Sistema de Administración de Materias Curriculares (SAMI)

Gudiño Santiago D.
Torres Cristina E.
Bernal Iván M.

***Resumen*— Se presenta una breve visión de las tecnologías ASP.NET (Active Server Pages .NET), Servicios Web y la Plataforma de Servicios de Reportes SQL, enfocada a la integración de las mismas, para la implementación de un sistema distribuido para el Internet. El sistema desarrollado empleando estas tecnologías se denomina SAMI (Sistema de Administración de Materias Curriculares), el mismo que permite la administración de trabajos y pruebas de materias curriculares. La información de la asignación de cursos a profesores y estudiantes, se extrae de manera automática de la base de datos del sistema existente SAE. Se presentan reportes con información y estadísticas de períodos, cursos, profesores, estudiantes, trabajos y pruebas. Además, se emplea un mecanismo de seguridad basado en roles de usuario mediante la implementación de una extensión de seguridad personalizada de la Plataforma de Servicios de Reportes SQL, desarrollada en lenguaje C#.**

***Términos para indexación* — ASP.NET, SAMI, Servicios de Reportes SQL, Servicios Web.**

I. INTEGRACIÓN DE LAS TECNOLOGÍAS

PARA poder presentar la integración de las tecnologías en cuestión, a continuación se presentan conceptos básicos de cada una de ellas.

Los Servicios Web son colecciones de funciones que se presentan como una sola entidad y pueden ser publicados en una red, como el Internet, y usados por otros programas. Son una muestra clara de la evolución de los sistemas distribuidos, debido a que ofrecen alta capacidad de interoperabilidad entre sistemas, debido a que emplean estándares, tales como: el *Protocolo Simple de Acceso a Objetos* (*SOAP, Simple Object Access Protocol*), el *Lenguaje de Marcas Extensible* (*XML*, *eXtensible Markup Language*) y HTTP (Hypertext Transfer Protocol).

Por otro lado, la tecnología *ASP.NET* presenta una plataforma de programación desarrollada por Microsoft, para la construcción de aplicaciones web, mediante un modelo de programación orientado a objetos y una nueva estructura que permite crear aplicaciones seguras, escalables y estables [1].

ASP.NET brinda ventajas en relación al modelo original de ASP (Active Server Pages .NET); esto se debe a que se beneficia de todas las características proporcionadas por la Infraestructura .NET e incluye nuevas particulares que permiten a los desarrolladores generar aplicaciones web mucho más sólidas.

Entre las principales ventajas de ASP.NET se pueden mencionar: mejoras en rendimiento (emplea técnicas de caché), compatibilidad con diversos lenguajes de programación, seguridad (brinda un conjunto de clases para realizar autenticación y autorización de usuarios), integración con XML, detección de las capacidades del navegador (los componentes pueden automáticamente producir una respuesta basándose en el tipo de navegador), facilidades para el desarrollador (proporciona un conjunto de controles web y permite separar en archivos diferentes, la lógica del programa del HTML y el interfaz gráfico), etc.

ASP.NET permite el uso de ADO.NET para el acceso a datos, esta última tecnología se caracteriza por brindar al programador una capa de abstracción que oculta las diferencias entre los distintos proveedores de datos, y que incluye objetos prefabricados y funciones que brindan un fácil acceso a datos; además es parte de la Infraestructura .NET, por lo que se beneficia de todas las características brindadas por la misma.

La *Plataforma de Servicios de Reportes SQL* es una iniciativa de Microsoft, que permite la administración eficiente de reportes a través de una arquitectura integrada. La Plataforma de reportes se provee como un aditamento a la plataforma de bases de datos SQL y fue liberada a inicios del año 2004 [2]. En general, la Plataforma de Servicios de Reportes SQL brinda una colección completa de servicios, herramientas e interfaces de programación de aplicaciones que soportan una infraestructura estructurada y organizada para el desarrollo de reportes.

Los reportes son generados en base al Lenguaje de Definición de Reportes (*RDL*, *Report Definition Language*), a partir de múltiples fuentes de datos (SQL Server, OLE DB, Open Database Connectivity (ODBC), Oracle) y pueden ser visualizados en varios formatos, tales como: HTML, PDF (Portable Document Format), Excel, XML, TIFF (Tag Image

Este proyecto se realizó en La Escuela Politécnica Nacional (EPN), en el Departamento de Electrónica, Telecomunicaciones, y Redes de Información.

S. D. Gudiño participó en el proyecto por la Escuela Politécnica Nacional (e-mail: santogudi@hotmail.com).

C. E. Torres participó en el proyecto por la Escuela Politécnica Nacional (e-mail: ctorresjaramillo@hotmail.com).

I. M. Bernal trabaja en la Escuela Politécnica Nacional en el Departamento de Electrónica, Telecomunicaciones, y Redes de Información, Ladrón de Guevara E11-253, Quito-Ecuador (teléfono: 5932-2507-144; fax: 5932-2547-175; e-mail: imbernal@mailfie.epn.edu.ec).

File Format) y CSV (Comma Separated Values).

Una de sus características más importantes de la esta plataforma de reportes es que está diseñada con una arquitectura de Servicios Web, lo cual le permite alcanzar altos niveles de conectividad e interoperabilidad de sistemas.

El desarrollo de la integración de Servicios Web, la tecnología ASP.NET y la Plataforma de Servicios de Reportes SQL, tiene como objetivo crear aplicaciones personalizadas, que cumplan con los requerimientos de un entorno determinado. Además, mediante la integración de estas tecnologías, es posible crear aplicaciones que se beneficien de todos los servicios y ventajas brindadas por cada una de ellas.

Existen dos ambientes claramente definidos para el desarrollo e implementación de aplicaciones que integren las tecnologías antes mencionadas, estos son: una Intranet o el Internet.

A. Integración de las tecnologías en una Intranet

En el caso de una Intranet, los Servicios Web, la tecnología ASP.NET y la Plataforma de Servicios de Reportes SQL, pueden integrarse para la presentación y administración de reportes con información organizacional o de negocios, por medio de una aplicación web personalizada, que brinde una funcionalidad diseñada específicamente para la red interna de una entidad.

La información organizacional se relaciona con los datos concernientes al funcionamiento interno de una entidad, como por ejemplo: horarios de trabajo, organización de reuniones, asignación de responsabilidades específicas, etc. Por otro lado, se encuentra la información de negocios; por ejemplo, en el caso de una empresa de venta de mercadería, será la información de ventas, compras, balances, precios al público, stock de productos, etc.

Debido a que, por defecto, la plataforma de reportes utiliza la seguridad integrada de Windows como método de autenticación, se presentan facilidades para su rápida incorporación a los sistemas manejados dentro de la red interna de las organizaciones, y para el caso de la integración con Aplicaciones Web ASP.NET, será beneficioso y natural el empleo del mismo método de autenticación.

El empleo de la autenticación mediante Windows puede emplear la estructura de la organización, manejada por un controlador de dominio (Directorio Activo, Active Directory), para el establecimiento de las políticas de seguridad en cuanto a la autenticación y a la autorización de usuarios y/o grupos.

El acceso a los reportes por medio de peticiones SOAP y peticiones URL al Servicio Web de la plataforma de reportes, puede ser incorporado en la lógica de las aplicaciones web nuevas o existentes.

Por medio de las peticiones SOAP, se puede tener acceso a toda la funcionalidad (petición de reportes, administración de contenido, suscripciones, caché, etc.) del Servidor de Reportes; por otro lado, el acceso URL está limitado a la petición de reportes y recursos.

Las peticiones de reportes mediante SOAP derivan en la pérdida de las características interactivas que pueden configurarse en los reportes, y que están disponibles mediante acceso URL [3].

Para abstraer de mejor forma la integración de las tecnologías en una Intranet, se puede visualizar el ejemplo de la Fig. 1, donde existe un Controlador de Dominio que se encarga del manejo de políticas de seguridad (autorización y autenticación) de la red.

El Servidor de Reportes expone su funcionalidad a través de un Servicio Web, el mismo que es consumido por una Aplicación Web ASP.NET que personaliza sus funciones de acuerdo a las necesidades de la empresa, e incorpora funcionalidad adicional como el procesamiento de información (cálculos, edición de datos, etc.). Los reportes que se manejan en este ambiente tienen como fuentes de información las bases de datos de la empresa, y la definición de los mismos se almacena en la base de datos del Servidor de Reportes. Para cuestiones de administración del Servidor de Reportes, se emplea un elemento que es parte de la plataforma de reportes denominado el Administrador de Reportes.

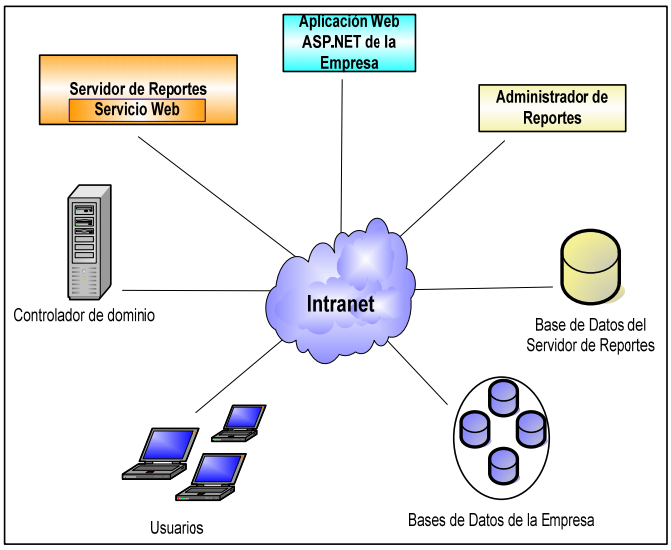

Fig. 1. Integración de las tecnologías en una Intranet

B. Integración de las tecnologías en el Internet

Debido a que la plataforma de reportes brinda soporte natural para el manejo de la seguridad integrada de Windows, a través de un controlador de dominio en ambientes de redes internas, debe tenerse en cuenta ciertas consideraciones para soportar el manejo de reportes a través del Internet.

En un ambiente de Internet, la seguridad es un ámbito de mucha importancia, por este motivo, es meritorio utilizar el API brindado por la plataforma de reportes, para desarrollar un complemento de seguridad (extensión personalizada de seguridad) que permita establecer y controlar las políticas de seguridad específicas en cuanto a la autenticación y autorización de usuarios.

Debido a que se busca incorporar el manejo de reportes en una aplicación web, dicha aplicación también debe considerar la seguridad a utilizarse. Debido a que el ambiente es el Internet, lo más recomendable será el empleo de la autenticación por medio de formularios, donde los usuarios al realizar peticiones a un recurso privado determinado, se redireccionan a un formulario que solicita sus credenciales, y solo en caso de que éstas sean válidas, los usuarios accederán al recurso solicitado.

De manera similar a lo antes mencionado en el ambiente de reportes en una Intranet, es posible tener acceso SOAP y URL al Servidor de Reportes, con las ventajas y limitaciones que cada uno posee.

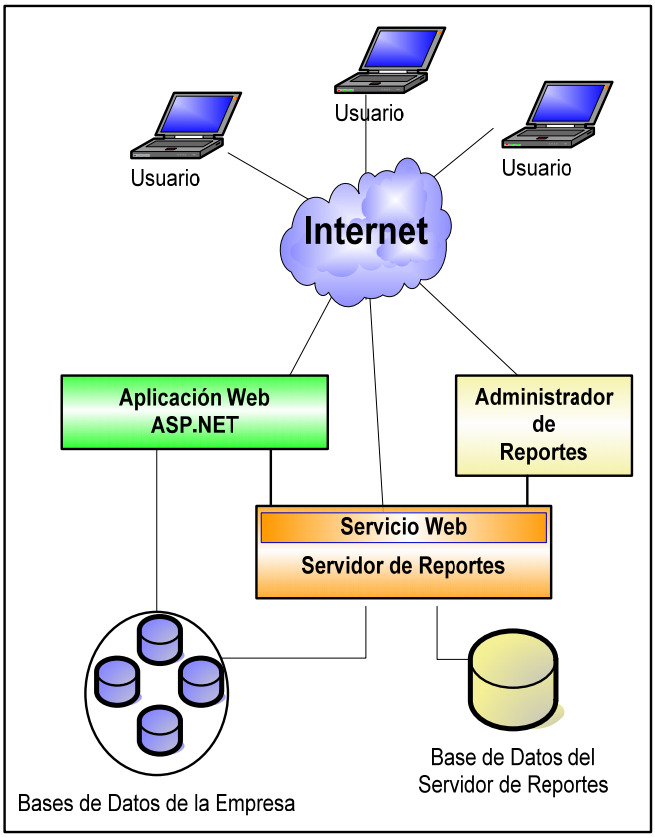

Fig. 2. Integración de las tecnologías en el Internet

En la Fig. 2 es posible observar la integración de las tecnologías en el Internet, donde la principal diferencia con el ejemplo en un ambiente de Intranet es que aquí no es posible emplear un controlador de dominio para el manejo de políticas de seguridad del sistema, por lo que el Servidor de Reportes maneja individualmente sus políticas de seguridad, por medio de su complemento de seguridad. Los otros elementos cumplen las mismas funciones que en el ejemplo de la Fig. 1.

Este tipo de integración se detallará a continuación, por medio de un sistema desarrollado para el Internet, que se beneficia de todas las tecnologías mencionadas y que incorpora a la plataforma de reportes una extensión de seguridad por formularios.

II. SISTEMA DE ADMINISTRACIÓN DE MATERIAS CURRICULARES (SAMI)

El objetivo principal del sistema desarrollado es presentar un ejemplo de la integración de las tecnologías descritas anteriormente en las secciones anteriores. Para este propósito, a continuación se presentan los requerimientos, el diseño y la implementación de un sistema distribuido que permite la administración de materias curriculares de la Carrera de Electrónica y Redes de Información de la Escuela Politécnica Nacional, con la finalidad de facilitar la tarea del docente en cuanto a la administración de trabajos y pruebas durante un periodo curricular. Además, el sistema presenta reportes actualizados de la información de materias, profesores, estudiantes, trabajos y pruebas. Para hacer referencia a este sistema, en lo posterior se empleará el nombre *SAMI* (*Sistema de Administración de Materias currIculares*).

Una de las grandes facilidades que brinda el sistema al docente, es proveer de forma automática, información académica de semestres como: listas de estudiantes, listas de profesores y su asignación de cursos. El sistema obtiene esta información de la base de datos del *Sistema de Administración Estudiantil (SAE)* y carga dicha información en las tablas correspondientes de la nueva base de datos creada para el sistema.

A. *Requerimientos*

Para que el Sistema SAMI permita la administración de materias curriculares, es necesario que cumpla con ciertos requerimientos en cuanto a funcionalidad; a continuación se citan los de mayor relevancia:

- Creación de periodos e importación automática de datos del Sistema SAE.
- Administración de trabajos y pruebas.
- Entrega de trabajos vía web.
- Visualización de reportes.

Es importante recalcar que la funcionalidad ofrecida por el Sistema SAMI se expone a través de una Aplicación Web ASP.NET, por lo que los usuarios tendrán un acceso sencillo y amigable a través de un navegador web.

B. *Diseño*

En esta sección se presentan las consideraciones de diseño del Sistema SAMI, en cuanto a su arquitectura, bases de datos y seguridad; además, se realiza el diseño de las capas lógicas de la Aplicación Web ASP.NET, la misma que en lo posterior se la denominará *SAMI Web*. Por último, se muestra un diagrama de secuencias, que permitirá complementar el diseño, al analizar la secuencia de eventos de determinadas peticiones.

1) *Arquitectura*

Para diseñar la arquitectura del Sistema SAMI es necesario considerar que los elementos empleados deben interactuar, de tal forma que brinden un sistema estructurado y funcional que

permita cubrir con los requerimientos expuestos en la sección anterior.

En la Fig. 3, es posible observar la arquitectura que se empleó en la implementación del Sistema SAMI, donde se puede apreciar la integración de las tecnologías mencionadas; además se puede visualizar el uso del servidor web, proporcionado por IIS.

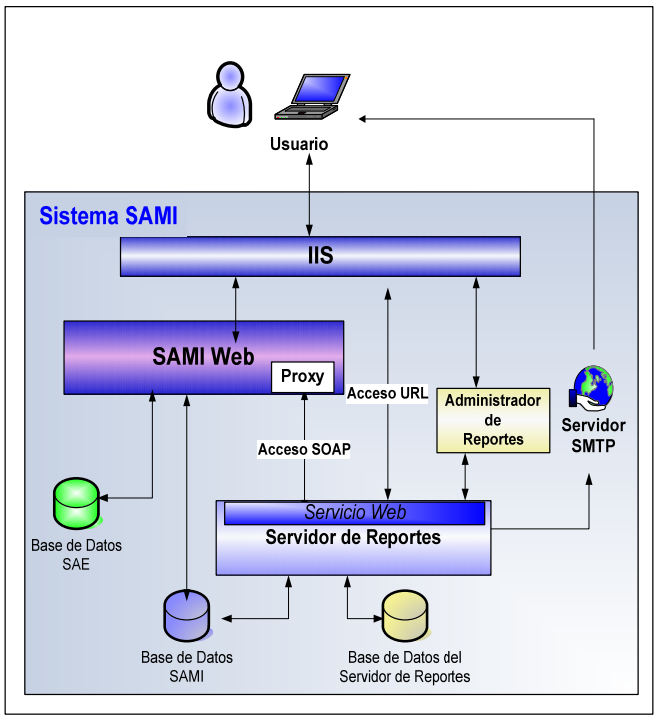

Fig. 3. Arquitectura del Sistema SAMI

Uno de los elementos fundamentales del Sistema es la SAMI Web, debido a que expone la funcionalidad del sistema a sus usuarios a través de Formularios Web ASP.NET. Se puede apreciar en la Fig. 3, que la SAMI Web debe interactuar directamente con las Bases de Datos SAE y SAMI, cabe recalcar que la interacción con la base de datos del Sistema SAE, será únicamente de lectura (con el fin de obtener la información a transferir a la base de datos SAMI). Para el caso de la base de datos SAMI, el acceso será tanto de escritura como de lectura; esto se debe a que toda la información de los usuarios (profesores y estudiantes) y la concerniente a la administración de trabajos y pruebas, será almacenada y podrá ser accedida a través de esta base de datos.

Otro elemento muy importante es el Servidor de Reportes, que es la unidad lógica funcional principal de toda la Plataforma de Servicios de Reportes SQL. Para poder emplear su funcionalidad desde la SAMI Web, es necesario que dicha aplicación implemente una clase proxy que permita acceder, mediante el protocolo SOAP, a la funcionalidad (principalmente procesos de autenticación y creación de suscripciones) expuesta por el Servicio Web del Servidor de Reportes.

El Servidor de Reportes tiene acceso a la base de datos SAMI y a su propia base de datos. En la base de datos del Servidor de Reportes se almacena la definición de los reportes publicados, los mismos que a través de la extensión de procesamiento de datos SQL, invocarán la información solicitada a la base de datos SAMI.

Como se visualiza en la Fig. 3, la arquitectura de este sistema incluye el acceso URL al Servidor de Reportes, debido a que el sistema pretende aprovechar las ventajas que brinda este tipo de acceso, principalmente respecto a la configuración de características interactivas de los reportes. Se puede visualizar en dicha figura, que en el acceso URL se establece una conexión directa entre el cliente y el Servidor de Reportes.

Otro de los componentes empleados en esta arquitectura es el Servidor SMTP, el cual debe cumplir con el objetivo de entregar, automáticamente, notificaciones a los correos electrónicos de los estudiantes, ya sean estos de: creación de pruebas o trabajos, entregas de notas, etc.; por este motivo, es necesario que el Servidor de Reportes, a través de su extensión de entrega vía correo electrónico, emplee el Servidor SMTP para enviar las notificaciones requeridas.

El Sistema SAMI debe permitir el uso del Administrador de Reportes, con la finalidad de realizar ciertas tareas administrativas del Servidor de Reportes, tales como: visualización del estado de las notificaciones a los usuarios, asignación de políticas de acceso de usuarios a los reportes, etc.

A pesar de que el Sistema SAMI comprende una arquitectura integrada por un conjunto de elementos, se pretende que toda su estructura y procesos internos sean transparentes al usuario; por este motivo, lo único que los diferentes usuarios (visitantes, profesores, estudiantes y administradores) necesitarán para acceder al sistema es un navegador web.

2) Bases de Datos

El sistema emplea una base de datos propia; accede a la base de datos del Sistema SAE, y a la base de datos del Servidor de Reportes (indirectamente a través del Servidor de Reportes). En lo que respecta a la base de datos SAMI, es necesario apoyarse en la información que el Sistema SAMI requiere para ofrecer sus servicios, como: listas de estudiantes, listas de profesores, asignación de cursos, información de trabajos y pruebas, etc.

Como se mencionó anteriormente, el sistema debe importar cierto conjunto de información de la base de datos SAE, por lo que para diseñar las tablas que almacenen dicha información, es necesario conocer ciertos aspectos de administración de información del Sistema SAE, que se presentan a continuación:

- El Sistema SAE emplea una base de datos por carrera y periodo curricular.
- Las bases de datos mantienen la misma estructura de tablas y relaciones en todos los periodos curriculares.
- Las bases de datos del Sistema SAE, emplean

un *Servidor de Base de Datos SQL Server 7.0*.

La particularidad del Sistema SAMI es el manejo de la información de semestres curriculares en una sola base de datos, para todos los periodos curriculares, a diferencia del Sistema SAE, que maneja una por cada periodo curricular. Esto brinda ventajas al sistema, debido a que almacenar información acumulada de periodos en una única base de datos, es más práctico y funcional; además, para el control de información innecesaria, el sistema da la posibilidad de eliminar la información completa de periodos.

Para el manejo de trabajos y pruebas, se emplean tablas específicas que almacenan información propia a la naturaleza de los mismos. Dichas tablas tienen las correspondientes relaciones para poder asociar la información de asignación de estudiantes a cursos a las respectivas pruebas y trabajos.

Para el manejo de la información de usuarios y roles se emplean tres tablas, éstas son: Usuario, Rol y RolUsuario. En la Fig. 4, se aprecia un diagrama de relaciones, en donde las tablas funcionan de forma independiente a las tablas de las secciones anteriores, debido a que la administración de seguridad es tratada de forma aislada a los datos del sistema.

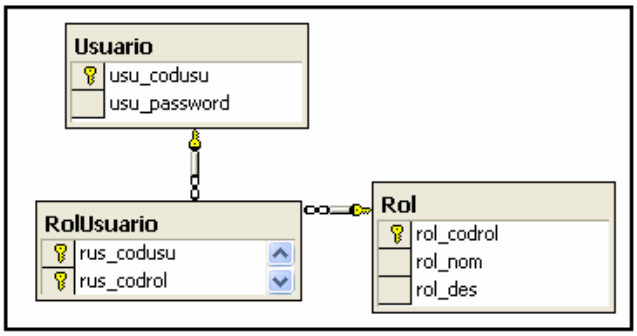

Fig. 4. Diagrama de campos y relaciones de las tablas con información de usuarios y roles

3) Seguridad

En base a los requerimientos del sistema establecidos, el sistema emplea un esquema de seguridad de autenticación y autorización basado en roles de usuario. Para ello, el Sistema SAMI almacena la información correspondiente a nombres de usuario, contraseñas y asignación de roles, en la base datos del sistema, en las tablas correspondientes.

Con la finalidad de contar con un entorno sencillo y amigable de administración, se establece como nombre de usuario al mismo código único que maneja y entiende el Sistema SAE, tanto para profesores como para estudiantes, este valor no podrá ser cambiado para facilitar la tarea administrativa. Así también, para asegurar la información de las contraseñas suministradas por los usuarios, se utiliza el algoritmo *3DES* (Triple Data Encryption Standard)[2] como método de encripción para almacenar dicha información; el usuario contará con la opción para realizar el cambio de su respectiva contraseña. El algoritmo 3DES también es usado para encriptar cierta información en el archivo de configuración de la SAMI Web.

Se definen entonces cuatro roles: visitante, profesor, estudiante y administrador. La SAMI Web maneja una parte pública que podrá ser accedida por cualquier usuario y otra parte privada para lo cual es necesario que el usuario suministre credenciales.

En lo que respecta a la base de datos del sistema, se emplea la autenticación SQL como método de autenticación, para lo cual se configura una cuenta en el servidor SQL. Estas credenciales son suministradas automáticamente por la SAMI Web del sistema, cuando requiera ejecutar tareas de lectura y escritura de información.

Debido a que el Sistema SAMI embebe la visualización de reportes a través de Formularios Web ASP.NET, es necesario tener en cuenta que la Plataforma de Servicios de Reportes SQL utiliza el método de autenticación y autorización Windows, como método predeterminado de seguridad, el mismo que es empleado en ambientes de Intranets; razón por la cual, es necesario diseñar e implementar una extensión de seguridad personalizada para la plataforma de reportes, que permita contar con un tipo de autenticación y autorización que pueda manejarse en el ambiente de Internet.

En este punto es necesario realizar una aclaración de la seguridad que maneja el Servidor de Reportes y la seguridad que manejará la SAMI Web del sistema SAMI. Ambos modelos de seguridad son acoplados de manera que el autenticarse en la Aplicación ASP.NET, implica también autenticarse en el Servidor de Reportes, de forma totalmente transparente al usuario. Esta estrategia es necesaria, ya que mediante un control de servidor ASP.NET personalizado, se prevé la invocación directa de reportes (acceso URL) por parte del usuario, al Servidor de Reportes.

Extensión de Seguridad de Autenticación por Formularios

La plataforma de reportes presenta gran flexibilidad y capacidad de crecimiento definiendo un conjunto de interfaces de programación, por medio de las cuales su funcionalidad puede ser extendida para brindar soluciones más robustas y complejas. Éste es el caso de las Extensiones de Seguridad; que por medio de las interfaces de programación de autenticación y autorización, es posible definir e implementar un modelo de seguridad personalizado que se acople de mejor manera a los requerimientos específicos de los usuarios.

Es así que se desarrolló una extensión de seguridad de autenticación por formularios, que permite suministrar credenciales de usuario para acceder a la funcionalidad del Servidor de Reportes. Así también, se prevé el manejo de roles de usuario que permitan definir permisos grupales a los recursos de la plataforma de reportes. Tanto las credenciales de usuario como la asignación de roles, son almacenadas en la base de datos del Sistema SAMI; por este motivo, la extensión de seguridad requiere tener acceso a la información

[2] ***3DES***: algoritmo de encripción simétrica que emplea tres iteraciones sucesivas del algoritmo estándar de encripción *DES*.

de esta base de datos, con la finalidad de obtener los datos necesarios para realizar el proceso de autenticación y autorización. Cabe recalcar que con la extensión de seguridad por formularios se pretende exponer al Servidor de Reportes a través de la SAMI Web.

Para el diseño de la extensión de seguridad que emplea el Sistema SAMI, se tomó en cuenta dos ejemplos muy ilustrativos, uno publicado por Microsoft [4] y otro de *Teo Lachev* [5]. Ambos ejemplos sirvieron como material de apoyo para el diseño e implementación de las necesidades específicas del sistema.

Similar al método de autenticación por formularios para Aplicaciones ASP.NET, la nueva extensión de seguridad valida y autentica las credenciales suministradas por los usuarios; luego de lo cual se emite un cookie, que se envía al usuario para que éste lo transmita en la cabecera de las posteriores peticiones al Servidor de Reportes. Así también, en las posteriores peticiones de los usuarios autenticados, la extensión de seguridad chequea los permisos que los usuarios poseen y en base a ellos, autoriza las acciones requeridas.

SAMI Web

La SAMI Web del sistema SAMI será expuesta al Internet, por lo que es necesario contar con un modelo se seguridad apropiado que proteja al sistema y asegure su óptimo funcionamiento. Por este motivo, se empleó la autenticación por formularios como método de autenticación de usuarios y la asignación de roles como método de autorización de grupos de usuarios.

Es importante considerar el hecho de que el sistema administra información de diferentes tipos de usuarios (visitantes, profesores, estudiantes y administradores), motivo por el cual es también necesario asegurar la información privada de cada uno de ellos (excepto para el perfil visitante) para evitar, que por ejemplo, estudiantes puedan obtener información que solo el profesor maneja.

La aplicación maneja *dos cookies*, uno suministrado por el Servidor de Reportes (empleando la extensión personalizada de seguridad por formularios) previa autenticación, y otro suministrado por la SAMI Web; los mismos que son enviados al cliente web.

En cuanto al perfil público, debe considerarse el hecho de que el visitante también puede acceder a reportes con información general de periodos, cursos, profesores, estudiantes, trabajos y pruebas. Por esta razón, el sistema considera el uso de una cuenta de visitante para realizar la autenticación con el Servidor de Reportes que permite el acceso a dichos recursos; esta consideración implica un procedimiento en el que se realiza la autenticación en el Servidor de Reportes (empleando SOAP) con una cuenta visitante pero no se produce ningún tipo de autenticación en la SAMI Web.

En cuanto a la estructura misma de la SAMI Web, ésta está diseñada en base a un modelo modular de capas: datos, negocios y presentación, que abstraen detalles de toda la implementación de la aplicación web. Con respecto a la organización de directorios, se definió una estructura funcional que separa lógicamente la información accedida por cada uno de los usuarios. Esta estructura de directorios permitió implementar políticas de seguridad específicas para cada uno de ellos, a través de los archivos de configuración web (Web.config).

4) Diagramas de secuencia

Cuando un usuario realiza una petición por un recurso privado, la aplicación web automáticamente redirige al usuario a un formulario de autenticación para que éste suministre sus credenciales y se inicie el proceso de autenticación.

Un aspecto importante a recalcar, es que inicialmente la SAMI Web es quien se autentica con el Servidor de Reportes y recibe el cookie correspondiente; pero ésta a su vez, es la encargada de retransmitir ese cookie de autenticación al cliente. Adicionalmente, la aplicación genera su propio cookie de autenticación que también lo enviará al cliente; de esta manera el mismo queda autenticado tanto en el Servidor de Reportes como en la SAMI Web. Ambos cookies serán empacados automáticamente por el navegador web en la cabecera de futuras peticiones, para ser analizados por la aplicación web y por el Servidor de Reportes, con el fin de comprobar los permisos concedidos y permitir o negar las acciones requeridas. En la Fig. 5 se presenta todo el procedimiento interno del sistema.

Después del proceso de autenticación, el usuario envía los cookies suministrados por la aplicación web (cookies: SAMI y Servidor de Reportes), en cada una de las subsiguientes peticiones.

La Fig. 6 presenta la interacción que se produce cuando un usuario invoca un formulario web que contiene de manera embebida la visualización de un reporte publicado en el Servidor de Reportes. Como ya se mencionó, se empleará un control de servidor ASP.NET para tal propósito. La aplicación web envía al usuario el formulario web pedido, el cual posee en su código HTML, una referencia a un reporte publicado en el Servidor de Reportes. Es así que el navegador web automáticamente genera la petición del reporte al Servidor de Reportes (acceso URL), quien chequea el cookie de autenticación y los permisos concedidos. Finalmente, si la petición es autorizada, se genera el reporte y es entregado al usuario.

La Fig. 7 explica la interacción del perfil visitante con el Sistema SAMI. En este caso, se realizará un proceso de autenticación de una cuenta visitante con el Servidor de Reportes; este procedimiento será ejecutado únicamente en la primera petición realizada por el usuario al sistema. Debido a que el perfil visitante accede a información pública, no requiere ningún tipo autenticación con la SAMI Web, pero sí deriva en la autenticación de la cuenta visitante con el Servidor de Reportes; procedimiento totalmente transparente al usuario.

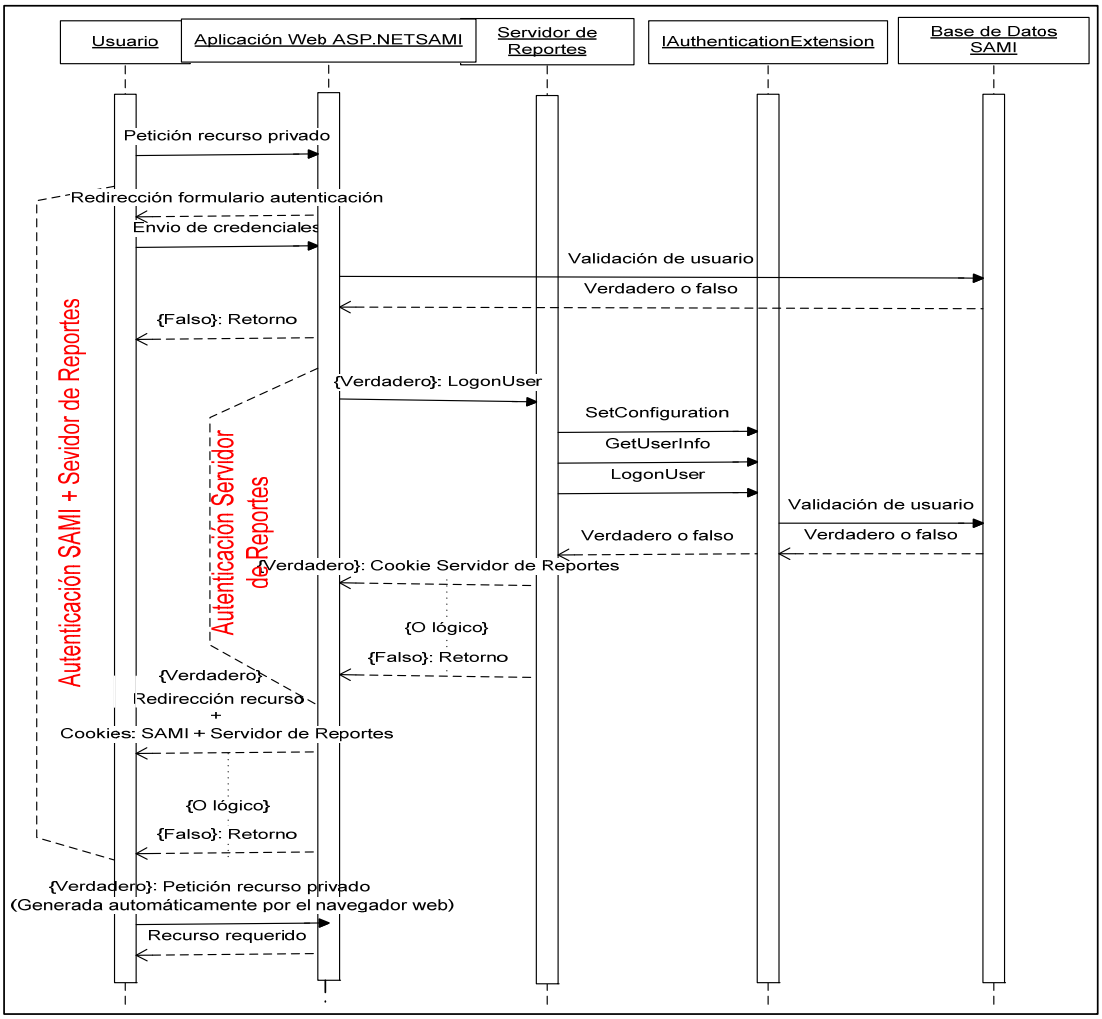

Fig. 5. Diagrama de secuencia de autenticación de usuarios

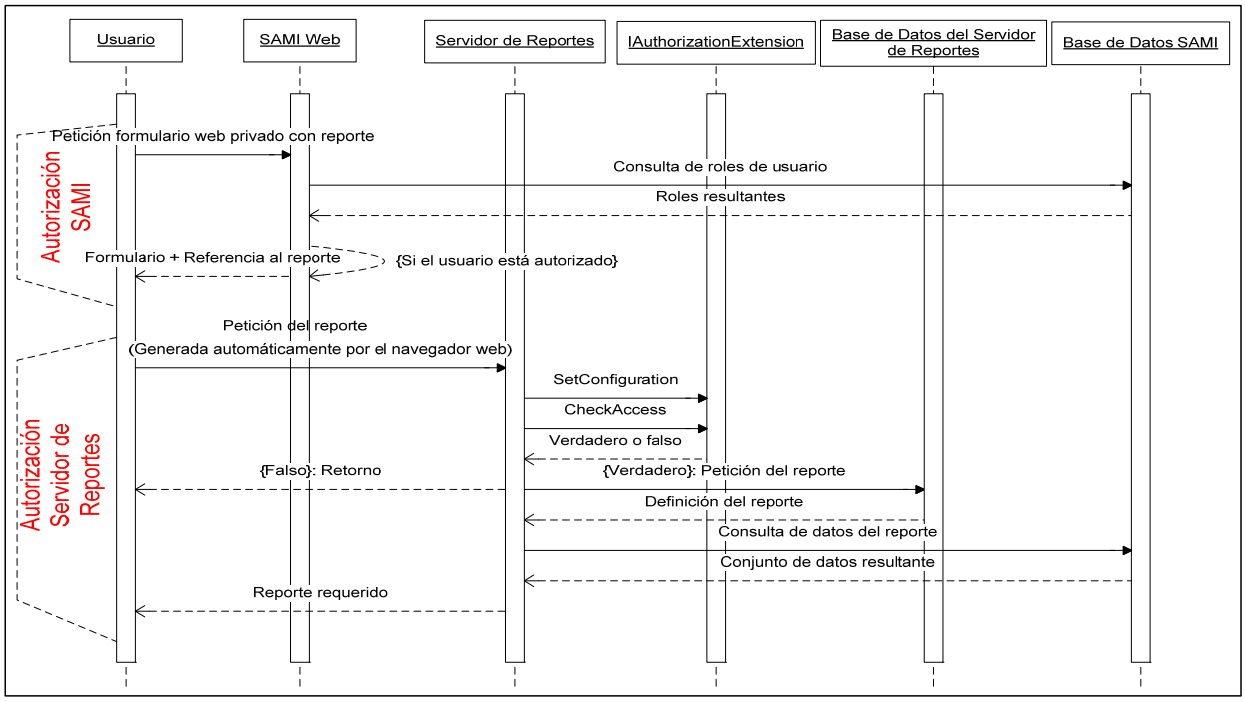

Fig. 6 Diagrama de secuencia de autorización de usuarios

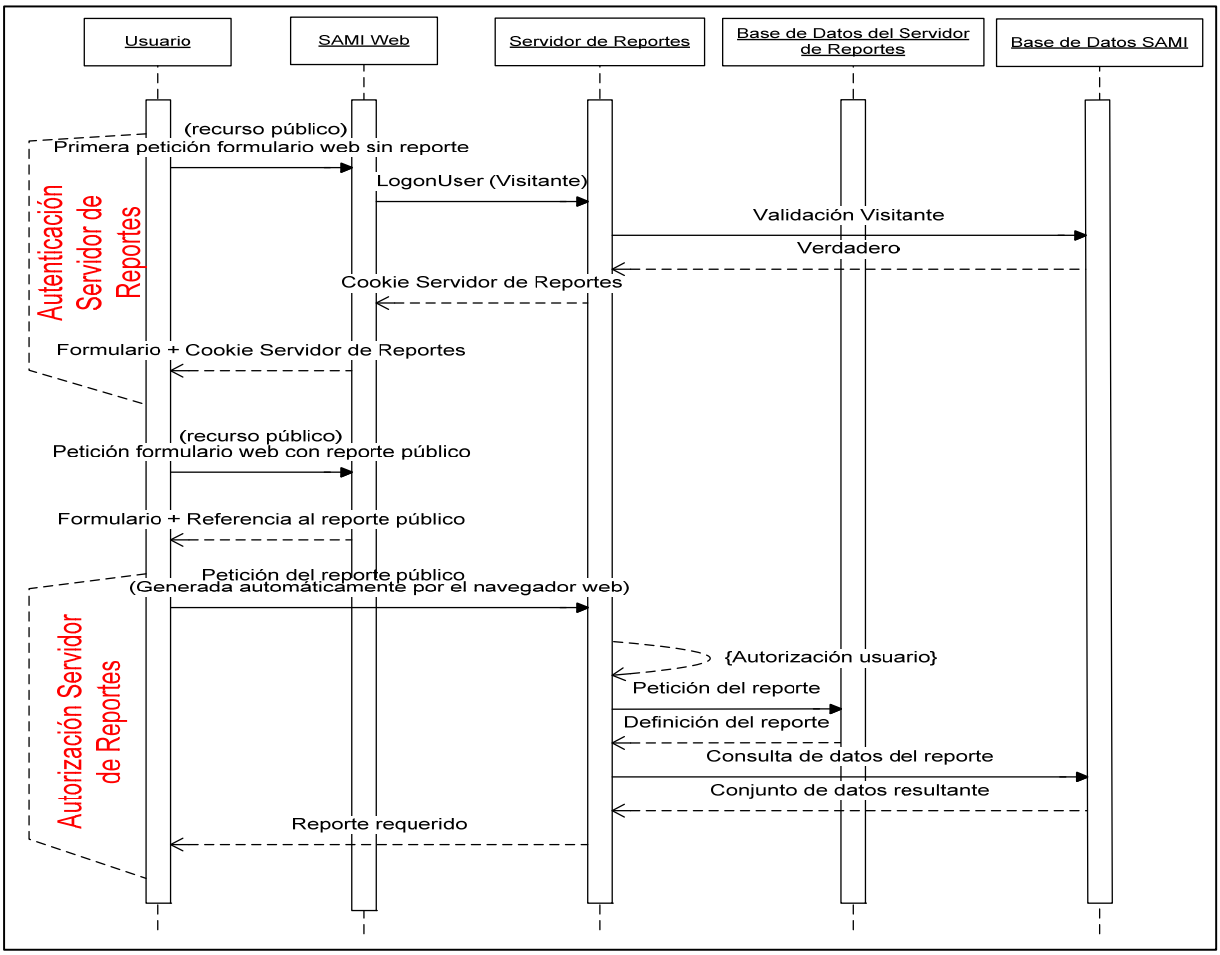

Fig. 7 Diagrama de secuencia de autenticación y autorización de visitantes

C. *Implementación y escenario de pruebas*

Para generar el entorno de prueba, se empleó el programa *Microsoft Virtual PC 2004* [6], para simular una máquina servidor con Windows Server 2003 Enterprise Edition, que albergará todos los componentes de programa que requiere el Sistema SAMI. La máquina que hospeda a este programa de emulación, es una máquina real con Windows XP Professional Edition como sistema operativo. El objetivo de emplear este programa es poder simular una red con dos computadoras distintas (empleando una sola máquina física), donde la una hace las veces de servidor y la otra de cliente.

En la máquina virtual (cuyo nombre será "server") se procedió a la instalación de los siguientes elementos computacionales: Windows Server 2003 Enterprise Edition [7], SQL Server 2000 Enterprise Edition [8], IIS, ASP.NET, SMTP Virtual Server, VS2003, SQL Server Reporting Services Enterprise Edition.

Como herramienta de apoyo, se empleó el conjunto de componentes de programa para .NET de Enterprise Library [9], que permiten reutilizar componentes específicos definidos para el manejo de archivos de configuración, acceso a datos, criptografía, seguridad entre otros.

Para el caso del Sistema SAMI se emplearon los componentes para el manejo de acceso a datos y el manejo de archivos de configuración especializados, motivo por el cual fue necesaria también la instalación de dichos componentes en la máquina "server". Estos componentes están incluidos en el conjunto de herramientas que proporciona *Enterprise Library* (ensamblados y ejecutables).

1) *Plataforma de Reportes*

Para la implementación de la extensión de seguridad, se definieron dos clases: *AuthenticationExtension* (que implementa la interfaz *IAuthenticationExtension*) y *AuthorizationExtension* (que implementa la interfaz *IAuthorizationExtension*). Se empleó además los componentes de programa de Enterprise Library para el manejo de archivos de configuración del Sistema SAMI, para el acceso a datos y para valores de configuración del sistema.

La fase final de la implementación de la extensión de seguridad en el Servidor de Reportes, se realiza a través de la edición de sus archivos de configuración para su declaración, presentados en la Tabla I.

TABLA I
ARCHIVOS DE CONFIGURACIÓN DEL SERVIDOR DE REPORTES

Archivo	Descripción
RSReportServer.config	Archivo principal de configuración del Servidor de Reportes.
Rssrvpolicy.config	Archivo de configuración de las políticas de seguridad del Servidor de
Web.config	Archivo de configuración del Servicio Web del Servidor de Reportes.

TABLA II
ARCHIVOS DE CONFIGURACIÓN DEL ADMINISTRADOR DE REPORTES

Archivo	Descripción
RSWebApplication.config	Archivo principal de configuración del Administrador de Reportes.
rsmgrpolicy.config	Archivo de configuración de las políticas de seguridad del Administrador de Reportes.
Web.config	Archivo de configuración del Administrador de Reportes.

La extensión de seguridad de autenticación por formularios debe ser también declarada en los archivos de configuración del Administrador de Reportes, los cuales son presentados en la Tabla II.

En lo referente a los reportes, el Sistema SAMI define reportes públicos (accedidos por todos los usuarios) y reportes privados (accedidos por estudiantes, profesores y administradores). El diseño e implementación de los mismos se lo realizó a través del Diseñador de Reportes empleando VS2003.

2) SAMI Web

Para consumir los servicios expuestos por el Servidor de Reportes desde la Aplicación Web ASP.NET, es necesario recordar que el acceso al mismo puede ser de dos formas: SOAP y URL.

Para el caso del acceso SOAP se implementó una clase Proxy denominada *ReportServerProxy* y para el acceso URL se empleó un componente de servidor ASP.NET denominado ReportViewer.

La presencia de la clase ReportServerProxy radica en el hecho de que la clase generada automáticamente por VS2003, carece de la funcionalidad necesaria para la administración de cookies que requiere el Sistema SAMI. Los métodos sobrescritos por la clase ReportServerProxy para el manejo de la funcionalidad mencionada, son: GetWebRequest y GetWebResponse.

Para realizar la visualización de reportes se empleó un control de servidor ASP.NET denominado ReportViewer. Este control es una versión mejorada de los ejemplos realizados por Teo Lachev [5] y de la ayuda de la Plataforma de Servicios de Reportes SQL. El ReportViewer brinda las facilidades correspondientes para la integración de reportes en una Aplicación Web ASP.NET y utiliza el acceso URL al Servicio Web del Servidor de Reportes; permite la presentación de reportes, desplazarse por ellos y exportarlos a distintos formatos.

Para el manejo de la configuración del Sistema SAMI, se definieron dos archivos de configuración, que son: *samiConfiguration.config* y *dataConfiguration.config* El primero almacenará información referente a la configuración en general del sistema, y el segundo almacenará información necesaria para la conexión con las bases de datos empleadas.

La necesidad de definir el archivo dataConfiguration.config, es aprovechar el soporte de Enterprise Library para el manejo de la información de las cadenas de conexión a las bases de datos, para lo cual define clases que encapsulan su lectura y escritura. En cuanto al archivo samiConfiguration.config, permite manejar de forma separada, variables de configuración específicas del sistema, para lo cual es necesario definir una clase que encapsule su lectura y escritura (Config).

III. CONCLUSIONES

A través del Sistema SAMI es posible demostrar la interacción de Servicios Web, la tecnología ASP.NET y la Plataforma de Servicios de Reportes SQL, brindando un sistema personalizado que se beneficia de todos los servicios brindados por cada una de ellas. El sistema SAMI está desarrollado de tal forma que las tecnologías se complementen entre sí, para ofrecer un sistema moderno y modular.

El modelo de autenticación por formularios, empleado por el Sistema SAMI, es el que más se adaptó al ambiente que maneja el sistema, el Internet, puesto que permite especificar con mayor personalización las políticas de seguridad, sin embargo, requiere mayor cantidad de tiempo en su implementación y desarrollo.

Mediante la extensión de seguridad de autenticación por formularios desarrollada para la plataforma de reportes para el Sistema SAMI, se presenta un ejemplo de la versatilidad y extensibilidad que brinda la plataforma de reportes en cuanto a la personalización de funciones.

Para el caso del Sistema SAMI, la personalización de funciones realizada, permite que el docente tenga una mayor interacción con sus estudiantes, al facilitarle un ambiente amigable para la administración de trabajos y planificación de pruebas. Por otro lado, el estudiante podrá tener información personalizada de las materias que toma, con sus pruebas y trabajos, lo cual resuelve gran parte de los problemas de aprendizaje por falta de información precisa.

Existen múltiples áreas en las que la integración de las

tecnologías puede emplearse, ya sea en sistemas de comercio electrónico, bancos, o como el caso del Sistema SAMI en el área de la educación; siendo este ejemplo una base para que futuros trabajos puedan tomarlo como referencia.

REFERENCIAS

[1] MICROSOFT, “Colección combinada de Visual Studio .NET 2003”, 2003.

[2] http://www.microsoft.com/presspass/features/2004/jan04/01-27SQLReporting.asp, 27/06/2005.

[3] WROX, “Professional SQL Server Reporting Services”, Wiley Publishing, USA, 2004.

[4] http://msdn.microsoft.com/library/?url=/library/en-us/dnsql2k/html/ufairs.asp?frame=true#ufairs_topic3, 19/08/2005.

[5] LACHEV T, “Microsoft Reporting Services in Action”, Manning Publications, USA, 2005.

[6] http://www.microsoft.com/windows/virtualpc/default.mspx, 19/08/2005.

[7] http://www.microsoft.com/windowsserver2003/default.mspx, 19/08/2005.

[8] http://www.microsoft.com/sql/default.mspx, 19/08/2005.

[9] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/entlib.asp, 19/08/2005.

Santiago D. Gudiño

Nació en Quito el 29 de Marzo de 1981. Realizó sus estudios secundarios en el Colegio San Gabriel, donde obtuvo el bachillerato. En 1999 ingresó a la Escuela Politécnica Nacional en donde varios semestres obtuvo reconocimientos por excelencia académica.

Perteneció al grupo de células universitarias .NET, incursionando en tecnologías como ASP.NET, ADO.NET, y Servicios Web.

Formó parte de varios cursos organizados por Microsoft, entre los que resaltan: arquitectura de software .NET y arquitectura Windows Server 2003. En los últimos tiempos ha tenido gran incursión en SQL Reporting Services. En el 2005 se tituló en la carrera de Ingeniería en Electrónica y Redes de Información de la EPN.

Cristina E. Torres

Nació en Riobamba-Ecuador, el 23 de enero de 1982. Realizó sus estudios secundarios en el Colegio María Auxiliadora y los finalizó en el año 1999, en el mismo año ingresó a la Escuela Politécnica Nacional (EPN). Formó parte del grupo de células universitarias .NET, lo cual le permitió incursionar en tecnologías como ASP.NET, ADO.NET, Servicios Web. Asistió a algunas exposiciones y cursos organizados por Microsoft que le permitieron familiarizarse con tecnologías como: SQL Reporting Services y Windows Server 2003. A sus 23 años se tituló en la carrera en Electrónica y Redes de Información de la EPN (2004).

Iván M. Bernal

Ingeniero en Electrónica y Telecomunicaciones, Escuela Politécnica Nacional (EPN), en Quito-Ecuador en 1992. Obtuvo los títulos de M.Sc. (1997) y Ph.D. (2002) en Computer Engineering en Syracuse University, NY, USA. Actualmente es docente de la EPN, en el Departamento de Electrónica, Telecomunicaciones y Redes de Información.