

# Sistema de Pedidos para el comercio electrónico

Lisette Toirac Perdomo<sup>1</sup>, Alejandro Garcés Calvelo<sup>2</sup>

<sup>1,2</sup> Departamento de Ciencia de la Computación, Universidad de Oriente  
Santiago de Cuba, 90500, Cuba

<sup>1</sup>[lissette@csd.uo.edu.cu](mailto:lissette@csd.uo.edu.cu)

<sup>2</sup>[garcescalvelo@yahoo.es](mailto:garcescalvelo@yahoo.es)

**Resumen.** El presente trabajo describe un sistema de transacciones de comercio electrónico, específicamente un Sistema de Pedidos. Este se encarga de mostrar los diferentes productos que se ofertan en una compañía y realizar todos los procesos de compras relacionados con esta actividad, teniendo en cuenta las solicitudes que realicen los clientes. Además brinda facilidades de búsqueda de productos por un tipo determinado, pues se tiene en cuenta que todos los productos estén agrupados por categorías; permite la realización de informes de ventas y control de existencia en inventario. El sistema se desarrolla a partir de una especificación formal, evitando así las inconsistencias y ambigüedades, además esto obliga a los diseñadores a pensar cuidadosamente cada aspecto del sistema. La especificación formal es escrita en el sistema de notación Z.

## 1 Introducción

Con la rápida evolución de las telecomunicaciones muchas de las actividades relacionadas con los procesos de compra y venta se realizan digitalmente. Así, si contamos con un sistema de comercio electrónico robusto y fiable que garantice su rápido acceso desde cualquier lugar del mundo, se mejorará considerablemente la competitividad global.

Muchos de los modelos que sirven de base para importantes aplicaciones del comercio electrónico, utilizan aproximaciones informales o la intuición humana. Sin embargo, por tal camino es muy difícil probar que los modelos de datos escogidos son los acertados. De ahí que la mayoría de los diseños contengan inconsistencias que se manifiestan cuando los mismos son utilizados. Las aproximaciones del diseño informal no proporcionan el grado de exactitud y fiabilidad requerido para las transacciones del comercio electrónico.

Precisamente, este trabajo describe un sistema de transacciones de comercio electrónico desarrollado a partir de una especificación formal. Este consiste en un Sistema de Pedidos el cual se encarga de realizar las actividades de compras dadas las solicitudes de los clientes, de acuerdo a un conjunto de productos que la compañía posee. La especificación formal escogida es de gran importancia pues evita las inconsistencias y ambigüedades, además obliga a los diseñadores a pensar cuidadosamente cada aspecto del sistema.

## 2 Especificación formal

En esta sección se brinda una especificación formal que constituye una modificación del modelo descrito por Ehikioya [2], la cual sirve de base para el sistema de transacciones de comercio electrónico que aquí se propone.

El sistema de transacciones que se describe en este trabajo es utilizado para mostrar los diferentes productos que se ofertan en una compañía y realizar todos los procesos de compras relacionados con esta actividad a través de un sistema de pedidos. Es importante destacar que este sistema posee una tarjeta de compra virtual en la cual se podrán adicionar y eliminar productos, los cuales una vez terminada la selección, conformaran el pedido del cliente. Además, brinda facilidades de búsqueda de productos por un tipo determinado, pues se tiene en cuenta que todos los productos estén agrupados por categorías; permite además la realización de informes de ventas y control de existencia en inventario.

Esta formalización se establece sobre la base de algunas abstracciones como por ejemplo, para referenciar al banco solo se necesita el nombre del banco y la cuenta, pues otros componentes y funciones no son especificados. Además se supone que existe otro módulo que se encarga de controlar los pedidos a los proveedores.

La especificación formal es escrita en el sistema de notación Z [4,5]. Los conceptos y modelos de datos principales de la especificación formal se describen a continuación. Estos resultan de gran importancia para poder comprender el sistema de pedidos que se presenta en la siguiente sección. En esta definición  $P(A)$  denota el conjunto potencia del conjunto  $A$ , y  $N$  representa al conjunto de los números naturales.

En este sistema se definen varios tipos aplicados al diseño, estos son:

[CHAR,EMAIL,EMAIL\_ADDRESS,FAX, DATE]

MONEY ==  $N_1$

DIGIT == 0..9

SECONDS ==  $N$

PHONE\_NUMBER == SeqDIGIT

ACCOUNT\_TYPE ::= Savings | Current

PAY\_MODE ::= Cheque | CreditCard | DebitCard | ApprovedCredit

APPROVED\_CREDIT\_STATUS ::= Available | Unavailable

SHOPPING\_CARD\_STATUS ::= Stopped | Completed

TRANSACTION\_STATUS ::= Waiting | Executing

OPERATION ::= NewOrder | CancelOrder

Entre los principales objetos encontramos:

BankAccount  $\cong$  [id\_BankAccount:  $N_1$ ; number :  $N_1$ ; balance : MONEY; type : ACCOUNT\_TYPE]

$\forall a,b: \text{BankAccount} \mid a \neq b \bullet a.\text{id\_BankAccount} \neq b.\text{id\_BankAccount}$

Existen dos tipos de cuentas: *Cuentas corrientes* y *Cuentas de Ahorro*. Si el cliente paga con cheque se decrementa la cuenta corriente, si paga con tarjeta de débito no importa el tipo de cuenta que posea.

CreditCard  $\cong$  [id\_CreditCard:  $N_1$ ; number :  $N_1$ ; balance : MONEY; limit : MONEY | balance  $\leq$  limit ]

$\forall a,b: \text{CreditCard} \mid a \neq b \bullet a.\text{id\_CreditCard} \neq b.\text{id\_CreditCard}$

Describe las tarjetas de crédito del cliente. Esta es otra forma de pago que admite el sistema.

$\text{Product} \cong [\text{id\_product} : \mathbb{N}_1; \text{name: seqCHAR}; \text{category: seqCHAR}; \text{description: seqCHAR}; \text{supplierCost: MONEY}; \text{OurCost: MONEY} \mid \text{suppliercost} > 0 \wedge \# \text{name} > 0 \wedge \text{ourcost} > \text{suppliercost} \wedge \# \text{categoria} > 0 \wedge \# \text{description} > 0]$

$\forall a, b: \text{Product} \mid a \neq b \bullet a.\text{id\_product} \neq b.\text{id\_product}$

Cada producto pertenece a una categoría, de esta forma los productos se mostrarán y buscarán por categorías. Además hay que tener en cuenta que cada producto en el inventario es único.

$\text{Supplier} \cong [\text{id\_supplier: } \mathbb{N}_1; \text{name: SeqCHAR}; \text{products: P (Product)}; \text{phone : PHONE\_NUMBER}; \text{fax : PHONE\_NUMBER}; \text{e\_mail : EMAIL\_ADDRESS} \mid \# \text{name} > 0 \wedge \text{products} \neq 0]$

$\forall a, b: \text{Supplier} \mid a \neq b \bullet a.\text{id\_supplier} \neq b.\text{id\_supplier}$

El proveedor suministra los productos que la compañía compra para luego vendérselos a los clientes. En este trabajo se asume que existe otro módulo que se encarga de realizar los pedidos al proveedor.

$\text{Bank} \cong [\text{id\_bank: } \mathbb{N}_1; \text{name: SeqCHAR}; \text{accounts: P (BankAccount)} \mid \text{accounts} \neq 0 \wedge \# \text{name} > 0]$

$\forall a, b: \text{Bank} \mid a \neq b \bullet a.\text{id\_bank} \neq b.\text{id\_bank}$

Cuando un cliente paga un pedido utilizando un cheque o una tarjeta de débito los fondos son decrementados de la cuenta bancaria. En este trabajo se asume una abstracción para el tratamiento de los bancos, donde solo se maneja el nombre del banco y las cuentas bancarias.

$\text{CreditCompany} \cong [\text{id\_CreditCompany: } \mathbb{N}_1; \text{name: SeqCHAR}; \text{cards: P (CreditCard)} \mid \text{cards} \neq 0 \wedge \# \text{name} > 0]$

$\forall a, b: \text{CreditCompany} \mid a \neq b \bullet a.\text{id\_CreditCompany} \neq b.\text{id\_CreditCompany}$

Cuando un cliente paga con tarjeta de crédito, el sistema se comunica con la compañía de crédito, para determinar si este cliente tiene el dinero suficiente para realizar el pago. Para esta operación se asume que la compañía de créditos posee medios electrónicos para poder acceder a la información de los clientes.

$\text{Client} \cong [\text{id\_Client: } \mathbb{N}_1; \text{address: SeqCHAR}; \text{name: SeqCHAR}; \text{fax: PHONE\_NUMBER}; \text{phone: PHONE\_NUMBER}; \text{e\_mail: EMAIL\_ADDRESS}; \text{defaultPayMode: PAY\_MODE}; \text{account: BankAccount}; \text{CreditCard: CreditCard} \mid \# \text{name} > 0 \wedge \# \text{address} > 0]$

$\forall a, b: \text{Client} \mid a \neq b \bullet a.\text{id\_client} \neq b.\text{id\_client}$

Los clientes interactúan con el sistema para realizar los pedidos a la compañía. Los clientes pueden tener cuentas bancarias, tarjetas de crédito o tarjetas de débito. El *modo de pago por defecto* que se define con anterioridad se utiliza cuando el cliente realiza un pedido y no especifica la modalidad de pago.

$\text{ApprovedCredit} \cong [\text{id\_Client: } \mathbb{N}_1; \text{balance: MONEY}; \text{limit: MONEY}; \text{status: APPROVED\_CREDIT\_STATUS} \mid \text{balance} \leq \text{limit}]$

$\forall a, b: \text{ApprovedCredit} \mid a \neq b \bullet a.\text{id\_client} \neq b.\text{id\_client}$

Crédito que proporciona la compañía de este sistema de pedidos a sus clientes. Cada cliente solo puede tener un único crédito aprobado por la compañía. El estado indica si está disponible este modo de pago para realizar un pedido.

ShoppingCard=[id\_ShoppingCard:  $N_1$ ; Client: Client; items: P ( Product x  $N_1$ );  
timelimit: SECONDS; status: SHOPPING\_CARD\_STATUS]

Para que el cliente pueda realizar un pedido se utiliza una tarjeta de compra, en la cual se almacena la lista de los productos a comprar. Aquí debe tenerse en cuenta que el cliente puede abandonar la operación sin indicarlo al sistema. De ahí que las tarjetas de compras posean un atributo que determina el tiempo en que se debe conservar la información. En caso que el cliente deba realizar un pedido muy grande y desee guardar los productos seleccionados para continuar con dicha selección más tarde, el estado de la tarjeta debe definirse como *Detenido* y en caso contrario como *Completado*.

Order  $\equiv$  [id\_order:  $N_1$ ; client: Client; items: P (Product x  $N_1$ ); paymode: PAY\_MODE;  
shipping: MONEY; subtotal: MONEY; tax: MONEY; total: MONEY; ShipMode:  
ShippingMode; destAddress: seqCHAR, date: DATE | #address >0  $\wedge$  items  $\neq$  0  $\wedge$   
total=subtotal+tax+shipping]

$\forall a,b: \text{Order} \mid a \neq b \bullet a.\text{id\_order} \neq b.\text{id\_order}$

Cada orden es diferente aunque sea el mismo cliente el que la realice, pues esta puede variar en cuanto a fecha, dirección de destino, tipo de pago entre otros.

ClientTransaction  $\equiv$  [id\_clientTransaction:  $N_1$ ; order: Order; action: OPERATION;  
status: TRANSACTION\_STATUS]

$\forall o: \text{Order}; t: \text{ClientTransaction} \mid t.\text{order} = o \bullet t.\text{client} = o.\text{client}$

En este objeto se representan las transacciones que realiza el Sistema. Varios clientes pueden estar conectados al Sistema de pedidos realizando operaciones, pero todas no pueden realizarse al mismo tiempo; de ahí que exista un campo estado que indica cual transacción está activa y cuales están en espera. Además se tiene en cuenta que operación está realizando la orden activa, ya que puede ser una solicitud de un *nuevo pedido* o una *cancelación de pedido* ya realizado.

Inventory  $\equiv$  [items: P (Product x  $N_1$ ); quantity: seqDIGIT; requested: seqDIGIT]

El inventario contiene la lista de todos los productos que oferta la compañía con las cantidades de cada producto y las cantidades comprometidas con pedidos realizados pero que aún no han sido pagados.

Los principales cambios realizados a la especificación descrita por Ehikioya [1] se encuentran en la definición de los siguientes objetos:

*ApprovedCredit*: Se adicionó un campo estado (status) que permite chequear la disponibilidad de este tipo de crédito en los clientes. Este campo lo controla un módulo externo relacionado con las cuentas por cobrar que de acuerdo al cumplimiento en el pago del crédito brindado al cliente, lo actualiza con los valores: Available o Unavailable.

*ShoppingCard*: En este objeto también se incluye un campo estado (status) para poder brindar soluciones ante la posibilidad de que el cliente necesite realizar pedidos muy grandes que requieran diferentes momentos de selección de productos. De esta forma un pedido inicial puede almacenarse para luego ser completado y creada la orden.

*Inventory*: En la especificación de Ehikiova [2], se define un objeto holdList que almacena los productos que han sido pedidos pero que aún no se han pagado. En este

modelo dicho objeto se une con el objeto Inventory, simplemente añadiendo un campo requested que controle la cantidad de producto comprometida pero que aún no ha sido pagada. Esta modificación evita las redundancias en el almacenamiento de la información y permite homogenizar los accesos a la información referida a la existencia de productos en inventario.

### 3 Implementación

En esta sección se discuten algunos de los principales casos de uso que han sido implementados y la arquitectura general del Sistema de Pedidos.

#### 3.1 Casos de usos

Estos casos de usos abarcan, desde la interacción de los clientes con el sistema de pedidos hasta los procesos de verificación con otros sistemas requeridos por la actividad que se esté realizando.

**Escenario del Caso de Uso:** *Solicitud de Pedido*. Un cliente realiza un pedido usando el sistema. A continuación se muestran los pasos necesarios para crear y procesar la orden:

1. Una orden se recibe a través de cualquier medio: fax, teléfono, correo electrónico, o página web.
2. La orden se convierte a su formato general.
3. Se chequea la corrección de la orden (es decir, se verifica que la información requerida este apropiadamente llena).
  - a. Si la orden no es correcta, entonces se notifica al cliente de los errores.
4. Se verifica la situación financiera del cliente.
5. Si el pago es por cheque o tarjeta de débito, se verifica que la cuenta de banco del cliente posea los fondos disponibles.
  - a. Si los fondos son insuficientes en la cuenta del banco, entonces se notifica al cliente del problema y se piden otros métodos de pago.
6. Si el pago es por la tarjeta del crédito, se chequea el crédito del cliente.
  - a. Si el crédito del cliente no es bueno, se notifica al cliente del problema y se piden otros métodos de pago.
7. Si el pago es por el crédito aceptado de nuestra compañía, se chequea el límite del crédito del cliente.
  - a. Si el límite del crédito no es suficiente para cubrir el pago, entonces se notifica al cliente del problema y se piden otros métodos de pago.
  - b. Si el cliente no ha aprobado el crédito, se notifica al cliente del problema y de dónde debe aprobarse el crédito o pedir otros métodos de pago.
8. Se registra la orden con el estado *Nueva Orden*, se inicializa la shoppingcard y se crea la transacción.
9. Se actualizan las cantidades de los productos comprometidos de la orden en el inventario.

**Escenario del Caso de Uso: *Cancelación de Pedido.*** Un cliente cancela un pedido usando el sistema. A continuación se muestran los pasos necesarios para realizar y procesar la orden:

1. Se especifica el cliente que realizó el pedido.
2. Se muestra un listado de todos los pedidos realizados por este cliente.
3. Se selecciona la orden a cancelar.
4. Se registra la orden con el estado *Cancelar Orden*.

**Escenario del Caso de Uso: *Procesar transacciones.*** El sistema constantemente se encuentra procesando las transacciones que se encuentran en espera. A continuación se muestran los pasos necesarios para el procesamiento.

1. Si no existe ninguna transacción activa se toma la siguiente transacción de la lista de transacciones que se encuentran en espera y se activa.
2. De acuerdo al modo de pago se actualiza el saldo del cliente.
  - a. Si la acción es una nueva orden entonces:
  - b. Si el pago es por cheque o tarjeta de débito, se decrementa el saldo de la cuenta en el importe total de la orden.
    - i. Si los fondos son insuficientes en la cuenta del banco, entonces se notifica al cliente del problema y se piden otros métodos de pago.
  - c. Si el pago es por tarjeta del crédito, se decrementa el balance de la tarjeta en el importe total de la orden.
    - i. Si el crédito del cliente no es bueno, se notifica al cliente del problema y se piden otros métodos de pago.
  - d. Si el pago es por el crédito aceptado de la compañía, se decrementa el balance del crédito en el importe total de la orden.
    - i. Si el límite del crédito no es suficiente para cubrir el pago, entonces se notifica al cliente del problema y se piden otros métodos de pago.
    - ii. Si el cliente no ha aprobado el crédito, se notifica al cliente del problema y de dónde debe aprobarse el crédito o pedir otros métodos de pago.
  - e. Si la acción es cancelar una orden entonces:
  - f. Si el pago es por cheque o tarjeta de débito, se aumenta el saldo de la cuenta en el importe total de la orden.
  - g. Si el pago es por la tarjeta del crédito, se aumenta el balance de la tarjeta en el importe total de la orden.
  - h. Si el pago es por el crédito aceptado de la compañía, se aumenta el balance del crédito en el importe total de la orden.
3. Se actualiza el inventario en las cantidades reales del producto y en las comprometidas.
  - a. Si la acción es una nueva orden entonces para cada producto de la orden se decrementa en el inventario la cantidad real y la comprometida en la cantidad solicitada en el pedido.
  - b. Si la acción es cancelar orden entonces para cada producto de la orden se aumenta en el inventario la cantidad real en la cantidad solicitada en el pedido.
4. Enviar la información apropiada a logística y otros departamentos.

**Escenario del Caso de Uso: *Solicitud de Pedido on line*:** Un cliente hace un pedido usando el sistema. A continuación se muestran los pasos necesarios para hacer y procesar la orden:

1. Se determina para el cliente que desea realizar el pedido si no existe alguna solicitud *No Completada* en su shoppingcard.
  - a. Si existe se brinda la posibilidad de completarla y en caso contrario se comienza un nuevo pedido.
2. Se llena la tarjeta de compra virtual seleccionando los productos deseados.
  - a. Para cada producto seleccionado se chequea su existencia en inventario y en esas cantidades.
3. Se muestra al cliente un formulario con toda la información requerida para la solicitud y sus productos seleccionados.
4. Se chequea la corrección de la orden (es decir, se verifica que la información requerida este apropiadamente llena).
  - a. Si la orden no es correcta, entonces se notifica al cliente de los errores.
5. Se verifica la situación financiera del cliente.
  - a. Si el pago es por cheque o tarjeta de débito, se verifica que la cuenta de banco del cliente posea los fondos disponibles.
    - i. Si los fondos son insuficientes en la cuenta del banco, entonces se notifica al cliente del problema y se piden otros métodos de pago.
  - b. Si el pago es por la tarjeta del crédito, se chequea el crédito del cliente.
    - i. Si el crédito del cliente no es bueno, se notifica al cliente del problema y se piden otros métodos de pago.
  - c. Si el pago es por el crédito aceptado de nuestra compañía, se chequea el límite del crédito del cliente.
    - i. Si el límite del crédito no es suficiente para cubrir el pago, entonces se notifica al cliente del problema y se piden otros métodos de pago.
    - ii. Si el cliente no ha aprobado el crédito, se notifica al cliente del problema y de dónde debe aprobarse el crédito o pedir otros métodos de pago.
6. Se registra la orden con el estado *Nueva Orden*, se inicializa la shoppingcard y se crea la transacción.
7. Se actualizan las cantidades de los productos comprometidos de la orden en el inventario.

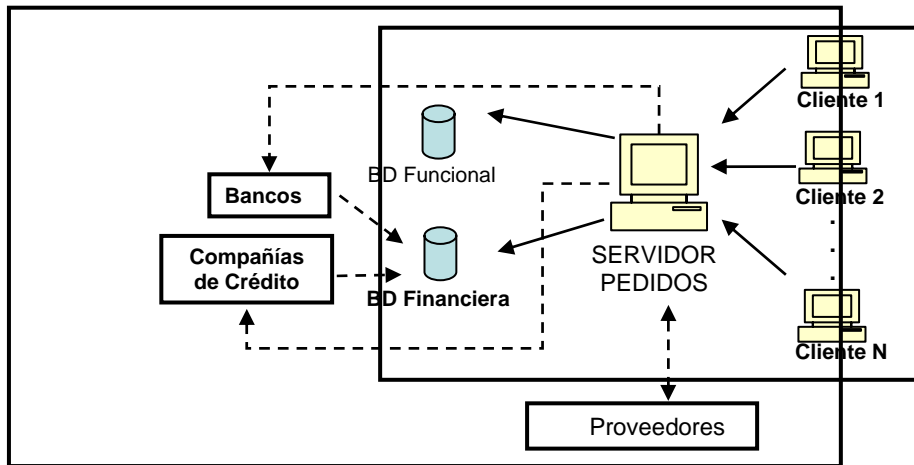
Cuando se realizan algunas de las operaciones: pedido o cancelación de ordenes, además de actualizar las tarjetas de los clientes se solicita una transferencia al banco por el monto total de la operación a la cuenta del sistema (ya sea para aumentar o decrementar el saldo).

### 3.2 Arquitectura del sistema

La arquitectura funcional del Sistema de Pedidos y sus relaciones de comunicación con los diferentes módulos externos se muestran en la Fig. 1.

Como puede verse, en la misma se describen dos tipos de aplicaciones principales, primero el Sistema de Pedidos propiamente dicho, el cual en esta figura se representa en

el cuadro interno y el segundo está formado por los módulos asociados a organizaciones externas y que se corresponden con los Bancos, Compañías de Crédito y Proveedores.



**Fig. 1.** Arquitectura Funcional del Sistema de Pedidos.

El Sistema de Pedidos posee tres componentes básicos: las bases de datos, el servidor de pedidos y los clientes. Se utilizan dos bases de datos: una con la información financiera relativa a los clientes y otra que abarca toda la información funcional, es decir, existencia de productos en inventario, solicitudes de pedidos, etc. Se decidió trabajar de esta forma debido a que la Base de Datos Financiera es controlada por el sistema de pedidos y por módulos externos asociados a los Bancos y Compañías de Crédito, entre otros, que garantizan la actualización de los balances en las tarjetas de créditos, tarjetas de débito, cheques, cuentas bancarias, etc. ante modificaciones que sufran por operaciones externas a este sistema de pedidos.

Se asume también que existe un módulo que controla los créditos ofrecidos por la compañía y que se encarga de actualizar el estado del crédito para que este sistema pueda determinar si puede aceptar este modo de pago.

Se supone que existe un módulo externo encargado de controlar la relación que existe entre este sistema y los proveedores, garantizando cantidades suficientes de productos para satisfacer los pedidos. Los clientes realizan las solicitudes al servidor, el cual se encarga de procesarlas, acceder a las Bases de Datos y retornar al cliente la información solicitada.

Existe una gran variedad de herramientas que pudieran utilizarse para la implementación de este sistema de pedidos. Teniendo en cuenta la necesidad de que este sistema sea de fácil acceso desde cualquier punto, se decidió desarrollar una aplicación Web y lograr así una mejor conexión con los consumidores.

Dentro de la Web existen varios lenguajes etiquetados que pudieran aplicarse a este sistema, sin embargo se seleccionó el Procesador de Páginas de Hipertexto (PHP) debido a que es un lenguaje de alto nivel, multiplataforma, que permite la creación de páginas dinámicas de una manera rápida y fácil; posee un lenguaje muy potente, con una integración a más de 8 servidores de HTTP y acceso a 20 tipos diferentes de Bases de Datos; posee además un diseño modular de fácil ampliación y licencia abierta [1].



Con respecto al almacenamiento de los datos se decidió utilizar como gestor de Base de Datos a Microsoft SQL Server [3] por todas las ventajas que brinda entre las que encontramos que es una plataforma de desarrollo fácil y abierta: integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server. Además es un gestor de base de datos que contiene de forma integrada la posibilidad de generar contenido HTML de forma automática y su rendimiento sobre Internet está publicado, ofreciendo registros importantes.

## **4 Conclusiones**

El desarrollo de aplicaciones de tecnologías de información en Internet puede introducir grandes avances en el desarrollo de las compañías. El sistema de pedidos aquí descrito, garantiza un rápido y eficiente acceso al mismo, facilitando a los clientes esquemas sencillos para la solicitud de pedidos. Además este trabajo ayuda a los programadores a diseñar e implementar sistemas de comercio electrónico utilizando una metodología de especificación formal.

## **Referencias**

1. Bakken S. S., Schmid E. : Manual de PHP (2004)
2. Ehikiova S.A.: A Formal perspective to modelling electronic commerce transactions. Revista Colombiana de Computación, Vol 2. (Dic 2001) 21-40
3. Hussey P.: Designing Efficient Application for Microsoft SQL Server. Microsoft Press. (1997)
4. Spivey J. M.: Introducing Z: A specification language and its semantics. Cambridge University Press (1988)
5. Spivey J. M.: The Z notation: A reference manual. 2da Edition, Prentice Hall International Series in Computer Science (1992)