



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

Análisis preliminar del sistema de software para la cadena de supermercados Mes%

1^{er} cuatrimestre de 2015

Ingeniería De Software I

Tutor: *Ezequiel Castellano*

Grupo 10

Integrante	LU	Correo electrónico
Barrios, Leandro Ezequiel	404/11	ezequiel.barrios@gmail.com
Benegas, Gonzalo Segundo	958/12	gsbenegas@gmail.com
Di Alessio, Adrian Alejandro	631/06	adrianalejandro86@hotmail.com
Rodriguez, Pedro	197/12	pedro3110.jim@gmail.com
Vanecek, Juan	169/10	juann.vanecek@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

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

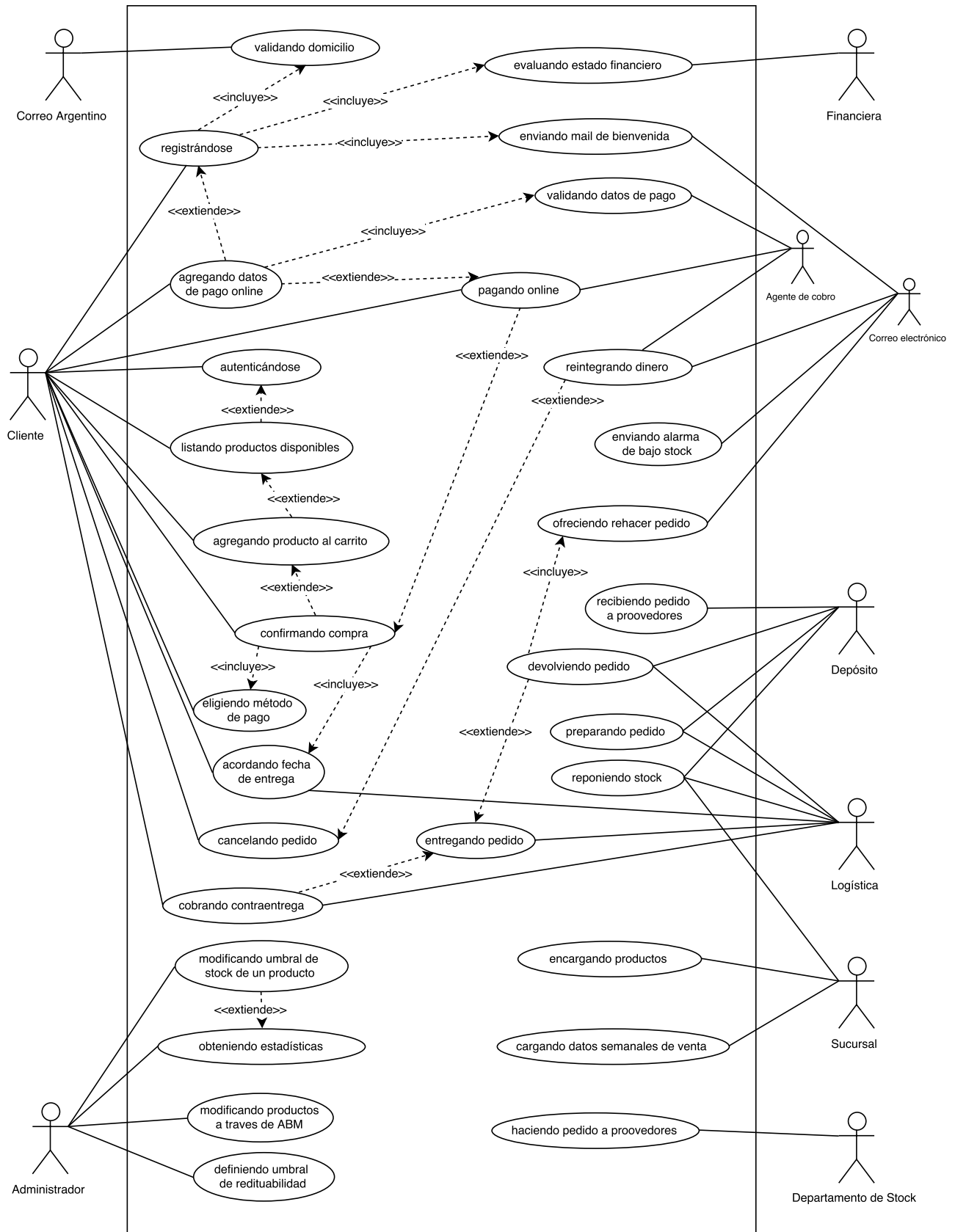
Esta página fue dejada en blanco intencionalmente.

Contents

1	Introducción	2
2	Casos de Uso	4
3	Modelo conceptual	18
3.1	OCL	19
4	Máquinas de estados finitos	23
5	Diagrama de actividad	25

Parte I. Introducción

Parte II. Casos de Uso



Caso de Uso: Registrándose**Actors:** Cliente**Pre:** -**Post:** El cliente se encuentra registrado**Curso Normal**

1. El cliente ingresa su usuario y contraseña
2. El sistema valida que el usuario no exista.
3. El sistema valida que la contraseña sea segura.
4. El cliente ingresa sus datos personales
5. Incluye CU: evaluando estado financiero
6. El cliente ingresa su dirección
7. Incluye CU: validando domicilio
8. Si el cliente desea: es extendido por CU: Agregando datos de pago online
9. El cliente ingresa su mail
10. Incluye CU: enviando mail de bienvenida
11. El cliente ingresa al link de bienvenida
12. El sistema marca al usuario como validado
13. FIN CU

Curso Alternativo

- 2.1 Si el usuario ya existe, mostrar mensaje e ir a 1.
- 3.1 Si la contraseña no es segura, informar al cliente e ir a 1
- 5.1 Si el cliente tiene deudas, denegarle el registro. FIN CU.
- 7.1 si el domicilio no es válido, mostrar mensaje e ir a 6.
- 11.1 Si luego de 10 días el usuario no ingresa al link de bienvenida, el registro se cancela, FIN CU.

Caso de uso 1: Registrándose

Caso de Uso: Evaluando estado financiero**Actors:** Financiera**Pre:** -**Post:** -**Curso Normal**

1. El sistema envía request a la API de estado financiero con el DNI del cliente
2. El sistema parsea la respuesta de la API

Curso Alternativo

Caso de uso 2: Evaluando estado financiero

Caso de Uso: Enviando mail de bienvenida**Actors:** Correo electrónico**Pre:** -**Post:** -**Curso Normal**

1. El sistema define el asunto, cuerpo y destinatario(nuevo cliente) del nuevo mensaje.
2. El sistema pide al servido de correo electrónico enviar el mensaje.
3. El servidor de correo electrónico envía el mensaje.

Curso Alternativo

Caso de uso 3: Enviando mail de bienvenida

Caso de Uso: Autenticándose**Actors:** Cliente**Pre:** -**Post:** El cliente se encuentra autenticado**Curso Normal**

1. El cliente ingresa el usuario y la contraseña
2. El sistema verifica los datos ingresados por el cliente
3. El cliente es redirigido al portal de bienvenida
4. FIN CU

Curso Alternativo

- 3.1 Si los datos son inválidos, se muestra un mensaje de error
- 4.1 FIN CU

Caso de uso 4: Autenticándose

Caso de Uso: Listando productos disponibles**Actors:** Cliente**Pre:** El cliente está autenticado**Post:** -**Curso Normal**

1. El cliente ingresa al listado de productos
2. El sistema obtiene los productos que están en stock
3. El sistema obtiene las recomendaciones para el usuario
4. El sistema muestra los productos y las recomendaciones que están en stock
5. Si el cliente lo desea, se extiende con CU: Agregando producto al carrito
6. FIN CU

Curso Alternativo

Caso de uso 5: Listando productos disponibles

Caso de Uso: Agregando producto al carrito**Actors:** Cliente**Pre:** El cliente está autenticado**Post:** El carrito contiene al producto agregado**Curso Normal**

1. El cliente hace click sobre el producto
2. El sistema muestra un dropdown con la cantidad de unidades que están disponibles en ese momento
3. El cliente elige la cantidad
4. El sistema agrega el producto al carrito y calcula el monto total
5. Si el cliente lo desea, es extendido por CU: confirmando compra 6. FIN CU

Curso Alternativo

Caso de uso 6: Agregando producto al carrito

Caso de Uso: Confirmando compra**Actors:** Cliente**Pre:** El cliente tiene un carrito armado**Post:** El cliente tiene un carrito reservado y confirmado**Curso Normal**

1. El sistema ratifica la disponibilidad de stock para cada producto, y los reserva para el cliente; el carrito se encuentra reservado
2. Incluye: Calcular costo de envío
(El sistema obtiene el costo de envío de la API de logística)
3. El sistema informa del costo total de la compra, incluyendo el envío.
4. Incluye: Acordando fecha de entrega
5. Incluye: Elijiendo método de pago
6. Si el pago es online: Es extendido por: Pagando online.
7. El cliente confirma el pedido, FIN CU

Curso Alternativo

- 1.1 Si algún producto ya no tiene disponibilidad, se resta del carrito y se le informa al usuario; vuelve al paso 1.
- 7.1 Si el cliente no confirma el pedido luego de 10 minutos, los productos son reingresados a stock, y el carrito deja de estar reservado, vuelve a paso 1

Caso de uso 7: Confirmando compra

Caso de Uso: Acordando fecha de entrega**Actors:** Cliente, Logística**Pre:** El cliente tiene un carrito reservado**Post:** El pedido tiene fecha tentativa de entrega**Curso Normal**

1. El sistema pregunta próximas fechas libres a la API de logística
2. El sistema presenta las posibles fechas al cliente
3. El cliente elige la fecha deseada
4. FIN CU

Curso Alternativo

Caso de uso 8: Acordando fecha de entrega

Caso de Uso: Eligiendo método de pago**Actors:** Cliente**Pre:** El cliente tiene un carrito reservado**Post:** El cliente eligió un método de pago permitido**Curso Normal**

1. El sistema determina si el cliente tiene autorizado el pago contraentrega
2. El sistema presenta los métodos de pagos disponibles
3. El cliente elige el método de pago deseado
4. FIN CU

Curso Alternativo

Caso de uso 9: Eligiendo método de pago

Caso de Uso: Pagando online**Actors:** Cliente, agente de cobro**Pre:** El cliente eligió el método de pago online**Post:** -**Curso Normal**

1. Si el cliente lo desea, es extendido por CU: agregando datos de pago online
2. El sistema muestra datos de pago asociados al cliente
3. El cliente indica método y datos de pago
4. El sistema abre una ventana del agente de cobro con los datos de la transacción.
5. El cliente realiza la operación a través del agente de cobro, generando un token comprobante del pago.
6. El sistema recibe el comprobante de pago, y lo verifica contra el agente de pago.
7. El sistema envía un mail al cliente, informando la compra exitosa y adjuntando un comprobante de pago de la operación a través del sistema de correo electrónico.
8. FIN CU

Curso Alternativo

- 3.1 Si el cliente no posee datos de pago, ir a paso 1
- 5.1 Si el agente de cobro rechaza el pago, ir al paso 1.
- 6.1 Si el token de pago es inválido, informar al usuario, e ir al paso 1

Caso de uso 10: Pagando online

Caso de Uso: Agregando datos de pago online**Actors:** Cliente**Pre:** El cliente está autenticado**Post:** El cliente posee un nuevo dato de pago asociado a su cuenta**Curso Normal**

1. El cliente elige el método de pago online de entre las opciones disponibles
2. Según el método de pago elegido, el cliente ingresa los datos de autenticación solicitados.
3. Incluye: validando datos de pago
4. FIN CU

Curso Alternativo

- Si los datos de pago son inválidos regresa a paso 1.

Caso de uso 11: Agregando datos de pago online

Caso de Uso: Cancelando pedido**Actors:** Cliente**Pre:** El cliente tiene un pedido sin confirmar, o confirmado, pero sin armar en depósito.**Post:** El pedido es cancelado.**Curso Normal**

1. Si el pedido está sin confirmar, lo cancela, FIN CU.
2. La reserva de productos se anula, y los mismos se reingresan a stock
3. Si fue pagado, es extendido por CU: Reintegrando dinero
4. FIN CU

Curso Alternativo

Caso de uso 12: Cancelando pedido

Caso de Uso: Modificando pedido**Actors:** Cliente**Pre:** El cliente tiene un pedido sin confirmar, o confirmado, pero sin armar en depósito.**Post:** El pedido es modificado.**Curso Normal**

1. El cliente quita los productos que desee y agrega los productos que desee, siempre y cuando estén en stock.
2. Si el pedido está sin confirmar, se registra la modificación, FIN CU.
3. Se cancela el pedido anterior: incluye CU Cancelando Pedido.
4. Se genera un nuevo pedido con los datos modificados.
5. Incluye CU: Confirmando compra.
6. FIN CU

Curso Alternativo

Caso de uso 13: Modificando pedido

Caso de Uso: Validando datos de pago**Actors:** Agente de cobro**Pre:** El cliente ingresó los datos de pago**Post:** Los datos de pago fueron validados**Curso Normal**

1. El sistema contrasta los datos de pago del cliente contra el agente de cobro correspondiente a través de una API.
2. FIN CU

Curso Alternativo

- 2.1 Si los datos de pago no son válidos, el sistema los marca como inválidos, FIN CU

Caso de uso 14: Validando datos de pago

Caso de Uso: Reintegrando dinero**Actors:** Agente de cobro, Correo electrónico**Pre:** El cliente canceló un pedido**Post:** El dinero correspondiente al pedido fue reintegrado al cliente**Curso Normal**

1. El sistema contacta al agente de cobro, solicitando la anulación de las operaciones correspondientes al pago del pedido.
2. El agente de cobro anula las operaciones de pago solicitadas, y entrega un número de operación y un comprobante de anulación para cada una de ellas.
3. El sistema envía un mail al cliente, informando que el pedido fue anulado, a través del sistema de correo electrónico, adjuntando los comprobantes de pago
4. FIN CU

Curso Alternativo

- 3.1 Si el pago no puede ser anulado, se le informa de la situación al cliente, brindándole los números de operación.

Caso de uso 15: Reintegrando dinero

Caso de Uso: Validando domicilio**Actors:** Correo argentino**Pre:** El cliente ingresó los datos de su domicilio**Post:** Los datos de su domicilio fueron validados**Curso Normal**

1. El sistema verifica los datos de domicilio del cliente mediante la API del Correo Argentino
2. FIN CU

Curso Alternativo

- 2.1 Si los datos de domicilio no son correctos, se marcan como inválidos. FIN CU

Caso de uso 16: Validando domicilio

Caso de Uso: Preparando pedido**Actors:** Depósito, Logística**Pre:** El cliente tiene un pedido confirmado**Post:** El cliente tiene un pedido preparado**Curso Normal**

1. El sistema brinda el listado de productos a preparar, junto con el domicilio y la fecha y hora de entrega
2. Un operario del depósito marca el pedido como preparado y se encarga de empaquetar los productos
3. Depósito informa del pedido a logística, y se pacta el envío para la fecha correspondiente; los datos de la operación se cargan en el sistema
4. La información correspondiente al envío: remito, hoja de ruta, factura, etcétera, se empaqueta junto con el mismo.
5. FIN CU

Curso Alternativo

Caso de uso 17: Preparando pedido

Caso de Uso: Entregando pedido**Actors:** Logística, Depósito**Pre:** El cliente tiene un pedido preparado**Post:** Se realiza un intento de entrega**Curso Normal**

1. Logística retira el pedido.
2. Logística entrega el pedido al cliente.
3. Si el pago es contraentrega, es extendido por CU: Cobrando contraentrega
4. Logística registra la entrega al cliente satisfactoria
5. FIN CU.

Curso Alternativo

2.1 Si el cliente no recibe el pedido, es extendido por CU: Devolviendo pedido.

Caso de uso 18: Entregando pedido

Caso de Uso: Cobrando contraentrega**Actors:** Logística, Cliente**Pre:** El cliente eligió contraentrega como método de pago**Post:** El pago del pedido fue efectuado**Curso Normal**

1. El transportista presenta la factura de compra
2. El cliente entrega el dinero correspondiente
3. FIN CU

Curso Alternativo

Caso de uso 19: Cobrando contraentrega

Caso de Uso: Devolviendo pedido**Actors:** Logística, Depósito**Pre:** La entrega del pedido fue fallida**Post:** El pedido está anulado y los productos aprobados fueron reingresados a stock**Curso Normal**

1. Logística devuelve stock al depósito
2. Encargados del depósito hacen un inventario de la mercadería malograda
3. La mercadería en buen estado es reingresada a stock
4. Depósito carga la falta del cliente y el costo generado a la empresa
5. El pedido es anulado.
6. Incluye: Ofreciendo rehacer pedido.
7. FIN CU

Curso Alternativo

Caso de uso 20: Devolviendo pedido

Caso de Uso: Ofreciendo rehacer pedido**Actors:** Correo electrónico**Pre:** El cliente tiene un pedido anulado.**Post:** Se le ofrece al cliente rehacer el pedido.**Curso Normal**

1. El sistema genera un link hacia una orden de compra con el mismo carrito del pedido anulado
2. El sistema de correo electrónico envía un mail al cliente con el link y una invitación a rehacer el pedido
3. FIN CU

Curso Alternativo

Caso de uso 21: Ofreciendo rehacer pedido

Caso de Uso: Enviando alarma de bajo stock**Actors:** Correo electrónico**Pre:** Existe al menos un producto con stock menor al límite estipulado**Post:** Se avisa de la falta de stock al Departamento de Stock**Curso Normal**

1. El sistema prepara una lista de todos los productos por debajo del límite estipulado y el stock necesario para reestablecerlos
2. El sistema de correo electrónico envía un mail con los datos al Departamento de Stock
3. FIN CU

Curso Alternativo

Caso de uso 22: Enviando alarma de bajo stock

Caso de Uso: Haciendo pedido a proveedores**Actors:** Departamento de stock**Pre:** Una alarma de bajo stock fue activada**Post:** Se realiza un nuevo pedido a un proveedor**Curso Normal**

1. El Departamento de Stock determina al mejor proveedor y efectúa una compra
2. El Departamento de Stock carga el comprobante de pedido al sistema
3. FIN CU

Curso Alternativo

Caso de uso 23: Haciendo pedido a proveedores

Caso de Uso: Recibiendo pedido a proveedores**Actors:** Depósito**Pre:** -**Post:** La mercadería es ingresada al depósito**Curso Normal**

1. Empleados del depósito guardan la mercadería
2. Depósito registra el ingreso y el sistema actualiza el stock de los nuevos productos
3. FIN CU

Curso Alternativo

Caso de uso 24: Recibiendo pedido a proveedores

Caso de Uso: Reponiendo stock**Actors:** Depósito, Logística, Sucursal**Pre:** Hay un pedido de reposición de la sucursal**Post:** Los productos requeridos son ingresados a la sucursal**Curso Normal**

1. Depósito reserva todo el stock necesario para completar el pedido, o el máximo posible en caso de que la reserva de stock sea menor a esta cantidad, informando la baja de los productos correspondientes.
2. Un empleado del depósito empaqueta los productos reservados y avisa a logística
3. Depósito entrega el pedido a logística junto con una hoja de ruta
4. Logística transporta el envío hasta la sucursal
5. La sucursal repone los productos en las góndolas
6. FIN CU

Curso Alternativo

- 1.1 Si no se ha reservado el stock necesario para el pedido, se espera a que llegue una reposición y se vuelve a 1

Caso de uso 25: Reponiendo stock

Caso de Uso: Definiendo umbral de redituabilidad**Actors:** Administrador**Pre:** -**Post:** Se redefine el umbral de redituabilidad**Curso Normal**

1. El administrador ingresa el nuevo umbral de redituabilidad
2. El sistema guarda el nuevo umbral
3. FIN CU

Curso Alternativo

Caso de uso 26: Definiendo umbral de redituabilidad

Caso de Uso: Modificando productos a través de ABM**Actors:** Administrador**Pre:** -**Post:** Los productos solicitados son modificados**Curso Normal**

1. El sistema muestra una interfaz para agregar, dar de baja, y modificar productos
2. El administrador ingresa la operación requerida
3. El sistema realiza los cambios necesarios y confirma la operación
4. FIN CU

Curso Alternativo

Caso de uso 27: Modificando productos a través de ABM

Caso de Uso: Obteniendo estadísticas**Actors:** Administrador**Pre:** -**Post:** Se envían las estadísticas al administrador**Curso Normal**

1. El administrador solicita las estadísticas a través del sistema
2. El sistema genera las estadísticas de venta de cada producto y compras de cada usuario, y las prepara para su adecuada visualización
3. El administrador descarga las estadísticas a través del sistema
4. Si administrador desea modificar límite de productos, es extendido por CU: Modificando umbral de stock de un producto.
5. FIN CU.

Curso Alternativo

Caso de uso 28: Obteniendo estadísticas

Caso de Uso: Modificando umbral de stock de un producto**Actors:** Administrador**Pre:** El producto existe**Post:** El producto tiene un nuevo umbral de stock**Curso Normal**

1. El administrador informa del nuevo límite para el producto
2. FIN CU

Curso Alternativo

Caso de uso 29: Modificando umbral de stock de un producto

Caso de Uso: Encargando productos**Actors:** Sucursal, Depósito**Pre:** La sucursal está autenticada, desea encargar productos**Post:** Los productos deseados fueron encargados**Curso Normal**

1. La sucursal ingresa el listado de productos que desea encargar.
2. El sistema informa a depósito del pedido.
3. El sistema le confirma a la sucursal que el pedido fue encargado.
4. FIN CU

Curso Alternativo

Caso de uso 30: Encargando productos

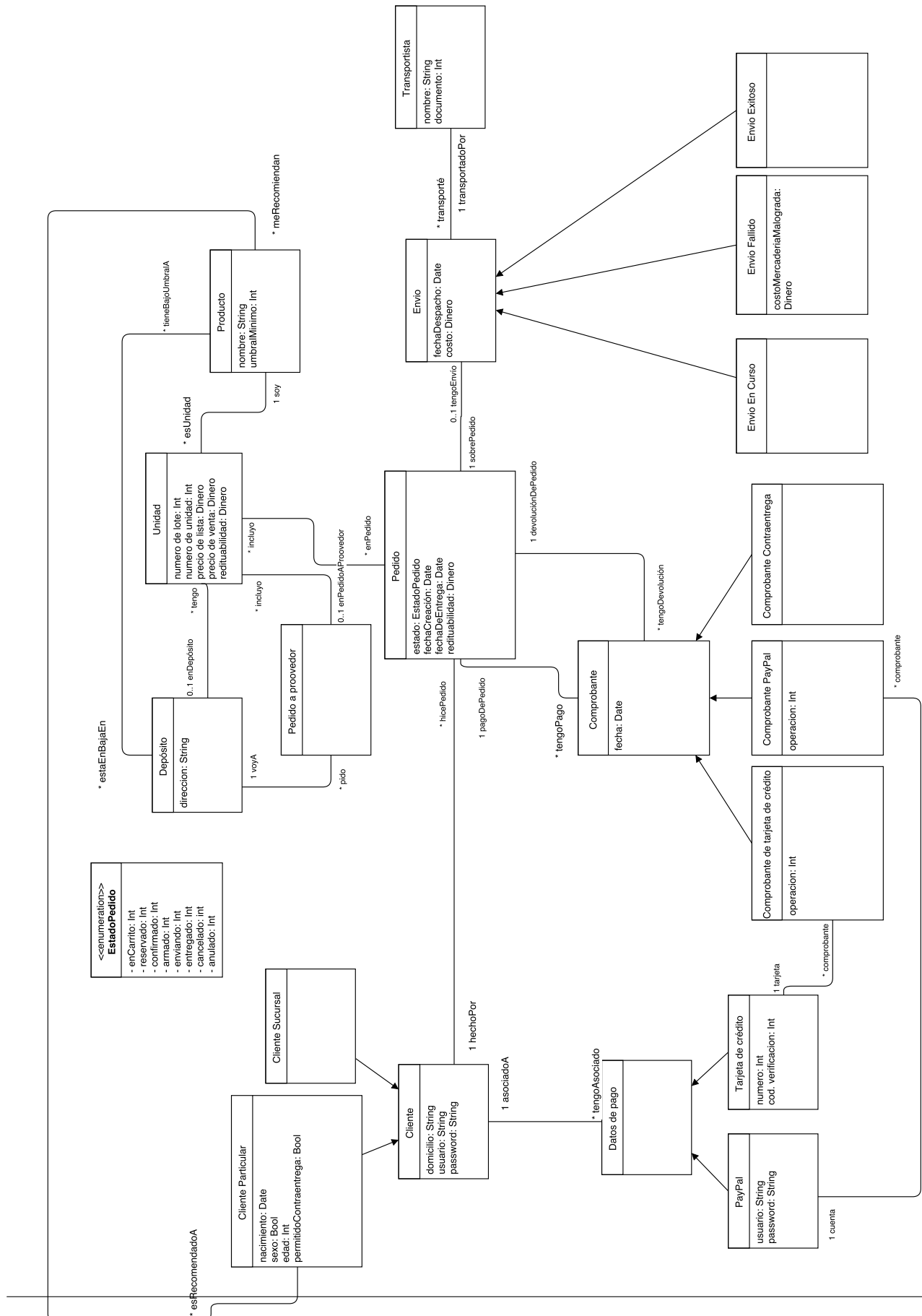
Caso de Uso: Cargando datos semanales de venta**Actors:** Sucursal**Pre:** La sucursal tiene datos de venta para informar**Post:** El sistema contiene datos de venta actualizados**Curso Normal**

1. La sucursal carga los datos de venta de la semana a través de una interfaz sencilla, por ejemplo una planilla de cálculos.
2. El sistema procesa los datos de venta, y los integra a su base de estadísticas
3. FIN CU

Curso Alternativo

Caso de uso 31: Cargando datos semanales de venta

Parte III. Modelo conceptual



3.1 OCL

1. La redituabilidad de una unidad es su precio de venta menos su precio de lista.

```
Context: Unidad
Inv: self.redituabilidad = self.precioDeVenta - self.precioDeLista
```

2. La redituabilidad de un pedido es la ganancia menos el costo.

```
Context: Pedido

ganancia = if self.estadoPedido == Concretado
  then _
    self.unidades->collect(precioDeVenta-precioDeLista)->sum()
  else 0
  endif
costo = if self.envio
  then self.envio.costo
  else 0
  endif
+ if self.envio.oclIsTypeOf(EnvioFallido)
  then self.envio.costoMercaderiaRechazada
  else 0
  endif
Inv: self.redituabilidad = ganancia - costo
```

3. Se le permite la contraentrega si la redituabilidad del cliente es mayor al umbral de redituabilidad.

```
Context: Cliente Particular

redituabilidadCliente = self.pedidos->collect(redituabilidad)->sum()
Inv: iff(permitidoContraentrega, redituabilidadCliente > _
  umbralRedituabilidad)
```

4. Un cliente no puede tener dos pedidos vigentes.

```
Context: Cliente Particular
Inv: self.pedidos->filter(x | x.estado != entregado and x.estado != _
  anulado and x.estado != cancelado).size() < 2
```

5. Un Producto está bajo en stock en un Depósito sii la cantidad de unidades de dicho producto en el Depósito está por debajo del umbral mínimo de dicho producto.

```
Context: Producto
Inv: self.estaEnBajaEn->forall(d | d.tengo->filter(u | u.soy = _
  self).size() < self.umbralMinimo)
```

6. $fechaCreacion < fechaDespacho < fechaEntrega$

```

Context: Envio
Inv: self.sobrePedido.fechaCreacion < self.fechaDespacho < _
self.sobrePedido.fechaEntrega

```

7. Una unidad está en solo uno de los siguientes lugares:

pedidoAProveedor

depósito

depósito y en pedido en estado enCarrito, anulado, cancelado

en pedido confirmado

```

Context: Unidad
Inv:
(self.enPedidoAProveedor->size() == 1 and self.enDeposito->size() == 0 _
and self.enPedido->size() == 0)
or
(self.enDeposito->size() == 1 and self.enPedidoAProveedor->size() == 0 _
and self.enPedido->size() == 0)
or
(self.enDeposito->size() == 1 and self.enPedidoAProveedor->size() == 0 _
and self.enPedido->size() > 0 and self.enPedido.estado in {enCarrito, _
cancelado, anulado})
or
(self.enDeposito->size() == 0 and self.enPedidoAProveedor->size() == 0 _
and self.enPedido->filter(p | p.estado notIn {enCarrito, cancelado, _
anulado}).size() == 1)

```

8. Un cliente C tiene un producto recomendado P si y solo si:

C nunca compro P y

P pertenece a la lista de 10 más comprados de algun cliente parecido a C

```

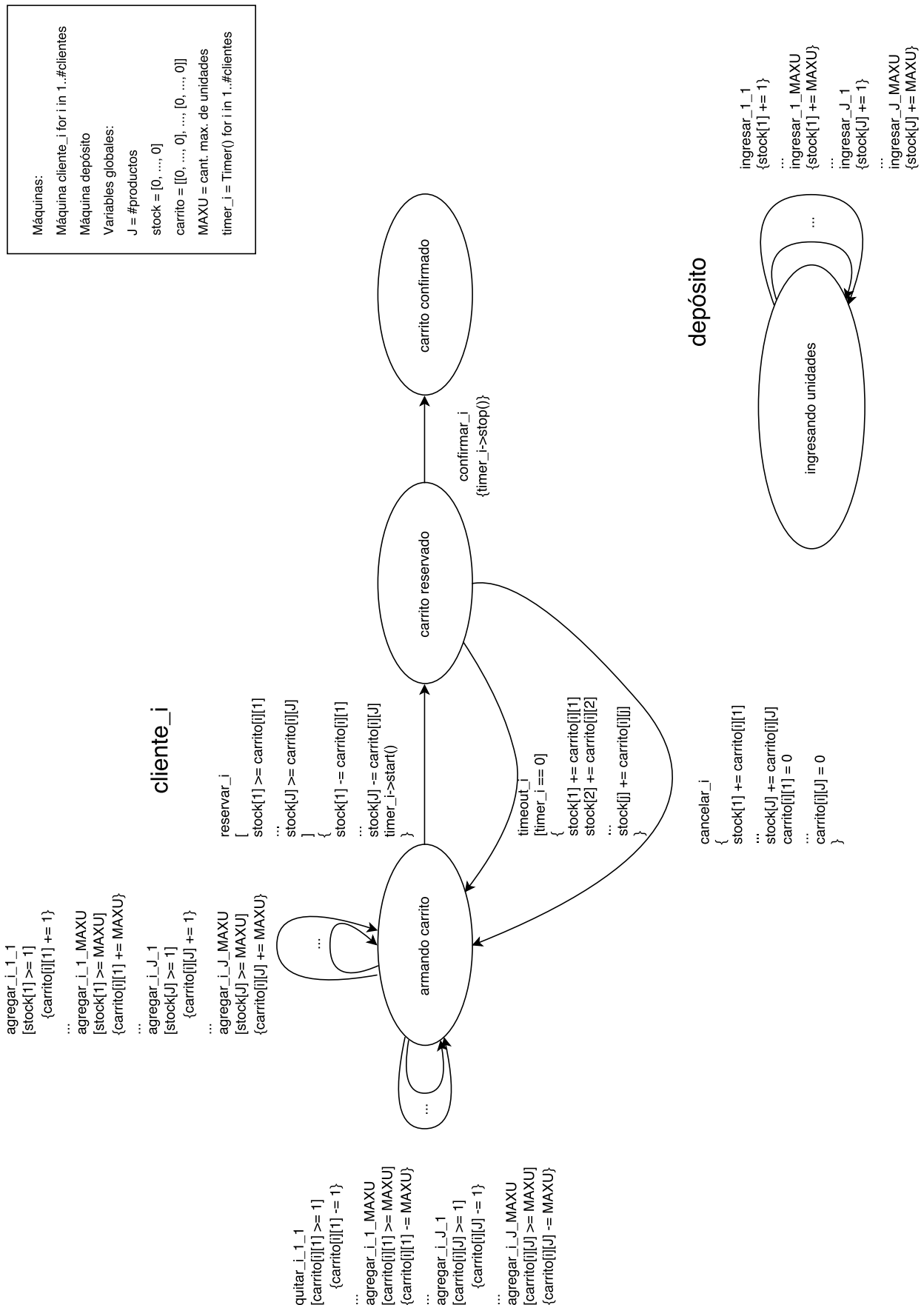
Context: Cliente Particular
edadParecida(c1, c2) = |c1.edad - c2.edad| < 4
productosComprados(c) = self.hicePedido.incluyo.soy
frecuenciaProducto(c, p) = productosComprados(c).filter(p1 | p1 == _
p).size()
frecuenciasProductosOrdenadas(c) = _
productosComprados(c).asSet().sortedBy(p | frecuenciaProducto(c, _
p)).reverse()
diezMasComprados(c) = frecuenciasProductosOrdenadas(c).subSequence(1, _
10)
clienteParecido(c1, c2) = edadParecida(c1, c2) and _
intersection(diezMasComprados(c1), diezMasComprados(c2)).size() >= 5
Inv: self.meRecomiendan->forall(p | p notIn productosComprados(self) _
and ClienteParticular.allInstances()->exists(c | clienteParecido(self, _
c) and p in diezMasComprados(c)))

```

9. Comprobantes deben tener asociados datos de pago registrados por el cliente.

```
Context: Comprobante de tarjeta de cr\'edito
Inv: self.tarjeta.asociadoA == self.pagoDePedido.hechoPor
Context: Comprobante PayPal
Inv: self.cuenta.asociadoA == self.pagoDePedido.hechoPor
```


Parte IV. Máquinas de estados finitos



Parte V. Diagrama de actividad

