"SWITCH TRANSACCIONAL BASADO EN EL PASO DE MENSAJES XML"

Diana Béjar Domínguez[1], Elena Luzuriaga Castro[2], Fabricio Echeverría Briones[3]

[1]Ingeniera en Computación 2006
[2]Ingeniera en Computación 2006
[4]Director de Tesis. Ingeniero en Computación, Escuela Superior Politécnica del Litoral, 1998. Profesor de ESPOL desde 2000

ABSTRACT

The Transactional Switch gets communicate several enterprises, being possible to relation different data models on line, this is possible thanks to facilities that brings the Remote Programming, in this case, using TCP, because this protocol permits communicate the Transactional Switch with other enterprises through an application that each enterprise has in their servers, with an exhaustive analysis in order to get an efficient interface, relation between two different data models is possible, XML is the best tool in order to get that. XML is used to save and analyze transactions in real time; XML permits give a specific format and functionality to a string. This article has the main steps that were made to get this solution.

RESUMEN

El Switch Transaccional logra la comunicación de varias empresas pudiendo relacionar modelos de datos diferentes en línea, esto es posible gracias a las facilidades que brinda la programación remota mediante el protocolo TCP que permite comunicar el Switch Transaccional con varias empresas a través de una aplicación que se alojara en el servidor de cada una, mediante un amplio análisis para conseguir la interfaz mas eficiente, el relacionar diferentes modelos de datos es posible y su respectivo almacenamiento y análisis en tiempo real depende de utilizar las ventajas que presenta el XML al permitir dar una funcionalidad especifica a un formato de datos. Este artículo contiene un recorrido breve por las diferentes etapas que llevaron a cabo el desarrollo de esta solución a las transacciones entre empresas.

INTRODUCCIÓN

Los negocios entre empresas han evolucionado con el paso del tiempo y el avance tecnológico, el negocio entre empresas a través de Internet toma el nombre de Business to Business (B2B), permitiendo que las empresas realicen transacciones entre si a través de un sitio confiable y sin intermediarios.
Poco a poco este tipo de negocio se ha ido introduciendo con gran fuerza en las organizaciones mundiales, pero aún en nuestro país no hay una confianza total en los negocios B2B, es por esto, que el propósito de este articulo es hacer un gran aporte en línea para dar a las empresas ecuatorianas una opción diferente y confiada de negociar.
Este artículo se basa en crear un Switch Transaccional genérico bajo la presentación de un middleware, que permita que empresas con diferentes plataformas y modelos de negocio puedan interactuar y realizar negocios entre sí.

CONTENIDO

Descripción del problema a resolver

Una transacción entre empresas va mas allá de aquello que primero surge (compra y venta), va mas allá del intercambio de información. En esta época hay cosas que no cambian, como por ejemplo el negocio entre empresas, esto es ejecutado con la ayuda de personas que sirvan de intermediarios entre los involucrados. El fin de este Switch Transaccional es seguir la línea tradicional de los mismos con la diferencia de hacerlos de forma automática sin personas que deban encargarse de realizar todos aquellos pasos previos para realizar una transacción.
De forma indirecta, el Switch Transaccional ayuda a la empresa que lo utilice a reducir costos, sean estos de movilización, suministros, incluso sueldos de los intermediarios que tradicionalmente

realizaban estos trámites y conseguir la información requerida de manera automática cada cierto intervalo de tiempo.
La principal ventaja de este Switch Transaccional es comunicar empresas con diferentes modelos de negocios y plataformas, haciendo posible que la transacción deseada se realice con éxito, puesto que su característica principal es permitir adaptar modelos de negocio diferentes.

Definición del Switch Transaccional
El Switch Transaccional es un middleware transaccional que permite el funcionamiento de B2B en nuestro medio pues es capaz de hacer transacciones que reflejan los resultados, obteniendo en las transacciones los depósitos de datos de las empresas participantes.

El Switch Transaccional utiliza "Programación Remota", lo que permite acceder mediante el protocolo TCP a aplicaciones que a su vez se conectan a sus respectivos servidores de base de datos

El Switch Transaccional maneja el concepto de servicios, que son transacciones configuradas por cualquier empresa que desee brindar un servicio a otras empresas que deseen suscribirse al mismo.

Por esto las empresas participantes tendrán dos roles claramente definidos dentro del Switch Transaccional, empresa que brinda servicios y empresa que se suscribe a un servicio.

Un servicio consta de una o más operaciones (Propiedad que brinda la empresa que lo ofrece al momento del registro), cada operación es una transacción realizada a la base de datos de sí misma, pudiendo consultar información o brindar la posibilidad de permitir insertar o actualizar datos.

Una suscripción es también un servicio, pero este es configurado en base al servicio elegido mediante este proceso, en esta etapa también intervienen el registro de las operaciones que son punto principal para que los resultados obtenidos sean los correctos.

Cada empresa tendrá la responsabilidad en el momento de la configuración de registrar sus estructuras de datos en una operación elegida, teniendo como diferencia que la empresa que se suscriba a un servicio tendrá la responsabilidad de asociar sus estructuras de datos con aquellas pertenecientes a la empresa a la cual se suscribió.

Definición de la arquitectura del Switch Transaccional
Para definir la arquitectura del Switch Transaccional se lo hará desde tres puntos diferentes.

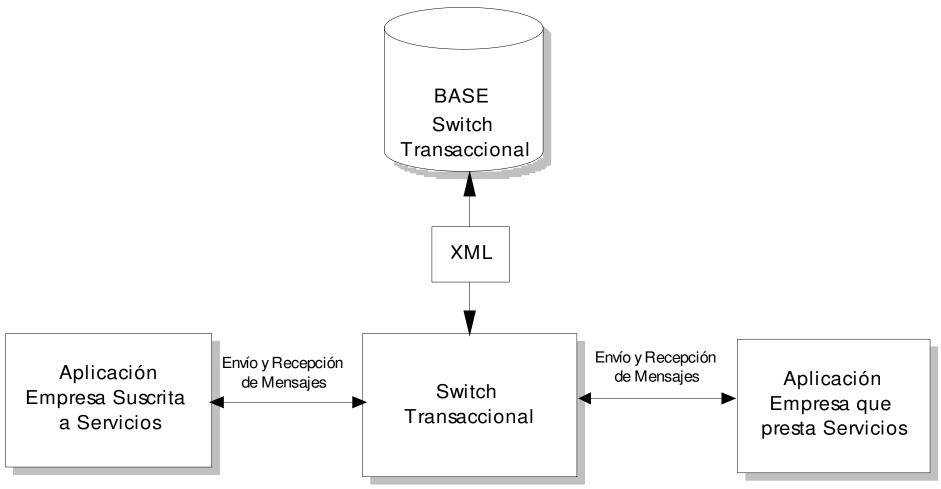

Figura No. 1
Arquitectura del Switch Transaccional

Definición de la arquitectura del Switch Transaccional desde el punto de vista de Empresa que presta servicios

- Una empresa que va a brindar un servicio empieza registrando sus datos generales dentro del Switch Transaccional, a continuación la empresa queda lista para registrar su servicio.

- La empresa debe definir el servicio dando al mismo, nombre, descripción, el número de operaciones a utilizar así como el nombre y funcionalidad de cada operación

- La empresa debe configurar cada operación según el orden y funcionalidad dados en el registro del servicio, para esto debe ingresar las estructuras correspondientes de acuerdo al tipo de operación que ingresó.

- Según las estructuras de datos ingresadas el Switch Transaccional contará con la sentencia transaccional a ser utilizada en la aplicación de la empresa.

Definición de la arquitectura del Switch Transaccional desde el punto de vista de Empresa que se suscribe a un servicio

- La empresa que suscribe un servicio debe registrar sus datos generales para luego pasar a elegir un servicio.

- Según el servicio elegido a la empresa observará un resumen del mismo, proporcionándole el nombre, descripción y funcionalidad por operación.

- La empresa debe ingresar el tiempo que ejecutan las transacciones.

- Se presenta a la empresa las operaciones que constan en el servicio al que se está suscribiendo, por lo que debe configurar sus operaciones en base a éstas.

- La empresa debe ingresar las operaciones transaccionales que van a relacionar con el servicio elegido.

- Por cada operación a registrar la empresa debe ingresar sus estructuras de datos.

- En la configuración presentan los campos que ha ingresado así como los que corresponden a la operación actual, por lo que debe asociar los campos, siendo responsable de la forma en la cual van a interactuar sus datos con los datos proporcionados por la empresa que brinda el servicio.

Definición de la arquitectura del Switch Transaccional desde el punto de vista del motor

El Switch Transaccional tiene registrado en su base de datos servicios a ofrecer y suscripciones a los mismos.

- El Switch Transaccional inicia su funcionamiento elaborando un cronograma de los servicios a ejecutar diariamente, ingresando todos éstos a una lista que registra el orden de ejecución. El Switch Transaccional elije el servicio a ejecutar, ejecutando uno a la vez.

- Una vez que empieza la ejecución de un servicio determina el número de operaciones y las ejecuta por operaciones según el orden registrado en la base de datos.

- Por cada par de operaciones a analizar, el Switch Transaccional determina cual de éstas es la que proporciona el origen o fuente de datos y cual va a ser el destino de los mismos.

- El Switch Transaccional hace un requerimiento a la base de datos, captura la sentencia transaccional a través del formato XML generado para la operación, el resultado del análisis del mismo es enviado a través de un URL a la aplicación que retornará los datos.

- Con la información obtenida el Switch Transaccional utiliza el formato XML que permite asociar los datos de la fuente con el destino, para hacer la respectiva transformación de acuerdo a los tipos de datos validados en la etapa de configuración o registro.

- El Switch Transaccional consulta otra vez a la base de datos para obtener el formato XML de la empresa destino que le permite formar el mensaje que será enviado a la aplicación de ésta a través de un URL con los datos ordenados, los cuales surgen del resultado de la transformación de los datos fuente en destino.

- El Switch Transaccional termina la ejecución registrando en el Historial los datos de la operación realizada y actualizando en el Cronograma de Ejecución la fecha de la próxima vez que el servicio sea requerido.

Análisis General
Para realizar el análisis del Switch Transaccional se realiza un análisis global del mismo así como un análisis más detallado.

Del análisis Global se obtuvo el siguiente gráfico que explica que una empresa accede al Switch Transaccional para suscribirse a un servicio brindado por otra empresa. El Switch Transaccional envía mensajes con requerimientos a las aplicaciones de ambas empresas. Y de forma transparente realiza las operaciones obteniendo y actualizando datos de cada una (Empresa suscriptora y Empresa que presta servicios) según sea necesario.

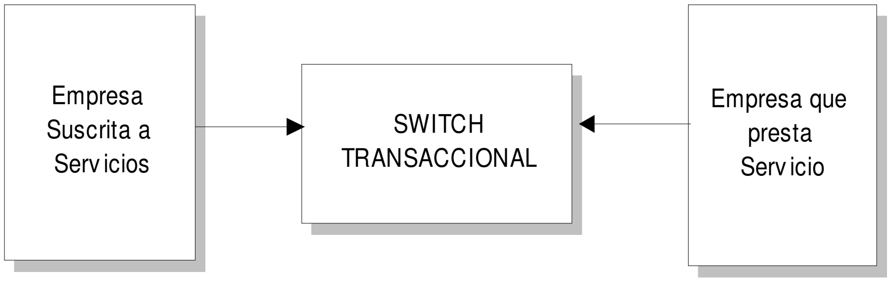

Figura No. 2
Análisis General del Switch

El Switch Transaccional consta de un conjunto de sentencias a bajo y alto nivel que van a ser llevadas a cabo en documentos XML:

Las sentencias a bajo nivel son formatos XML que van a permitir registrar las operaciones básicas de una Base de Datos (SELECT, INSERT, DELETE o UPDATE).

Por otro lado las sentencias a alto nivel son las operaciones compuestas que van a dar como resultado de la configuración de los servicios y las suscripciones a los mismos, tanto suscripciones como servicios se van a formar de una o más sentencias a bajo nivel.

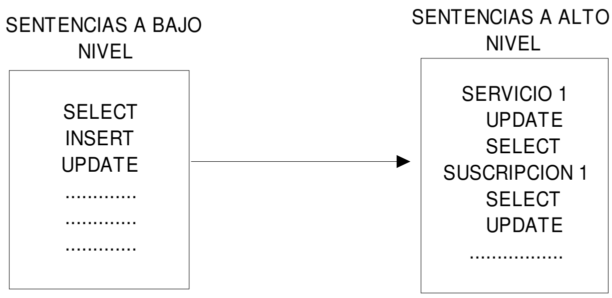

Figura No. 3
Sentencias de bajo y alto nivel

Cada servicio consta de una o más operaciones y una suscripción, se adapta a un servicio y a su número de operaciones. Al momento de ejecutar un servicio (una suscripción), el Switch Transaccional consulta las operaciones que intervienen tanto para la suscripción como para el servicio y obtiene una operación de cada uno a la vez.

Las operaciones elegidas van a ser buscadas en la base de datos, donde está la información necesaria para armar el mensaje a la aplicación remota a través de la plantilla XML correspondiente que va a interactuar a lo largo de la ejecución de esas dos operaciones relacionadas, mediante estas plantillas se van a realizar la ejecución de las sentencias correspondientes a cada empresa que interviene en la transacción. El Switch Transaccional se encarga de identificar cual es la empresa que contiene los datos que se van a enviar a la otra empresa, estas consultas y envíos de datos se realizan mediante aplicaciones que se van a encontrar funcionando en cada empresa, por lo que el Switch Transaccional debe identificar la aplicación fuente y la aplicación destino

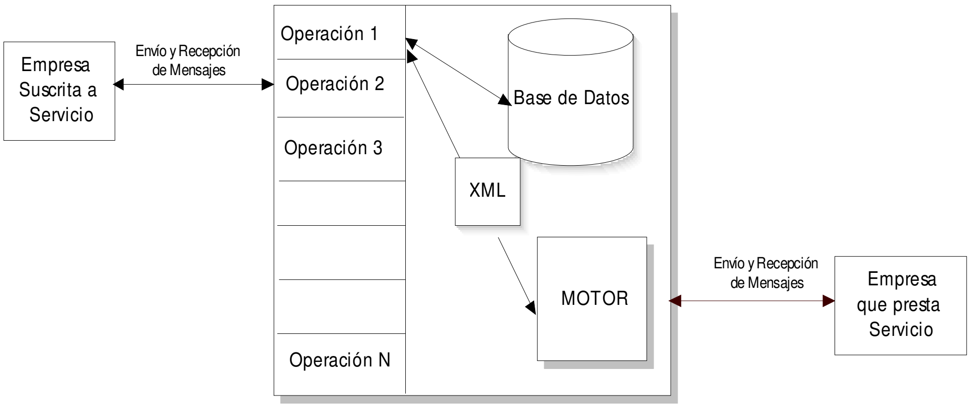

Figura No. 4
Análisis detallado del Switch

Estos formatos XML representan de forma general operaciones estándares para configurar de acuerdo a los propósitos del servicio, estas operaciones son: INSERT, UPDATE y SELECT. Pero es necesario otro formato XML que registre el comportamiento de los datos de una empresa con respecto a la otra, esta plantilla es la de ASOCIACION que permite relacionar los datos que intervienen en las operaciones que pertenecen a un servicio, en otras palabras permite traducir o transformar los datos fuentes en los datos destino.
Los formatos o plantillas que el Switch Transaccional utiliza son los siguientes:

Formato XML SELECT

```
<OPER_SELECT>
  <SELECT>none
<CAMPO IdCampo="Orden del Campo" TipoDato="tipo de dato">Nombre del Campo</CAMPO>
</SELECT>
  <FROM>
    <TABLA IdTabla="Orden de la Tabla">Nombre de la Tabla</TABLA>
  </FROM>
  <WHERE>
    <CONDICIONES IdCondicion="Orden de la Condición">Condición para la consulta
</CONDICIONES>
  </WHERE>
</OPER_SELECT>
```

Figura No. 6
Formato SELECT

Este formato tiene como función representar sentencias SELECT que en una base de datos son aquellas encargadas de consultar información y presentarla de acuerdo a como ha sido codificada la sentencia.

En esta plantilla los elementos representan parte de la sentencia SQL SELECT de la siguiente forma:
<OPER SELECT>: Es el encabezado del formato.
<SELECT>: Este elemento indica el inicio del comando, permite colocar propiedades como none (un select simple, DISTINCT (Sólo se recupere un registro de cada campo) o COUNT (Cuenta de los registros).
<CAMPO>: Su función es registrar los campos que serán consultados asignándoles un identificador y guardando el tipo de dato que le corresponde.
<FROM>: Inicia la segunda parte de la sentencia en la cual se indica las tablas a ser consultadas.
<TABLA>: Registra las tablas que van a ser consultadas.
<WHERE>: Inicia la tercera parte de la sentencia que contiene las condiciones de la consulta de datos.
<CONDICIONES>: Registra las condiciones o restricciones aplicadas a la consulta de datos.

Formato XML INSERT

```
<OPER_INSERT>
  <INSERT_INTO>
    <TABLA IdTabla="Orden de la Tabla">Nombre de la Tabla</TABLA>
    <CAMPOS>
            <CAMPO IdCampo="Orden del Campo" TipoDato="tipo de dato">Nombre del
            Campo</CAMPO>
    </CAMPOS>
  </INSERT INTO>
</OPER_INSERT>
```

Figura No. 7
Formato INSERT

Las sentencias INSERT se representan a través de esta plantiílla. En una base de datos estos comandos son aquellos encargados de ingresar nueva información al servidor en la Tabla que se le indique.

Los elementos que representan parte del comando INSERT son los siguientes:
<OPER INSERT>: Es el encabezado del formato.
<INSERT_INTO>: Este elemento indica el inicio de la sentencia.
<TABLA>: Registra la tabla en la cual se van a insertar datos.
<CAMPOS>: Indica el inicio del registro de los campos que se van a insertar en la tabla indicada anteriormente.
<CAMPO>: Su función es registrar los campos que serán insertados asignándoles un identificador y guardando el tipo de dato que le corresponde.

Formato XML UPDATE

```
<OPER_UPDATE>
  <UPDATE>
    <TABLA IdTabla="Orden de la Tabla">Nombre de la Tabla</TABLA>
    <SET>
            <CAMPO IdCampo="Orden del Campo" TipoDato="tipo de dato">Nombre
            del Campo</CAMPO>
    </CAMPOS>
    </SET>
<WHERE>
      <CONDICIONES IdCondicion="0">Edad = 12</CONDICIONES>
    </WHERE>
  </UPDATE>
</OPER_UPDATE>
```

Figura No. 8
Formato UPDATE

UPDATE permite como su nombre lo indica actualizar, en este caso aquellos registros que cumplan con las condiciones o restricciones presentadas por la sentencia.

Los elementos utilizados para representar sentencias SQL UPDATE son los siguientes:
<OPER_UPDATE>: Es el encabezado del formato.
<UPDATE>: Este elemento indica el inicio de la sentencia.
<TABLA>: Registra la tabla que se va a actualizar.
<SET>: Inicia la segunda parte de la sentencia en la cual se indica los campos que se van a actualizar.
<CAMPO>: Registra los campos a actualizar.
<WHERE>: Inicia la tercera parte de la sentencia que contiene las condiciones para la actualización de datos.
<CONDICIONES>: Registra las condiciones o restricciones requeridas en la sentencia

Formato XML DELETE

```
<OPER_DELETE>
  <DELETE>
  <FROM>
    <TABLA IdTabla="Orden de la Tabla">Nombre de la Tabla</TABLA>
  </FROM>
    <WHERE>
      <CONDICIONES IdCondicion="0">Edad = 12</CONDICIONES>
    </WHERE>
  </DELETE>
```

Figura No. 9
Formato DELETE

Para eliminar registros de una base de datos se utiliza la sentencia DELETE, en este caso se eliminan aquellos registros que cumplan con las condiciones o restricciones presentadas por la sentencia.

Los elementos utilizados para representar sentencias SQL DELETE son los siguientes:
<OPER_DELETE>: Es el encabezado del formato.
<DELETE>: Este elemento indica el inicio de la sentencia.
<FROM>: Este elemento que es el paso previo a colocar la tabla en donde algunos registros serán eliminados.
<TABLA>: Registra la tabla en la que se va a eliminar.
<WHERE>: Inicia la tercera parte de la sentencia que contiene las condiciones para la actualización de datos.
<CONDICIONES>: Registra las condiciones o restricciones requeridas en la sentencia

Formato XML ASOCIACIÓN DE DATOS

```
<ASOCIACIÓN>
  <RELACIÓN>
    <FUENTE Tipo_Dato= ipodatos>Nombre del  Campo o
Concatenación</FUENTE>
    <DESTINO Tipo_Dato= ipodatos >Nombre del Campo Destino
    </DESTINO>
  </ RELACIÓN >
</ASOCIACIÓN>
```

Figura No. 11
Formato ASOCIACION

Este formato de Asociación de datos permite realizar la transformación de los datos fuente con los datos destino, está diseñado de la siguiente forma:
<ASOCIACION>: Es el encabezado del formato.
<RELACION>: Este elemento indica el inicio de una relación de los datos fuentes con los datos destino
<FUENTE>: Contiene el campo o campos concatenados que van a ser obtenidos desde la fuente.
<DESTINO>: Contiene el campo del destino que adquiere valores luego dedo de la consulta a la fuente.

CONCLUSIONES

- Es de suma importancia tener claro el concepto del proyecto a realizar y no asumir su nombre literalmente sino buscar todas las implicaciones que este pueda tener. La palabra Transacción puede no significar lo mismo para todas las personas y este hecho resulto crítico en cuanto a la elaboración de esta Tesis, pues al principio se le dio a Transacción el significado de operaciones de compra y venta, cuando su significado va más allá de operaciones financieras. Una transacción implica intercambio de información crítica, informaciones en las cuales esta involucrado el dinero pero no precisamente de la forma interpretada anteriormente, sino en la necesidad de adquirir cierta información de la manera más rápida y sin muchos trámites.

- La planificación, parte del total entendimiento del problema a resolver, tener ideas claras es lo más importante, porque por muchos conocimientos que se tengan si no se cumplen los requerimientos establecidos todo el trabajo realizado no tendrá los resultados esperados.

- Una vez claro el producto a desarrollar es importante establecer correctamente los objetivos, puesto que la fase de Análisis depende completamente de ellos. De la misma forma debe delimitarse los alcances y limitar las soluciones. Tener claro las partes que formarán el sistema y el comportamiento exacto de las mismas. La utilidad que brinda el lenguaje UML es vital en esta etapa pues ayuda a reconocer cada proceso que debe analizarse así como sus actividades y con que procesos debe relacionarse.

- La etapa de Diseño sirve para definir como hacer el Sistema, es la etapa donde se da solución a aquellos planteamientos hechos en la etapa de Análisis. De esta etapa se puede concluir que es necesario buscar todas las soluciones posibles a un problema por muy pequeño o fácil que parezca, muchas veces la primera solución es la más óptima, aunque esta sea una solución compleja o muy sencilla debe analizarse a fondo para no perder tiempo al tener que regresar de la fase de Implementación a ésta o peor aún a la de Análisis porque la solución elegida no satisface o no cumple completamente con los objetivos planteados inicialmente.

- La plataforma .NET es una buena opción para llegar a las soluciones pues brinda tantas funcionalidades que es muy posible encontrar más de una solución para un mismo problema.

- En la etapa de desarrollo la información recopilada es de vital importancia, esta etapa debería centrarse sólo en codificación y pruebas, pero por errores previos se debe regresar a fases anteriores. En esta tesis en un principio se planteó el uso de Web Services pero al momento de implementar esta tecnología se encontró que no era la solución adecuada pues no se encontró una forma de referenciar a los mismos dinámicamente, por lo que se tuvo que investigar otra tecnología que se adapte al diseño planteado originalmente.

- Una tecnología que cumplió con los objetivos del Switch Transaccional fue .NetRemoting la cual permite realizar programación distribuida permitiendo comunicar a un punto con otro a través de un servidor y teniendo como puerta de comunicación una dirección URL. Los Web Services son muy útiles también, pero la desventaja fue no encontrar la forma de acceder a ellos dinámicamente, pero si el diseño fuese para empresas específicas hubiese sido la solución más próxima.

- El XML resultó ser una herramienta de gran utilidad pues mediante él se pueden realizar muchas operaciones que en programación normal hubiesen requerido más tiempo de implementación. El XML brinda su utilidad en poder contar con nodos los cuales puedan recorrer de una forma más rápida que si el archivo fuese un simple documento de texto en el cual se hubiese tenido que leer carácter por carácter, lo cual inclusive incrementaría el tiempo de ejecución.

REFERENCIAS

a) JEFF PROSISE, Programming Microsoft .Net, 2002.
b) FABIO ARCINIEGAS A., Guía de desarrolladores XML, 2002
c) TOM ARCHER, Inside C#, 2002
d) CRAIG LARMAN, UML y Patrones – Introducción al análisis y diseño orientado a objetos. Primera Edición. 1999