

# Modelo Devs para simulación integrada de transacciones electrónicas y plataforma informática

Carlos María Chezzi<sup>1,2</sup>

*CIDISI - Facultad Regional Concordia  
Universidad Tecnológica Nacional  
Concordia, Argentina*

Ricardo Lerman Federico Baroni Ana R. Tymoschuk<sup>3</sup>

*CIDISI - Facultad Regional Santa Fe  
Universidad Tecnológica Nacional  
Santa Fe, Argentina*

---

## Resumen

La evaluación del desempeño de los procesos de comercio electrónico requiere una visión integrada de las capacidades ofrecidas por los recursos tecnológicos, las funcionalidades del sitio y los resultados económicos obtenidos. La simulación es una herramienta que permite experimentar configuraciones alternativas de negocio previo a su implementación. Para ello, se necesitan modelos que representen la interacción de las transacciones de un sitio, el patrón navegacional de clientes y la plataforma tecnológica que implementa los servicios. DEVS es un framework de modelado para simulación de evento discreto, basado en la teoría de sistemas y con recursos para modelar complejidad. El objetivo del trabajo es proponer la construcción de modelos de simulación en un framework DEVS para predecir el comportamiento de operaciones comerciales electrónicas a nivel de negocios y de recursos tecnológicos. DEVSJAVA es una herramienta orientada a objetos que implementa el framework DEVS en el lenguaje Java. Con el uso de esta herramienta se muestra un modelo de comercio electrónico B2C y el análisis de los resultados de simulación.

*Keywords:* Framework DEVS, Simulación, Desempeño, E-Commerce, DEVSJAVA.

---

---

<sup>1</sup> Los autores queremos agradecer especialmente a la Dra. Ana Rosa Tymoschuk por su vocación y compromiso constante en nuestra formación.

<sup>2</sup> Email: [carlos\\_chezzi@frcon.utn.edu.ar](mailto:carlos_chezzi@frcon.utn.edu.ar)

<sup>3</sup> Email: [anrotym@santafe-conicet.gov.ar](mailto:anrotym@santafe-conicet.gov.ar)

## 1. Introducción

Las tecnologías de Internet proponen recursos a la organización para implementar transacciones de comercio electrónico, tales como disponer información, establecer comunicaciones electrónicas y llegar a conducir operaciones en línea. Además, es fuente de creación de valor agregado a la empresa ya que permite alcanzar un número significativo de clientes y establecer comunicaciones interorganizacionales con reducción de costos [10]. En otro sentido, mediante la utilización de redes para la interconexión de servidores y sistemas distribuidos se configuran plataformas de alto rendimiento, que aseguran la disponibilidad y funcionalidad de los negocios basados en transacciones electrónicas [3].

En este contexto, se identifican dos aspectos fundamentales: uno de negocios que comprende metas y estrategias propias del proceso y otro de infraestructura tecnológica, que aporta las componentes de hardware, software y comunicaciones que implementan las operaciones electrónicas [7]. La evaluación integrada de estos aspectos, preferentemente previa a su implementación, orienta las decisiones sobre las configuraciones adecuadas de tecnología en relación con las transacciones requeridas de negocios [11].

La simulación en el proceso de negocios constituye una estrategia utilizada para comprender la esencia del sistema de negocios, identificar oportunidades para el cambio y evaluar su impacto a través de los indicadores de desempeño. Posibilita experimentar decisiones de mercados con configuraciones alternativas de negocios sin necesidad de interrumpir las operaciones del sistema actual [1]. De ahí la importancia de contar con modelos de simulación de transacciones de comercio electrónico que midan el desempeño de sus procesos, tanto en el nivel de negocios como en el de infraestructura informática y de comunicaciones.

Una característica fundamental de estos modelos es la necesidad de interacción entre unidades heterogéneas de estudio, como por ejemplo la integración de transacciones de negocios y la plataforma informática donde se ejecutan. Para esto se requieren técnicas de representación con capacidades de modelización de sistemas complejos [4]. Un framework de formalización que permite interoperabilidad, reusabilidad y flexibilidad es DEVS (Discrete EVents System Specification), cuyo fundamento en la teoría de sistemas y su metodología centrada en el modelado jerárquico y modular permiten representar complejidades de los componentes del sistema y sus acoplamientos.

El objetivo de este trabajo es proponer la construcción de modelos de simulación de eventos discretos en un framework DEVS para predecir el comportamiento de operaciones comerciales electrónicas a nivel de negocios y de recursos tecnológicos. Se presenta una Arquitectura de Referencia para el diseño de procesos de comercio electrónico y el Framework DEVS a través de la con-

ceptualización de su formalismo. Luego se muestra su utilización en la construcción de modelos de procesos de comercio electrónico y se implementa en la herramienta DEVSJAVA un caso ejemplo para transacciones Business to Consumer (B2C). Se proponen los resultados de simulación y se obtienen las conclusiones del trabajo.

## 2. Arquitectura de Referencia del Modelo de Negocios Electrónicos

Las operaciones de comercio electrónico se pueden modelar mediante una arquitectura de referencia [9] con cuatro vistas, como se muestra en la Figura 1, en donde las dos superiores contienen el proceso de negocios, según la estrategia organizacional sobre la cual se planifican las transacciones del sitio. Los bloques inferiores especifican la infraestructura tecnológica que soporta las operaciones de comercio electrónico. Por tanto, se identifican los servicios que operan las transacciones y los sistemas informáticos que los ejecutan.

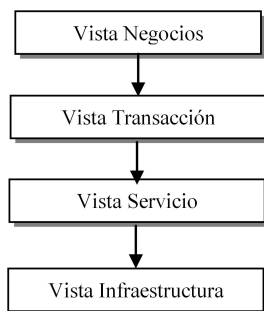


Figura 1. Vistas del Modelo de Comercio Electrónico.

Con base en la arquitectura de referencia, la identificación del segmento de negocios, las metas y los potenciales clientes, se deben diseñar las operaciones del sitio para la ejecución de las transacciones electrónicas, tales como navegar, buscar, seleccionar, comprar, registrar, entre otras. Una herramienta para determinar un patrón navegacional del sitio es el Customer Behavior Model Graph (CBMG) [7], que mediante monitorizaciones en el sitio se pueden establecer las frecuencias de acceso a las diferentes transacciones y las interacciones con los recursos informáticos.

En las vistas inferiores de la arquitectura de referencia se incluyen los recursos físicos de la plataforma informática, cuya representación se hace generalmente con teoría de redes de cola [6]. Cada recurso se representa como un centro de servicio o de demora, con un tiempo de atención a los requerimientos y contemplando una cola de espera si el centro está ocupado en el primer caso. El desempeño de la red de centros se estima mediante leyes operacionales, según el comportamiento de los clientes en el sitio y se obtienen métricas de los recursos y del negocio.

### 3. Formalismo DEVS

El formalismo DEVS [12] para sistemas de eventos discretos permite especificar un objeto matemático, llamado modelo atómico, definido por sus entradas (Inputports), salidas (Outputports), estados, una base de tiempos y funciones de transición internas y externas que coordinan los cambios de estado y ejecutan las salidas. Un modelo DEVS atómico se representa por la estructura (1).

$$(1) \quad N = \langle X_p, Y_p, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

Donde:

- *InputPorts* es el conjunto de puertos de entradas.
- *OutputPorts* es el conjunto de puertos de salidas.
- $X_p$  es el conjunto de valores permitidos en un puerto de entrada.
- $Y_p$  es el conjunto de valores permitidos en un puerto de salida.
- $X_n = \{(p, v) | (p \in InputsPort, v \in X_p)\}$  es el conjunto de puertos de entrada y valores de recepción permitidos desde un evento externo.
- $Y_n = \{(p, v) | (p \in OutputPorts, v \in Y_p)\}$  es el conjunto de puertos de salida y valores de emisión permitidos como evento de salida.
- $S$  es el conjunto de estados internos.
- $\delta_{int} = S \rightarrow S$  es la función de transición de estado interna.
- $\delta_{ext} = QxX \rightarrow S$  es la función de transición de estado externa.
  - $con Q = \{(s, e) | (s \in S, 0 < e < ta(S))\}$  es el conjunto de estados totales.
  - $e$  es el tiempo transcurrido desde la última transición.
  - $s$  es el conjunto parcial de estados para una duración  $ta(s)$  si no ocurre un evento externo.
- $ta = S \rightarrow R_{0, \infty}^+$  es la función de avance de tiempo. Para casos especiales: 0 identifica un estado transitorio de duración cero e  $\infty$  identifica un estado permanente de duración infinita. El modelo permanece en el estado  $\infty$  hasta el arribo de un evento externo.
- $\lambda = QxS \rightarrow S$  es la función de salida.

La semántica de un modelo atómico se basa en el siguiente razonamiento: los cambios de estados  $S$  se producen a través de las funciones de transición. A cada estado se le asocia un tiempo de permanencia calculado por la función de avance del tiempo  $ta$ . Si no ocurren eventos externos, los estados pueden cambiar ejecutando la función de transición interna  $\delta_{int}$ . Antes de cada transición interna se puede generar un evento de salida con la ejecución de la función de salida  $\lambda$ . Frente al ingreso de un evento de entrada externo, la función de transición de estado externa  $\delta_{ext}$  es quien determina el nuevo estado sobre la

base del estado corriente, el tiempo transcurrido  $e$  y la nueva entrada. Una característica a destacar es que la función  $\delta_{ext}$  no produce salida. Luego del cambio de estado  $e$  vuelve a cero.

Los modelos atómicos pueden interconectarse a través de puertos de entradas y salidas, generando modelos acoplados. Esta construcción jerárquica y modular permite representar sistemas complejos. Un modelo DEVS acoplado se representa por la estructura (2).

$$(2) \quad M = \langle X_m, Y_m, D_m, \{M_i \in D_m\}, EIC_m, IC_m, EOC_m \rangle$$

Donde:  $X_m$  es el conjunto de puertos de entrada y valores de entrada permitidos;  $Y_m$  es el conjunto de puertos de salida y valores de salida permitidos;  $D_m$  es el conjunto de referencias a componentes (modelos básicos o acoplados);  $M_i$  es el conjunto de componentes para cada  $i \in D_m$ .  $EIC_m$  (External Input Coupling) es el conjunto de enlaces de entrada que conecta las entradas de un modelo acoplado a una o más entradas de sus componentes.  $IC_m$  (Internal Coupling) es el conjunto de enlaces internos que conectan puertos de salidas con entradas de componentes del modelo acoplado.  $EOC_m$  (External Output Coupling) es el conjunto de enlaces de salidas que conecta las salidas de una o más componentes a la salida del modelo acoplado.

Los modelos pueden presentar eventos concurrentes, es decir varias componentes puedan activarse a la vez y enviar las salidas producidas a otras componentes. De este modo el receptor se enfrenta a una o más entradas y por tanto debe estar preparado para interpretarlas. En este caso se incorpora el concepto de DEVS Paralelos que produce una modificación al modelo DEVS tradicional (1) mediante el soporte de conjuntos de entradas y salidas que aceptan múltiples eventos y el agregado de la función de confluencia que organiza la ejecución concurrente.

Si alguna de las funciones  $\delta_{int}$ ,  $\delta_{ext}$ ,  $\lambda$  y  $ta$  es no determinística, el modelo DEVS es no determinístico.

#### 4. Framework DEVS para modelado de transacciones electrónicas

Un framework DEVS de modelado y simulación se define por tres elementos básicos: sistema real, modelo y simulador. El sistema real es la fuente de captura de datos de entrada y salida, el modelo es el conjunto de instrucciones que definen el modo de procesar las entradas y de producir las salidas y el simulador es el responsable de la ejecución del modelo de acuerdo a un protocolo formal abstracto [8]. De la observación del sistema fuente se obtienen datos que posibilitan la construcción de una base de datos de comportamiento. A partir de esta base de datos se diseña un diagrama de estructura que plantea componentes individuales y compuestos.

En el diagrama de estructura de la Figura 2 se esquematiza el modelo de proceso de comercio electrónico de acuerdo a una metodología top-down. Como subestructuras principales se plantean el Marco Experimental y el Modelo de Comercio Electrónico. El Marco Experimental consta de un Generador y un Transductor, mientras que el Modelo de Comercio Electrónico se compone de los modelos de Transacciones de Negocios y Plataforma Tecnológica. Para elaborar los modelos DEVS se utiliza, en otro sentido, una metodología bottom-up, de modo que desde los nodos hojas se identifica cada modelo atómico y del nodo padre sus acoplamientos.

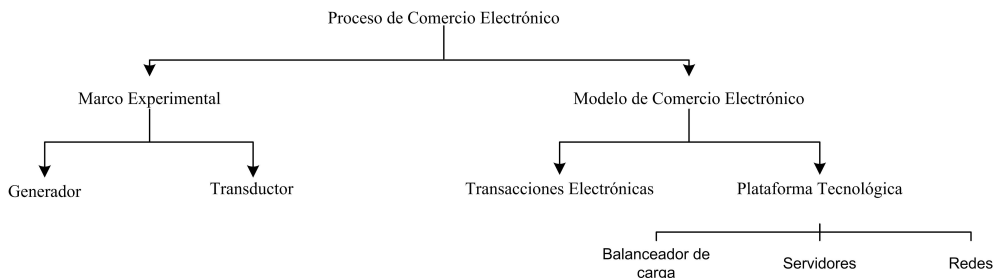


Figura 2. Diagrama de Estructura del Proceso de Comercio Electrónico.

Para construir la semántica DEVS se dispone del diagrama de especificación Statechart [5]. Es una herramienta orientada a gráficos, extensión de los diagramas de transición de estados, que permite especificar sistemas reactivos, con capacidades de modelado jerárquico, ortogonalidad (representación de actividades paralelas) e interdependencia (comunicación broadcast). En la Figura 3 (a) se muestra una transición interna que consiste en una línea punteada, cuyo rótulo se interpreta como: dada una condición y cumplido un tiempo en el estado  $i$  se produce una transición al estado  $j$ , con la salida de un valor por un puerto de salida y la ejecución de una operación a tener en cuenta en el algoritmo. En la Figura 3 (b) se observa una transición externa que consiste en una línea completa, cuyo rótulo se interpreta como: frente a un valor de un evento externo ingresado por un puerto de entrada se produce una transición al estado  $j$ , sin salida y se ejecuta una operación a tener en cuenta en el algoritmo.

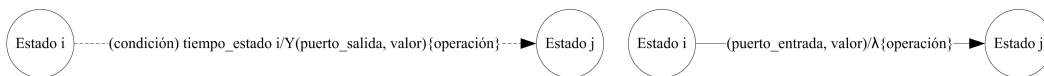


Figura 3. (a) transición interna; (b) transición externa.

#### 4.1. Protocolo de Simulación DEVS Paralelo

El protocolo de simulación DEVS construye un árbol AVL como se muestra en la Figura 4. Esta estructura contiene como elementos hojas los modelos atómicos. Con esta estructura se construye una cola desde los elementos hojas

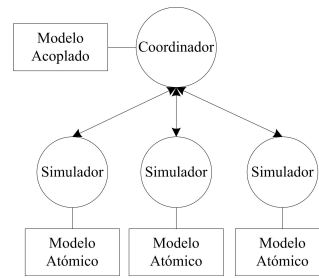


Figura 4. Estructura AVL del Protocolo DEVS

y se almacenan datos claves como nombre del elemento y tiempo del próximo evento. El ciclo de simulación [12] consiste en:

1. Buscar el tiempo menor en la cola y actualizar el reloj de simulación.
2. Seleccionar todos los modelos hojas cuyo tiempo del próximo evento coincide con el reloj. Con los elementos seleccionados formar el conjunto de modelos inminentes llamado I.
3. Ejecutar las funciones de salidas de los modelos del conjunto I y enviar los mensajes de salida a los puertos de entrada correspondientes. Los mensajes enviados son recolectados en un conjunto de múltiples eventos por los puertos de entrada.
4. Seleccionar todos los modelos que contienen un conjunto de múltiples eventos de mensajes esperando en sus entradas y formar el conjunto M.
5. Encontrar la intersección de los elementos de los conjuntos I y M, al cual se lo llama conjunto confluencia. Para todos los modelos del conjunto confluencia, ejecutar la función de transición de confluencia, que primero atiende las transiciones internas y luego las externas.
6. Para el resto de los elemento del conjunto I que no pertenecen a la intersección, ejecutar la función de transición interna asociada a cada modelo.
7. Para el resto de los elementos del conjunto M que no pertenecen a la intersección, ejecutar la función de transición externa asociada a cada modelo.

Una vez ejecutados los pasos anteriores se actualizan los próximos eventos de tiempo de acuerdo a las funciones de avance del tiempo asociadas a cada modelo. Se reconstruye la cola y se repiten los pasos 1 a 7 hasta que no existan modelos en el conjunto I o se produzca una condición de fin de la simulación.

#### 4.2. Marco Experimental

El Marco Experimental es un modelo acoplado de los modelos atómicos Generador y Transductor. El Generador es capaz de proveer las entradas al sistema y el Tansductor es responsable de monitorizar la ejecución del proceso

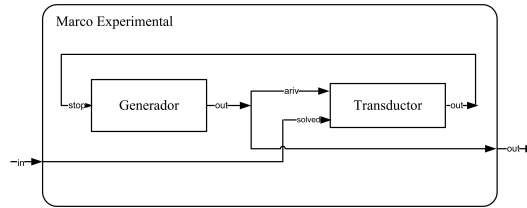


Figura 5. Modelo Acoplado del Marco Experimental.

de comercio electrónico y computar las métricas de salida. Su diseño refleja las condiciones de experimentación sobre el modelo del sistema. En la Figura 5 se presenta un gráfico general con un Generador que a través del puerto *out* produce los requerimientos al sistema de acuerdo a un tiempo entre arribos. Las salidas del Generador se envían al modelo del sistema y al puerto *ariv* del Transductor para que este las registre. Por el puerto *solved* el Transductor recibe los requerimientos atendidos, los que compara con los registrados para calcular las métricas. Una vez finalizado el tiempo de simulación el Transductor envía por su puerto de salida *out* un mensaje de *fin* al Generador. Este mensaje es recibido por el puerto de entrada *stop*. La descripción del formalismo DEVS de los modelos atómicos y acoplados se detalla en el trabajo [13].

#### 4.3. Modelo Transaccional

El modelo transaccional representa las sesiones de clientes en el sitio de negocios electrónicos. Para construir este modelo se toma como referencia el diagrama CBMG de la Figura 6, propuesto en el trabajo [7]. Este patrón representa el comportamiento de clientes frecuentes sobre un sitio que ofrece las transacciones Visitar Página Principal, Navegar, Buscar, Seleccionar productos para ver sus datos, Agregar al carro de compras y Pagar. La transición entre cada transacción se efectúa de acuerdo a una probabilidad. La organi-

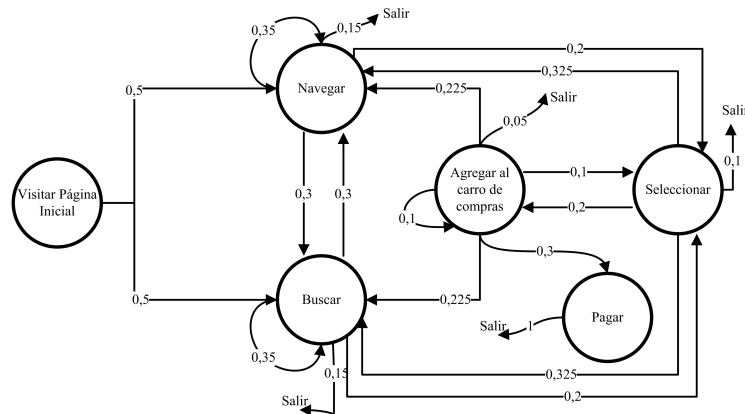


Figura 6. Grafo de comportamiento de clientes.

zación del modelo transaccional comprende dos funcionalidades diferentes: (i)



atender el requerimiento por parte de la plataforma tecnológica (recibir requerimientos de clientes a la transacción y enviarlos a procesar a la plataforma tecnológica) y (ii) recibir requerimientos atendidos por la plataforma tecnológica y enviarlos a la próxima transacción.

En la segunda función del modelo la selección del puerto de salida a la siguiente transacción es aleatoria. Por tanto, la función de salida se transforma en no determinística y de este modo el modelo DEVS es no determinístico. En la Figura 7 se especifica el modelo atómico transaccional en un diagrama Statechart. Se observa que consta de dos puertos de entradas, uno *in* que recibe los requerimientos a la transacción y otro *fromIT* que recibe los requerimientos atendidos por la tecnología informática. Respecto a los puertos de salida se considera un grupo de *n* puertos *out* cuya selección depende de la probabilidad de transición a la próxima transacción y un puerto *toIT* que envía los requerimientos a ser atendidos desde la tecnología informática. Los estados posibles son *passive* y *sending*, los cuales corresponden a los casos especiales llamados permanente (*tiempo*  $\infty$ ) y transitorio (*tiempo* 0).

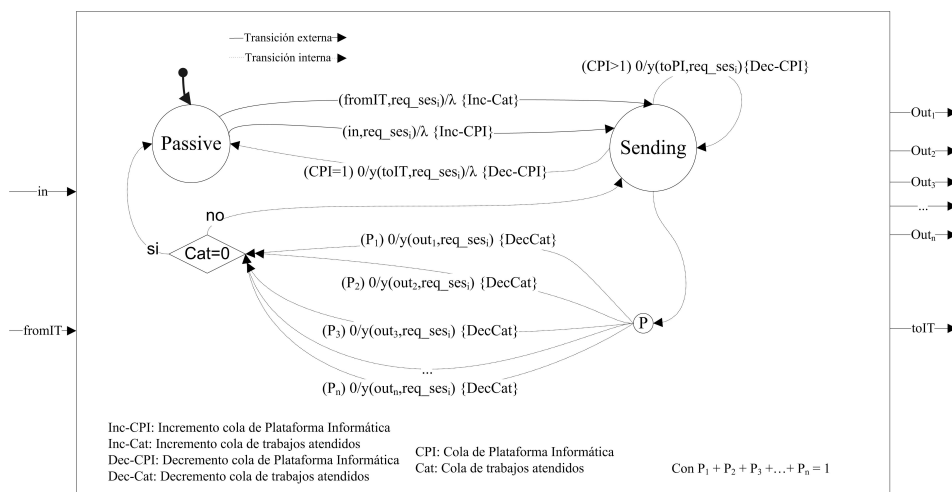


Figura 7. Modelo Atómico Transaccional.

En esta representación se incorpora el tratamiento de las colas para enriquecer la construcción de la lógica del algoritmo. Los requerimientos arribados por el puerto *in* se operan con la Cola de Plataforma Informática, la que se incrementa con una recepción y se decrementa con su procesamiento. Del mismo modo se opera la Cola de Trabajos Atendidos, que se incrementa al recibir un requerimiento atendido por la plataforma informática y se decrementa al ser enviado a la próxima transacción.

El modelo atómico transaccional no determinístico se define como:

$$\text{Modelo Transaccional} = \langle X_{trans}, Y_{trans}, S_{trans}, P_{\lambda, out}, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

Donde:

Conjunto de puertos y valores permitidos

$InputPorts = \{in, fromIT\}$  conjunto de puertos de entrada.

$OutputPorts = \{out_1, out_2, out_3, \dots, out_n, toIT\}$  conjunto de puertos de salida.

$X_{in} = \{reqses_i\}$  con  $i = 1..n$  es el conjunto de valores permitidos para el puerto de entrada  $in$  y representa los requerimientos a una transacción de negocios.

$X_{fromIT} = \{reqses_i\}$  es el conjunto de valores permitidos para el puerto de entrada  $fromIT$  y representa los requerimientos procesados por la Plataforma Informática.

$Y_{out_i} = \{reqses_i\}$  conjunto de valores permitidos para el puerto de salida  $out_i$  y representa el envío de un requerimiento a la próxima transacción de negocios de acuerdo a la probabilidad de transición.

$Y_{toIT} = \{reqses_i\}$  conjunto de valores permitidos para el puerto de salida  $toIT$  y representa el envío de un requerimiento para su proceso en la Plataforma Informática.

Conjunto de eventos de entrada y salida.

$X_{trans} = \{(in, reqses_i), (fromIT, reqses_i)\} \mid in, fromIT \in InputPorts, reqses_i \in X_{fromIT}$ .

$Y_{trans} = \{(out_i, reqses_i), (toIT, reqses_i)\} \mid out_i, toIT \in OutputPorts, reqses_i \in Y_{toIT}$ .

Conjunto de estados.

$S_{trans} = \{passive, sending\}$

Conjunto de probabilidades de selección de puerto de salida.

$P_{\lambda, out} = \{(P_1, trans_1, out_1), (P_2, trans_2, out_2), \dots, (P_n, trans_n, out_n)\}$  es el conjunto de probabilidades de  $P_i$  de continuar a la próxima transacción por el puerto de salida  $out_i$ .

Funciones:

Función de Transición Interna  $\delta_{int}$  :

$\delta_{int}(sending) = sending$  Si  $CPI > 1$

$\delta_{int}(sending) = passive$  Si  $CPI = 1$

$\delta_{int}(sending) = sending$  Si  $Cat > 1$

$\delta_{int}(sending) = passive$  Si  $Cat = 1$

Función de Transición Interna  $\delta_{ext}$  :

$\delta_{ext}(passive, e, (in, reqses_i)) = sending$

$\delta_{ext}(passive, e, (fromIT, reqses_i)) = sending$

$\delta_{ext}(sending, e, (in, reqses_i)) = sending$

$\delta_{ext}(sending, e, (fromIT, reqses_i)) = sending$

Función de Salida  $\lambda$ :

$r = random U[0, 1]$

$\lambda(sending) = (out_1, reqses_i)$  Si  $r \leq P_1$

$\lambda(sending) = (out_2, reqses_i)$  Si  $r \leq P_1 + P_2$

$\lambda(sending) = (out_n, reqses_i)$  Si  $r \leq P_1 + P_2 + \dots + P_n$

Función de Transición de Confluencia:  $\delta_{conf}(s, ta(s), x) = \delta_{ext}(\delta_{int}, 0, x)$

Función de Avance de Tiempo  $ta$  :

$$ta(passive) = \infty$$

$$ta(sending) = 0$$

En la función de salida  $\lambda$  se observa que se dividen los intervalos en  $n$  partes, donde cada intervalo corresponde a una suma de probabilidades. Se genera un número aleatorio  $r$  con una distribución  $U[0, 1]$ , se determina el intervalo que contine a  $r$  y se selecciona el puerto de salida. Frente al arribo de requerimientos concurrentes la función de confluencia es la responsable de organizar su ejecución, de modo que primero habilita la función de transición interna y luego la externa.

#### 4.4. Modelo de Plataforma Tecnológica

El diseño de modelos de infraestructura tecnológica se basa en modelos atómicos que representan centros de servicios. Estos centros pueden recibir requerimientos en una frecuencia superior a su capacidad de servicio y por tanto necesitan de una cola. Un modelo DEVS atómico de Recurso Tecnológico con cola FIFO se muestra en la Figura 8.

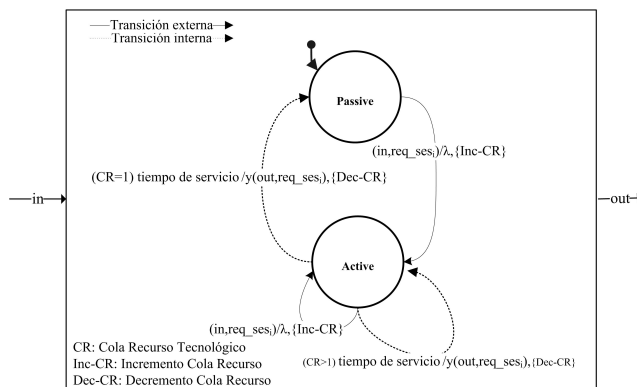


Figura 8. Modelo Atómico del Recurso Tecnológico

El puerto de entrada *in* recibe los requerimientos a ser atendidos y el puerto de salida *out* entrega los requerimientos completados. Los estados posibles son *active* y *passive*. En el estado inicial se encuentra *passive*. Su semántica consiste en: dado el arribo de un requerimiento por el puerto *in* la función de transición externa cambia al estado *active* e incrementa la cola de requerimientos. Una vez finalizado el tiempo de servicio se produce una salida de requerimiento atendido por el puerto *out* y el cambio de estado, por parte de la función de transición interna, está sujeto a los trabajos en cola. De modo

que si se encuentra un sólo trabajo en cola cambia al estado *passive* y si hay más de uno reinicia el estado en *active*. Si en estado *active* se produce el arribo de un evento externo, la función de transición externa almacena en cola el trabajo arribado. Frente al arribo de eventos concurrentes es la función de confluencia la que resuelve el problema.

Un modelo DEVS atómico de Recurso Tecnológico con cola FIFO se representa por la tupla:

$$\text{Recurso Tecnológico} = \langle X_r, Y_r, S_s, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

Donde:

Conjunto de puertos y valores permitidos.

$InputPorts = \{in\}$  conjunto de puertos de entrada.

$OutputPorts = \{out\}$  conjunto de puertos de salida.

$X_{in} = \{reqses_i\}$  con  $i = 1..n$  es el conjunto de valores permitidos para el puerto de entrada  $in$ .

$Y_{out} = \{reqses_i\}$  con  $i = 1..n$  es el conjunto de valores permitidos para el puerto de salida  $out$ .

Conjunto de eventos de entrada y salida.

$X_r = \{(in, reqses_i)\} \mid in \in InputPort \text{ y } reqses_i \in X_{in}$ .

$Y_r = \{(out, reqses_i)\} \mid out \in OutputPort \text{ y } reqses_i \in Y_{out}$ .

Conjunto de estados.

$S = \{active, passive\}$  conjunto de estados.

Funciones

Función de Transición Interna  $\delta_{int}$  :

$\delta_{int}(active) = passive \text{ si } CR = 1$

$\delta_{int}(active) = active \text{ si } CR > 1$

Función de Transición Externa  $\delta_{ext}$  :

$\delta_{ext}(passive, e, (in, reqses_i)) = active$

$\delta_{ext}(active, e, (in, reqses_i)) = active$

Función de Salida  $\lambda(active) = Y(out, reqses_i)$

Función de Transición de Confluencia:  $\delta_{conf}(s, ta(s), x) = \delta_{ext}(\delta_{int}, 0, x)$

Función de Avance de Tiempo  $ta$  :

$ta(active) = \text{tiempo de servicio}$

$ta(passive) = \infty$

De modo análogo se representa un centro de servicio sin cola. A través del acoplamiento de modelos atómicos, ya sea en forma jerárquica o plana, se pueden construir redes de recursos que caractericen diferentes configuraciones

de Plataformas Informáticas.

## 5. Desarrollo del Modelo de Simulación

### 5.1. Descripción del problema y su modelo DEVSJAVA

Para mostrar la utilización del Framework DEVS se propone un modelo de comercio electrónico B2C y se lo implementa en la herramienta simulación DEVSJAVA [13]. La vista transaccional del modelo se construye de acuerdo al digrama CBMG de la Figura 6 y representa un patrón navegacional de clientes a un sitio de venta electrónica de libros [7]. Respecto a los servicios y la plataforma informática se plantean dos grupos: (i) un cluster de servidores que ejecuta las transacciones: Visitar Página Inicial, Buscar, Navegar, Agregar al Carro de Compras y Pago; (ii) un servicio Web externo al cluster que realiza el cobro electrónico. El modelo DEVSJAVA se muestra en la Figura 9 y consiste en: (i) modelo acoplado Experimental Frame que genera los arribos al sitio, monitoriza el comportamiento y computa las métricas de salida, (ii) Modelo Transaccional que consiste en una red acoplada de transacciones de tipo plana con probabilidades de transiciones entre ellas, (iii) Cluster es un modelo acoplado jerárquico de plataforma informática cuyo nodo principal es un balanceador de carga y sus hijos los servidores Web, (iv) un modelo atómico asociado a la transacción Pagar que se corresponde con el servicio de ejecución del pago electrónico.

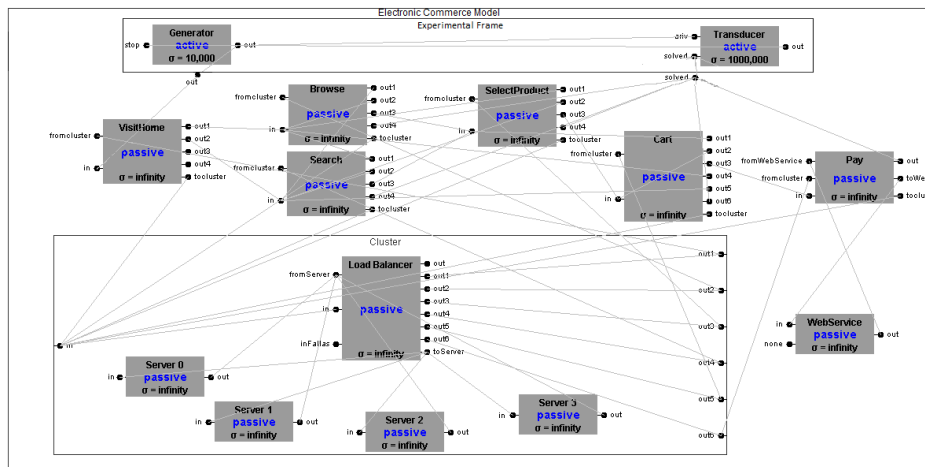


Figura 9. Ventana Simulador DEVSJAVA Modelo de Comercio Electrónico.

### 5.2. Parámetros del Modelo

El modelo del Cluster consiste en un centro de cola que representa al balanceador de cargas. El mismo gestiona la recepción de los requerimien-

tos arribados con una asignación secuencial en los servidores. El balanceador de carga es responsable de mantener en cola los arribos que no pueden ser atendidos, por tanto los servidores se modelan como centros de demora.

En la hipótesis de simulación se propone evaluar el rendimiento de la plataforma tecnológica en relación con la productividad de negocios del proceso de comercio. Se trabaja con dos alternativas de diseño tecnológico: incrementar el número de servidores de la misma capacidad o cambiar el sistema incorporando servidores de mayor capacidad. Para estas alternativas se calcula la cantidad de clientes que concretan la compra y de este modo se proyecta el rendimiento económico.

La configuración del modelo requiere tiempos de servicios para cada uno de los centros correspondientes a los recursos tecnológicos. Para ello se construye un sistema controlado que implementa las transacciones del sitio en un servidor Web y se realizan experimentaciones para diferentes cargas de trabajo. De estas experimentaciones se obtienen los siguientes tiempos de servicios promedio (en segundos): Visitar Página Principal: 0.0437, Navegar: 0.0156, Buscar: 0.025, Seleccionar producto: 0.0125, Agregar al carro de compras: 0.0063 y Pagar 0.037.

El cobro electrónico se implementa a través de una asociación a un servicio externo, cuya operación consiste en enviar los datos del comprador en un mensaje SOAP para que se ejecute el pago. SoapUI es una herramienta de software libre que posibilita el testeado de tecnologías con servicios Web (<http://www.soapui.org>). Con su uso se mide el tiempo de ejecución de un pago directo y de sus experimentaciones se calcula el tiempo promedio de servicio en la transacción de pago. Este tiempo es de 5.43 segundos.

## 6. Resultados de Simulación

Como métrica de evaluación de la tecnología informática se toma el tiempo de respuesta. De la investigación de indicadores del tiempo de respuesta en el sitio <http://www.keynote.com> se observa que son *recomendables* tiempos menores a 7,9 seg., *satisfactorios* los menores a 10,6 seg. e *insatisfactorios* en caso contrario. En un primer escenario se simula el comportamiento del sistema de acuerdo a los parámetros establecidos, con un único servidor y para diferentes tiempos entre arribos de clientes. En la Figura 10 se grafican los tiempos de respuesta del escenario 1 para diferentes tiempos entre arribos y se observan valores *insatisfactorios*.

Se plantea un segundo escenario que consiste en incorporar servidores al cluster de la misma capacidad que en el escenario 1 y simular para un tiempo entre arribos de 0.5 segundos. La Figura 11(a) muestra que para un cluster con 4 servidores el tiempo respuesta es de 23,69 seg., lo cual reduce a la mitad el tiempo de respuesta del escenario anterior, sin conseguir un valor adecuado.

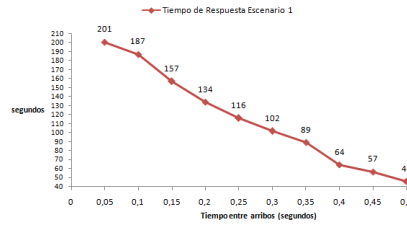


Figura 10. Escenario 1: tiempos de respuesta para diferentes velocidades de arribos de clientes.

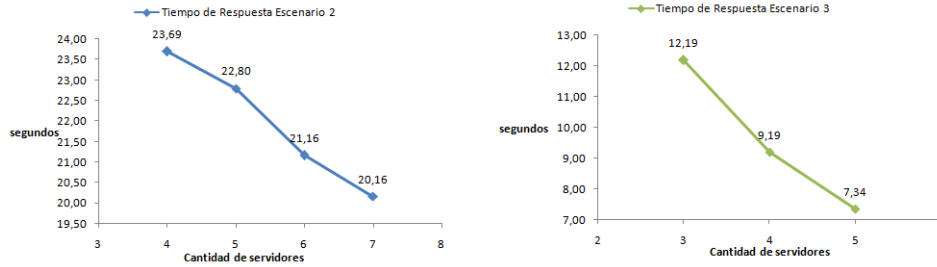


Figura 11. (a) Escenario 2; (b) Escenario 3

Además, con el incremento en el número de servidores a 5, 6 y 7 se produce un decremento de un segundo por servidor agregado. Como escenario 3 se considera un cambio de servidores que duplique la capacidad de los planteados. De la Figura 11(b) se observa que para un cluster con cuatro servidores se alcanza un tiempo de respuesta *satisfactorio* (9,19 seg.) y para un cluster de 5 servidores un tiempo de respuesta *recomendable* (7,34 seg.).

Para decidir la configuración a elegir se trabaja sobre las métricas de negocios. En el trabajo [2] se propone un Framework de Métricas de Negocios del cual se va a considerar la cantidad de clientes que efectúan compras. De la simulación se obtiene la tasa de ventas por segundo  $X_{Pay}$ . A partir de esta métrica se puede calcular un ingreso promedio de ventas por segundo (3).

$$(3) \text{ Ingreso por Ventas}[\$/seg] = X_{Pay}[\text{clientes/seg}] \times \text{Ingreso Promedio por Ventas}[\$/cliente]$$

En función del ingreso calculado por ventas se analiza un equilibrio entre la inversión requerida, los ingresos estimados del modelo de negocios y el tiempo en el cual se obtienen ganancias.

## 7. Conclusiones

En este trabajo se muestra el Formalismo DEVS como una estrategia para el diseño de modelos de procesos de comercio electrónico. A través del Framework DEVS se proponen los requerimientos necesarios para la construcción del modelo de simulación. Con un diagrama de estructura se abstraen las principales componentes del modelo. A partir de este diagrama se especifica con la herramienta gráfica Statechart cada modelo atómico. Sobre la base de los diagramas se construye el formalismo, con el cual se diseñan los algoritmos.

Estos algoritmos son implementados en la herramienta DEVSJAVA para su simulación.

De la simulación se obtienen tiempos de respuesta para evaluar el desempeño de la plataforma tecnológica y la tasa de ventas por segundo. Con dicha tasa se puede calcular el ingreso promedio por ventas y de este modo analizar el punto de equilibrio entre las inversiones, los costos y los ingresos.

Como trabajo futuro se proyecta considerar el tiempo de pensado del cliente en cada transacción y estudiar otras configuraciones de tecnología.

## Referencias

- [1] Bosilj-Vuksic, V., M. I. Stemberger, J. Jaklic, and A. Kovacic, *Assessment of E-Business Transformation Using Simulation Modeling*, Simulation. **83**, N 1, December (2007), 851–861.
- [2] Chezzi, C. M., A. Villamonte, and A. R. Tymoschuk, A. R., *Herramienta para Análisis de Configuraciones de Sistemas Eficientes de e-business*, XXXIV Conferencia Latinoamericana de Informática (CLEI 2008), (2008), 940–949.
- [3] Faour, A., and N. Mansour, *Weblins: A scalable WWW cluster-based server*, Advances in Engineering Software. **37**, N 1, January (2006), 11–19.
- [4] Filippi, J.-B., and P. Bisgambiglia, *JDEVS: an implementation of a DEVS based formal framework for environmental modelling*, Environmental Modelling and Software. **19**, N 3, March (2004), 261–274.
- [5] Francs, C., E. J. da Luz Oliveira, J. Costa, M. Santana, S. Bruschi and S. Carvalho, *Performance evaluation base on system modeling using Statecharts extensions*, Simulation Modelling Practice and Theory, **13**, (2005). 584–618.
- [6] Lazowska, E., J. Zahorjan, S. Graham and K. Sevcik, “Quantitative System Performance. Computer System Analysis Using Queueing Network Models,” Ed. Prentice Hall Inc., 1984.
- [7] Menascé, D., V. A. F. Almeida, R. Fonseca, and M. Mendes, *Business-oriented resource management policies for e-commerce servers*, Performance Evaluation. **42**, N 2, September (2000), 223–239.
- [8] Palaniappan, S., A. Sawhney., H. S. Sarjoughian, *Application of the DEVSS Framework in Construction Simulation*, Proceedings of the 2006 Winter Simulation Conference. (2006), 2077–2086
- [9] Schmid, B., and M. Lindemann, *Elements of a Reference Model for Electronic Markets*, Proceedings of the 31st Hawaii International Conference on System Sciences, Blanning R. and King D., Hawaii, HI, January (1998), 193–201.
- [10] Soto-Acosta, P., and A. Meroño-Cerdan, *Analyzing e-business value creation from a resource-based perspective*, International Journal of Information Management. **28**, N 1, February (2008), 46–60.
- [11] Soto-Acosta, P., and A. Meroño-Cerdan, *Evaluating Internet Technologies Business Effectiveness*, Telematics and Informatics. **26**, N 2 (2009), 211–221.
- [12] Zeigler, B. P., *DEVS Today: Recent Advances in Discrete Event-Based Information Technology*, 11TH IEEE International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems. (2003).
- [13] Zeigler, B. P. and H. S. Sarjoughian, *Introduction to DEVS Modeling and Simulation with JAVA*, ACIMS, (2003), URL: <http://www.acims.arizona.edu/EDUCATION/refCourse.html>.