



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN
Año 2017 - 2^{do} Cuatrimestre

SIMULACIÓN DE EVENTOS DISCRETOS

TRABAJO PRÁCTICO N° <1>
TEMA:<Manejo del Inventario de una Industria>
FECHA:<27 de noviembre de 2017>

INTEGRANTES:

CERVETTO, Marcos	- #FIUBA
<cervettomarcos@gmail.com>	
MARCHI, Marcos	- #FIUBA
<edgardo.marchi@gmail.com>	
PECKER MARCOSIG, Ezequiel	- #FIUBA
<ezepecker@gmail.com>	

Índice

1. Objetivo y Enunciado	2
2. Modelo Conceptual	2
2.1. Motivación	2
2.2. Descripción funcional	3
2.2.1. Demanda (cliente)	3
2.2.2. Atención al cliente y control de calidad	5
2.2.3. Control de inventario	6
2.2.4. Inventario	7
2.2.5. Proveedores	8
3. Descripción Formal	10
3.1. <i>Top Model</i>	10
3.2. Tipo de Cliente A	12
3.3. Tipo de Cliente B	13
3.4. Tipo de Cliente C	14
3.5. Control de inventario	16
3.6. Control de Calidad	18
3.7. Atención al cliente	21
3.8. Inventario	22
3.9. Industria	26
3.10. Proveedor Inmediato	26
3.11. Proveedor por múltiplo de cantidad fija	28
3.12. Proveedor por Encargo	29
4. Modelado y simulación	31
4.1. Pruebas parciales	31
4.1.1. Demandas de los Clientes	31
4.1.2. Cliente A	33
4.1.3. Cliente B	33
4.1.4. Cliente C	34
4.1.5. Control de Inventario	34
4.1.6. Control de Calidad	35
4.1.7. Atención al cliente	36
4.1.8. Inventario	37
4.1.9. Proveedor Inmediato	39
4.1.10. Proveedor por Múltiplo Fijo	39
4.1.11. Proveedor por Encargo	40
4.2. Pruebas de integración	40
4.3. Reporte y conclusiones	41
4.3.1. Reporte de resultados	41
4.3.2. Conclusiones	45
A. Código Implementado	46

1. Objetivo y Enunciado

El objetivo de este Trabajo Práctico es demostrar la comprensión del modelado y simulación utilizando el formalismo DEVS, junto a la aplicación de las técnicas mediante la herramienta CD++. Se deberá identificar un sistema del mundo real que pueda ser representado usando DEVS y construir luego un modelo del mismo. Finalmente se ejecutarán simulaciones del sistema analizado. El sistema estudiado puede ser natural o artificial, y puede existir en la realidad o no.

2. Modelo Conceptual

2.1. Motivación

Una compañía que vende un único producto está interesada en estimar cuántas unidades del mismo debería tener en inventario en los próximos n meses (n es un parámetro de entrada fijo).

La compañía cuenta básicamente con tres proveedores de diferentes materias primas. Cada producto se fabrica usando 1 unidad de cada materia prima.

La planta está compuesta por un departamento de control de inventario, un departamento de atención al cliente y control de calidad de producto (ya que los productos son perecederos), un stock de productos finalizados y tres stocks de materias primas (uno para cada material). Las demandas de la clientela se producen aleatoriamente en pedidos de 1 a 4 unidades (también aleatorios). Existen 3 tipos diferentes de demanda.

En la Figura 1 se propone un esquema del proceso. Se puede observar que el top-model consiste de tres jerarquías. En la jerarquía top se encuentran: los tres proveedores, los tres clientes y la industria en sí. En la siguiente se encuentran el inventario, el departamento de atención al cliente y el departamento de control de inventario. Finalmente, en el último nivel (dentro de Atención al Cliente) se encuentran la ventanilla de atención al cliente y el departamento de Control de Calidad del producto. Cada cliente consiste de un modelo atómico distinto para representar tres comportamientos distintos. De la misma manera se utilizan tres modelos atómicos para modelizar cada tipo de proveedor. Finalmente, para los departamentos de control de calidad, ventanilla de atención al cliente y control de inventario se utilizarán tres modelos atómicos.

La pregunta a responder mediante simulaciones es: *¿Qué política de reposición nos llevará a un manejo óptimo del nivel de stock logrando la mayor satisfacción del cliente y el menor costo de mantenimiento y reposición del inventario?*

Una pregunta adicional que surge a partir de la anterior es: *¿Cómo ponderar adecuadamente ambos requisitos en conflicto?*

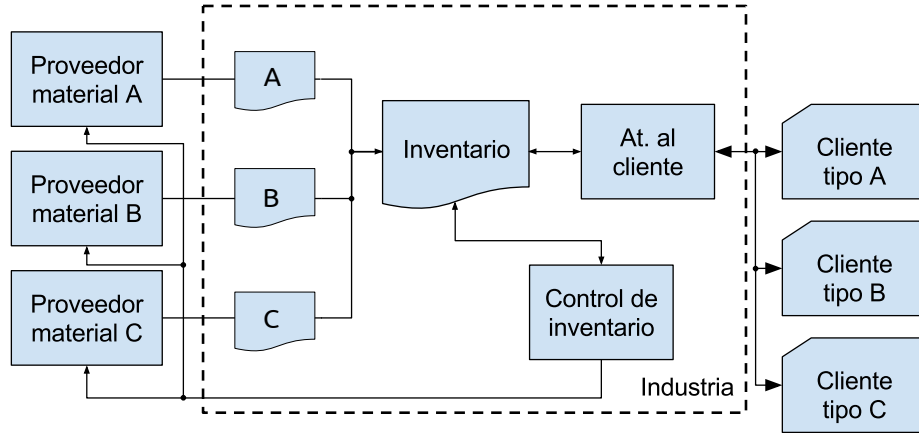


Figura 1: Esquema del problema planteado.

2.2. Descripción funcional

A continuación se describen cada uno de los bloques de la Figura 1.

2.2.1. Demanda (cliente)

Los tiempos entre demandas por tipo de cliente son variables aleatorias exponenciales e independientemente distribuidas con una media de 0,12 mes. Los tamaños de las demandas (que pueden ir de uno a cuatro unidades del producto), D , son variables aleatorias independientemente distribuidas e independientes, cuya probabilidad de ocurrencia se muestra en la Tabla 1.

D (# Unidades)	Probabilidad
1 unidad	$1/6$
2 unidades	$1/3$
3 unidades	$1/3$
4 unidades	$1/6$

Tabla 1: Demanda.

Existen 3 tipos de políticas de demanda: A, B y C. Un cliente tipo A encarga siempre n productos y los retira a principio de mes. Un cliente tipo B pide n productos pero se lleva solamente los que haya disponibles en stock en ese momento y los restantes para completar su pedido quedan encargados. Finalmente un cliente tipo C pide n productos pero solamente compra si están disponibles, sino rechaza la compra.

El modelo que representa el comportamiento de los clientes tipo A se puede observar en la Figura 2, donde la salida E_OUT indica la cantidad de productos encargados.

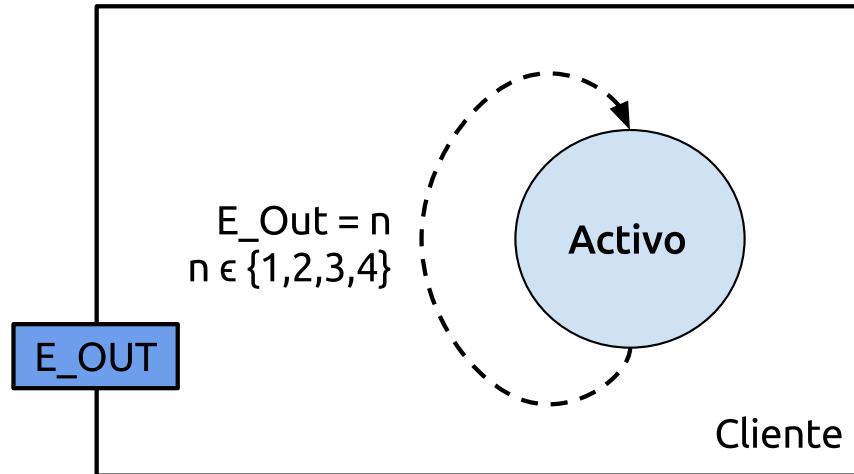


Figura 2: Modelo de cliente tipo A.

El modelo utilizado para el comportamiento de los clientes de tipo B se muestra en la Figura 3. Las salidas P_OUT, E_OUT y Q_OUT indican respectivamente la cantidad de productos retirados, la cantidad de productos encargados y la cantidad de productos pedidos.

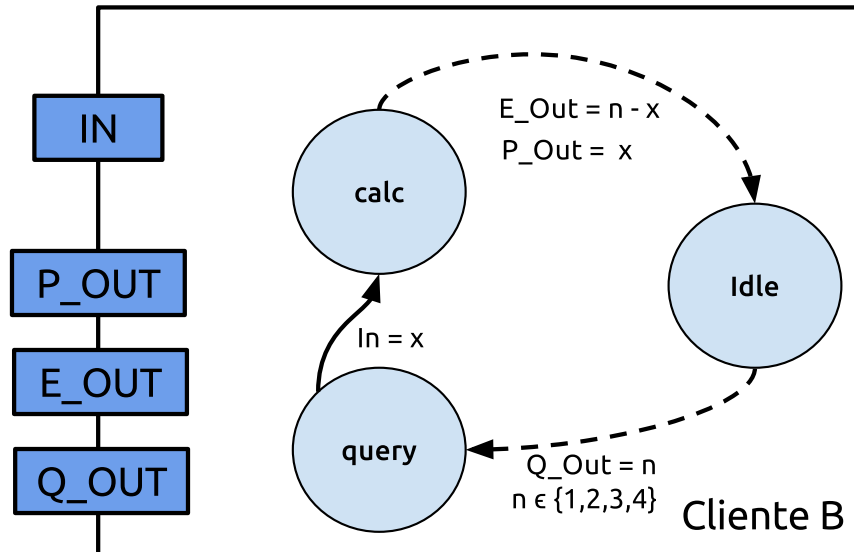


Figura 3: Modelo de cliente tipo B.

Finalmente, el modelo del comportamiento de los clientes de tipo C se muestra en la Figura 4.

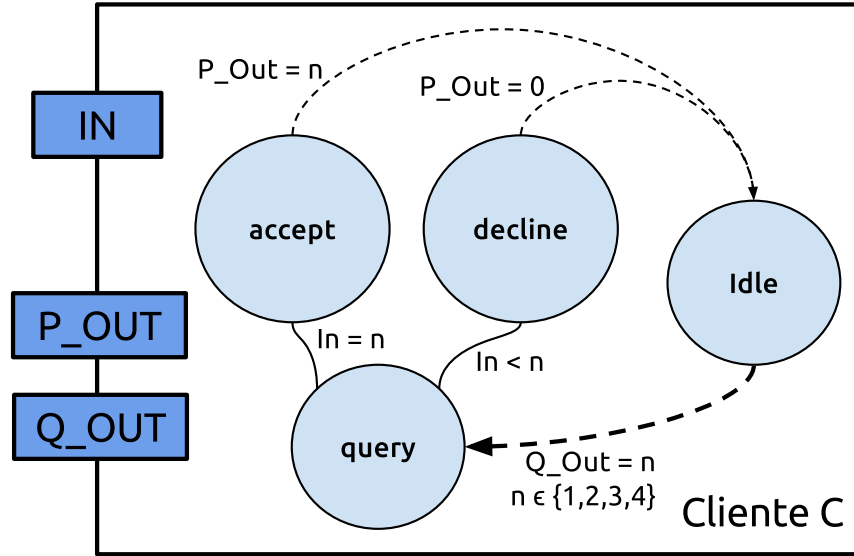


Figura 4: Modelo de cliente tipo C.

2.2.2. Atención al cliente y control de calidad

En la Figura 5 se puede observar la interfaz del módulo de Atención al Cliente y Control de Calidad. El departamento de Control de Calidad se encarga de examinar los productos al momento de venta y verificar su fecha de vencimiento. Los insumos con los que se elabora el producto son perecederos y la fecha de vencimiento del mismo se calcula con la fecha de vencimiento mínima de los insumos con los que se lo elaboró. La compañía descubre que una unidad no sirve más solamente cuando la examina debido a una venta. Si una unidad no sirve más, se desecha y se examina la siguiente unidad existente en el inventario. El departamento de atención al cliente se encarga de realizar la negociación con los distintos tipos de clientes. Es decir, consulta al inventario la cantidad de productos disponibles al recibir una consulta del cliente y le informa dicho dato al mismo. También procesa el pedido de compra de cada cliente.

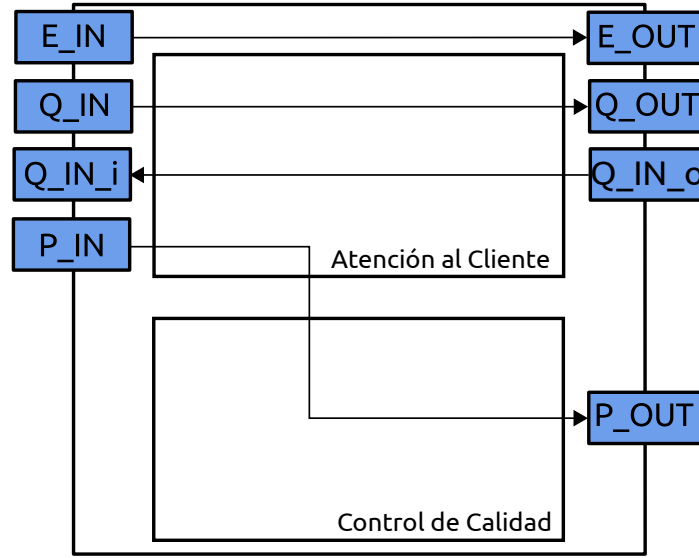


Figura 5: Atención al cliente.

2.2.3. Control de inventario

Al comienzo de cada mes, la compañía revisa el nivel de inventario y decide cuántas nuevas unidades pedir a sus proveedores. La compañía ordena Z unidades a todos los proveedores por igual. El costo será $K + pZ$, donde $K = 350$ pesos es el costo fijo del pedido y $p = 25$ pesos es el costo incremental por unidad. Naturalmente, si $Z = 0$ no se incurre en ningún costo. La compañía usa la siguiente política para decidir cuántas unidades pedir: si I es el nivel de inventario al comienzo del mes, S el inventario máximo posible (capacidad de almacenamiento) y n el parámetro de política de reposición, Z estará dado por:

$$Z = \begin{cases} S-I & \text{si } I < n \\ 0 & \text{si } I \geq n \end{cases} \quad (1)$$

El modelo está dado por la Figura 6, donde se observa que hay tres estados: Query, Calc y Wait.

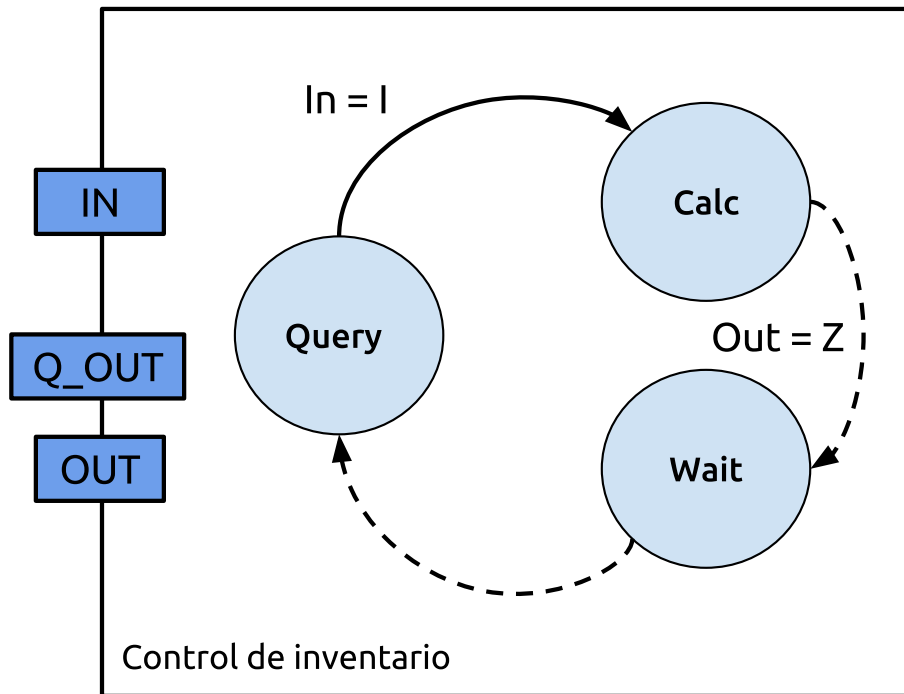


Figura 6: Esquema del control de inventario.

2.2.4. Inventario

Ante una consulta de un cliente el inventario responde con la cantidad de unidades disponibles en ese momento. En base a esto y a la política de cada cliente, al inventario se le pide una cantidad determinada de unidades para retiro y/o encargo. El inventario lleva una lista de unidades disponibles y una de unidades encargadas. Cuando una reposición llega, se usa primero para eliminar todos los pedidos pendientes (si hay) posibles; el resto de la reposición (si queda) se suma al inventario.

En la Figura 7 se observa el modelo con que se representa al Inventario. El mismo consiste de una cola FIFO (*first in - first out*). Se denomina N al largo de la cola. Suponemos también que la compañía incurre en un costo de mantenimiento del producto $h = 8$ pesos por unidad por mes mantenido en inventario (positivo). Este costo incluye alquiler del depósito de almacenamiento, seguro, impuestos y mantenimiento, al igual que costo de oportunidad de tener capital inmovilizado en inventario en vez de invertido en otra parte. Supongamos además que hay un costo de *demanda insatisfecha* $n = 50$ pesos por unidad pedida por cliente y no suministrada y por mes. Este costo contabiliza tanto el costo de mantener registros adicionales para unidades ya asignadas a clientes y aún no suministradas como un cálculo de costo por pérdida de la buena voluntad del cliente.

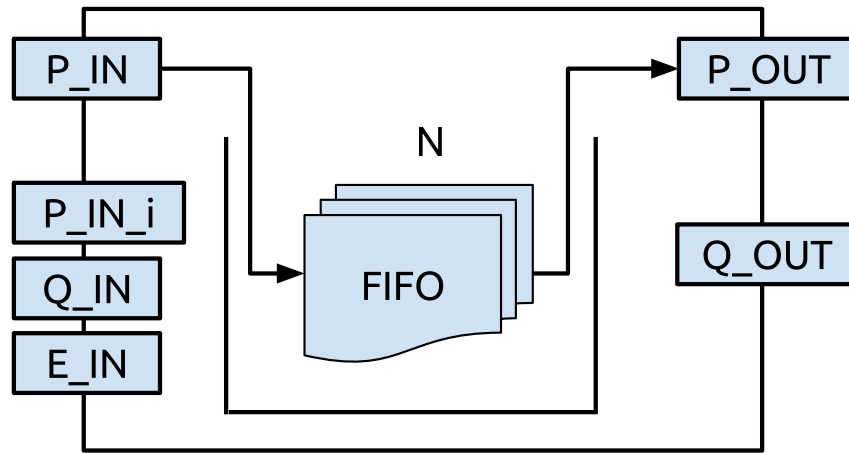


Figura 7: Inventario.

2.2.5. Proveedores

Los proveedores se comportan de 3 formas diferentes:

- i) inmediato.
 - ii) por múltiplo de cantidad fija.
 - iii) por encargo.
- i) Al hacer un pedido al proveedor inmediato de n unidades, entrega las n unidades en forma instantánea. Las n unidades asociadas tienen un tiempo de vencimiento asociado (Figura 8).
 - ii) El proveedor que funciona por múltiplo de cantidad fija, sólo entrega pedidos múltiplos de un número entero prefijado, digamos r (Figura 9). Es decir, que al hacer un pedido de n productos, pueden pasar 2 cosas:
 - 1) El número pedido n equivale a un múltiplo entero de r , $n = kr$, para algún $k \geq 0$, en cuyo caso se entregan exactamente n unidades.
 - 2) El número pedido n es mayor a un múltiplo entero de r , es decir, $n > kr$ y además $n < (k + 1)r$, para algún $k \geq 0$, en cuyo caso se entregan $(k + 1)r$ unidades.

Los productos de este proveedor no tienen vencimiento.

- iii) El tercer proveedor (Figura 10) funciona por encargo para el mes siguiente. Es decir, todo pedido será servido en el siguiente mes. En ese momento, además, se realizará el nuevo pedido.

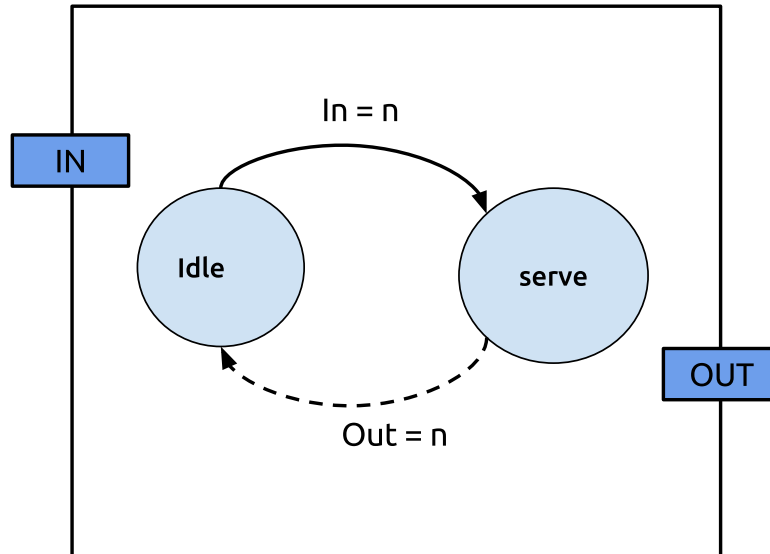


Figura 8: Proveedor Inmediato.

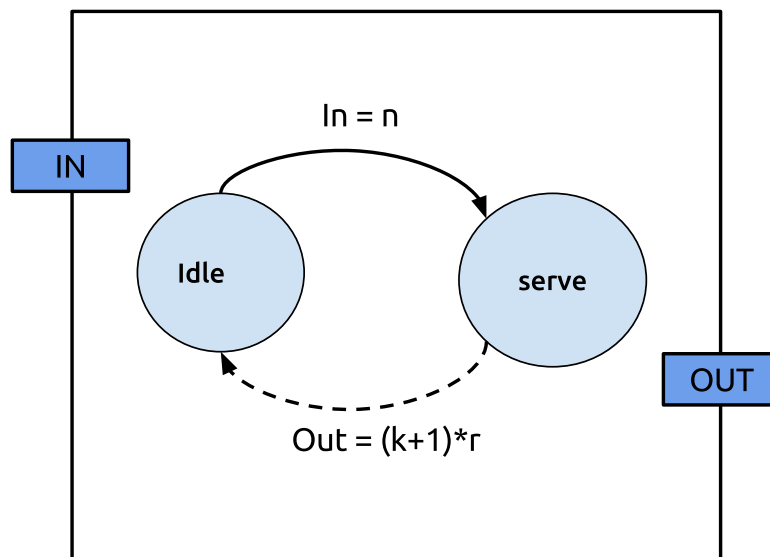


Figura 9: Proveedor por múltiplo de cantidad fija.

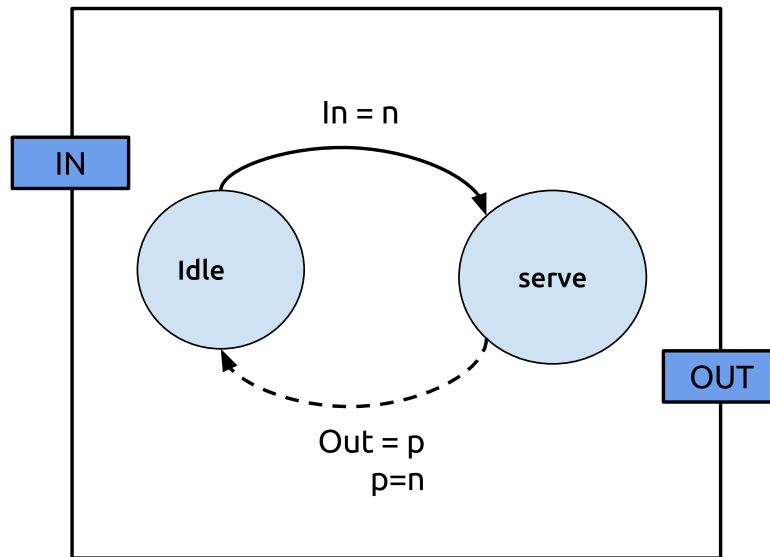


Figura 10: Proveedor por encargo.

3. Descripción Formal

3.1. *Top Model*

El diagrama de bloques correspondiente a cada uno de los atómicos descritos en la especificación funcional se puede ver en la figura 11.

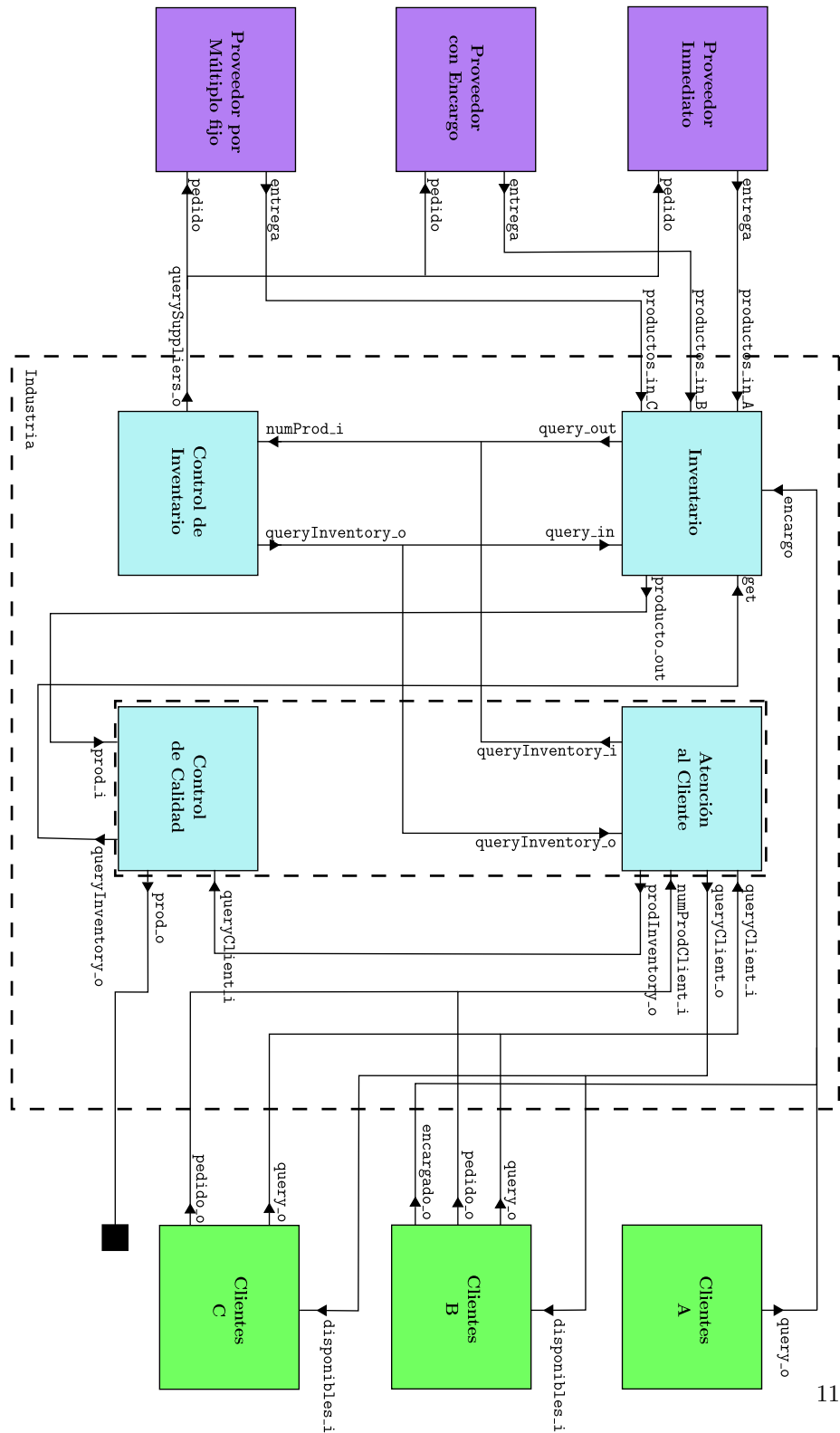


Figura 11: Interconexión de bloques en el top model.

3.2. Tipo de Cliente A

El modelo atómico correspondiente al bloque del cliente tipo A (CTA) queda definido de la siguiente manera:

$$\begin{aligned}
 CTA &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{QUERY\} \\
 X &= \{\emptyset\} \\
 Y &= \{query_o\} \in \mathbb{N}_0^+ \\
 \delta_{int} &: \delta_{int}(QUERY) = QUERY \\
 \delta_{ext} &: \emptyset \\
 \lambda &: \lambda(QUERY) = \{query_o = N\} \\
 t_A(QUERY) &= EXP_VAR
 \end{aligned}$$

En las figuras 12 y 13 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.

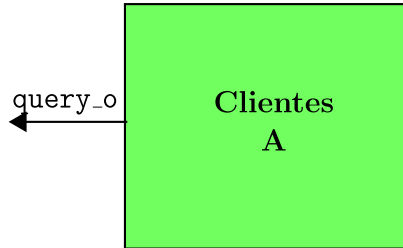


Figura 12: Entradas y salidas del modelo del cliente tipo A.

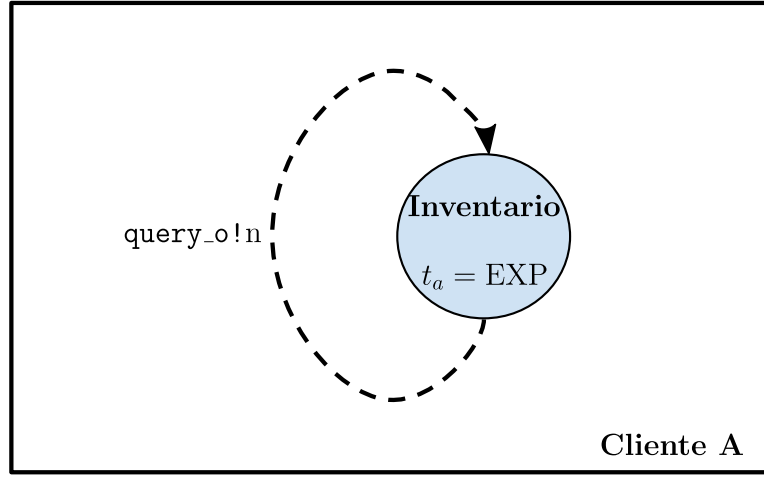


Figura 13: Diagrama de estados para el modelo del cliente tipo A.

3.3. Tipo de Cliente B

El modelo atómico correspondiente al bloque del cliente tipo B (CTB) queda definido de la siguiente manera:

$$\begin{aligned}
 CTB &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{IDLE, QUERY, CALC\} \\
 X &= \{disponibles_i\} \in \mathbb{N}_0^+ \\
 Y &= \{query_o, pedido_o\} \in \mathbb{N}_0^+ \\
 \delta_{int} : \delta_{int}(S) &\begin{cases} \delta_{int}(IDLE) = QUERY \\ \delta_{int}(CALC) = IDLE \end{cases} \\
 \delta_{ext} : \delta_{ext}(S, X) &\begin{cases} Si \ S = QUERY \wedge disponibles_i = P \\ \rightarrow \delta_{ext}(QUERY, disponibles_i) = CALC \end{cases} \\
 \lambda : \lambda(S) &\begin{cases} \lambda(IDLE) = \{query_o = N\} \\ \lambda(CALC) = \{pedido_o = P\} \end{cases} \\
 t_A(S) &= \begin{cases} 0 \ si \ S = \{CALC\} \\ \infty \ si \ S = \{QUERY\} \\ EXP_VAR \ si \ S = \{IDLE\} \end{cases}
 \end{aligned}$$

En las figuras 14 y 15 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.

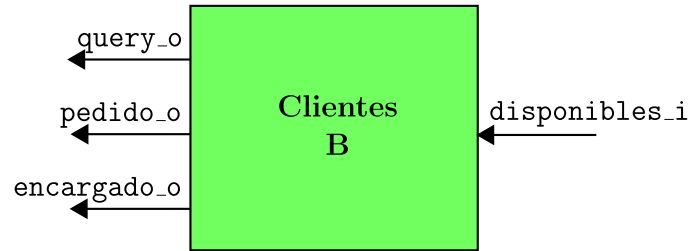


Figura 14: Entradas y salidas del modelo del cliente tipo B.

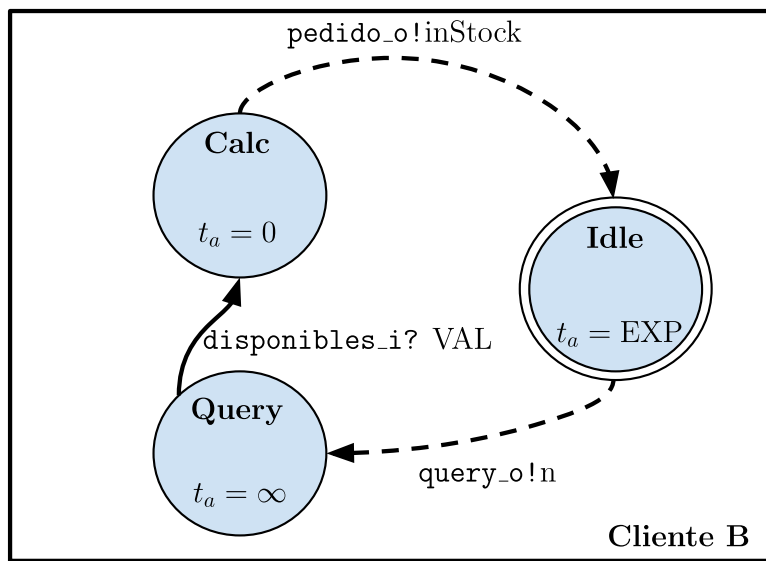


Figura 15: Diagrama de estados para el modelo del cliente tipo B.

3.4. Tipo de Cliente C

El modelo atómico correspondiente al bloque del cliente tipo C (CTC) queda definido de la siguiente manera:

$$\begin{aligned}
 CTC &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{IDLE, QUERY, ACCEPT, DECLINE\} \\
 X &= \{disponibles_i\} \in \mathbb{N}_0^+ \\
 Y &= \{query_o, pedido_o\} \in \mathbb{N}_0^+ \\
 \delta_{int} : \delta_{int}(S) &\begin{cases} \delta_{int}(IDLE) = QUERY \\ \delta_{int}(DECLINE) = IDLE \\ \delta_{int}(ACCEPT) = IDLE \end{cases} \\
 \delta_{ext} : \delta_{ext}(S, X) &\begin{cases} Si\ S = QUERY \wedge disponibles_i = N \\ \quad \rightarrow \delta_{ext}(QUERY, disponibles_i) = ACCEPT \\ Si\ S = QUERY \wedge disponibles_i < N \\ \quad \rightarrow \delta_{ext}(QUERY, disponibles_i) = DECLINE \end{cases} \\
 \lambda : \lambda(S) &\begin{cases} \lambda(IDLE) = \{query_o = N\} \\ \lambda(DECLINE) = \{pedido_o = 0\} \\ \lambda(ACCEPT) = \{pedido_o = N\} \end{cases} \\
 t_A(S) &= \begin{cases} 0\ si\ S = \{ACCEPT, DECLINE\} \\ \infty\ si\ S = \{QUERY\} \\ EXP_VAR\ si\ S = \{IDLE\} \end{cases}
 \end{aligned}$$

En las figuras 16 y 17 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.

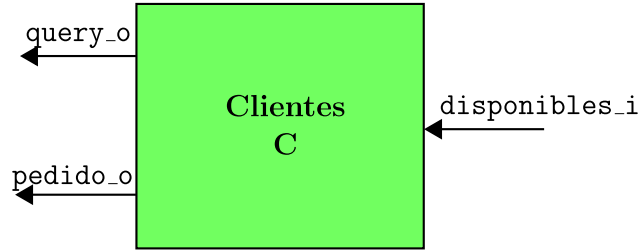


Figura 16: Entradas y salidas del modelo del cliente tipo C.

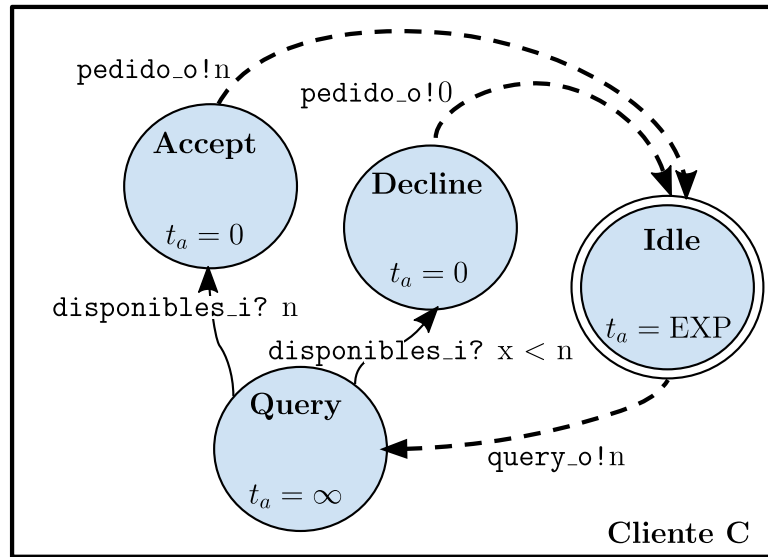


Figura 17: Diagrama de estados para el modelo del cliente tipo C.

3.5. Control de inventario

El modelo atómico correspondiente al bloque de control de inventario (CI) queda definido de la siguiente manera:

$$\begin{aligned}
 CI &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{WAITING, QUERY, CALC\} \\
 X &= \{numProd_i\} \in \mathbb{N}_0^+ \\
 Y &= \{queryInventory_o, querySuppliers_o\} \in \mathbb{N}_0^+ \\
 \delta_{int} : \delta_{int}(S) &\begin{cases} \delta_{int}(WAITING) = QUERY \\ \delta_{int}(CALC) = WAITING \end{cases} \\
 \delta_{ext} : \delta_{ext}(S, numProd_i) &\begin{cases} Si S = QUERY \wedge numProd_i = N \\ \rightarrow \delta_{ext}(S, numProd_i) = CALC \end{cases} \\
 \lambda : \lambda(S) &\begin{cases} \lambda(WAITING) = \{queryInventory_o = 1\} \\ \lambda(CALC) = \{querySuppliers_o = \begin{cases} S - N & si \ numProd_i \\ 0 & \forall \ numProd_i \end{cases} < s \ \} \end{cases} \\
 t_A(S) &= \begin{cases} 1 & si \ S = WAITING \\ 0 & si \ S = CALC \\ \infty & si \ S = QUERY \end{cases}
 \end{aligned}$$

En las figuras 18 y 19 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.



Figura 18: Entradas y salidas del modelo control de inventario

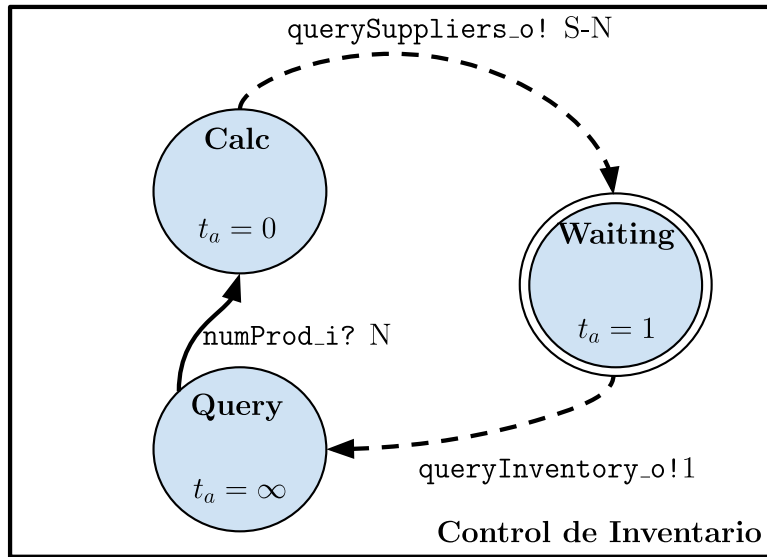


Figura 19: Diagrama de estados para el modelo de control de inventario.

3.6. Control de Calidad

El modelo atómico correspondiente al bloque de control de calidad (CC) queda definido de la siguiente manera:

$$\begin{aligned}
 CC &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{WAITING, QUERY, INV_WAIT, CHECK, SEND\} \\
 X &= \{prod_i \in \mathbb{R}^+, queryClient_i \in \mathbb{N}_0^+\} \\
 Y &= \{prod_o \in \mathbb{R}^+, queryInventory_o \in \mathbb{N}_0^+\} \\
 \delta_{int} : \delta_{int}(S) &= \begin{cases} \delta_{int}(QUERY) = INV_WAIT \\ \delta_{int}(CHECK) = \begin{cases} QUERY \\ si (numPassProd < \\ numClientQuery) \wedge \\ (\neg invEmpty) \\ SEND \forall otra condicion. \end{cases} \\ \delta_{int}(SEND) = WAITING \end{cases} \\
 \delta_{ext} : \delta_{ext}(S, X) &= \begin{cases} Si S = WAITING \wedge queryClient_i = N \\ \rightarrow \delta_{ext}(QUERY, queryClient_i) = QUERY \\ Si S = CHECK \wedge size(prod_i) = P \\ \rightarrow \delta_{ext}(CHECK, prod_i) = SEND \end{cases} \\
 \lambda : \lambda(S) &= \begin{cases} \lambda(QUERY) = \{queryInventory_o = 1\} \\ \lambda(CHECK) = \{\emptyset\} \\ \lambda(SEND) = \{prod_o = K\} \end{cases} \\
 t_A(S) &= \begin{cases} 0 si S = \{QUERY, CHECK, SEND\} \\ \infty si S = \{WAITING, INV_WAIT\} \end{cases}
 \end{aligned}$$

Donde:

$numPassProd$ = es el número de productos que pasan el control de calidad (i.e. su fecha de expiración es mayor al tiempo en que se produce el control).

$numClientQuery$ = es el número de productos que solicitó el cliente.

$invEmpty$ = es un Flag que indica si el inventario está vacío (i.e. devolvió un producto con fecha de expiración 0,0)¹

En las figuras 20 y 21 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.

¹el símbolo \neg refiere a la condición lógica de negación

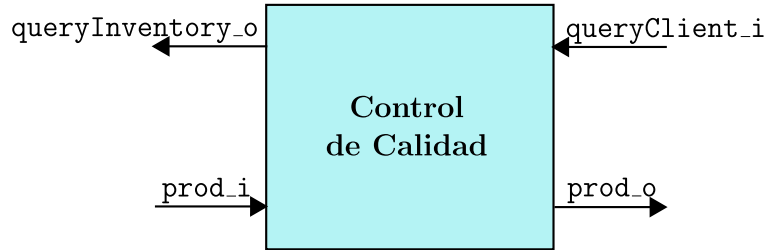


Figura 20: Entradas y salidas del modelo control de calidad.

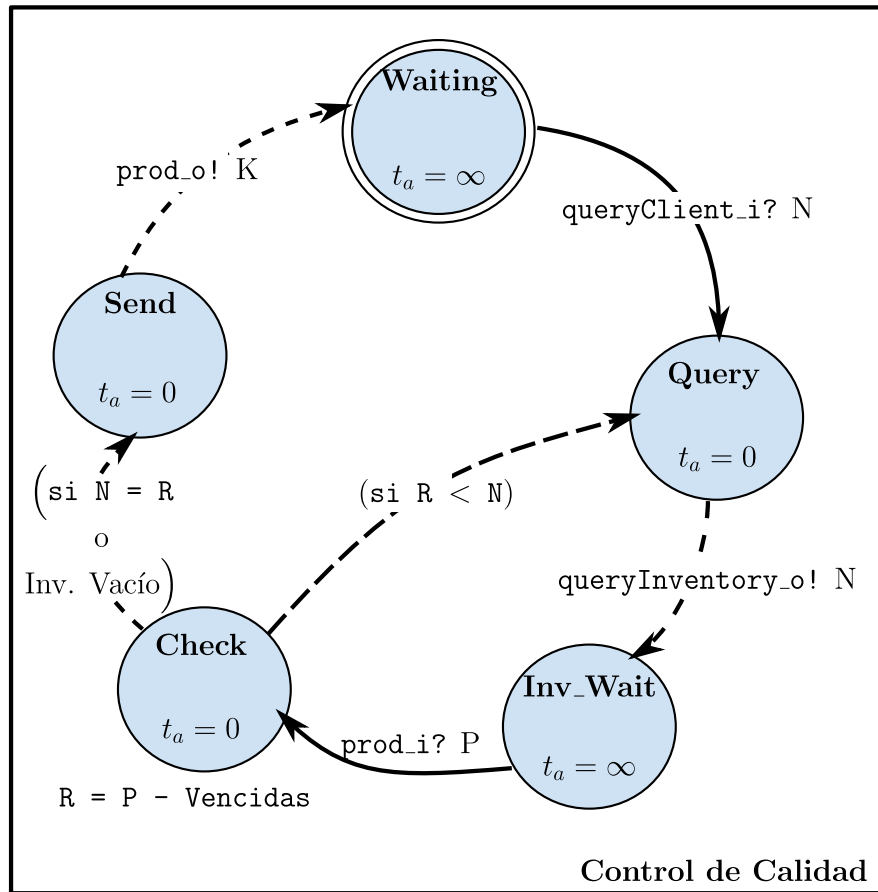


Figura 21: Diagrama de estados para el modelo de control de calidad.

3.7. Atención al cliente

El modelo atómico correspondiente al bloque de atención al cliente (AC) queda definido de la siguiente manera:

$$\begin{aligned}
 AC &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{WAITING, QUERY, INV_WAIT, \\
 &\quad CLI_RPLY, CLI_WAIT, INV_GET\} \\
 X &= \{numProdClient_i, queryClient_i, queryInventory_i\} \in \mathbb{N}_0^+ \\
 Y &= \{prodInventory_o, queryInventory_o, queryClient_o\} \in \mathbb{N}_0^+ \\
 \delta_{int} : \delta_{int}(S) &\begin{cases} \delta_{int}(QUERY) = INV_WAIT \\ \delta_{int}(CLI_RPLY) = CLI_WAIT \\ \delta_{int}(INV_GET) = WAITING \end{cases} \\
 \delta_{ext} : \delta_{ext}(S, X) &\begin{cases} Si S = WAITING \wedge queryClient_i = N \\ \quad \rightarrow \delta_{ext}(WAITING, queryClient_i) = QUERY \\ Si S = INV_WAIT \wedge queryInventory_i = P \\ \quad \rightarrow \delta_{ext}(INV_WAIT, queryInventory_i) = CLI_RPLY \\ Si S = CLI_WAIT \wedge numProdClient_i = K \\ \quad \rightarrow \delta_{ext}(CLI_WAIT, numProdClient_i) = INV_GET \end{cases} \\
 \lambda : \lambda(S) &\begin{cases} \lambda(QUERY) = \{queryInventory_o = N\} \\ \lambda(CLI_RPLY) = \{queryClient_o = P\} \\ \lambda(INV_GET) = \{prodInventory_o = K\} \end{cases} \\
 t_A(S) &= \begin{cases} 0 & si S = \{QUERY, CLI_RPLY, INV_GET\} \\ \infty & si S = \{WAITING, INV_WAIT, CLI_WAIT\} \end{cases}
 \end{aligned}$$

En las figuras 22 y 23 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.



Figura 22: Entradas y salidas del modelo atención al cliente.

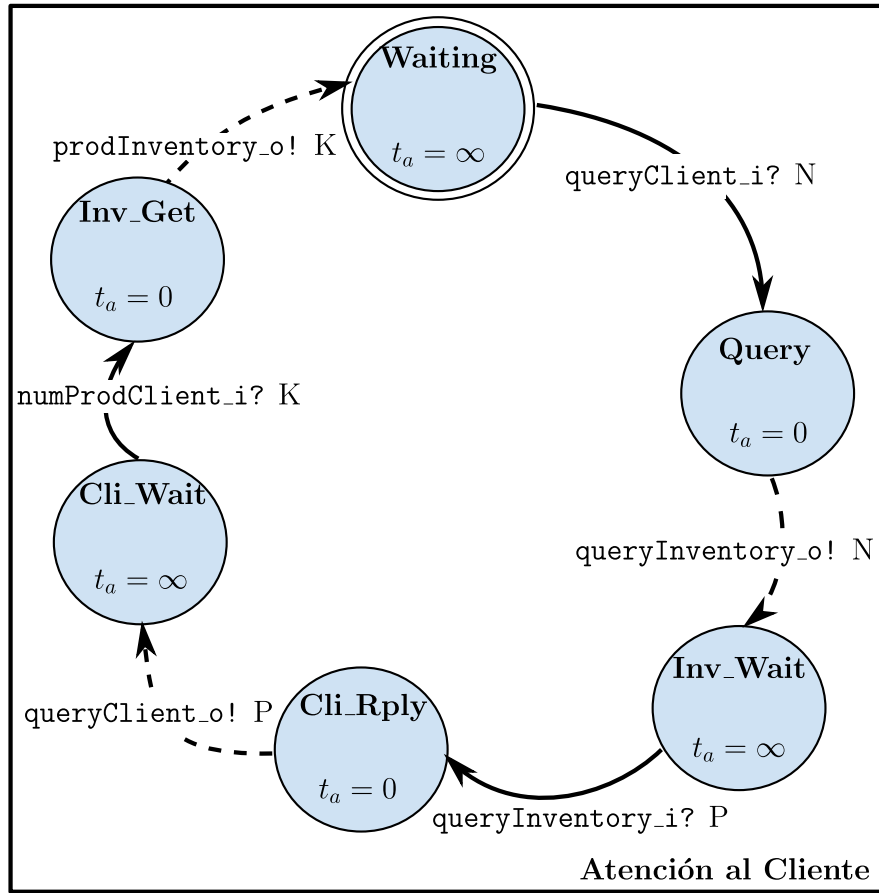


Figura 23: Diagrama de estados para el modelo de atención al cliente.

3.8. Inventario

El modelo atómico correspondiente al bloque de inventario (I) queda definido de la siguiente manera:

$$\begin{aligned}
 I &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{IDLE, QUERY, ENCARGO, PUSH, GET\} \\
 X &= \{query_in, encargo, producto_in_A, \\
 &\quad producto_in_B, producto_in_C, encargo, get\} \in \mathbb{N}_0^+ \\
 Y &= \{producto_out, query_out\} \in \mathbb{N}_0^+ \\
 \delta_{int} : \delta_{int}(S) &\begin{cases} \delta_{int}(QUERY) = IDLE \\ \delta_{int}(ENCARGO) = IDLE \\ \delta_{int}(PUSH) = IDLE \\ \delta_{int}(GET) = IDLE \end{cases} \\
 \delta_{ext} : \delta_{ext}(S, X) &\begin{cases} Si S = IDLE \wedge \begin{cases} product_in_A = Na \\ product_in_B = Nb \\ product_in_C = Nc \end{cases} \\
 \quad \wedge size(col_i) == 0 \exists i \in \{A, B, C\} \\
 \rightarrow \delta_{ext}(IDLE, \{product_in_A, product_in_B \\
 \quad product_in_C\}) = QUERY \\
 Si S = IDLE \wedge encargo = Ne \\
 \rightarrow \delta_{ext}(IDLE, encargo) = ENCARGO \\
 Si S = IDLE \wedge \begin{cases} product_in_A = Na \\ product_in_B = Nb \\ product_in_C = Nc \end{cases} \\
 \quad \wedge size(col_i)! = 0 \forall i \in \{A, B, C\} \\
 \rightarrow \delta_{ext}(IDLE, \{product_in_A, product_in_B \\
 \quad product_in_C\}) = PUSH \\
 Si S = IDLE \wedge get = Ng \\
 \rightarrow \delta_{ext}(IDLE, get) = GET \end{cases} \\
 \lambda : \lambda(S) &\begin{cases} \lambda(QUERY) = \{query_out = size(col) - encargos_q\} \\ \lambda(ENCARGO) = \{\emptyset\} \\ \lambda(PUSH) = \{producto_out = \min(get_q, size(col))\} \\ \lambda(GET) = \{producto_out = \min(encargos_q, size(col))\} \end{cases} \\
 t_A(S) &= \begin{cases} 0 \text{ si } S = \{QUERY, ENCARGO, PUSH, GET\} \\ \infty \text{ si } S = \{IDLE\} \end{cases}
 \end{aligned}$$

Donde:

cola = es la cola de productos terminados.

encargo_q = es la cantidad de productos encargados por clientes que no estaban disponibles para entrega inmediata.

get_q = es la cantidad de productos pedidos después de consultar la cantidad de productos disponibles.

La negociación entre los estados IDLE y GET es una *negociación segura*, es decir, se produce después de consultar al inventario la cantidad de productos disponibles. En caso que entre esta consulta y la compra ya no estén disponibles, porque en el medio se produjo una venta o porque alguno de los productos estaba vencido, se indica al control de calidad devolviendo un 0. Esos productos indicados con 0 no se reintegran a futuro.

En las figuras 24 y 25 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.



Figura 24: Entradas y salidas del modelo inventario.

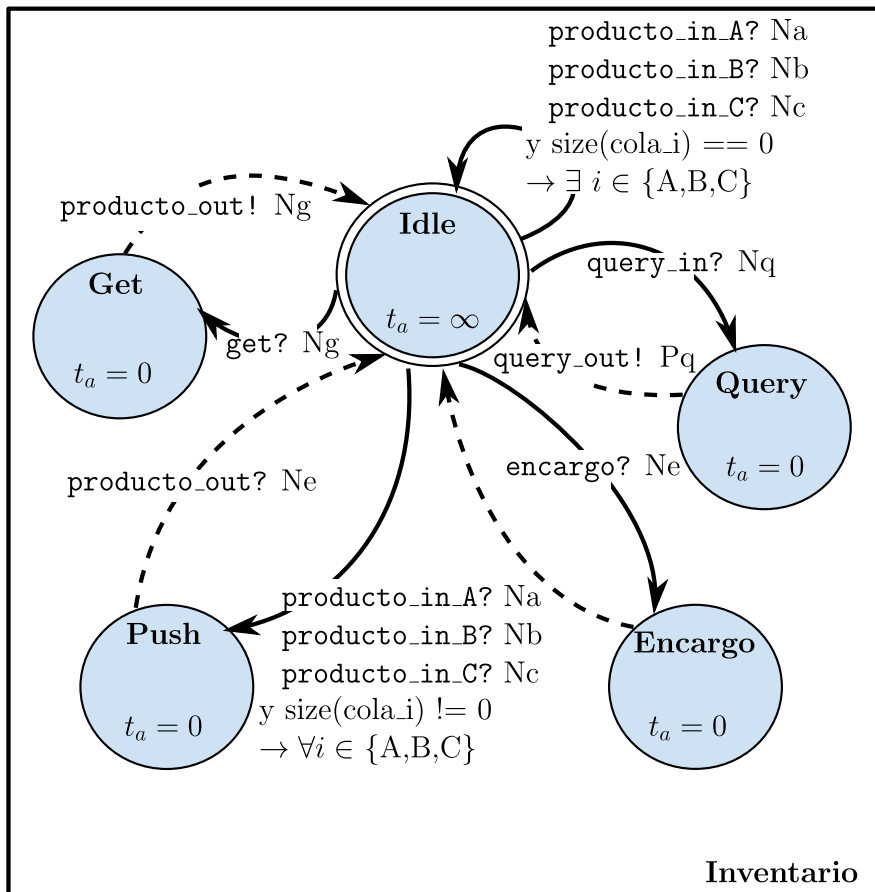


Figura 25: Diagrama de estados para el modelo de inventario.

3.9. Industria

La industria se instancia como un modelo acoplado definido de la siguiente manera:

$$\begin{aligned}
 IND &= \langle X, Y, M_i, EIC, EOC, IC, Select \rangle \\
 X &= \{prod_i_A, prod_i_B, prod_i_C, queryClient_i, \\
 &\quad encargo_i, numProdClient_i\} \in \mathbb{N}_0^+ \\
 Y &= \{prod_o\} \in \mathbb{R}^+ \cup \{queryClient_o, querySuppliers_o\} \in \mathbb{N}_0^+ \\
 M &= \{I, CI, CC, AC\} \\
 EIC &= \{(prod_i_A, I.products_in_A), (prod_i_B, I.products_in_B), \\
 &\quad (prod_i_C, I.products_in_C), (queryClient_i, AC.queryClient_i), \\
 &\quad (encargo_i, I.encargo), (numProdClient_i, AC.numProdClient_i)\} \\
 IC &= \{(AC.queryInventory_o, I.query_i), \\
 &\quad (AC.queryInventory_i, I.query_out), \\
 &\quad (CI.queryInventory_o, I.query_i), (CI.numProd_i, I.query_out) \\
 &\quad (CC.prod_i, I.product_o), (CC.queryInventory_o, I.get), \\
 &\quad (AC.prodInventory_o, queryClient_i)\} \\
 EOC &= \{(prod_o, CC.prod_o), (queryClient_o, CI.prod_o), \\
 &\quad (querySuppliers_o,)\} \\
 Select &= \{\{I, CI, AC, CC\} = CI \\
 &\quad \{I, AC, CC\} = I \\
 &\quad \{AC, CC\} = AC\}
 \end{aligned}$$

3.10. Proveedor Inmediato

El modelo atómico correspondiente al bloque del proveedor inmediato (PI) queda definido de la siguiente manera:

$$\begin{aligned}
 PI &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{IDLE, SERVE\} \\
 X &= \{pedido\} \in \mathbb{N}_0^+ \\
 Y &= \{entrega\} \in \mathbb{N}_0^+ \\
 \delta_{int} &: \delta_{int}(SERVE) = IDLE \\
 \delta_{ext} &: \delta_{ext}(S, X) \begin{cases} Si S = IDLE \wedge pedido = N \\ \rightarrow \delta_{ext}(IDLE, pedido) = SERVE \end{cases} \\
 \lambda &: \lambda(SERVE) = \{entrega = N\} \\
 t_A(S) &= \begin{cases} 0 & si S = \{SERVE\} \\ \infty & si S = \{IDLE\} \end{cases}
 \end{aligned}$$

En las figuras 26 y 27 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.

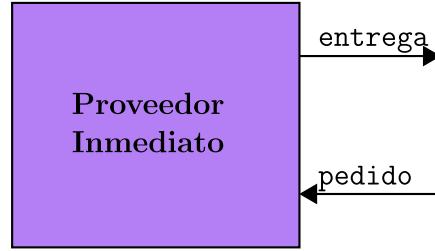


Figura 26: Entradas y salidas del modelo del proveedor inmediato

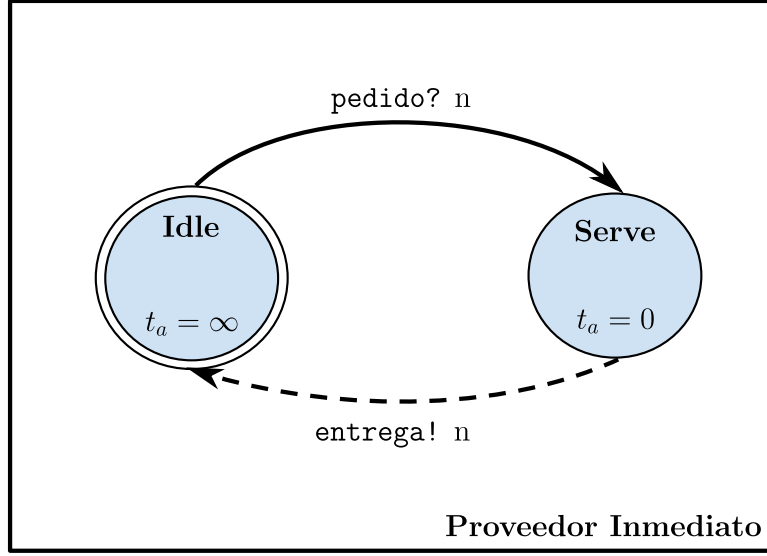


Figura 27: Diagrama de estados para el modelo del proveedor inmediato.

3.11. Proveedor por múltiplo de cantidad fija

El modelo atómico correspondiente al bloque del proveedor por múltiplo de cantidad fija (PF) queda definido de la siguiente manera:

$$\begin{aligned}
 PF &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{IDLE, SERVE\} \\
 X &= \{pedido\} \in \mathbb{N}_0^+ \\
 Y &= \{entrega\} \in \mathbb{N}_0^+ \\
 \delta_{int} &: \delta_{int}(SERVE) = IDLE \\
 \delta_{ext} &: \delta_{ext}(S, X) \begin{cases} Si \ S = IDLE \wedge pedido = N \\ \rightarrow \delta_{ext}(IDLE, pedido) = SERVE \end{cases} \\
 \lambda &: \lambda(SERVE) = \{entrega = \begin{cases} N \ Si \ N \% numProdPkg = 0 \\ \lfloor N / numProdPkg \rfloor \\ + N \% numProdPkg \ \forall \ N \end{cases} \} \\
 t_A(S) &= \begin{cases} 0 \ si \ S = \{SERVE\} \\ \infty \ si \ S = \{IDLE\} \end{cases}
 \end{aligned}$$

Donde:

$numProdPkg$ = es el número de productos por paquete cerrado que comercializa este proveedor.

En las figuras 28 y 29 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.

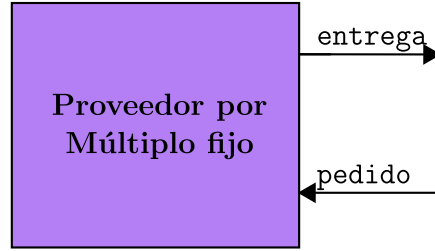


Figura 28: Entradas y salidas del modelo del proveedor por múltiplo de cantidad fija

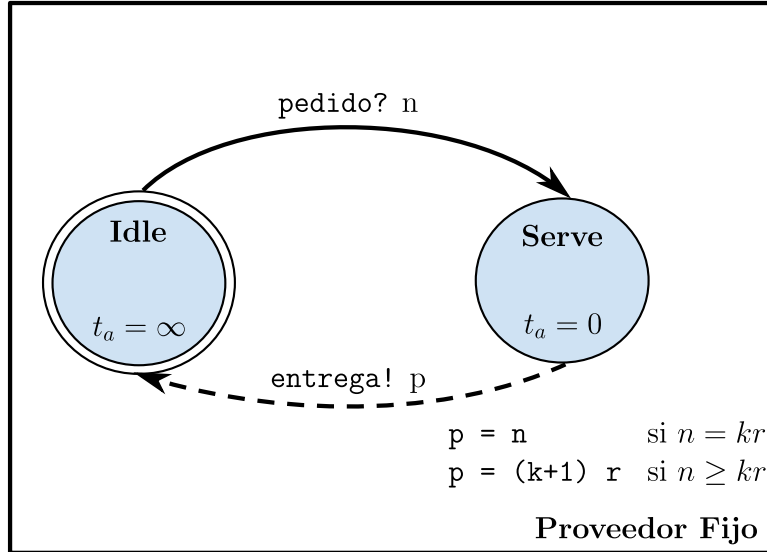


Figura 29: Diagrama de estados para el modelo del proveedor por múltiplo de cantidad fija.

3.12. Proveedor por Encargo

El modelo atómico correspondiente al bloque del proveedor por encargo (PE) queda definido de la siguiente manera:

$$\begin{aligned}
 PE &= \langle S, X, Y, \delta_{int}, \delta_{ext}, \lambda, t_A \rangle \\
 S &= \{IDLE, SERVE\} \\
 X &= \{pedido\} \in \mathbb{N}_0^+ \\
 Y &= \{entrega\} \in \mathbb{N}_0^+ \\
 \delta_{int} &: \delta_{int}(SERVE) = IDLE \\
 \delta_{ext} &: \delta_{ext}(S, X) \begin{cases} \text{Si } S = IDLE \wedge pedido = N \\ \rightarrow \delta_{ext}(IDLE, pedido) = SERVE \end{cases} \\
 \lambda &: \lambda(SERVE) = \{entrega = N\} \\
 t_A(S) &= \begin{cases} 1 \text{ si } S = \{SERVE\} \\ \infty \text{ si } S = \{IDLE\} \end{cases}
 \end{aligned}$$

En las figuras 30 y 31 se pueden observar los puertos de entrada/salida y el diagrama de estados respectivamente.

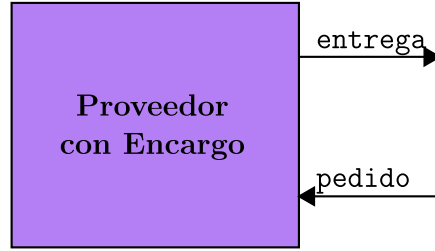


Figura 30: Entradas y salidas del modelo del proveedor por encargo

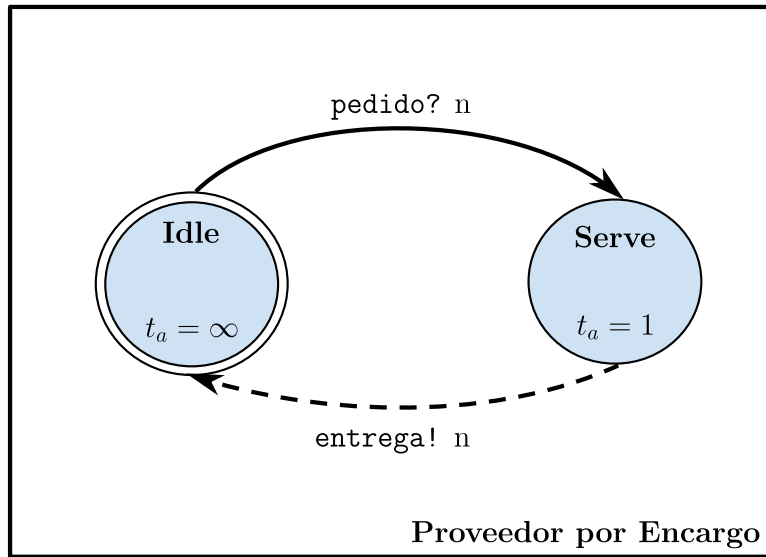


Figura 31: Diagrama de estados para el modelo del proveedor por encargo.

4. Modelado y simulación

4.1. Pruebas parciales

4.1.1. Demandas de los Clientes

En la Figura 32 y Figura 33 se observan los histogramas que representan la tasa de arribos de pedidos de clientes y la cantidad de unidades pedidas. Se puede notar que la distribución de los tiempos de arribos sigue una distribución de probabilidad exponencial y el valor medio del tiempo entre arribos es de alrededor de 0,12. La distribución de probabilidad de la cantidad de unidades pedidas corresponde a la mencionada en la sección 2.2.1.

En ambas figuras el tiempo se mide en segundos, aunque cada segundo representa un mes.

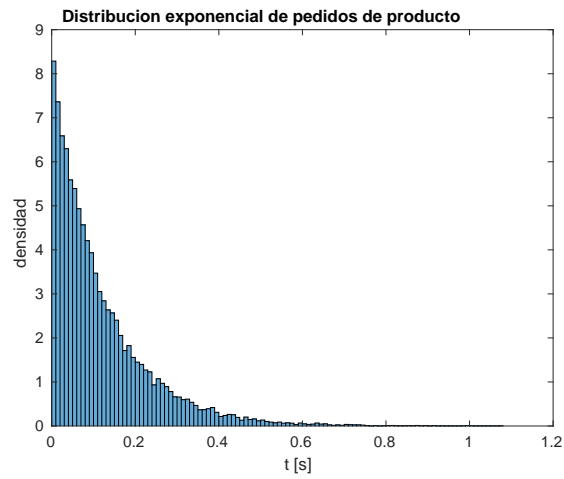


Figura 32: Función de densidad de probabilidad del tiempo entre pedidos de clientes.

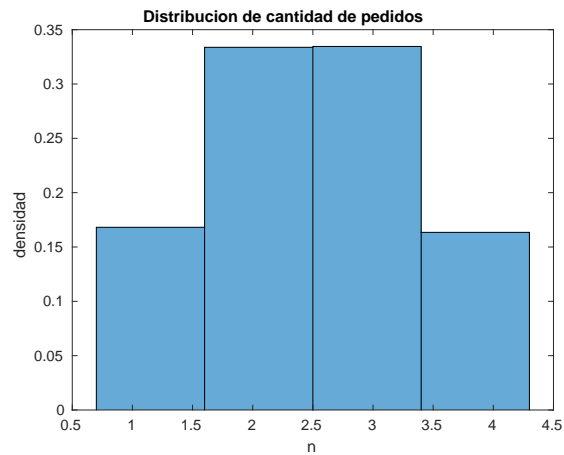


Figura 33: Función de densidad de probabilidad del número de unidades pedidas n .

4.1.2. Cliente A

Para la prueba del bloque de clientes tipo A no se necesitan eventos de entrada porque este tipo de cliente encarga productos sin importar si están disponibles. La salida de la prueba es la siguiente:

```
00:00:00:073:0.401741 Client A — encargo: 1
00:00:00:101:0.768183 Client A — encargo: 4
00:00:00:322:0.663614 Client A — encargo: 4
00:00:00:385:0.395575 Client A — encargo: 2
00:00:00:585:0.052725 Client A — encargo: 2
00:00:00:604:0.393726 Client A — encargo: 3
```

Con trazas como ésta se obtuvieron la Figura 32 y Figura 33.

4.1.3. Cliente B

Para la prueba del bloque de clientes tipo B se propuso el siguiente estímulo:

```
00:00:05:00 in_port 1
00:00:30:00 in_port 0
00:00:45:00 in_port 10
00:01:20:00 in_port 1
```

Dada la política de pedidos, se busca probar las condiciones:

- La cantidad de productos disponibles en el inventario es menor al valor consultado por el cliente, por lo tanto se piden las unidades disponibles y se encargan las restantes.
- La cantidad de productos en el inventario es mayor o igual al valor consultado por el cliente, por lo tanto piden todas las unidades por las que se consultó.

La salida de la prueba es la siguiente:

```
00:00:00:073:0.401741 Client B — pregunta por: 1 productos
00:00:05:000:0 Client B — le dicen que hay: 1 productos
00:00:05:000:0 Client B — pide: 1
00:00:05:119:0.451668 Client B — pregunta por: 4 productos
00:00:30:000:0 Client B — le dicen que hay: 0 productos
00:00:30:000:0 Client B — pide: 0
00:00:30:000:0 Client B — encarga: 4
00:00:30:009:0.329481 Client B — pregunta por: 4 productos
00:00:45:000:0 Client B — le dicen que hay: 10 productos
00:00:45:000:0 Client B — pide: 4
00:00:45:003:0.96502 Client B — pregunta por: 2 productos
00:01:20:000:0 Client B — le dicen que hay: 1 productos
00:01:20:000:0 Client B — pide: 1
00:01:20:000:0 Client B — encarga: 1
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.3, ya que si el número de unidades disponibles para entrega inmediata es menor a la cantidad consultada la compra se realiza igual.

4.1.4. Cliente C

Para la prueba del bloque de clientes de tipo C se propuso el siguiente estímulo:

```
00:00:05:00 in_port 1
00:00:30:00 in_port 0
00:00:45:00 in_port 10
00:01:20:00 in_port 1
```

Dada la política de pedidos, se busca probar las condiciones:

- La cantidad de productos disponibles en el inventario es menor al valor consultado por el cliente, por lo tanto se rechaza la compra.
- La cantidad de productos en el inventario es mayor o igual al valor consultado por el cliente, por lo tanto piden todas las unidades por las que se consultó.

La salida de la prueba es la siguiente:

```
00:00:00:073:0.401741 Client C – pregunta por: 1 productos
00:00:05:000:0 Client C – Peticion aceptada
00:00:05:000:0 Client C – pide: 1 productos
00:00:05:119:0.451668 Client C – pregunta por: 4 productos
00:00:30:000:0 Client C – Peticion rechazada
00:00:30:000:0 Client C – No pide nada
00:00:30:009:0.329481 Client C – pregunta por: 4 productos
00:00:45:000:0 Client C – Peticion aceptada
00:00:45:000:0 Client C – pide: 4 productos
00:00:45:003:0.96502 Client C – pregunta por: 2 productos
00:01:20:000:0 Client C – Peticion rechazada
00:01:20:000:0 Client C – No pide nada
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.4, ya que si el número de unidades disponibles para entrega inmediata es menor a la cantidad consultada entonces la compra no se realiza.

4.1.5. Control de Inventario

Para la prueba del bloque de control de inventario se propuso el siguiente estímulo:

```
00:00:01:10 numProd_i 8
00:00:02:10 numProd_i 190
00:00:03:10 numProd_i 49
```

Dada la política de pedidos, se busca probar las condiciones:

- La cantidad de productos en el inventario es menor al valor requerido por la política de reposición, por lo tanto se debe pedir materiales a los proveedores.
- La cantidad de productos en el inventario es menor al máximo pero no alcanza el mínimo de la política de reposición por lo que no se deben pedir materiales.

La salida de la prueba es la siguiente:

```
00:00:01:000:0 Control Inventario - Consulta a Inventario
00:00:01:010:0 Control Inventario - Inventario responde: 8
00:00:01:010:0 Control Inventario - Pedido a proveedores: 92
00:00:02:010:0 Control Inventario - Consulta a Inventario
00:00:02:010:0 Control Inventario - Inventario responde: 190
00:00:02:010:0 Control Inventario - Pedido a proveedores: 0
00:00:03:010:0 Control Inventario - Consulta a Inventario
00:00:03:010:0 Control Inventario - Inventario responde: 49
00:00:03:010:0 Control Inventario - Pedido a proveedores: 51
00:00:04:010:0 Control Inventario - Consulta a Inventario
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.5, ya que si el número de unidades se encuentra por debajo del umbral de política, el bloque pide, y de lo contrario no.

4.1.6. Control de Calidad

Para la prueba del bloque de control de calidad se propuso el siguiente estímulo:

```
00:00:01:00 queryClient_i 2
00:00:01:10 prod_i [1.2,1.3]
00:00:02:00 queryClient_i 3
00:00:02:10 prod_i [1.5,2.2,2.3]
00:00:02:20 prod_i [2.3]
00:00:03:00 queryClient_i 3
00:00:03:10 prod_i [3.2,3.3,0]
```

Se busca probar las condiciones:

- Un cliente compra N productos y el inventario devuelve N productos correctos. Por lo tanto se deberían enviar todos.
- Un cliente compra N productos y el inventario devuelve $N - 1$ productos correctos y uno vencido. Se debe descartar ese producto y pedir uno nuevo al inventario. El inventario devuelve un producto válido. Se deberían enviar los N productos válidos.

- Un cliente compra N productos pero el inventario dispone sólo de $N - 1$ y devuelve 0 en la última posición. El control de calidad debe interpretar la salida y despachar sólo los dos productos correctos.

La salida de la prueba es la siguiente:

```
00:00:01:000:0 Control Calidad - le piden: 2 productos
00:00:01:000:0 Control Calidad - pide: 2 productos
00:00:01:010:0 Control Calidad - recibe 2 productos
00:00:01:010:0 Control Calidad - se queda con 2 productos
00:00:01:010:0 Control Calidad - envia: 2 productos
00:00:02:000:0 Control Calidad - le piden: 3 productos
00:00:02:000:0 Control Calidad - pide: 3 productos
00:00:02:010:0 Control Calidad - recibe 3 productos
00:00:02:010:0 Control Calidad - se queda con 2 productos
00:00:02:010:0 Control Calidad - pide: 1 productos
00:00:02:020:0 Control Calidad - recibe 1 productos
00:00:02:020:0 Control Calidad - se queda con 3 productos
00:00:02:020:0 Control Calidad - envia: 3 productos
00:00:03:000:0 Control Calidad - le piden: 3 productos
00:00:03:000:0 Control Calidad - pide: 3 productos
00:00:03:010:0 Control Calidad - recibe 3 productos
00:00:03:010:0 Control Calidad - se queda con 2 productos
00:00:03:010:0 Control Calidad - el inventario esta vacio
00:00:03:010:0 Control Calidad - envia: 2 productos
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.6.

4.1.7. Atención al cliente

Para la prueba del bloque de atención al cliente se propuso el siguiente estímulo:

```
00:00:01:00      queryClient_i 1
00:00:01:10 queryInventory_i 2
00:00:01:15 numProdClient_i 1
00:00:02:00      queryClient_i 3
00:00:02:10 queryInventory_i 2
00:00:02:15 numProdClient_i 2
00:00:03:00      queryClient_i 3
00:00:03:10 queryInventory_i 2
00:00:03:15 numProdClient_i 0
```

Se busca probar las condiciones:

- Un cliente solicita N productos y el inventario devuelve una cantidad $> N$ que se le informa al cliente. Por lo tanto el cliente compra N productos.
- Un cliente solicita N productos. El inventario responde con $P < N$ y el cliente decide comprar P productos.

- Un cliente solicita N productos. El inventario responde con $P < N$ y el cliente decide no comprar ningún producto.

La salida de la prueba es la siguiente:

```
00:00:01:000:0 Atencion al cliente - le piden 1 productos
00:00:01:000:0 Atencion al cliente - pregunta por 1 productos
00:00:01:010:0 Atencion al cliente - le dicen que hay 2 stock
00:00:01:010:0 Atencion al cliente - informa a cliente que hay 2 stock
00:00:01:015:0 Atencion al cliente - cliente pide 1 productos
00:00:01:015:0 Atencion al cliente - pide al inventario 1 productos
00:00:02:000:0 Atencion al cliente - le piden 3 productos
00:00:02:000:0 Atencion al cliente - pregunta por 3 productos
00:00:02:010:0 Atencion al cliente - le dicen que hay 2 stock
00:00:02:010:0 Atencion al cliente - informa a cliente que hay 2 stock
00:00:02:015:0 Atencion al cliente - cliente pide 2 productos
00:00:02:015:0 Atencion al cliente - pide al inventario 2 productos
00:00:03:000:0 Atencion al cliente - le piden 3 productos
00:00:03:000:0 Atencion al cliente - pregunta por 3 productos
00:00:03:010:0 Atencion al cliente - le dicen que hay 2 stock
00:00:03:010:0 Atencion al cliente - informa a cliente que hay 2 stock
00:00:03:015:0 Atencion al cliente - cliente pide 0 productos
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.7

4.1.8. Inventario

Para la prueba del bloque de inventario se propuso el siguiente estímulo:

```
00:00:05:00 query_in 1
00:00:09:00 producto_in_A [1.5,1.2,0.9]
00:00:51:00 producto_in_B [0.9,0.9,0.9]
00:00:54:00 query_in 1
00:00:55:00 producto_in_C [0.9,0.9,0.9]
00:00:58:00 query_in 1
00:00:59:00 get 2
00:00:60:00 encargo 1
00:00:61:00 query_in 1
00:00:62:00 producto_in_C [0.9,0.9,0.9]
00:00:63:00 producto_in_A [1.5,1.2,0.9]
00:00:64:00 producto_in_B [0.9,0.9,0.9]
00:00:65:00 query_in 1
00:00:66:00 get 2
00:00:67:00 get 2
```

Se busca probar las condiciones:

- Un proveedor entrega unidades de un insumo. El inventario lo almacena en la cola de insumo correspondiente. Cuando haya unidades de todos los

insumos se obtendrán productos automáticamente.

- Un cliente encarga Ne productos. El inventario incrementa la cantidad de unidades encargadas E .
- Un cliente consulta la disponibilidad de Nq productos. El inventario responde con $N - E$, siendo N la cantidad de productos disponibles y E la cantidad de productos previamente encargados.
- Un cliente consulta la disponibilidad de Nq productos. El inventario responde con $N \geq Nq$ y el cliente compra $M > Nq$, si están disponibles se le entregan las M unidades y sino se le entregan Nq .

La salida de la prueba es la siguiente:

```
00:00:05:000:0 Inventario - Query out(N - E): 0
00:00:05:000:0 Inventario - Cola(N): 0
00:00:54:000:0 Inventario - Query out(N - E): 0
00:00:54:000:0 Inventario - Cola(N): 0
00:00:55:000:0 Inventario - Entran 3 productos
00:00:55:000:0 Inventario - Hay en cola N:3
INVENTARIO , 00:00:55:000:0 , 3 , 0 , 0 , 0 , 0
00:00:58:000:0 Inventario - Query out(N - E): 3
00:00:58:000:0 Inventario - Cola(N): 3
00:00:59:000:0 Inventario - Entrega: [0.9 , 0.9]
INVENTARIO , 00:00:59:000:0 , 1 , 0 , 0 , 0 , 0
00:01:00:000:0 Inventario - Encargos(E): 1
INVENTARIO , 00:01:00:000:0 , 1 , 0 , 0 , 0 , 1
00:01:01:000:0 Inventario - Query out(N - E): 0
00:01:01:000:0 Inventario - Cola(N): 1
00:01:04:000:0 Inventario - Entran 3 productos
00:01:04:000:0 Inventario - Encargos entregados:1
00:01:04:000:0 Inventario - Encargos restantes:0
00:01:04:000:0 Inventario - Hay en cola N:3
INVENTARIO , 00:01:04:000:0 , 3 , 0 , 0 , 0 , 0
00:01:05:000:0 Inventario - Query out(N - E): 3
00:01:05:000:0 Inventario - Cola(N): 3
00:01:06:000:0 Inventario - Entrega: [0.9 , 0.9]
INVENTARIO , 00:01:06:000:0 , 1 , 0 , 0 , 0 , 0
00:01:07:000:0 Inventario - Entrega: [0.9 , 0]
INVENTARIO , 00:01:07:000:0 , 0 , 0 , 0 , 0 , 0
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.8. Se puede notar que hasta que no haya disponibles unidades de los insumos A, B y C no se pueden producir productos y que en caso contrario se producen instantáneamente. Se puede notar que cuando se solicita una cantidad de productos mayor a la que se había consultado previamente se entregan unidades con fecha de vencimiento 0, que representan que ese producto no está disponible al momento de la compra y no se va a encargar.

4.1.9. Proveedor Inmediato

Para la prueba del bloque proveedor inmediato se propuso el siguiente estímulo:

00:00:05:00 pedido 1
00:00:09:00 pedido 4
00:00:17:00 pedido 3
00:00:50:00 pedido 2

Se busca probar las condiciones:

- El control de inventario solicita N unidades de insumo y el proveedor responde instantáneamente.

La salida de la prueba es la siguiente:

00:00:05:000:0 Proveedor Inmediato – Pedido: 1 productos
00:00:05:000:0 Proveedor Inmediato – Entrega: 1 productos
00:00:09:000:0 Proveedor Inmediato – Pedido: 4 productos
00:00:09:000:0 Proveedor Inmediato – Entrega: 4 productos
00:00:17:000:0 Proveedor Inmediato – Pedido: 3 productos
00:00:17:000:0 Proveedor Inmediato – Entrega: 3 productos
00:00:50:000:0 Proveedor Inmediato – Pedido: 2 productos
00:00:50:000:0 Proveedor Inmediato – Entrega: 2 productos

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.10.

4.1.10. Proveedor por Múltiplo Fijo

Para la prueba del bloque proveedor por múltiplo fijo se propuso el siguiente estímulo:

00:00:05:00 pedido 100
00:00:09:00 pedido 40
00:00:17:00 pedido 30
00:00:50:00 pedido 200

Se busca probar las condiciones:

- El control de inventario solicita N unidades de insumo. El proveedor entrega instantáneamente con una cantidad múltiplo del número de unidades por paquete.

La salida de la prueba es la siguiente:


```
00:00:05:000:0 Proveedor Fijo – Pedido: 100 productos
00:00:05:000:0 Proveedor Fijo – Entrega: 120 productos
00:00:09:000:0 Proveedor Fijo – Pedido: 40 productos
00:00:09:000:0 Proveedor Fijo – Entrega: 60 productos
00:00:17:000:0 Proveedor Fijo – Pedido: 30 productos
00:00:17:000:0 Proveedor Fijo – Entrega: 30 productos
00:00:50:000:0 Proveedor Fijo – Pedido: 200 productos
00:00:50:000:0 Proveedor Fijo – Entrega: 210 productos
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.11. La cantidad de unidades de insumo por paquete se fijó en 30.

4.1.11. Proveedor por Encargo

Para la prueba del bloque proveedor por encargo se propuso el siguiente estímulo:

```
00:00:05:00 pedido 100
00:00:06:00 pedido 40
00:00:07:00 pedido 30
00:00:50:00 pedido 200
```

Se busca probar las condiciones:

- El control de inventario solicita N unidades de insumo. El proveedor entrega la cantidad pedida un mes después.

La salida de la prueba es la siguiente:

```
00:00:05:000:0 Proveedor por Encargo – Pedido: 100 productos
00:00:05:999:0 Proveedor por Encargo – Entrega: 100 productos
00:00:06:000:0 Proveedor por Encargo – Pedido: 40 productos
00:00:06:999:0 Proveedor por Encargo – Entrega: 40 productos
00:00:07:000:0 Proveedor por Encargo – Pedido: 30 productos
00:00:07:999:0 Proveedor por Encargo – Entrega: 30 productos
00:00:50:000:0 Proveedor por Encargo – Pedido: 200 productos
00:00:50:999:0 Proveedor por Encargo – Entrega: 200 productos
```

Donde se observa que se cumplen las condiciones mencionadas en la sección 3.12.

4.2. Pruebas de integración

Una vez que se validaron los bloques con las pruebas realizadas en la Sección 4.1 se procedió a su interconexión como se muestra en la Figura 11.

4.3. Reporte y conclusiones

4.3.1. Reporte de resultados

Para responder a las preguntas de la Sección 2.1 se realizaron una serie de simulaciones para distintas políticas de reposición de inventario. Para cada una se analizaron los costos de reposición y la satisfacción del cliente, siendo esta última proporcional a la inversa de la cantidad de unidades encargadas. Los datos devueltos por la simulación se *parsearon* con un script de python para su presentación en pantalla.

A continuación se muestran los gráficos de evolución de tamaño del inventario, costo de reposición y mantenimiento de unidades en stock, y de cantidad de encargos para un parámetro $n = 50$ y $S = 85$ correspondientes a la política de reposición (Sección 2.2.3). En estos gráficos el tiempo está expresado en segundos, donde cada segundo representa un mes.

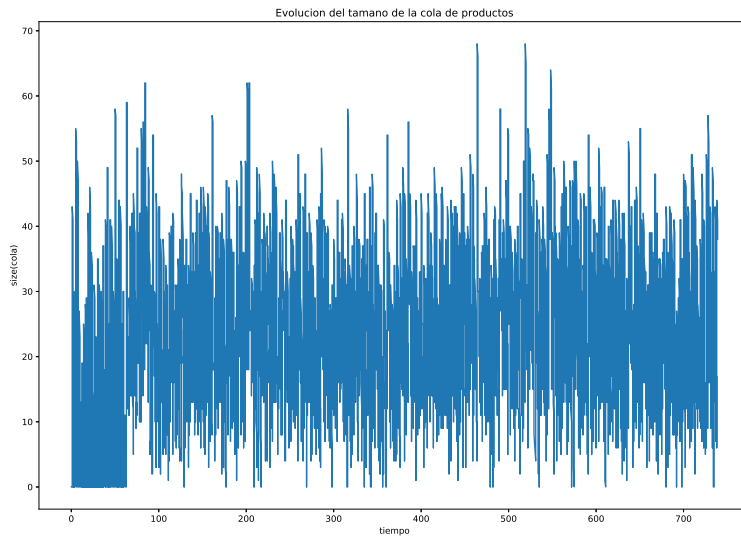


Figura 34: Evolución del tamaño del inventario para $n = 50$.

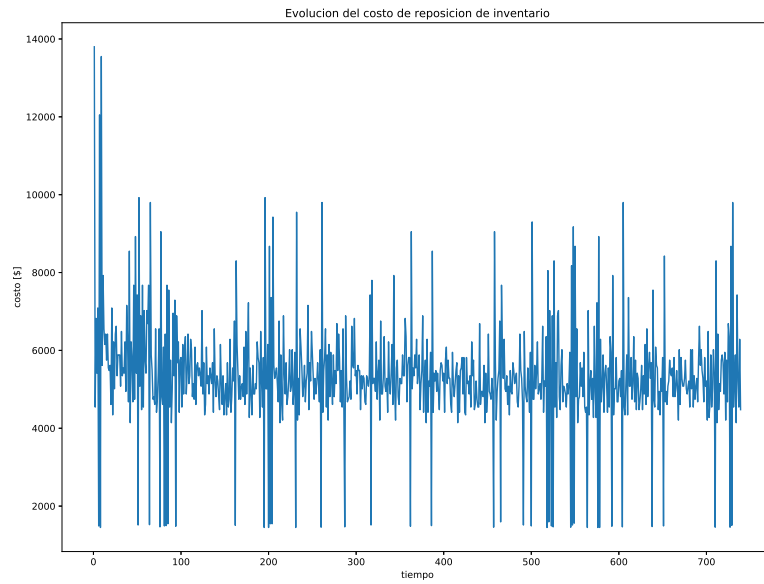


Figura 35: Evolución del costo por reposición de inventario para $n = 50$.

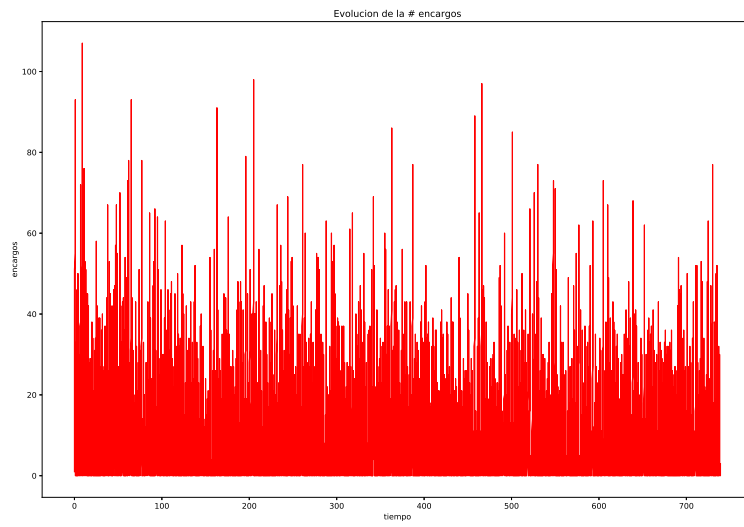


Figura 36: Evolución de la cantidad de encargos para $n = 50$.

Como siguiente análisis se procedió a disminuir la política de reposición a $n = 10$, con el objeto de evaluar la evolución de los parámetros anteriores. A diferencia del caso anterior se realizarán encargos de materia prima a los proveedores sólo en caso que el nivel del inventario sea menor a 10. Se espera que el costo baje si el inventario estaba sobredimensionado para la política de reposición anterior o que se produzca una menor cantidad de encargos si por el contrario estaba subdimensionado.

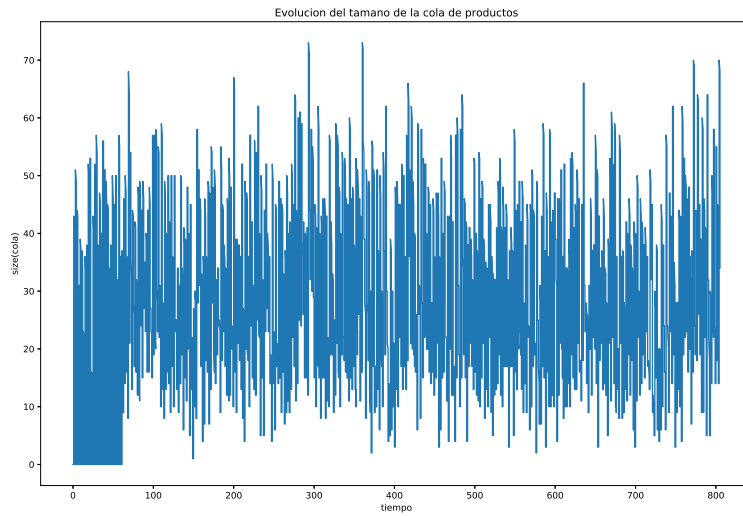


Figura 37: Evolución del tamaño del inventario para $n = 10$.

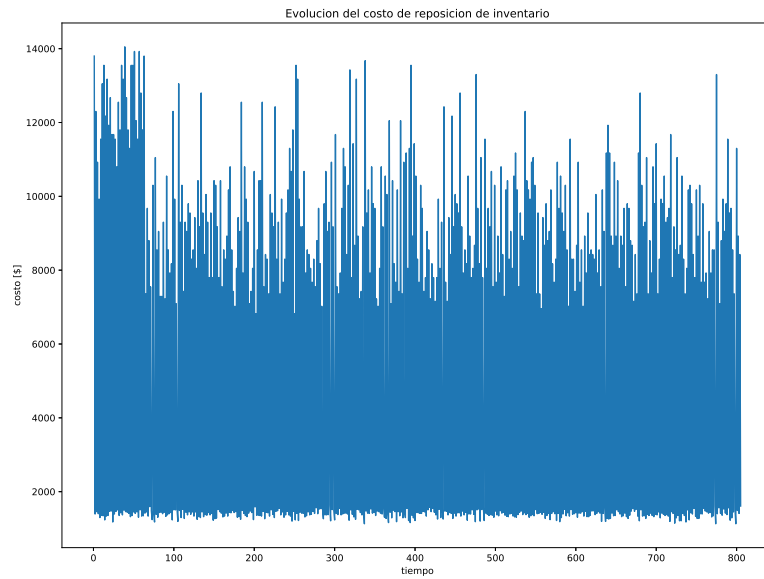


Figura 38: Evolución del costo por reposición de inventario para $n = 10$.

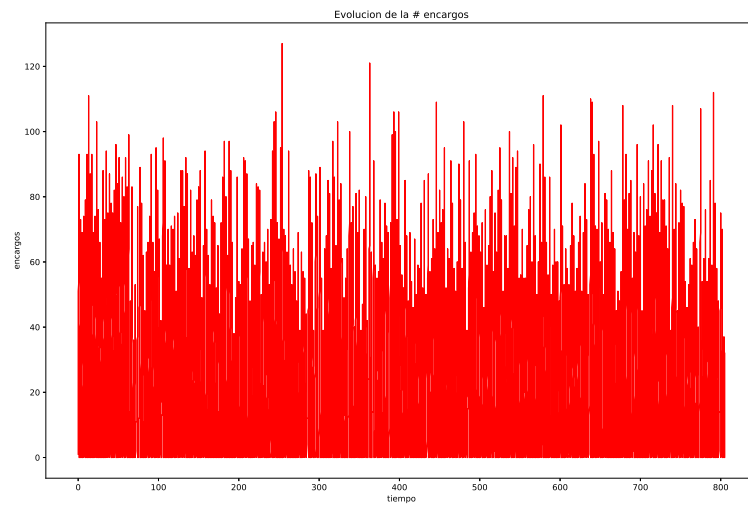


Figura 39: Evolución de la cantidad de encargos para $n = 10$.

Se observa que el costo por reposición y mantenimiento de unidades en el inventario se torna mucho más inestable, aumentando levemente en promedio. También se observa un aumento de la cantidad de encargos como así también una mayor variabilidad de la cantidad de productos en el inventario. Estos últimos dos comportamientos justifican el aumento en el costo. De esto se puede concluir que una política de reposición con un parámetro n tan bajo no resulta eficiente.

Para la elaboración de una unidad de producto es necesario que haya disponibles una unidad de cada insumo. Cada uno de los insumos de los que se compone el producto es provisto por un tipo de proveedor distinto, i.e. el insumo A es provisto por un proveedor inmediato, el insumo B por un proveedor por múltiplo de cantidad fija y el insumo C por un proveedor por encargo que entregue al mes siguiente de efectuado el pedido. En las simulaciones se pudo observar que cada vez que arriba el insumo C se consume al instante en la elaboración de productos pues los insumos A y B ya estaban previamente disponibles para ellos. Por este motivo, en todo momento la cola C de la Figura 1 tiene largo 0. También se puede notar que debido a la política de entrega del proveedor por múltiplo de cantidad fija, en la cola del insumo B siempre quedan remanentes de los pedidos anteriores y por lo tanto el tamaño de esta cola crece sostenidamente.

4.3.2. Conclusiones

Se comprobó la utilidad del formalismo DEVS y en particular de la herramienta CD++ para el modelado y simulación de sistemas con dinámicas por eventos. Al producirse eventos de manera aleatoria no es posible modelizarlos utilizando simuladores basados en una base de tiempo discreta, pues la ocurrencia de un evento de entrada es impredecible y por lo tanto el paso de simulación debería ser lo suficientemente pequeño como para coincidir con los eventos. En caso de adaptar el paso de simulación a la aparición de eventos hace que el tiempo necesario para realizar una simulación crezca exponencialmente.

Se simuló el comportamiento de un sistema de inventario de una industria considerando clientes de distinto tipo y proveedores con distintos comportamientos. Los insumos entregados por los proveedores se utilizan para elaborar el único producto de esta industria. Cada insumo y cada producto se representa a partir de su fecha de vencimiento. Los instantes en los que los clientes realizan pedidos, como la cantidad de unidades que piden y las fechas de vencimiento de los insumos y los productos responden a variables aleatorias con determinadas distribuciones de probabilidad y cuyos. En particular, para este trabajo se consideraron tasas de arribo de pedidos de clientes que siguen una distribuciones exponenciales.

Se descompuso al problema en tres niveles jerárquicos: a) un nivel superior o *top model* donde se encuentran la industria, los clientes y los proveedores, b) un segundo nivel dentro de la industria compuesto por el inventario, el control de inventario y la atención del cliente, y c) un tercer nivel de atención al cliente conformado por la atención al cliente y el control de calidad.

Para validar el comportamiento del sistema, en la sección 4.1 se verificó de

forma independiente el comportamiento de cada bloque excitándolos con entradas conocidas y analizando si los resultados obtenidos correspondían con los esperados.

Durante el desarrollo del trabajo nos encontramos con un problema de simultaneidad de eventos. Por un lado, el control de inventario realiza un pedido a sus proveedores a principio de cada mes. Por el otro lado, el proveedor por encargo entrega en ese mismo momento los insumos que le fueron encargados al inicio del mes. Esta simultaneidad fue *evitada* haciendo que el proveedor entregue cada 0,999 mes.

Por último, a lo largo del desarrollo de las simulaciones con la herramienta CD++ se encontraron algunos inconvenientes: a) la variable `nextChange()` no representa el tiempo entre la última transición interna y la próxima sino el tiempo que falta hasta dicha transición, b) el archivo `.ma` no distingue parámetros en mayúsculas y minúsculas (no es *case-sensitive*) y c) el archivo de salida de la simulación no guarda toda la información de los mensajes intercambiados por los bloques a lo largo de la simulación. Por este último motivo, en las funciones de salida de los atómicos se agregaron mensajes propios que utilizamos para analizar el comportamiento de la simulación y filtrar información que luego procesamos con un script de python. Para este trabajo práctico se utilizó el simulador CD++ en su versión *standalone* por consola.

A. Código Implementado

<https://github.com/TwinT/DEVS-Inventario>