pago == "Tarjeta" or self.Forma pago == "Paypal") implies self.Estado pago == "Pagado"</code>

15. El stock reservado no puede superar al stock real del deposito.

CONTEXTTO	Stock Deposito
INVARIANTE	<code>self.CantidadReservada <= self.CantidadReal</code>

16. Los pedidos se entregan, a lo sumo, siete días después de haber sido realizados.

CONTEXTTO	Pedido
INVARIANTE	<code>self.FechaEntrega - self.FechaRealizacion <= 7</code>

3.2.3. Generación de los reportes de ventas

Para mostrar como se podría generar un reporte de ventas en este modelo conceptual presentamos tres ejemplos:

- **Productos mas vendidos:** Primero, tomamos los objetos de la clase “Cantidad Pedida” y, para cada producto, sumamos el valor del campo “Cantidad” de los objetos de la clase “Cantidad Pedida” que estén relacionados con el objeto de la clase “Producto” que representa al producto en cuestión. Luego, ordenamos esta lista de forma decreciente.
- **Depósitos con mas pedidos:** Dado un deposito podemos obtener todos los pedidos realizados a este mediante su relación con la clase “Pedido”. Luego, haciendo *size()* al conjunto obtenido por esta relación podemos obtener la cantidad de pedidos de cada deposito. Así podemos armar un lista, con la cantidad de pedidos de cada deposito y, luego, ordenarla de forma decreciente.
- **Métodos de pago:** Dado todo el conjunto de instancias de “Pedido Cliente”, filtramos por cada método de pago ofrecido a los pedidos cuyo campo “Forma de pago” se corresponda con el método de pago deseado y, luego, contamos cuantas instancias tenemos para cada uno de los métodos.

3.2.4. Trazabilidad

A continuación listamos los requerimientos que se modelan con este diagrama.

- **Agregar depósitos al sistema:** se agrega una nueva Clase Deposito
- **Cerrar un pedido desde la pagina web:** el objeto pedido cambia su estado a cerrado
- **Tomar productos del pedido =>Notificar pedido armado:** El estado del pedido cambia a armado
- **Configurar umbrales por producto en el sistema:** stockDeposito.umbral = umbral configurado
- **Incrementar stock cuando recibo reposición del proveedor:** stockDeposito.cantidadReal se le incrementa el stock que se pidió
- **Coordinar fecha de entrega:** Pedido.fechaDeEntrega = fecha que el cliente acuerda
- **Se puede consultar los registros de las ventas: informe**
- **Se puede configurar los parámetros del reporte: informe**
- **Mantener listado de productos comprados:** Pedido.productos si Pedido.estado = Pagado
- **Mantener información de los medios de pagos utilizados:**
- **Los locales puedan registrarse:** PedidoCliente.metodoDePago
- **Los locales pueden modificar los pedidos no cerrados:** si Pedido.estado != Cerrado entonces se modifican las relaciones con productos
- **Los clientes pueden modificar los pedido no cerrados:** si Pedido.estado != Cerrado entonces se modifican las relaciones con productos
- **Un cliente pueda registrarse:** se crea un nuevo objeto CuentaCliente
- **Un cliente pueda seleccionar la fecha de entrega:** Pedido.fechaDeEntrega = fecha que el cliente acuerda
- **Un cliente pueda ingresar sus datos de contacto:** atributos de clase Cuenta
- **Reservar stock:** StockDeposito.cantidadReservada igual al stock reservado

- **Consultar pedidos armados:** se muestran los pedidos tal que Pedido.estadoPedido = Armado
- **Seleccionar pedido entregado:** se modifica Pedido.estadoPedido = Entregado
- **El cliente pago en contra-entrega =>se notifica pago a contra-entrega:** se modifica Pedido.estadoPedido = Pagado
- **El cliente pago desde la pagina =>se notifica el pago online:** se modifica Pedido.estadoPedido = Pagado
- **Llevar el conteo en base a los pedidos realizados de cuantos productos quedan:** StockDeposito.cantidadDisponible
- **Mantener historial de pedidos entregados y no entregados de un cliente:** Instancias de la clase Pedido asociadas con una instancia de la clase Cuenta
- **Rechazar pedido durante un tiempo:** clase Penalizacion
- **Pedido llega a la casa del cliente:** PedidoCliente.estadoPedido = Entregado
- **Cliente pague el pedido:** PedidoCliente.estadoPedido = Pagado
- **Ofrecer pago con PayPal:** PedidoCliente.metodoDePago = PayPal
- **Ofrecer pago con credito:** PedidoCliente.metodoDePago = Tarjeta
- **Ofrecer pago con debito:** PedidoCliente.metodoDePago = Tarjeta
- **Ofrecer pago con efectivo:** PedidoCliente.metodoDePago = ContraEntrega

3.3. Finite State Machine - FSM

3.3.1. Diagramas de FSM

Con los diagramas de máquinas de estado (Finite State Machine - FSM) decidimos modelar los comportamientos del sistema que **requerían sincronización**. Estos comportamientos se representan mediante eventos y estados, y flechas indicando que evento lleva a que estado. Estos eventos pueden realizarse bajo condiciones y modificar el valor de variables predefinidas. También se utilizaron variables de tipo timer que representan a cronómetros (el tiempo de cada timer transcurrirá en el estado que haga referencia a éste).

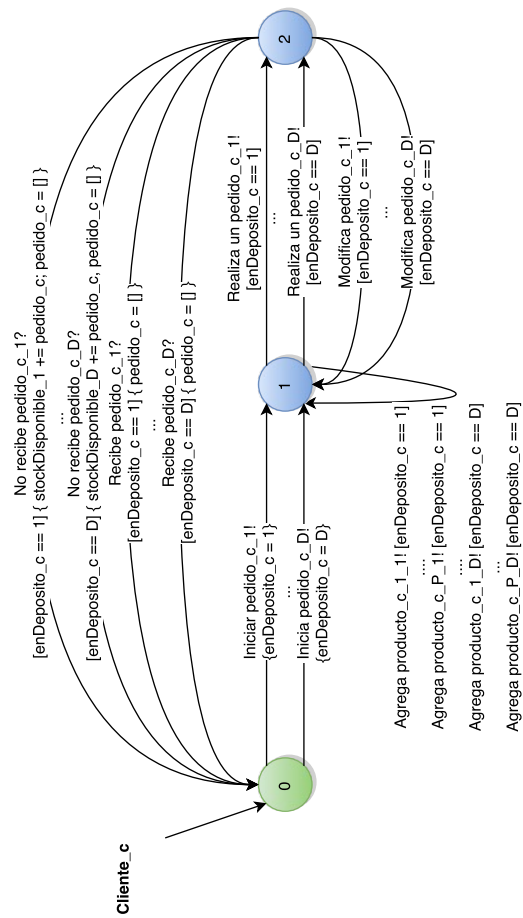
Los comportamientos modelados son:

- El procesamiento de los pedidos de los clientes (creación, modificación, restricciones en las cantidades pedidas, etc).
- La reserva del stock del depósito en respuesta al pedido de un cliente.
- El funcionamiento de las alarmas de cada producto en cada depósito.
- La penalización de los usuarios cuando no reciben un pedido.
- La actualización del stock luego de un pedido al proveedor.

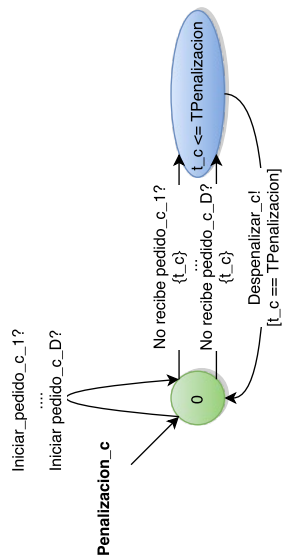
Tres cosas importantes que se deben ver en este modelo son:

- Cuando un cliente confirma su pedido (mediante “Realiza un pedido”), el sistema reserva en el depósito el stock necesario para satisfacer dicho pedido y, en caso de no poder satisfacerlo, dicho pedido no es aceptado.
- Al igual que en el caso de uso “Modificando pedido”, no se permite que un cliente quite productos o unidades de un producto que ya pidió. Solo se le permite agregar productos o unidades. Esto es así porque los productos del pedido original pudieron ya haber sido cobrados al cliente.
- Cuando un cliente está en su periodo de penalización, no es capaz de iniciar nuevos pedidos, es decir, el sistema los rechaza.

A continuación mostraremos las máquinas de estado realizadas, las cuales se sincronizan todas entre si para dar lugar al comportamiento completo del sistema:

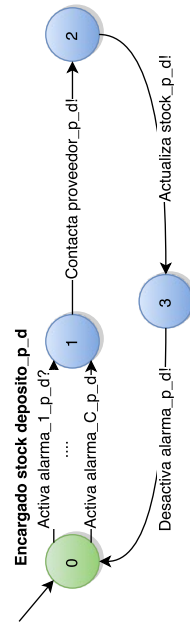
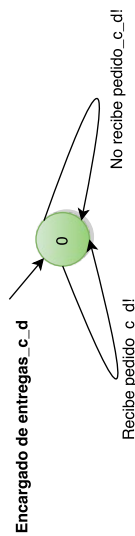


FSM para la penalización del usuario:



C = #CLIENTES
 P = #PRODUCTOS
 D = #DEPÓSITOS
 M = #MAXPROD
 c entre 1 y #CLIENTES
 p entre 1 y #PRODUCTOS
 d entre 1 y #DEPÓSITOS
 t.c: timer en días
 TPenalizacion es un entero en [0..365]

Para todo d:
 stockDisponible_d[1..P] arreglo de enteros
 stockReservado_d[1..P] arreglo de enteros
 Para todo p:
 proveedor_p_d: Es la cantidad de productos provista por el proveedor al depósito d
 umbral_p_d: umbral del producto p en el depósito d en [1, M]
 Para todo c:
 enDeposito_c es un entero en [1, D]
 pedido_c(p): cantidad del producto p que tiene el pedido del cliente c en [1, M]
 pedidoOrig_c(p): arreglo que almacena el pedido original cuando el cliente desea modificar su pedido.
 Para todo p:
 prod_p_c: productos de tipo p que agrega el cliente c en [1, M]



El comportamiento del sitio web con respecto a un deposito esta dado por la composición paralela de:

$$Deposito_d = Deposito_1_d \parallel Deposito_2_d \parallel \dots \parallel Deposito_C_d.$$

El comportamiento del sistema de penalización de usuarios esta dado por la composición paralela de:

$$Penalizacion = Penalizacion_1 \parallel Penalizacion_2 \parallel \dots \parallel Penalizacion_C$$

El comportamiento del encargado de entregas de un deposito esta dado por la composición paralela de:

$$Encargado\ de\ entregas_d = Encargado\ de\ entregas_1_d \parallel Encargado\ de\ entregas_2_d \parallel \dots \parallel Encargado\ de\ entregas_C_d$$

El comportamiento del encargado de stock de un deposito esta dado por la composición paralela de:

$$Encargado\ de\ stock\ deposito_d = Encargado\ de\ stock\ deposito_1_d \parallel Encargado\ de\ stock\ deposito_2_d \parallel \dots \parallel Encargado\ de\ stock\ deposito_P_d$$

El comportamiento completo del sistema esta dado por la composición paralela de:

$$Sistema = Penalizacion \parallel Deposito_1 \parallel \dots \parallel Deposito_D \parallel Encargado\ de\ entregas_1 \parallel \dots \parallel Encargado\ de\ entregas_D \parallel Encargado\ de\ stock\ deposito_1 \parallel \dots \parallel Encargado\ de\ stock\ deposito_D \parallel Cliente_1 \parallel \dots \parallel Cliente_C$$

3.3.2. Trazabilidad

A continuación listamos los requerimientos que se modelan con este diagrama:

- **Incrementar stock cuando recibo reposición del proveedor:** en la maquina encargado stock *deposito_d*, la acción *actualizarStock_d*
- **Los clientes pueden y ver y seleccionar los productos que desean recibir:** maquina *Cliente_c*, acción *agregaProducto_c_p_d*
- **Reservar stock:** maquina *Deposito_d*, acción *realizaPedido_c_d*
- **Los productos en falta son filtrado del listado que ve el cliente:** maquina *Deposito_d*, acción *agregaProducto_c_p_d*
- **La cantidad máxima permitida para pedir un producto es igual a la que hay en el deposito:** maquina *Deposito_d*, acción *agregaProducto_c_p_d*
- **Proveedor envía producto:** maquina *EncargadoStockDeposito_d*, acción *actualizarStock_d*
- **Llevar el conteo en base a los pedidos realizados de cuantos productos quedan:** array *stockDisponible*
- **Se disparan alarmas cuando la cantidad de productos cae debajo del umbral configurado:** maquina *Deposito_d*, acción *activarAlarma_p_d*
- **Contactar al proveedor:** *EncargadoStockDeposito_d*, acción *contactarProveedor_d*
- **Rechazar pedido durante un tiempo:** *Penalizacion_c*, segundo estado
- **Pedido llega a la casa del cliente:** *EncargadoEntregas_d*, acción *pedidoEnviado_c_d*

3.4. Modelo de comportamiento

3.4.1. Diagrama de actividad

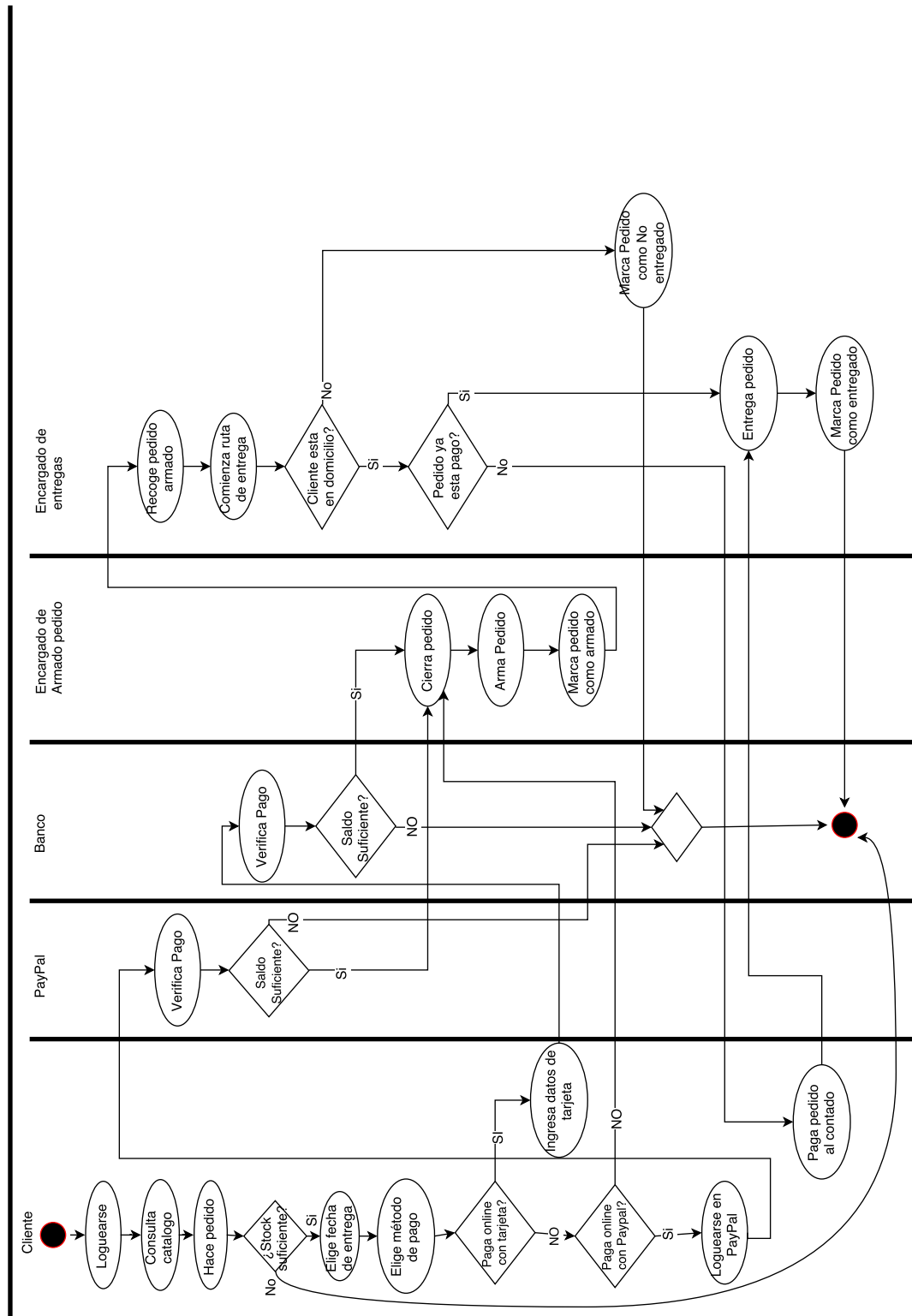
En este ultimo modelo mostraremos las actividades realizadas con su respectivo orden. Este orden lo indicaremos con flechas. También mostraremos los responsables de cada actividad. Las actividades también pueden suceder de forma concurrente. Para esto usaremos una barra (que llamaremos fork) y colocaremos las actividades concurrentes debajo de esta. Luego de esto hace falta que las actividades esperen a las que iniciaron el fork con ésta. Usaremos también una barra para esto, colocando las actividades por encima (que llamaremos join). Además se colocará condiciones if que indican a dónde se dirigirá un flujo que depende de esa condición.

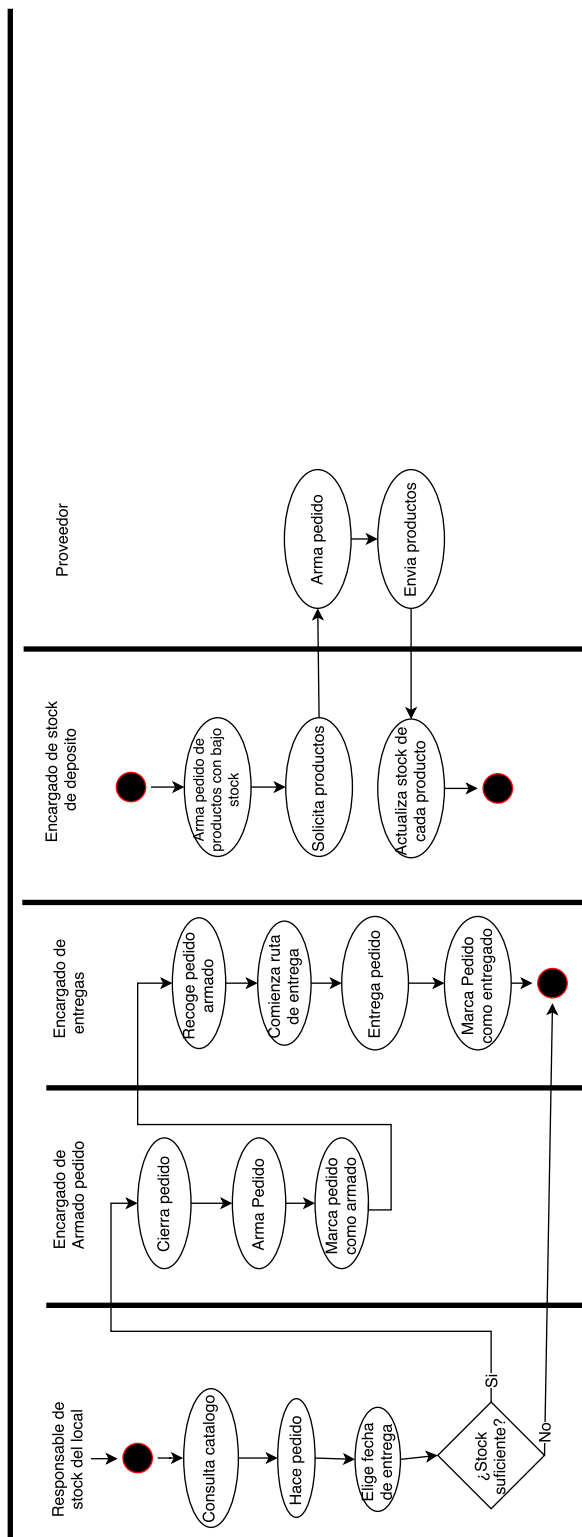
Los flujos que se observan son:

- El flujo de un pedido de un cliente y de un local desde que es realizado hasta que es entregado o no a su domicilio.
- El flujo de reponer el stock de un depósito desde que se arma el pedido para el proveedor hasta que este los envía y se actualiza el stock en consecuencia.

En estos tres flujos se puede observar el orden en el que son realizados muchos de los casos de uso que se mostraron en el primero modelo.

A continuación presentamos los tres diagramas de actividad realizados:





3.4.2. Trazabilidad

A continuación listamos los requerimientos que se modelan con este diagrama.

- **Los usuarios se pueden autenticar:** Actividad loguearse
- **Cerrar un pedido desde la pagina web:** Actividad Cierra pedido
- **Tomar productos del pedido =>Notificar pedido armado:** Actividad Marca pedido como armado
- **Coordinar fecha de entrega:** Actividad Elije fecha
- **Los locales pueden y ver y seleccionar los productos que desea recibir:** Actividad Consultar Catalogo
- **Un cliente pueda seleccionar la fecha de entrega:** Actividad Elije fecha de entrega
- **Entregar pedidos al local:** Flujo de un pedido de un local
- **Seleccionar pedido entregado:** Actividad Marca pedido como entregado
- **El cliente pago en contra-entrega =>se notifica pago a contra-entrega:** Actividad Paga pedido al contado
- **El cliente pago desde la pagina =>se notifica el pago online:** Actividad Loguearse en PayPal e Ingresando datos tarjeta
- **Coordinar fecha de entrega con proveedor:** Actividad Solicita productos
- **Ofrecer pago con PayPal:** Actividad Logueandose en PayPal
- **Ofrecer pago con crédito:** Actividad Ingresando datos tarjeta
- **Ofrecer pago con débito:** Actividad Ingresando datos tarjeta
- **Ofrecer pago con efectivo:** Actividad Paga pedido al contado

3.5. Trazabilidad

Para mostrar la trazabilidad y la completitud de nuestros diagramas, armamos una tabla con los requerimientos y las presunciones del dominio aclarando para cada uno en que diagrama se ve reflejado. Luego en cada diagrama en particular se detalla cada requerimiento o presunción del dominio.

#	Requerimiento	Diagrama
1	Reponer los productos en las góndolas	
2	Agregar depósitos al sistema	CU - MC
3	Los usuarios se pueden autenticar.	CU - DA
4	Que el cliente envíe una copia del DNI mediante el sitio.	CU
5	Cerrar un pedido desde la pagina web	CU - MC - DA
6	Tomar productos del pedido =>Notificar pedido armado.	CU - MC - DA
7	Configurar umbrales por producto en el sistema.	CU - MC
8	Incrementar stock cuando recibo reposición del proveedor.	CU - MC - FSM
9	Coordinar fecha de entrega.	CU - MC - DA
10	Se puede consultar los registros de las ventas.	CU - MC
11	Se puede configurar los parámetros del reporte.	CU - MC
12	Mantener listado de productos comprados.	MC
13	Mantener información de los medios de pagos utilizados.	MC
14	Los locales puedan registrarse.	CU - MC
15	Los locales pueden y ver y seleccionar los productos que desea recibir.	CU - DA
16	Los locales pueden modificar los pedidos no cerrados.	CU - MC
17	Los clientes pueden y ver y seleccionar los productos que desean recibir.	CU - FSM
18	Los clientes pueden modificar los pedido no cerrados.	CU - MC
19	Un cliente pueda registrarse.	CU - MC
20	Un cliente pueda seleccionar la fecha de entrega.	CU - MC - DA
21	Un cliente pueda ingresar sus datos de contacto.	CU - MC
22	Reservar stock.	CU - MC - FSM
23	Los productos en falta son filtrado del listado que ve el cliente.	CU - FSM
24	La cantidad máxima permitida para pedir un producto es igual a la que hay en el deposito.	CU - FSM
25	Lograr que la interfase funcione correctamente en computadoras de escritorio	
26	Lograr que la interfase funcione correctamente en dispositivos móviles	
27	Entregar pedidos al local	DA
28	Consultar pedidos armados	CU - MC
29	Seleccionar pedido entregado	CU - MC - DA
30	El cliente pago en contraentrega =>se notifica pago a contraentrega.	CU - MC - DA
31	El cliente pago desde la pagina =>se notifica el pago online.	CU - MC - DA
32	Coordinar fecha de entrega con proveedor.	CU - DA
33	Proveedor envía producto	FSM
34	Llevar el conteo en base a los pedidos realizados de cuantos productos quedan.	CU - MC - FSM
35	Se disparan alarmas cuando la cantidad de productos cae debajo del umbral configurado	FSM
36	Contactar al proveedor.	FSM

37	Poder definir si un cliente es malo en función de un porcentaje de pedidos no entregados.	
38	Mantener historial de pedidos entregados y no entregados de un cliente.	MC
39	Incrementar el precio del próximo pedido.	
40	Rechazar pedido durante un tiempo	MC - FSM
41	Pedido llega a la casa del cliente	MC - FSM
42	Cliente pague el pedido	CU - MC
43	Ofrecer pago con PayPal	CU - MC - DA
44	Ofrecer pago con crédito	CU - MC - DA
45	Ofrecer pago con débito	CU - MC - DA
46	Ofrecer pago con efectivo.	CU - MC - DA

CU: Diagrama de casos de uso.

MC: Diagrama de clases - Modelo conceptual.

FSM: Maquina de estados.

DA: Diagrama de actividad.

4. Discusión

4.1. Modificación de los pedidos

Decidimos que tanto clientes como locales no puedan quitar productos de su pedido originales al momento de modificarlos.

En el caso de los pedidos de los clientes, lo que motivo esta decisión fue el hecho de que, si el cliente selecciono pagar su pedido original de forma online con tarjeta o Paypal, entonces todos los productos de dicho pedido ya fueron comprados. Entonces, si permitiéramos que el cliente quite productos de su pedido, podría darse el caso en que el cliente modificara su pedido de forma tal que el nuevo pedido tenga un importe menor que el abonado originalmente. En dicho caso tendríamos dos formas posibles de actuar:

- Reembolsar al cliente la diferencia.
- Quedarnos con la diferencia de dinero.

La primer opción nos parece un tanto ilógica e incomoda desde un punto de vista comercial y la segunda opción nos parece deshonesta.

A pesar de contar con estas dos opciones, nosotros consideramos que el hecho de que el cliente realice un pedido establece un **contrato o compromiso** entre las dos partes (*el cliente y Mes %*), en el cual el cliente se compromete a pagar los productos y la cadena Mes % garantiza tener en stock los productos pedidos y entregarlos en tiempo y forma. Por ende, no permitimos que los clientes quiten productos de sus pedidos.

Agregar productos al pedido no implica ninguna modificación al *contrato* previo, solo es un nuevo contrato al cual Mes % se compromete a entregar en la misma fecha que el anterior.

Por otra parte, tampoco permitimos que los locales quiten productos de su pedido y esto es debido a como funciona el sistema de alarmas (*avisos por mail*) de falta de stock de los productos. **Este mismo motivo también aplica a los pedidos de los clientes.**

Supongamos el siguiente escenario: Un pedido de un local causa que el stock de el producto X caiga por debajo del umbral configurado. Esto causara que se active la alarma asociada a dicho producto y que se envíe un mail al encargado de stock del deposito. Al recibir la notificación el encargado de stock realiza un pedido al proveedor para reponer las unidades del producto en falta. Supongamos ahora que el local es capaz de quitar productos de su pedido y decide quitar **todas** las unidades que solicito del producto X, causando así que el stock disponible en el deposito supere al umbral. Ahora, tenemos dos escenarios posibles:

- Los nuevos pedidos que arriban al sistema causan que el stock del producto X vuelva a caer por debajo del umbral, enviándose un nuevo aviso al encargado de stock del deposito. Esto causaría que el sistema de alarmas se vuelva confuso para el encargado del stock, ya que podría darse el caso en que el stock del producto este cruzando de un lado al otro del umbral constantemente (*por modificaciones en los pedidos*) y que el encargado reciba una cantidad absurda de notificaciones.
- No se reciben nuevos pedidos solicitando al producto X. Luego, arriba el pedido hecho al proveedor y ahora se tiene una cantidad muy grande de unidades del producto X que bien podrían: no venderse, producir problemas de espacio en el deposito, echarse a perder (*si es que es algo perecedero*), etc.

Ninguno de los dos escenarios nos parece tolerable, de modo que este motivo nos llevo a no permitir que los locales quiten productos de sus pedidos y es un motivo mas para que los clientes no puedan hacerlo.

4.2. Rechazo de los pedidos

Si el sistema no cuenta con el stock suficiente (*definido como stock real menos stock reservado*) para satisfacer el pedido de un cliente en su totalidad, entonces se rechazara su pedido.

Esto se desprende directamente del enunciado, el cual nos indica que: “No podemos tener problemas de stock cuando un pedido se realiza de forma online”.

Esto nos causa un pequeño inconveniente, el cual es: El usuario ve el stock disponible con el que cuenta el deposito al momento de **iniciar** su pedido, pero este valor puede diferir (*haber disminuido*) al momento en que el usuario quiere confirmar su pedido, causando que el sistema se lo rechace y que el usuario haya perdido tiempo de su vida.

Creemos que este inconveniente no es muy grave y, si el deposito cuenta con cantidades de stock lo suficientemente grandes (*y umbrales bien configurados*) como para que el pedido de un usuario no represente una diferencia para el resto de los usuarios comprando, entonces este escenario no sucederá frecuentemente.

4.3. Pedidos no entregados

Si al enviar un pedido a un cliente este no se encuentra en su domicilio, entonces se informa que el pedido no pudo ser entregado y los productos vuelven al deposito. Ahora, existe la posibilidad de que el cliente haya pagado online el pedido antes de recibirlo, en cuyo caso tenemos tres formas de actuar:

- Devolverle al cliente su dinero.
- Quedarnos con el dinero del cliente y con los productos del pedido.
- Quedarnos con el dinero del cliente y hacer que este vaya al deposito a buscar su pedido.

Nosotros optamos por la segunda opción, ya que es la mas conveniente para Mes %. Pero, puede que esto no sea legal en algunos lugares, en cuyo caso creemos que se debería optar por la tercer opción, ya que es la que menos gastos y molestias le ocasionarían a Mes %. En caso de elegir esta opción, seria necesario agregar una interacción entre el cliente y el encargado del deposito donde este le solicita su pedido y el encargado se lo entrega. Además de que se debería tener en mente el problema de espacio que esto podría ocasionar y de que forma el encargado de stock del deposito sabe que pedido corresponde a que cliente. También seria necesario agregar un caso de uso en donde, luego de entregarle el pedido al cliente, el encargado de stock del deposito lo marque como “Pedido entregado”.

La primer opción queda descartada por lo ya dicho en la sección “Modificación de los pedidos”, sobre que los pedidos son **contratos**. Además, desde un punto de vista comercial, seria realmente contraproducente devolver el dinero del pedido y además haber perdido dinero en el envío fallido del mismo.

4.4. Fechas de entrega de los pedidos

El sistema propone a los usuarios tres fechas al azar, en días hábiles, dentro de los siete días siguientes a la fecha en que se realizo el pedido.

Esto podría ocasionar que muchos pedidos se agenden para un mismo día pero, dado que la logística de la entrega de los pedidos no corre por nuestra cuenta y que no se nos impuso ninguna restricción sobre la cantidad de pedidos que se pueden entregar por día, asumimos que esto no ocasiona un problema y que los pedidos van a poder ser entregados.

En caso de que esto ocasionara un problema, podría resolverse fácilmente estableciendo un limite a la cantidad de pedidos (*valor que desconocemos*) agendados por día y rechazando los

pedidos de los usuarios si, a partir de la fecha actual, sucede que los próximos siete días alcanzaron su tope de pedidos para entregar.

5. Conclusiones

Este trabajo no resulto nada sencillo ya que implicaba realizar muchos diagramas, los cuales estaban estrechamente relacionados, y debían ser consistentes entre si y con lo realizado en el trabajo anterior.

Desde nuestro punto de vista, el diagrama de **casos de uso** debería haberse realizado en el primer trabajo conjuntamente con los diagramas de contexto, ya que estos tienen una muy fuerte relación y permiten detectar fácilmente errores o inconsistencias entre si.

También creemos que el escribir los detalles de cada operación de los casos de uso ayuda enormemente a determinar el alcance del sistema, las cosas que necesitamos especificar, cómo se realizan y las cosas que podemos delegar a terceros, etc; y todo esto habría ayudado a la hora de diseñar el diagrama de objetivos, ya que mientras se escribe el detalle de una operación se piensa en diferentes formas de llevarla a cabo (*posibles o-refinamientos*).

El diagrama de clases nos ayudo mucho a comprender con que información iba a trabajar nuestro sistema, que información necesitábamos guardar y cual debíamos mantener actualizada.

Las máquinas de estado fueron un verdadero ~~dolor de cabeza~~ reto, ya que realizar la sincronización de los pedidos de los usuarios de forma que todos puedan realizar pedidos sin importar el estado de los otros usuarios fue, como mínimo, muy complicado. A esto, además, se suma el hecho de que la máquina debía reflejar como el sistema mantenía actualizado el stock disponible y stock reservado del depósito en el que se hacía el pedido y el manejo de las alarmas una vez que el stock disponible caía por debajo de cierto umbral. Todo el proceso de desarrollar la FSM nos permitió notar algunas falencias en los detalles de algunos casos de uso (*en el que el cliente realiza el pedido*).

Los diagramas de actividad no aportaron mucho al entendimiento general del problema ni a detectar algún error. Además fueron los primeros diagramas en estar terminados y, por mucha diferencia, los que más rápido se hicieron.

Finalmente, todo este proceso de muchas, muchas y muchas horas nos llevo a entender la dificultad de encarar un proyecto similar a un proyecto real (*aunque claramente acotado*) y de la gran cantidad de decisiones que se deben tomar durante la etapa de diseño.

Creemos que sería sumamente útil contar con un software que se encargara de validar los diferentes diagramas (*al menos hasta cierto punto*), chequear inconsistencias entre estos y/o sugerir cambios.

Como nota final, creemos que el tener que atar nuestra solución al diagrama de objetivos realizado en el primer trabajo nos limitó en este trabajo a la hora de tomar decisiones, ya que al momento de realizar el diagrama de objetivos hubo muchas situaciones (*como el hecho de que no entregar un pedido ni devolver el dinero fuera ilegal o como asignar las fechas de entrega a los pedidos*) que no tuvimos en cuenta.