

# **ESCUELA POLITECNICA DEL EJÉRCITO**

## **SEDE LATACUNGA**

### **FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA**

**DESARROLLO DE SERVICIOS DE INFORMACION PARA INTRANETS BAJO EL PARADIGMA XML/WEB SERVICES/ JAVA CASO PRACTICO”CONFERENCIA WEB EMPRESARIAL”**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TITULO DE INGENIERO EN SISTEMAS E INFORMATICA**

**CRISTIAN SANTIAGO MOLINA HERRERA**

Latacunga, Abril-2005

## **CERTIFICACION**

Los suscritos Ing. José Luis Carrillo e Ing. Raúl Cajas certifican que el siguiente trabajo teórico- práctico, fue desarrollado íntegramente por el señor Molina Herrera Cristian Santiago, bajo nuestra supervisión.

---

Ing. José Luis Carrillo  
DIRECTOR DE TESIS

---

Ing. Raúl Cajas  
CODIRECTOR DE TESIS

## **AGRADECIMIENTO**

En la vida tuve muchas personas importantes mis padres, mi hermano, abuelitos, amigos quienes han sido mi fuerza y mi aliento para salir adelante.

Es por esto que mi agradecimiento pero en primer lugar a Dios por darme salud y llenarme de fortaleza para sobresalir en los momentos más difíciles de mi vida a mi abuelito que lamentablemente no puede estar pero desde el lugar que se encuentre estará feliz, a mi hermano por su amistad y apoyo, a mis familiares por sus consejos y buenos deseos, a los amigos por los buenos momentos compartidos durante toda nuestra carrera estudiantil.

A la inolvidable Escuela Politécnica del Ejercito Sede Latacunga en especial a la facultad de Ingeniería en Sistemas e Informática y a sus profesores por la oportunidad que nos brindaron para consolidar nuestros estudios y por ende nuestros sueños.

A nuestro director, codirector y al Ingeniero Marco Jarrin por ser guías facilitadores de todos sus conocimientos con lo cual hicieron posible el desarrollo de este trabajo.

***Cristian***

## **DEDICATORIA**

El presente trabajo lo dedico a mis padres por su apoyo incondicional y por su lucha para ver reflejados sus deseos de salir adelante en mí, ya que sin el apoyo económico pero principalmente moral no podría haber llegado a culminar mi carrera.

También lo dedico a mi hermano quien ha estado en los momentos más difíciles que tuvimos que pasar a mis tíos y mis abuelitos que me inculcaron a ser una persona de bien y sobre todo sencilla.

Dejo un refrán que las personas que quieren salir adelante tienen que llevar presente “Si todo está bajo control es porque no vamos suficientemente deprisa “.

***Cristian***

# CAPITULO 1

## I.- LOS SERVICIOS WEB – WEB SERVICES

### 1.1- ESTADO DEL ARTE DE LOS WEB – SERVICES

#### 1.1.1- INTRODUCCIÓN

En algún momento de nuestra vida, todos hemos ofrecido o contratado un servicio. Mensualmente, el pago de nuestra tarjeta de crédito es debido a que compramos o contratamos a alguien para que realice cierta labor que nos es necesaria, ya sea por obligación o placer. Dentro de los servicios que contratamos están, por ejemplo, el agua, el teléfono, la luz, el pago de impuestos, etc.

Algunos de los servicios que buscamos por necesidad podrían ser la compra de boletos para ir a un concierto, el pago de mercancías que adquirimos en un catálogo de ropa, o tal vez, los boletos de avión para ir de vacaciones. En fin, siempre habrá algo nuevo que comprar, rentar o alquilar.

El mundo moderno nos exige, además, un ritmo de vida mucho más ágil. La mayoría de nosotros trabajamos y no tenemos tiempo que perder en el tráfico para ir a un centro comercial, ni tampoco tenemos mucho tiempo para ir al banco, pues las colas son muy grandes. La vida de hoy nos exige poder realizar estos menesteres lo más ágil y rápidamente posible.

Con el nacimiento de Internet se puede realizar ya muchas de estas labores desde la comodidad de nuestra casa u oficina. Tareas como la compra por catálogo o los famosos carritos que existen en innumerables sitios. Un ejemplo claro es “Amazon”,

donde se puede escoger libros de un catálogo interminable, los depositamos en un carrito de compras y pasamos a pagar con nuestra tarjeta de crédito.

A la vista del público, es una labor sencilla que involucra sólo ciertos pasos:

1. Buscar un sitio que ofrezca un servicio de interés.
2. Elegir en un catálogo, en algunos casos.
3. Pagar el servicio.

Sin embargo, en el aspecto computacional, se requiere de: una infraestructura de comunicaciones muy amplia, que permita la conectividad con el servicio; un conjunto de servidores donde resida el software; una base de datos que registre los catálogos y mantenga la información generada por el servicio; un servidor de aplicaciones que haga públicos los servicios; tecnologías que permitan las transacciones con estos servidores.

Mucho se ha avanzado en la publicación de aplicaciones vía web para ofrecerlos. Sin embargo, de estas aplicaciones, la mayoría han sido pensadas para resolver servicios que sólo competen a la institución que los ofrece. Aunque en la vida real, para realizar, por ejemplo, la solicitud de un pasaporte, se requiere llevar una serie de documentos para seguir el trámite. Se puede mencionar: acta de nacimiento, libreta de servicio militar, cédula de identidad.

¿Que sucede cuando no contamos con uno de ellos? Supongamos que no tenemos la cédula de identidad. Entonces, tenemos que ir a un módulo autorizado, presentar la documentación necesaria (más tiempo si nos falta algún requisito) y esperar unos días para que nos sea entregada. Cuando contamos con este documento, continuamos con el procedimiento normal del trámite del pasaporte que, entre otras cosas, implica levantarnos muy temprano para evitar grandes colas, esperar y rogar porque no se nos haya olvidado algo. Si no fue así, entonces pasaremos a pagar los servicios al banco y

regresaremos por la tarde, para sentarnos a esperar a que nos sea entregado nuestro documento.

En la vida agitada que tenemos, ya no es concebible la pérdida de tiempo, y menos aún, si consideramos que existe la tecnología que permite automatizar estas labores.

¿Qué sucedería si en lugar de trasladarnos a realizar el trámite del pasaporte a una oficina de Relaciones Exteriores, pudiéramos hacerlo desde la comodidad de nuestra casa? Si existiera un lugar en Internet donde llevarlo a cabo con un simple “click” a nuestro ratón.

Supongamos que nos conectamos a Internet y navegamos hasta el portal de la Secretaría de Relaciones Exteriores y escogemos realizar el trámite del pasaporte. Como en cualquier sitio de Internet, llenaremos una forma que sólo nos pide nuestra Cédula y nuestra firma digital, acostumbrados a llenar las solicitudes de una página. Después de introducir los datos, mágicamente el sistema nos informa que nuestro trámite ha sido aprobado, que debemos pagar un monto, y aparece una casilla donde se puede introducir nuestro número de tarjeta de crédito. Una vez dado ese paso, nos informa que el pasaporte será enviado a nuestra dirección actual. Increíble, ¿no?

Absortos, trataríamos de idear y razonar cómo lo hizo. Si antes nos pedían nuestra acta de nacimiento para verificar nuestra edad y nacionalidad; nuestra libreta militar, requerida para hombres mayores de 18 años; nuestra credencial para probar nuestra identidad; se debía ir al banco a pagar un formato por el trámite. ¿Qué pasó entonces? ¿Acaso en este mágico y maravilloso sistema ya no es necesario?

La respuesta es no, sí es necesaria, pero se debe al sistema tan inteligente, que hizo todo esto por nosotros. ¿Cómo? Primero con nuestra Número de Cédula y nuestra firma digital se conectó al portal, también automatizado, del registro civil, y verificó que estemos registrados. Una vez localizados y verificados, se dio cuenta que pertenecemos al sexo masculino, y dado que verificó nuestra edad en el acta de

nacimiento, se conectó a la Secretaría de la Defensa Nacional, para verificar la libreta militar liberada. Verificado lo anterior, nos indicó el monto a pagar. Cuando le introdujimos nuestra tarjeta de crédito, pasó al banco y retiró el monto necesario y lo transfirió a la Secretaría. Posteriormente, accedió al portal del Registro de Población y preguntó dónde vivíamos, para facilitar la información al sistema automatizado de entrega de pasaportes.

En nuestros días este proceso es realizable, gracias al surgimiento de tecnologías que permiten a sistemas de naturaleza heterogénea comunicarse unos con otros. Los estándares como XML (Extensible Markup Language) y SOAP (Simple Object Access Protocol) permiten el nacimiento de lo que actualmente se conoce como “Web Services”, los cuales pueden publicar servicios automatizados por medio del intercambio de documentos XML.

En general, un sitio que ofrece un servicio tiene que tener una infraestructura mínima que, como ya se mencionó, consta de un servidor de base de datos, un servidor de web y un servidor de aplicaciones. Es común que estos servidores estén sobre un sistema operativo UNIX o Windows y que la base de datos, el servidor de web y las aplicaciones sean de los líderes en el mercado. Es decir, toda una infraestructura de naturaleza heterogénea.

Como hemos visto, los servicios web nos pueden ayudar a automatizar los trámites al hacerlos disponibles en Internet.

### **1.1.2- HISTORIA**

La Web en sus orígenes fue pensada como un medio para desplegar información, esta reposa de manera estática en los servidores la cual es accedida a través de una consulta hecha por un navegador valiéndose del protocolo HTTP (Hyper Text Transfer Protocol). Actualmente se maneja el mismo concepto en la comunicación cliente-



servidor (navegador - webserver) solo que no necesariamente el resultado de la comunicación debe provenir de la carga de una página estática, esta puede ser el resultado de la ejecución en el servidor de alguna lógica de programación.

Esto último no necesariamente lo llamamos una aplicación Web, pero nos acerca al concepto. Consideraremos una aplicación Web a un sitio Web donde la navegación a través de él y la entrada de datos por parte de un usuario, afectan el estado de la lógica del negocio. En esencia, una aplicación Web usa un sitio Web como entrada (front-end) a una aplicación típica. Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web. Bajo este concepto las aplicaciones Web no solo se encargan de desplegar información, sino que también, deben contener una lógica asociada que permita apoyar algún proceso propio del negocio para el cual fue diseñada.

Para el desarrollo de aplicaciones Web se han generado múltiples tecnologías entre ellas se encuentran:

- CGI. Common Gateway Interface fue la primera técnica utilizada para que el contenido de las páginas web se generará de manera dinámica, es común encontrar en los diferentes servidores web el modulo que soporta la ejecución de CGIs. De manera resumida se puede decir que el CGI es un mecanismo de comunicación entre el servidor WEB y una aplicación externa, esta aplicación puede estar desarrollada en casi cualquier lenguaje, este solo debe cumplir la condición de ser soportado por el servidor http, las aplicaciones CGIs se encuentran desarrolladas en lenguajes como c , PERL, etc.

Este mecanismo tiene falencias que evita su uso a gran escala, la más conocida es en cuanto a rendimiento, ya que por cada petición que se realice en el servidor se crea un nuevo proceso, lo cual tiene un costo muy alto en lo que a recursos del sistema se refiere.

- Fast- CGI. Esta es una solución similar al CGI mencionado anteriormente, solo que propone la creación de un solo proceso persistente por cada programa FastCGI en lugar de por cada solicitud del cliente. Es una solución viable pero también tiene inconvenientes de proliferación de procesos en el caso de peticiones concurrentes.
- Páginas dinámicas en servidor. Con la aparición de esta tecnología se entra a una nueva forma de trabajo, la cual está orientada al trabajo del diseñador web, quien no necesariamente conoce de lenguajes de programación. Este nuevo enfoque consiste en insertar pequeños fragmentos de lógica de programación en la estructura HTML de la página, al contrario de lo que se hacía en los CGIs, que era en el lenguaje de programación utilizar sentencias de impresión para generar salidas HTML. En este sentido se conocen diferentes alternativas como PHP, ASP, JSP, entre otros.
- Servlets. El servlet se puede considerar como una evolución de los CGIs desarrollada por SUN Microsystems como parte de la tecnología JAVA. De forma general consiste en la ejecución de aplicaciones Java en el motor de servlets (Servlet engine) el cual hace parte del servidor web, algo que lo hace ventajoso con respecto a los CGIs es que por cada petición de usuario no se crea un proceso sino un hilo, el cual es mucho más económico para el sistema. Esta tecnología hace parte de la arquitectura propuesta por SUN en su plataforma J2EE (Java 2 Enterprise Edition).
- Servicios Web. La arquitectura de servicios WEB plantea algo más que una técnica para el desarrollo de aplicaciones Web, representa un modelo de computación distribuida para Internet basado en XML (eXtensible Markup Language). Bajo este concepto ya no solo se trata la comunicación usuario - aplicación, sino que de manera adicional se maneja la interacción aplicación - aplicación. Para aclarar un poco más el concepto tomemos como ejemplo una rutina de programación, como sabemos una rutina es como una caja negra, la

cual encierra un proceso y que cumple una función claramente definida, luego para construir una aplicación llamamos dichas rutinas enviando parámetros y recibiendo la respuesta respectiva. Un servicio web se puede considerar como una rutina a la cual se le envían los parámetros utilizando XML encapsulados en el protocolo HTTP.

Durante los últimos diez años, nada ha tenido una influencia tan grande sobre el mundo del software como Internet y la World Wide Web. Los esfuerzos de las organizaciones por adaptarse a este nuevo entorno han generalizado el uso de aplicaciones para navegadores. Pero a pesar de su innegable utilidad, ofrecer servicios en navegadores no es la única forma en que las aplicaciones pueden explotar la tecnología de la Web. Si el software que se ejecuta en otros sistemas **podiera poner** en marcha directamente operaciones en la Web, las aplicaciones Web podrían ofrecer mejores servicios a clientes más diversos.

Esto es posible gracias a los servicios Web. Con esta nueva tecnología, el software que se ejecuta en sistemas de escritorio, servidores, asistentes digitales personales, teléfonos móviles o cualquier otro dispositivo inteligente puede poner en marcha directamente operaciones ofrecidas por aplicaciones Web. Aunque las aplicaciones para Internet serán mayoría en este nuevo mundo, las aplicaciones que se ejecuten en redes internas también podrán beneficiarse del uso de los servicios Web.

Lo más importante sobre el origen de las tecnologías de servicios Web es que ninguna de ellas es la solución de un solo proveedor. De hecho, no hay ninguna que esté vinculada a tecnologías exclusivas como Windows, Unix, el modelo de objetos componentes (COM) de Microsoft o Enterprise JavaBeans (EJB). Por el contrario, los servicios Web basados en WSDL, SOAP y UDDI se pueden utilizar prácticamente con cualquier plataforma, lenguaje y modelo de objetos.

### **1.1.3- DEFINICIÓN DE SERVICIOS WEB Y SERVICOS WEB XML**

### 1.1.3.1- Definición de Servicios Web

Un servicio Web es una colección de funciones que son presentadas como una sola entidad y es anunciada en la red para ser usada por otros programas. Los servicios de Web son los bloques de construcción para crear sistemas distribuidos abiertos. Es un servicio ofrecido a través de la Web, que usa protocolos y APIs para recibir peticiones y enviar las correspondientes respuestas, el cuál tiene las siguientes características:

- Es accesible a través de la interfaz del protocolo.
- Su interfaz se describe en un documento de Descripción de Servicios.

Ejemplos de servicios:

- Passport de Microsoft, servicio de autenticación.
- Servicio de Bolsa de Valores, que regresa el valor o cotización de una acción de acuerdo a un símbolo designado.
- Servicio de compra, que permite a los sistemas de cómputo adquirir artículos de oficina cuando se proporciona un código de artículo y una cantidad.

Un servicio Web puede agregar otros servicios Web para proveer un conjunto de mayores características. Por ejemplo, un servicio Web pudiera proveer características de alto nivel en viajes (cumpla con múltiples requerimientos de cada usuario), al ofertar servicios Web de renta de autos, boletos de avión, y hoteles. Las aplicaciones futuras se realizarán con servicios Web que serán seleccionados dinámicamente en tiempo real basados en costo, calidad, y disponibilidad.

Entre las razones por las cuales los servicios Web jugarán un rol principal en la siguiente generación de sistemas distribuidos, están:

- Interoperabilidad. Cualquier servicio Web puede interactuar con cualquier otro servicio Web. El protocolo estándar SOAP permite que cualquier servicio pueda

ser ofrecido o utilizado independientemente del lenguaje o ambiente en que se haya desarrollado.

- Omnipresencia. Los servicios Web se comunican utilizando HTTP y XML. Cualquier dispositivo que trabaje con éstas tecnologías puede tanto ser huésped y acceder a los servicios Web. Por ejemplo, serán utilizados en teléfonos, automóviles o aún en máquinas vendedoras de refrescos (una máquina de venta de refrescos puede comunicarse vía inalámbrica con el servicio Web de un proveedor local y ordenar un pedido de suministro).
- Barrera mínima de participación. Los conceptos detrás de los servicios de Web son fáciles de comprender y se ofrecen Herramientas de Desarrollo (ToolKits) por IBM, Sun Microsystems y la Organización de Apache, permiten a los desarrolladores crear e implementar rápidamente servicios de Web.
- Apoyo de las Industrias. Todas las compañías apoyan el protocolo SOAP y la tecnología derivada de los servicios Web

#### **1.1.3.2- Definición de Servicios Web -XML**

Los servicios Web XML son la base que permitirá la computación distribuida en Internet. El uso de estándares abiertos y la orientación hacia la comunicación y colaboración entre personas y aplicaciones han creado un entorno en el que los servicios Web XML funcionan como plataforma para la integración de aplicaciones. Las aplicaciones se crean con varios servicios Web XML de distintas fuentes que colaboran entre sí, independientemente de dónde residan o de cómo se pusieran en marcha.

- Los servicios Web XML utilizan un protocolo estándar de Web para poner funciones útiles a disposición de los usuarios de la Web.
- Los servicios Web XML ofrecen una forma de describir sus interfaces con suficiente detalle para que el usuario pueda crear una aplicación de cliente para comunicarse con ellas. Esta descripción se ofrece normalmente en un

documento XML que recibe el nombre de documento de lenguaje de descripción de servicios Web.

- Los servicios Web XML están registrados para que los posibles usuarios puedan encontrarlos con facilidad. Esto se hace con descripción, descubrimiento e integración universales.

La tecnología de servicios Web se puede dividir en áreas, cada una de las cuales trata un aspecto concreto del problema. Estas áreas son:

- Descripción de la información enviada por la red. La puesta en marcha de una operación remota suele implicar el paso de parámetros y la recepción de algún tipo de resultados. Con los servicios Web, esta información se describe mediante el lenguaje de marcado extensible (XML), que ha sido admitido universalmente como el lenguaje común para la descripción de datos. XML permite describir e intercambiar todo tipo de datos.
- Acceso a servicios Web. Una vez definida una interfaz, los clientes tienen que usar algún protocolo para poner en marcha las operaciones en esa interfaz. No hay ningún protocolo específico para ello (de hecho, WSDL permite explícitamente el uso de distintos protocolos para poner en marcha las operaciones en una interfaz). No obstante, la opción más importante es el protocolo de acceso simple a objetos (Simple Object Access Protocol SOAP), que se define con XML y permite identificar la operación que se va a poner en marcha, pasar los parámetros de entrada de esa operación como datos definidos por XML, y devolver los posibles resultados también como datos definidos por XML. SOAP define únicamente un ajustador sencillo para comunicar esta información, que se puede transportar de distintas maneras. Por ejemplo, las llamadas de SOAP se pueden transportar con el protocolo de transferencia de hipertexto (HTTP), con una tecnología de mensajería como el protocolo simple de transferencia de correo (SMTP), o con otros métodos.

### **1.1.3.3- Capacidades de los Servicios Web**

Los proveedores de servicios Web necesitan algún mecanismo que les permita especificar con exactitud los datos técnicos de los servicios ofrecidos. Como ocurre con otros tipos de servicios, lo más útil es agrupar operaciones relacionadas en interfaces y luego utilizar algún método para describir cada una de esas operaciones. En el caso de los servicios Web, esto se hace con el lenguaje de descripción de servicios Web (WSDL). Cada una de las interfaces definidas con WSDL contiene una o más operaciones, y cada operación tiene datos de entrada y salida definidos con XML.

### **1.1.3.4- Búsqueda de Servicios Web**

Para que los desarrolladores puedan crear clientes que utilicen servicios Web es necesario que conozcan cuáles son los servicios disponibles, qué es lo que ofrecen y cómo son sus interfaces. La mejor opción consiste en utilizar Internet para crear un registro estándar donde se pueda almacenar este tipo de información y acceder a ella. Esto es justamente lo que hace la tecnología definida para descripción, descubrimiento e integración universal (UDDI). UDDI permite a los proveedores de servicios Web dar a conocer sus ofertas de una forma estándar, lo que hace que los clientes puedan saber cuáles son los servicios ofrecidos por cada proveedor y que los creadores de software de cliente dispongan de toda la información necesaria para crear esos clientes.

Hemos definido un servicio Web XML como un servicio de software que se ofrece en la Web por medio de SOAP, que se describe con un archivo WSDL y que se registra en UDDI. Los primeros servicios Web XML eran básicamente fuentes de información que se podían incluir fácilmente en aplicaciones (índices bursátiles, pronósticos del tiempo, marcadores deportivos, etc.). Es fácil imaginar la creación de una clase de aplicaciones que analizarían y añadirían la información que quisiera y la presentarían de distintas formas; por ejemplo, podría tener una hoja de cálculo Excel de Microsoft que

resumiera toda su información económica (acciones, 401K, cuentas bancarias, préstamos, etc.). Si es posible acceder a esta información a través de servicios Web XML, Excel puede actualizarla continuamente. Parte de esta información puede ser gratuita, mientras que para acceder a otra parte podría ser necesario suscribirse al servicio. La mayor parte de esta información ya está disponible en la Web, pero los servicios Web XML pueden hacer que el acceso mediante programación sea más sencillo y fiable.

El hecho de ofrecer aplicaciones existentes como servicios Web XML permitirá a los usuarios crear aplicaciones nuevas y más potentes a partir de servicios Web XML. Por ejemplo, un usuario podría desarrollar una aplicación de compras para conseguir automáticamente información sobre los precios de distintos proveedores, lo que le permitiría elegir un proveedor, cursar el pedido y supervisar el envío hasta recibirlo. La aplicación del proveedor, junto con la oferta de sus servicios en la Web, podría utilizar a su vez servicios Web XML para comprobar el crédito del cliente, cargar el precio en su cuenta y preparar el envío con una empresa de transportes.

En el futuro, algunos de los servicios Web XML más interesantes serán compatibles con aplicaciones que usen la Web para cosas que son imposibles de hacer hoy en día. Por ejemplo, el proyecto My Services de Microsoft .NET ofrece un servicio de calendario. Si su dentista o su mecánico ofrecieran su calendario a través de este servicio Web XML, usted podría concertar directamente una cita en línea en su calendario para una limpieza bucal o para un mantenimiento de rutina.

Es fácil imaginarse cientos de aplicaciones que se podrían crear si se pudiera programar la Web.

### **1.1.3- COMPONENTES DE LOS SERVICIOS WEB**



- **Servicio.** La aplicación es ofrecida para ser utilizada por solicitantes que llenan los requisitos especificados por el proveedor de servicios. La implementación se realiza sobre una plataforma accesible en la red. El servicio se describe a través de un lenguaje de descripción de servicio. Tanto la descripción como las políticas de uso han sido publicadas de antemano en un registro.
- **Proveedor de Servicio.** Desde el punto de vista comercial, es quien presta el servicio. Desde el punto de vista de arquitectura, es la plataforma que provee el servicio.
- **Registro de Servicios.** Es un depósito de descripciones de servicios que puede ser consultado, donde los proveedores de servicios publican sus servicios y los solicitantes encuentran los servicios y detalles para utilizar dichos servicios.
- **Solicitante de servicios.** Desde el punto de vista comercial, la empresa que requiere cierto servicio. Desde el punto de vista de la arquitectura, la aplicación o cliente que busca e invoca un servicio.

La siguiente figura muestra las relaciones operacionales entre los componentes:

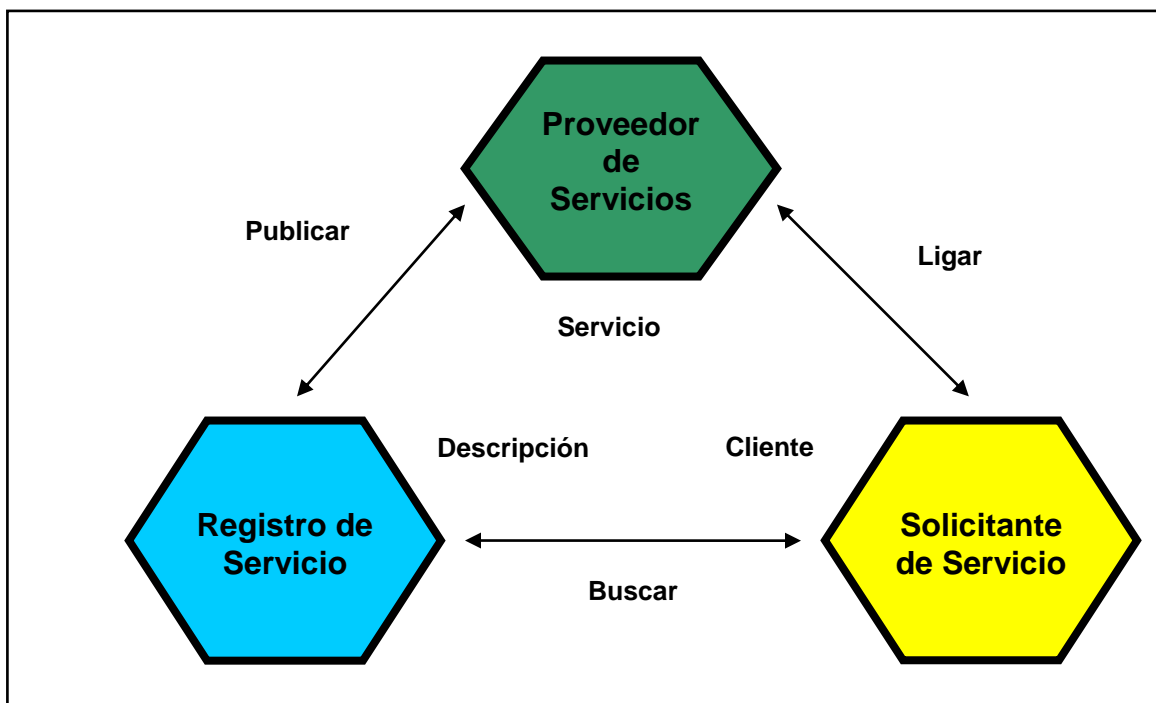


Figura 1.1 Componentes de un Servicio Web

Operaciones de Servicios Web:

- **Publicar/Cancelar.** Los proveedores de servicios publican (publicitan) la disponibilidad de su servicio comercial (e-business) a uno o más Registros de servicios, o cancelan la publicación de su servicio.
- **Búsqueda.** Los solicitantes de servicios interactúan con uno o más Registros de servicios para descubrir un conjunto de servicios comerciales con los que pueden interactuar para encontrar una solución.
- **Ligar, Unir (Bind).** Los solicitantes de servicios negocian con los proveedores de servicios para acceder e invocar servicios comerciales (e-business)

#### 1.1.4- ARQUITECTURA DE SERVICIOS WEB

La siguiente figura representa el funcionamiento general de los servicios web:

Figura 1. Arquitectura general servicios web

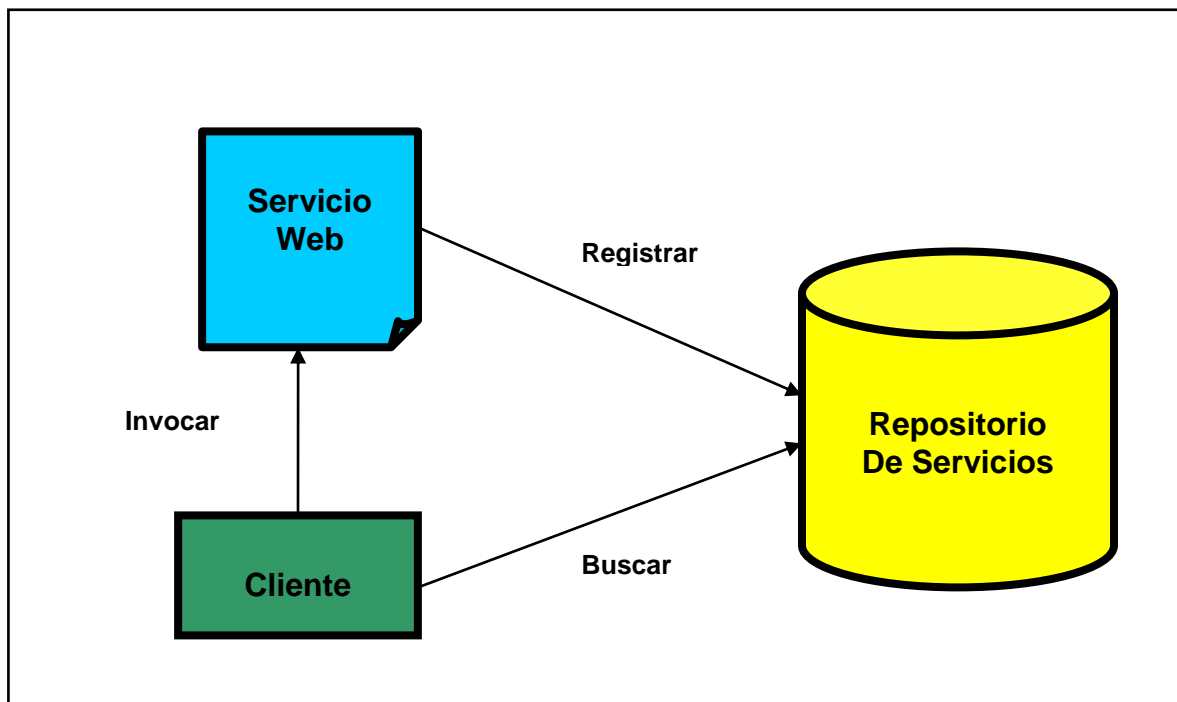


Figura 1.2 Esquema General del funcionamiento de los Servicios Web

Según Figura 1.2, se puede deducir que un servicio Web se registra en un repositorio de servicios, el cliente busca en el repositorio el servicio que necesita y luego lo invoca. De manera más detallada "La arquitectura de los servicios Web es una meta-arquitectura que permite que ciertos servicios de red sean dinámicamente descritos, publicados, descubiertos e invocados en un ambiente de cómputo distribuido.

Los servicios Web son aplicaciones auto-contenidas y modulares que pueden ser:

- Descritas mediante un lenguaje de descripción de servicio, como el lenguaje WSDL (Web Service Description Language)
- Publicadas al someter las descripciones y políticas de uso en algún Registro bien conocido, utilizando el método de registro UDDI (Universal Description, Discovery and Integration).
- Encontradas al enviar peticiones al Registro y recibir detalles de ligamiento (binding) del servicio que se ajusta a los parámetros de la búsqueda.
- Asociadas al utilizar la información contenida en la descripción del servicio para crear una instancia de servicio disponible o proxy.
- Invocadas sobre la red al utilizar la información contenida en los detalles de ligamiento de la descripción del servicio.
- Compuestas con otros servicios para integrar servicios y aplicaciones nuevas.

Bajo esta definición más precisa de la arquitectura se puede generar la siguiente gráfica:

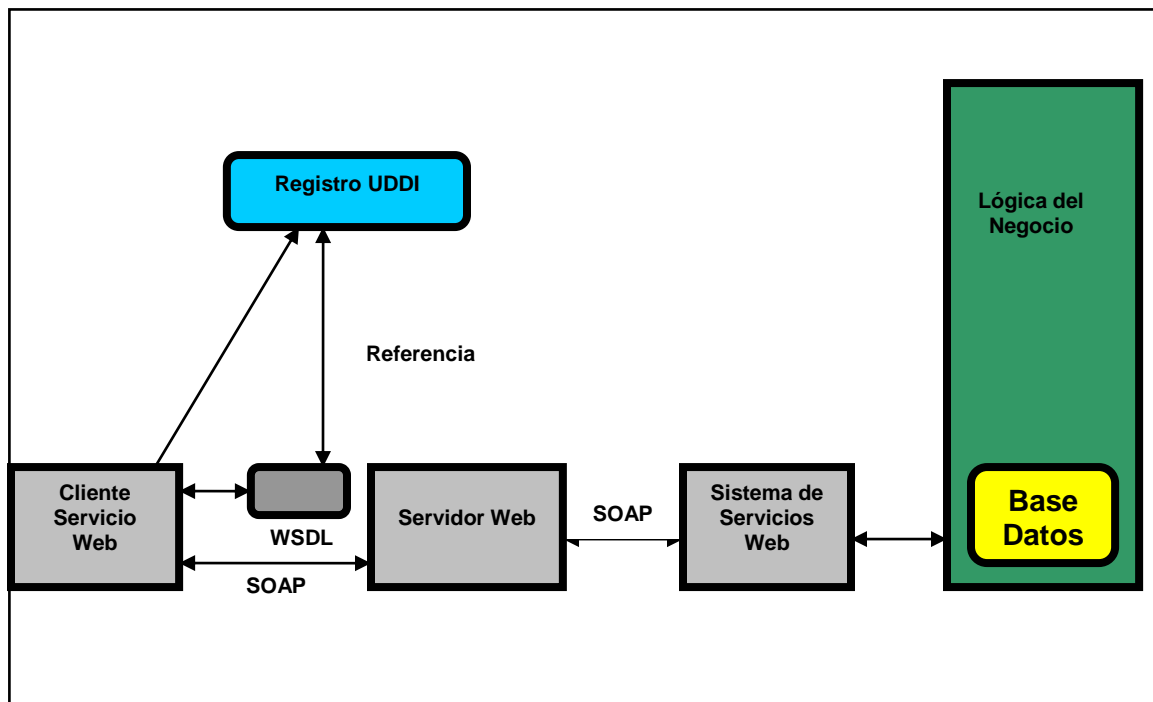


Figura 1.3. Arquitectura detallada de los servicios Web

Ahora se puede dar una descripción más real del funcionamiento de los servicios web. El servicio web es construido y luego descrito por medio de WSDL y registrado bajo el estándar UDDI, el cliente busca en el registro UDDI (como si fuese un motor de búsqueda al estilo de Google) y obtiene el descriptor WSDL del servicio que necesita, lo invoca haciendo uso de SOAP el cual también es utilizado para comunicar la petición entre los diferentes componentes del servidor que aloja el servicio WEB, para entregar una respuesta utilizando nuevamente SOAP.

### 1.1.5- IMPLEMENTANDO UN SERVICIO WEB

Se presenta un breve esquema paso a paso de cómo se implementa un servicio Web en el siguiente diagrama:

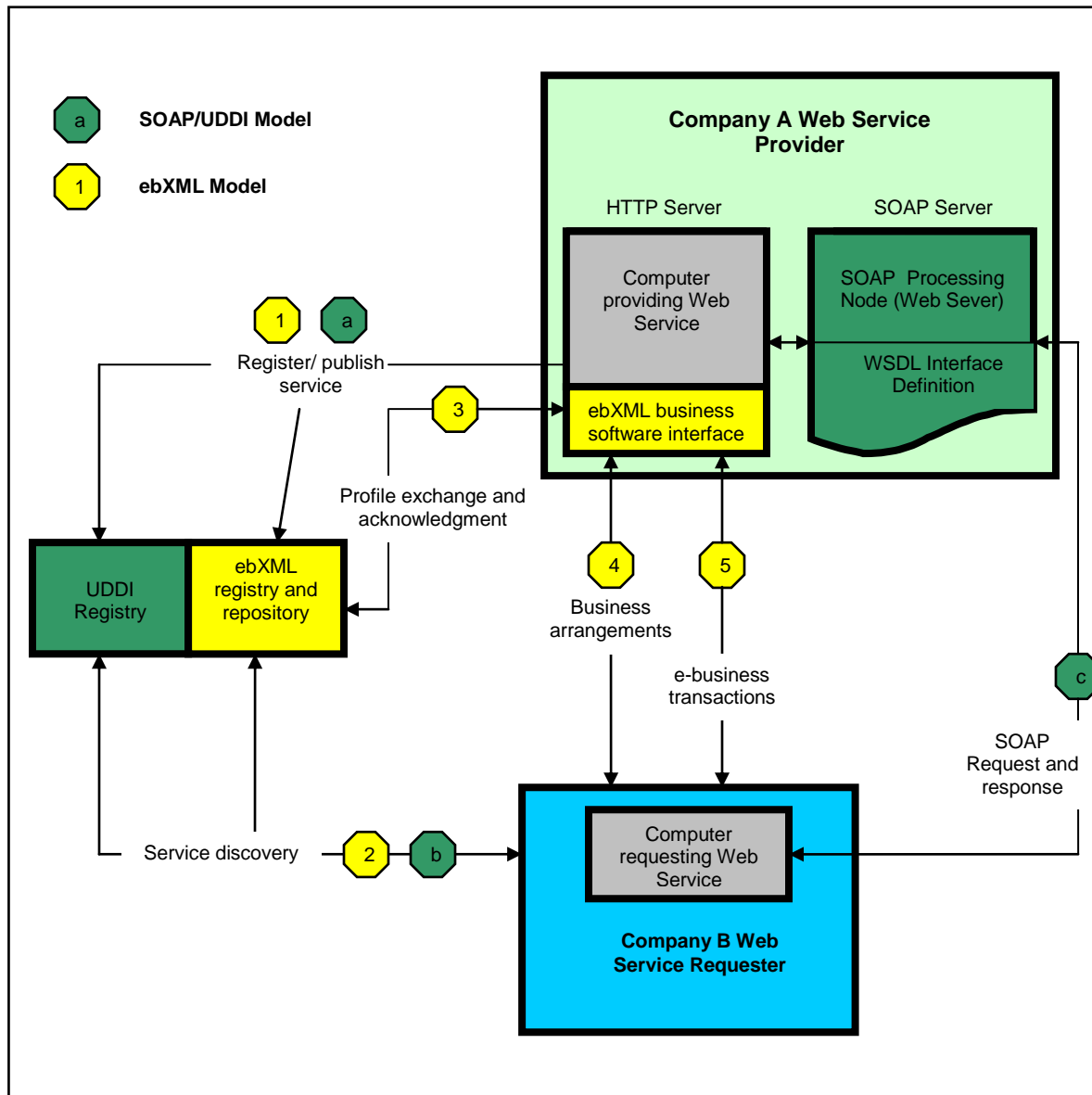


Figura 1.4 Implementando un Servicio Web

1. Un proveedor de servicio crea un servicio Web
2. El proveedor de servicio utiliza WSDL para describir el servicio (a un registro UDDI)
3. El proveedor de servicio registra el servicio (en un registro UDDI y/o a un registro/depósito ebXML )
4. Otro servicio o usuario localiza y solicita el servicio registrado al consultar los registros UDDI y/o ebXML

5. El servicio o usuario solicitante escribe una aplicación que liga el servicio registrado utilizando SOAP (en el caso de UDDI) y/o ebXML
6. Se intercambian datos y mensajes XML sobre http

### **1.1.6- HERRAMIENTAS PARA EL DESARROLLO DE SERVICIOS WEB**

El primer paso para construir un servicio Web es el de seleccionar las herramientas de desarrollo de servicios Web, para los cuales existen las siguientes opciones:

- Proyecto XML de Fuente Abierta de la Fundación de Software Apache (<http://www.apache.org/>):
- SOAP (Simple Object Access Protocol) 3.0 de Apache ó AXIS (Apache eXtensible Interaction System)
- Servidor de Web de fuente abierta en Java Jakarta Tomcat 4.0.3 de Apache.
- Tomcat es un recipiente de servlets (Catalina) utilizado en el desarrollo de tecnologías de Java Servlet 2.3 y JavaServer Pages 1.2.
- Segmentador de componentes de texto XML de fuente abierta Xerces (Java, Perl, C++) de Apache
- Herramientas de servicios Web de IBM:
- IBM WSTK 3.01. Web Service ToolKit que incluye SOAP de Apache, un generador de WSDL, cliente UDDI, procesador Xalan de hojas de estilo XML, y otros.
- UUDI4J, biblioteca de clases de Java con API para interactuar con un registro UDDI.
- Sun Open Net Environment (Sun ONE) de Sun Microsystems
- Sun ONE Starter Kit 2.0
- Java Web Services Developer Pack, de Sun Microsystems
- <bigwig> Project

### **1.1.7- SERVICIOS WEB EN JAVA**

Aunque los servicios Web inicialmente fueron promovidos por Microsoft e IBM, con la liberación del J2EE 1.4, Sun ha proporcionado soporte total para los servicios web.

En el entorno J2EE los servicios Web se construyen sobre JAX-RPC (Java API for XML-based RPC). Este es un API para construir servicios web y clientes que utilizan llamadas a procedimientos remotos (RPC) y XML.

En JAX-RPC, una llamada a un procedimiento remoto está representada por un protocolo basado en XML, como SOAP. La especificación SOAP define la envoltura estructural, las reglas de codificación y la convención para representar llamadas y respuestas a procedimientos remotos. Estas llamadas y respuestas se transmiten como mensajes SOAP (archivos XML) sobre HTTP. Aunque los mensajes SOAP son un poco complejos, esta complejidad es transparente para el desarrollador. Cuando se desarrollan las implementaciones del lado del cliente y del servidor, los desarrolladores no tienen que generar mensajes SOAP porque estos los generan las llamadas al API JAX-RPC.

En el lado del servidor, el desarrollador especifica los procedimientos remotos definiendo métodos de interfase escrito en Java. El desarrollador también codifica una o más clases que implementan esos métodos. Los programas cliente también son fáciles de codificar. Un cliente crea un Proxy, un objeto local que representa el servicio, y luego simplemente llama a los métodos del Proxy. Con JAX-RPC, el desarrollador no tiene que generar o analizar mensajes SOAP. Es el sistema de ejecución JAX-RPC el que convierte las llamadas y respuestas a/desde el API a mensajes SOAP.

#### **1.1.7.1 Implementación de un Servicio Web en Java**

Los pasos para crear un sencillo Servicio Web son:

- Crear la lógica del negocio del Servicio Web. Primero necesitamos escribir una clase Java que implemente la lógica de negocio del servicio Web.
- Desplegar la clase Java en el servidor SOAP. Luego necesitamos convertir la clase Java en un Servicio Web. Mostraremos como desplegar la clase Java en un servidor SOAP usando la herramienta JWSDP.
- Generar las clases de acceso para los clientes. Una aplicación cliente usa un objeto proxy para acceder al Servicio Web. Cuando se realiza la solicitud, el proxy acepta una llamada a un método Java desde la aplicación y la traduce en un mensaje XML. Devolviendo una respuesta, el proxy recibe el mensaje de respuesta SOAP, lo traduce en objetos Java y devuelve el resultado a la aplicación cliente.

- Desarrollar la aplicación cliente. La aplicación cliente trata el proxy como un objeto Java estándar que facilita la comunicación con un Servicio Web.

### **1.1.7.2 Interoperabilidad de los Servicios Web**

La interoperabilidad es una de las principales promesas de los Servicios Web. Los servicios web están diseñados para ser independientes del sistema operativo subyacente y del lenguaje de programación.

Al ser SOAP extremadamente extensible y versátil, para conseguir la interoperabilidad al nivel del protocolo de transporte. Las dos partes implicadas en un intercambio de mensajes deben ponerse de acuerdo para utilizar el mismo protocolo de transporte, como HTTP, SMTP, o JMS. Pero la compatibilidad con el protocolo de transporte no garantiza necesariamente la interoperabilidad. Hay otros problemas que afectan a la interoperabilidad. Un ejemplo es el tipo de codificación de los datos. Un estilo de codificación define cómo se codifican programáticamente en XML los tipos de datos. La especificación SOAP no obliga a ningún estilo de codificación particular, y los desarrolladores son libres para usar el estilo de codificación que quieran. En teoría virtualmente hay infinidad de estilos de codificación incompatibles entre sí. Afortunadamente hay una convención estándar.

## **1.2- PROTOCOLO DE COMUNICACIÓN DE WEB-SERVICES: SOAP**

### **1.2.1- ACCESO A SERVICIOS WEB: SOAP**

SOAP es un protocolo basado en XML, para el intercambio de información en un ambiente descentralizado y distribuido. El SOAP permite la intercomunicación entre objetos de cualquier tipo - sobre cualquier plataforma, en cualquier lenguaje.

El SOAP es un protocolo que opera en un contexto más amplio, el de los servicios Web, aparejado con el estándar UDDI (Universal Description, Discovery, and Integration) para proporcionar servicios de registro y mensajería entre los negocios.



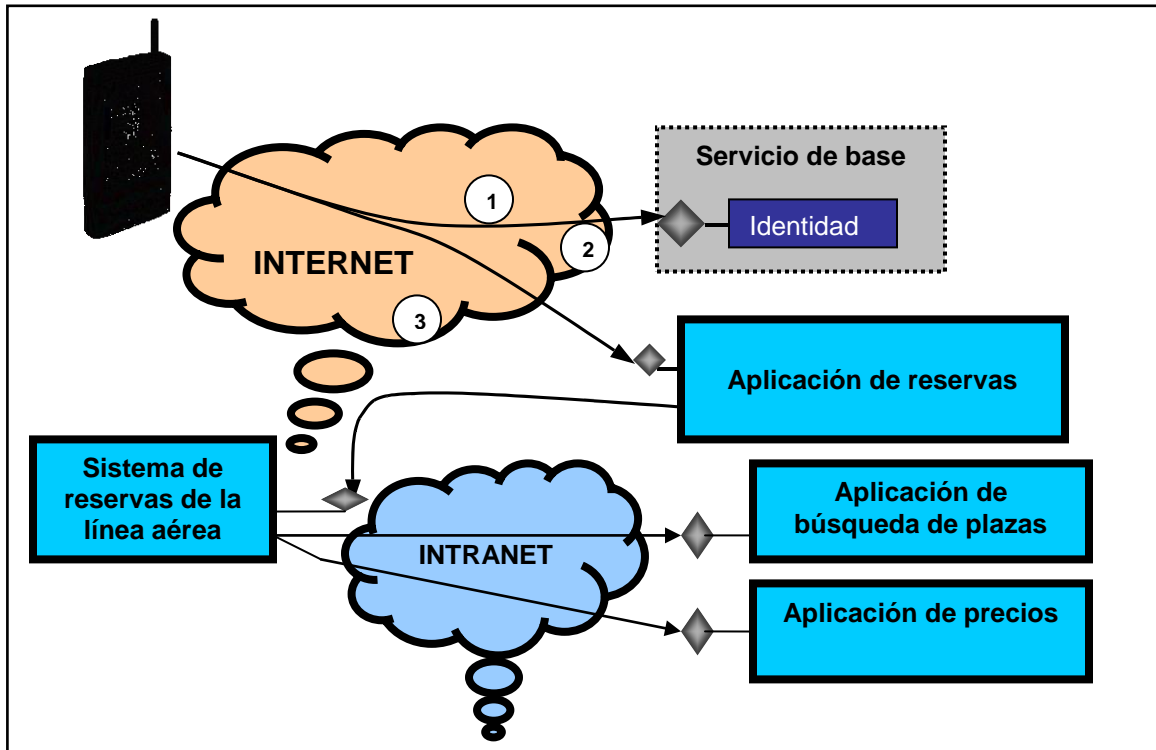


Figura 1.5 SOAP

La última parte opcional de la especificación SOAP define cómo es un mensaje HTTP que contiene un mensaje SOAP. Este enlace HTTP es importante, ya que HTTP se puede utilizar en casi todos los sistemas operativos actuales (y en otros que no lo son tanto). El enlace HTTP es opcional, pero casi todas las versiones de SOAP son compatibles con él ya que es el único protocolo normalizado para SOAP. Por esta razón se ha extendido la creencia errónea de que SOAP requiere HTTP. Algunas versiones admiten transportes por MSMQ, MQ Series, SMTP o TCP/IP, pero la mayor parte de los servicios Web XML actuales utilizan HTTP debido a su universalidad. HTTP es el protocolo fundamental de la Web, por lo que la mayor parte de las organizaciones cuentan con una infraestructura de red compatible con HTTP y personal que sabe cómo administrarla. Las infraestructuras de seguridad, supervisión y equilibrio de carga para HTTP están ya muy extendidas.

La característica más atractiva de SOAP es que se ha introducido en muchas plataformas de hardware y software diferentes. Esto significa que se puede utilizar

SOAP para vincular sistemas distintos dentro y fuera de una organización. En el pasado se han hecho muchos intentos para conseguir un protocolo de comunicaciones común que se pudiera utilizar para la integración de sistemas, pero ninguno se ha difundido tanto como SOAP. SOAP es mucho más pequeño y mucho más fácil de aplicar que muchos de los protocolos anteriores.

La universalidad de HTTP y la simplicidad de SOAP los convierte en la base ideal para poner en marcha servicios Web XML que se pueden activar desde prácticamente cualquier entorno.

### **1.2.2- ARQUITECTURA DE UNA PETICION/RESPUESTA DE SOAP**

En principio una petición/respuesta de SOAP puede viajar sobre cualquier protocolo de transporte (HTTP, SMTP, etc.). Los encabezados diferirán de protocolo a protocolo, pero el contenido de SOAP es el mismo. La petición de SOAP en XML consiste de tres partes:

El sobre ó envoltura de SOAP define un marco de referencia general para expresar qué hay en el mensaje, quién debe de atenderlo, y si es opcional u obligatorio. .

Las reglas de codificación de SOAP definen un mecanismo de serialización que puede ser utilizado para intercambiar instancias de tipos de datos definidos por la aplicación.

La representación RPC de SOAP define una convención que puede ser utilizada para representar los llamados y respuestas de procedimientos remotos.

Un mensaje SOAP/HTTP es aceptado típicamente por un servlet que corre en el servidor Web de Apache (Tomcat). En su momento el servlet revisa si hay alguna petición de SOAP y la transfiere al motor SOAP de Apache. El motor analiza el contenido de texto de XML y utiliza la dirección destino del servicio Web para buscar e invocar el servicio y brindar el resultado.

La respuesta SOAP/HTTP es regresada como un documento XML dentro de una respuesta HTTP estándar. El documento XML está estructurado como cualquier respuesta excepto que el Cuerpo contiene inmerso el resultado método.

En cuanto al rendimiento de los mensajes de SOAP utilizando HTTP y XML, comparado con el rendimiento de mensajes binarios utilizados por CORBA, DCOM y RMI, se puede agregar que en el caso de los últimos, tanto el destinatario y remitente tienen conocimiento completo del contenido del mensaje y no codifican meta información como los nombres o tipo de argumentos. Quizás este método sea eficiente, pero dificulta a los intermediarios el procesamiento de mensajes. Y debido a que cada sistema utiliza una codificación binaria distinta dificulta la construcción de sistemas inter operables.

Dado que SOAP utiliza XML para codificar mensajes, es relativamente sencillo procesar los mensajes en cada paso del proceso de llamado. Además, la facilidad de depuración de mensajes SOAP permite la convergencia rápida de las diversas implementaciones de SOAP, la cual es importante en la interoperabilidad a gran escala.

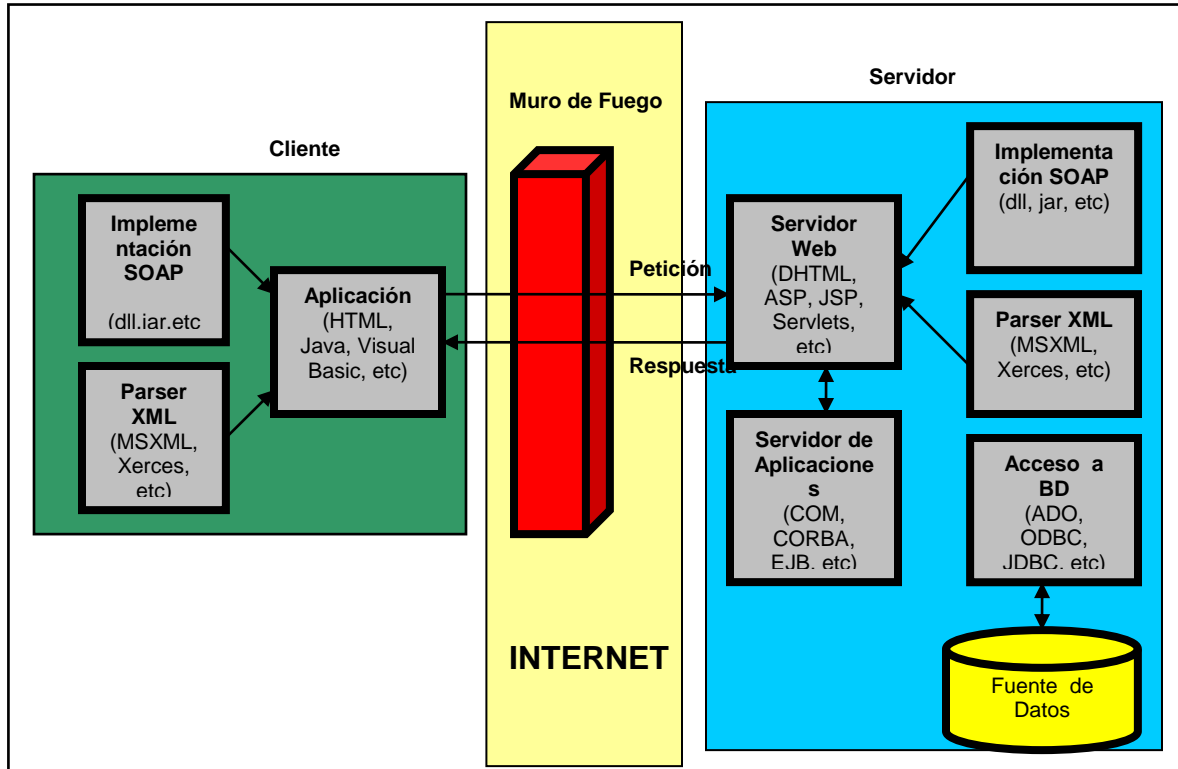


Figura 1.6 Arquitectura de una Petición / Respuesta SOAP

### 1.2.3- CAPAS DE RED

En la evolución de servicios de Web han resaltado tres capas de red: TCP/IP, HTTP/HTML, y XML. Estas capas están colocadas una arriba de otra, y permanecen de esta manera hoy en día.

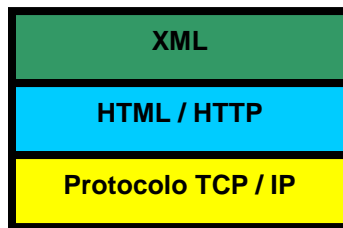


Figura 1.7 Capas de Red

El primer nivel, el protocolo TCP/IP, se encarga primordialmente de pasar los paquetes de datos por el cable. El TCP/IP es un protocolo que garantiza la transmisión a través de redes públicas, enfatizando en confiabilidad del transporte de datos y la conectividad física.

El segundo nivel, HTML sobre HTTP, es una capa de presentación y se ocupa de búsquedas, recuperación y compartición de información basadas en navegadores. El énfasis aquí es la navegación a través de una interfaz gráfica y la manipulación de formatos de presentación. A pesar de que el HTML carece del potencial de ser programable y de ser expandido, la compartición de documentos hipertextos en un ambiente orientado a navegadores gráficos revolucionó la forma en la comunicación humana. Los ambiente de trabajo en red, cargados de sistemas operativos propietarios y software dependiente de la plataforma, están cambiando y abriendo camino a los sistemas abiertos de cómputo en el Internet que son basados en estándares.

Al frente de este nuevo mundo basado en estándares se encuentra el XML, que es el tercer nivel y quizás el cohesionador más importante del Internet. El XML, es un formato de intercambio de datos fuertemente estructurados, que proporciona una nueva dimensión al nivel HTTP/HTML, en el cual la comunicación máquina-a-máquina

se realiza por la interfaces estándar. Esta capa permite a los programas intercambiar datos formateados en forma independiente de plataforma y presentación. Se pueden agregar hojas de estilo XSLT opcionales como componente de presentación y/o transformación

### **1.3- LENGUAJE DE DEFINICIÓN DE SERVICIOS WEB: WSDL**

#### **1.3.1- DEFINICION DE INTERFACES PARA SERVICIOS WEB: WSDL**

WSDL significa lenguaje de descripción de servicios Web. Un archivo WSDL es un documento XML que describe un conjunto de mensajes SOAP y la forma en que se intercambian esos mensajes. En otras palabras, WSDL es a SOAP lo que IDL es a CORBA o COM. Puesto que WSDL es XML, se puede leer y editar en la mayor parte de los casos y es generado y consumido por software.

La notación que utiliza un archivo WSDL para describir formatos de mensajes se basa en el estándar de esquemas XML, lo que significa que es independiente del lenguaje de programación y que está basada en estándares. Esto hace que sea adecuada para describir interfaces de servicios Web XML a las que se puede acceder desde una gran variedad de plataformas y lenguajes de programación. Además de describir los contenidos de los mensajes, WSDL define dónde se puede acceder al servicio y qué protocolo de comunicaciones se utiliza para ello. Es decir, el archivo WSDL define todo lo necesario para escribir un programa para trabajar con un servicio Web XML. Existen varias herramientas para leer un archivo WSDL y generar el código necesario para comunicarse con un servicio Web XML.

Muchos de los actuales kits de herramientas de SOAP incluyen herramientas para generar archivos WSDL a partir de interfaces de programas existentes, pero hay pocas que permitan escribir archivos WSDL directamente. La gama de herramientas para WSDL no es todo lo completa que debiera, pero no pasará mucho tiempo antes de que herramientas para crear archivos WSDL y luego generar proxys y códigos auxiliares, muy similares a las herramientas IDL de COM, pasen a formar parte de la mayoría de

versiones de SOAP. Esto convertirá a WSDL en la mejor forma de crear interfaces SOAP para servicios Web XML.

### 1.3.2- ARQUITECTURA DE UN DOCUMENTO WSDL

Los componentes de Descripción de Servicio son:

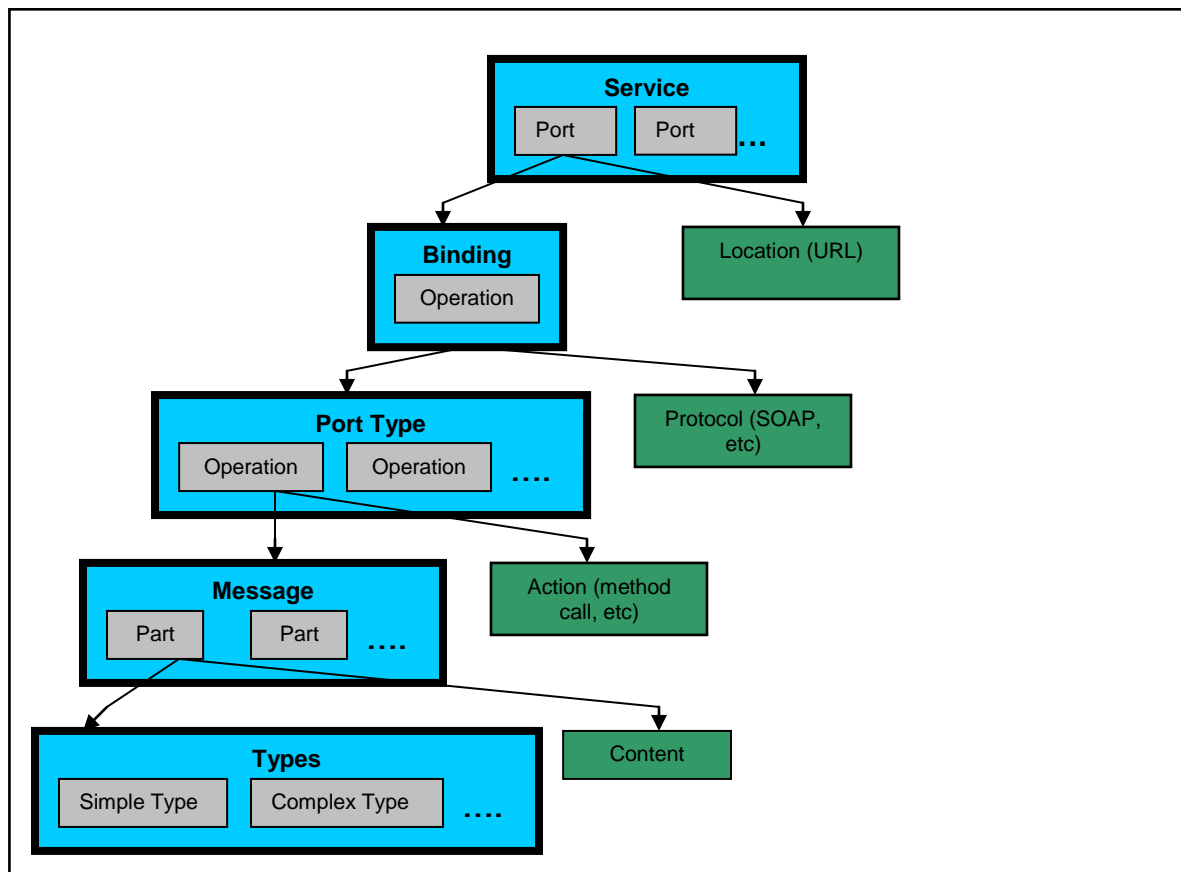


Figura 1.8 Arquitectura de un Documento WSDL

- Service = Conjunto de “ports” relacionados que implementan el servicio.
- Port = “Port type” + “Binding”
- Es un descripción abstracta de una acción soportada por el servicio. Cada operación se corresponde a un mensaje de input o de output

- Port type = Colección de “operations” o “signatures” de los métodos que definen el intercambio ordenado de los mensajes.
- Bindings = especifica los protocolos usa cada “port” y el “encoding”.
- Message = descripción de los datos que van a ser transmitidos. Son una colección de “data values” de un tipo particular (utilizando XML Schema como mecanismo de tipación).

## **1.4- DIRECTORIO UNIVERSAL DE OBJETOS: UDDI**

### **1.4.1- DEFINICION DE UDDI**

UDDI (descripción, descubrimiento e integración universales) es como las páginas amarillas de los servicios Web donde se puede buscar una empresa que ofrezca cualquier tipo de servicios, leer sobre el servicio ofrecido y ponerse en contacto con alguien para conseguir más información. Por supuesto, puede ofrecer un servicio Web sin registrarlo en UDDI, igual que puede abrir una empresa en su sótano y recurrir al boca a boca para la publicidad, pero si realmente quiere llegar al mercado necesitará UDDI para que sus clientes puedan conocerle.

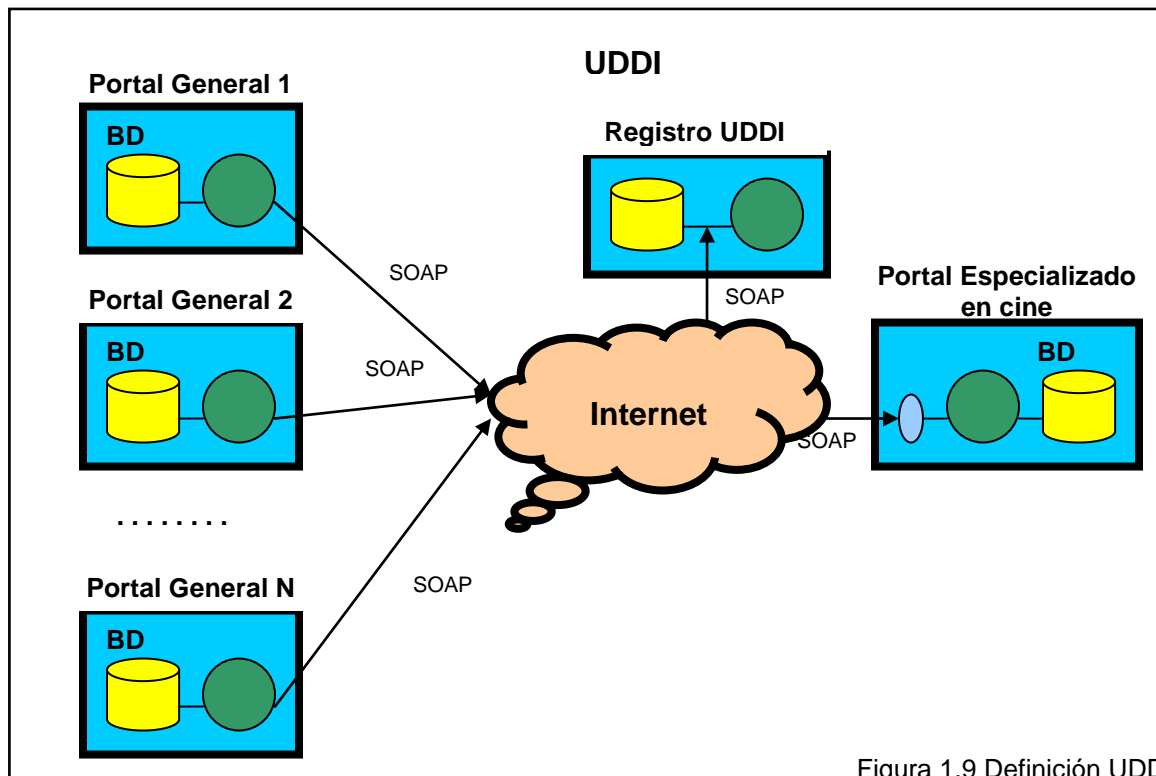


Figura 1.9 Definición UDDI

#### 1.4.2- BUSQUEDA DE SERVICIOS WEB: UDDI

Una organización que desee ofrecer servicios Web puede crear un registro de empresa UDDI, que es un documento XML cuyo formato se especifica en un esquema definido por UDDI. Una vez creados, estos registros de empresa se almacenan en una base de datos replicada conocida como registro de empresa UDDI. Cada sitio de operador UDDI mantiene una copia del registro de empresa. Estos sitios se comunican entre sí según las necesidades para mantener la coherencia de la información que almacenan. Para acceder a la información del registro UDDI, el software de cliente utiliza siempre SOAP por HTTP (de hecho, UDDI es también un servicio Web que proporciona información sobre otros servicios Web).



El directorio UDDI también le ofrece varias formas de buscar los servicios que necesita para crear sus aplicaciones. Por ejemplo, puede buscar los proveedores de un servicio en una zona geográfica concreta o una empresa de un tipo determinado. El directorio UDDI le proporcionará información, contactos, enlaces y datos técnicos que le permitirán decidir cuáles son los servicios que cumplen sus condiciones.

UDDI le permite encontrar empresas cuyos servicios Web podrían interesarle.

Con la especificación WS-Inspection puede explorar una colección de servicios Web XML ofrecidos en un servidor concreto para encontrar los que más se adapten a sus necesidades.

UDDI es una pieza importante en el puzzle de los servicios Web. Sin él, las organizaciones que tienen servicios Web no podrían dar a conocer a sus posibles clientes qué es lo que ofrecen, y los clientes no podrían conocer fácilmente los datos que necesitan para acceder a esos servicios. Aunque uno de los principales objetivos de UDDI es hacer que los clientes puedan localizar servicios en Internet, también puede resultar muy útil dentro de una organización. Los servicios Web suponen un avance significativo, pero serían mucho menos útiles sin un servicio como UDDI.

### **1.4.3- SECCIONES BLANCA, AMARILLA Y VERDE**

Conceptualmente, la información proporcionada de un negocio en un registro UDDI consta de tres componentes:

- La Sección Blanca es muy similar a la información que aparece en el directorio telefónico, que incluye nombre, teléfono, y dirección.
- La Sección Amarilla es muy similar a su equivalente telefónico, e incluyen categorías de catalogación industrial tradicionales, ubicación geográfica, etc. Mediante el uso de códigos y claves predeterminadas, los negocios se pueden

registrar y así facilitar a otros servicios la búsqueda usando estos índices de clasificación.

- La Sección Verde contiene la información técnica acerca de los servicios ofrecidos por los negocios. Se incluyen referencias de especificaciones de servicios Web así como también información complementaria para los mecanismos diversos de búsqueda basados en URL.

#### **1.4.4- ARQUITECTURA CENTRAL DE UDDI**

La información que compone un registro UDDI consiste por el momento de cuatro tipos de estructuras de datos.

- **businessEntity:** Esta estructura captura la información sobre un negocio o entidad y que es utilizada por el negocio para publicar información descriptiva sobre si mismo y los servicios que ofrece.
- **businessService:** Esta estructura representa los servicios o procesos de negocios que provee la estructura businessEntity.
- **bindingTemplate:** Esta estructura representa los datos importantes que describen las características técnicas de la implementación del servicio ofrecido.
- **tModel:** El papel principal de esta estructura es la de representar una especificación técnica.

#### **1.4.5- CARACTERISTICAS DE UDDI**

El UDDI ofrece 2 interfaces: una para los proveedores de servicios y otra para quien consume los servicios.

La interfase de publicación, provee el mecanismo para que los proveedores de servicios se registren ellos mismos y sus servicios en el Registro UDDI.

La interfase de consulta permite a los subscriptores de servicios buscar los servicios disponibles. El cuál provee dos tipos de llamados, un mecanismo de búsqueda y un mecanismo de obtención, cuando ya se tiene disponible toda la información referente a la disponibilidad de un servicio

## **1.5- JAVA WEB SERVICES DEVELOPER PACK: JWSDP**

### **1.6.1- JWSDP**

#### **1.6.1.1- Definición**

Es un kit de herramientas gratis integrada que permite a diseñadores de Java construir e implementar aplicaciones de XML, Servicios Web, aplicaciones Web con la última Tecnología y estándares de implementación.

#### **1.6.1.1- Instalación del Software**

##### **REQUERIMIENTOS:**

- Instalar una versión del SDK superior a la 1.3 en el directorio C:\j2sdk
- Instalar JWSDP -1.3 o superior C:\jwsdp-1.3
- Disponer de un navegador HTTP estándar
- Definir las siguientes variables de entorno

<b>Nombre Variable</b>	<b>Valor Variable</b>
JAVA_HOME	C:\j2sdk
ANT_HOME	C:\jwsdp-1.3\apache-ant
JWSDP_HOME	C:\jwsdp-1.3
Path	C:\j2sdk\bin;C:\jwsdp-1.3\apache-ant\bin;%JWSDP_HOME%\bin;%ANT_HOME%\bin;

Si eliges nombres diferentes de directorios, cambia las variables de entorno definidas en la tabla anterior para que apunten respectivamente al directorio de instalación de SDK Y JWSDP.

### 1.6.1.1- Funcionamiento de JWSDP

Para poner en funcionamiento este servidor se puede hacer a través del menú Inicio-> Programas->Java (TM) Web Services Developer Pack 1.3 -> Start Tomcat.

Esto arrancará el proceso, que pondrá un servidor a escuchar en el puerto 8080 de TCP. También se puede ejecutar el script `c:\jwsdp-1.3\bin\start.bat`  
Para pararlo se irá al menú inicio-> Programas ->Java(TM) Web Services Developer Pack 1.3 -> Stop Tomcat. También se puede ejecutar el script `c:\jwsdp-1.3\bin\stop.bat`

Para comprobar que el servidor está funcionando, acceda con un navegador al UR: `http://localhost:8080/`, donde deberá salir una página como:



Las aplicaciones consisten en una colección de servlets, páginas HTML, JSP, beans y otros recursos que se pueden empaquetar y ejecutar en distintos contenedores Web, de distintos fabricantes y que ofrecen una determinada funcionalidad a la que los clientes acceden típicamente a través de un navegador.

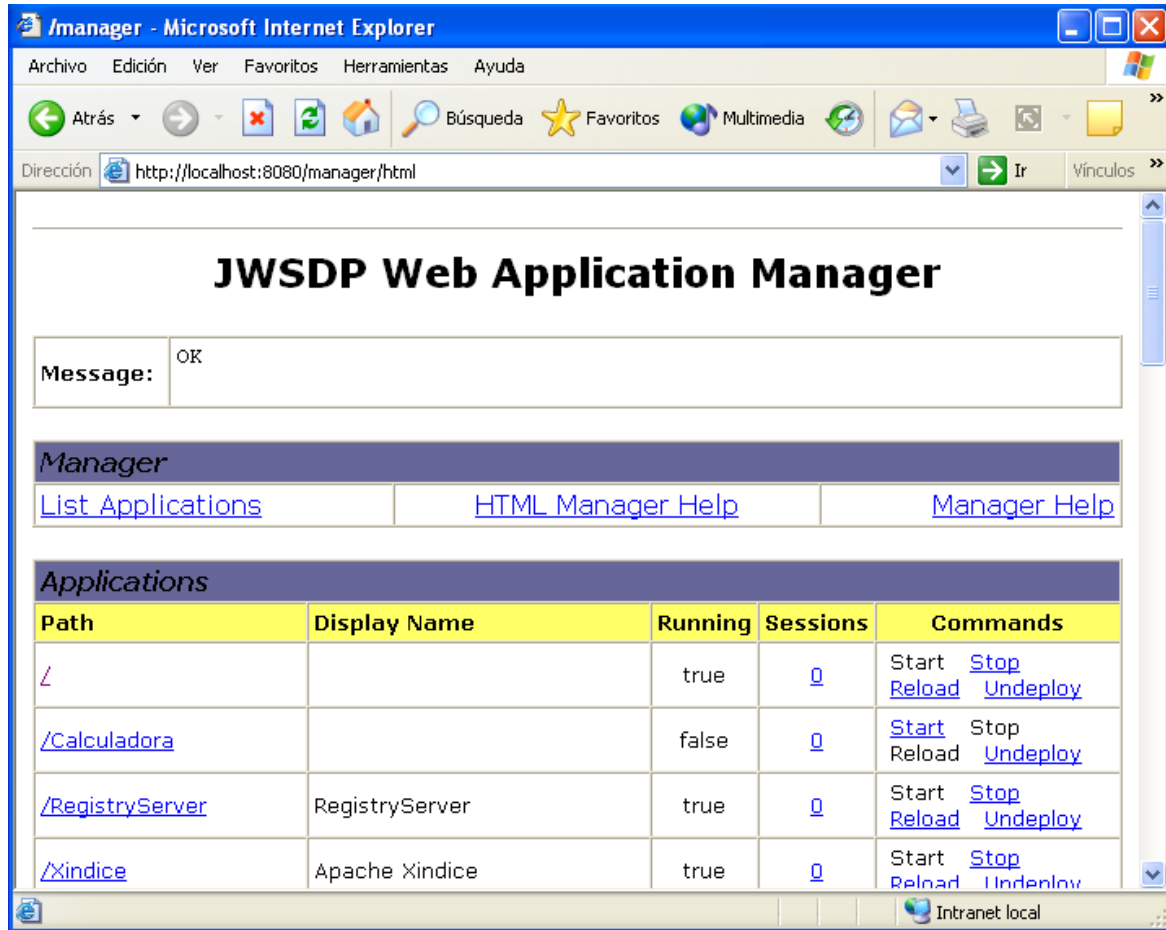
El ciclo de vida de desarrollo típico de aplicaciones Web es:

- Codificar y compilar
- Empaquetar y desplegar
- Ejecutar y visualizar

Las Aplicaciones Web se instalarán por defecto en el directorio `c:\jwsdp-1.3\webapps`

Para acceder a un servlet, el URL del navegador deberá ser:  
`http://localhost:8080/contexto aplicación Web/servlet/nombre del servlet`

Para gestionar las aplicaciones que se encuentran en el contenedor (arrancarlas, pararlas, desplegarlas, etc.) se puede utilizar el gestor Web que incluye Tomcat, en la dirección `http://localhost:8080/manager/html`. Para acceder a dicha página se solicitará un usuario y contraseña que por defecto son "tomcat" en ambos casos, pero dependerá del nombre q usted ingrese en la instalación.



### 1.6.1.1- Paquete de Aplicaciones

JWSDP incluye:

- JavaServer Faces (JSF) v1.0 EA4
- XML and Web Services Security v1.0 EA2
- Java Architecture for XML Binding (JAXB) v1.0.2
- Java API for XML Processing (JAXP) v1.2.4
- Java API for XML Registries (JAXR) v1.0.5
- Java API for XML-based RPC (JAX-RPC) v1.1
- SOAP with Attachments API for Java (SAAJ) v1.2
- JavaServer Pages Standard Tag Library (JSTL) v1.1 EA

- Java WSDP Registry Server v1.0\_06
- Ant Build Tool 1.5.4
- Apache Tomcat v5 development container
- Ws-I Supply Chain Management Sample Application 1.0

JAX-RPC es el API de java para la creación de Servicios Web utilizando llamadas al procedimiento remoto. Está incluido en el JWSDP (Java Web Services Developer Pack), perteneciente a la plataforma J2EE.

Como hemos dicho anteriormente, JAX-RPC trabaja con llamadas al procedimiento remoto, es decir, el cliente invoca un proceso remoto desplegado en un servidor o contenedor Web, y el servicio le responde mediante un mensaje. Para esta comunicación, utiliza los mensajes SOAP, que viajan por el protocolo HTTP.

Una de las principales características de esta API, es la posibilidad de enviar archivos completos como respuestas, o archivos fragmentados, y no solo mensajes.

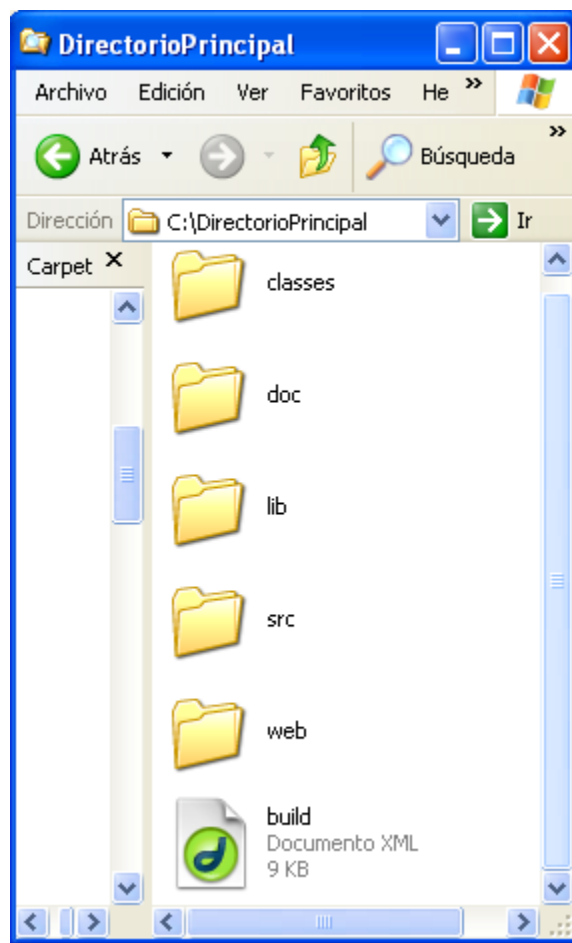
También cabe destacar que JAX-RPC respeta una de las características fundamentales de los Servicios Web, y es la interoperabilidad entre servicios y clientes que utilizan distintas plataformas. Es decir, un cliente que utiliza JAX-RPC, y por tanto está implementado en java, puede comunicarse con un servicio que no esté implementado en java, y viceversa. Esto es debido a que JAX-RPC se centra en los mensajes SOAP.

**Ant.** Este programa es como el tradicional make que se usa para compilar principalmente las aplicaciones que se desarrollan en C y C++. Ant es una herramienta diseñada especialmente para el desarrollo de aplicaciones java

### **1.6.1- FUNDAMENTOS PARA EL DESARROLLO DE APLICACIONES USANDO JWSDP**

Para el desarrollo de aplicaciones o servicios Web usando la tecnología JWSDP es necesario tener en cuenta que ahora las aplicaciones publicadas en un servidor Web deben tener una jerarquía de directorios que corresponde a la estructura que conforma una aplicación Web los cuáles son:

- src en este directorio se localizaran los archivos fuente codificados en java.
- web en este directorio se localizaran los archivos en formato html o jsp.
- docs en este directorio se localizaran la documentación relacionada a la aplicación que se desarrollara.
- lib en este directorio se localizaran las librerías o archivo .jar utilizados en la aplicacoión.





A su vez es necesario que se definan unos archivos que van a ser usados para compilar e instalar los programas fácilmente. Estos archivos son:

build.xml este archivo contiene las etiquetas que permiten la compilación de los programas, su instalación, su remoción de un contenedor Web (eg. Tomcat). Este archivo es esperado por el comando ant.

Recuerde que el build.xml es un archivo similar al makefile o Makefile del programa make.

Ahora dentro de las propiedades del archivo build las más importantes son:

- app.path escoja una ruta de localización de su aplicación, por ejemplo /demo.
- app.name nombre de la aplicación para este caso demo.
- catalina.home es el directorio donde esta instalado el jwsdp-1.3
- manager.url es el url del sitio donde se administran las páginas WEB. Se recomienda hacer el desarrollo de sus aplicativos en la misma maquina donde opera el JWSDP, por lo tanto el valor de esta variable puede ser `http://localhost:8080/manager`.
- manager.username el nombre del usuario administrador. Este usuario es creado por el instalador de JWSDP.
- manager.password password del administrador de aplicativos en JWSDP. Suministrado a la hora de la instalación.

## **1.6- WASP**

### **1.6.1- INTRODUCCION**

Systinet WASP Server para Java es un Servidor Web que soporta SOAP 1.1, SOAP 1.2, y WSDL 1.1.

Puede correr como un servicio auto suficiente o puede desplegar en J2EE aplicaciones servidores como son BEA Weblogic, IBM Websphere ,y SUN ONE Application Servers además de una gran variedad de Servlets construidos.

WASP esta muy optimizado y es ínter operable con otras plataformas de Servidores Web como son .NET.

Además también posibilita la creación de un unificado Servicio Web seguro y manejable

### 1.6.2- Arquitectura

**Figure 1. Systinet WASP Server para la Arquitectura Java**

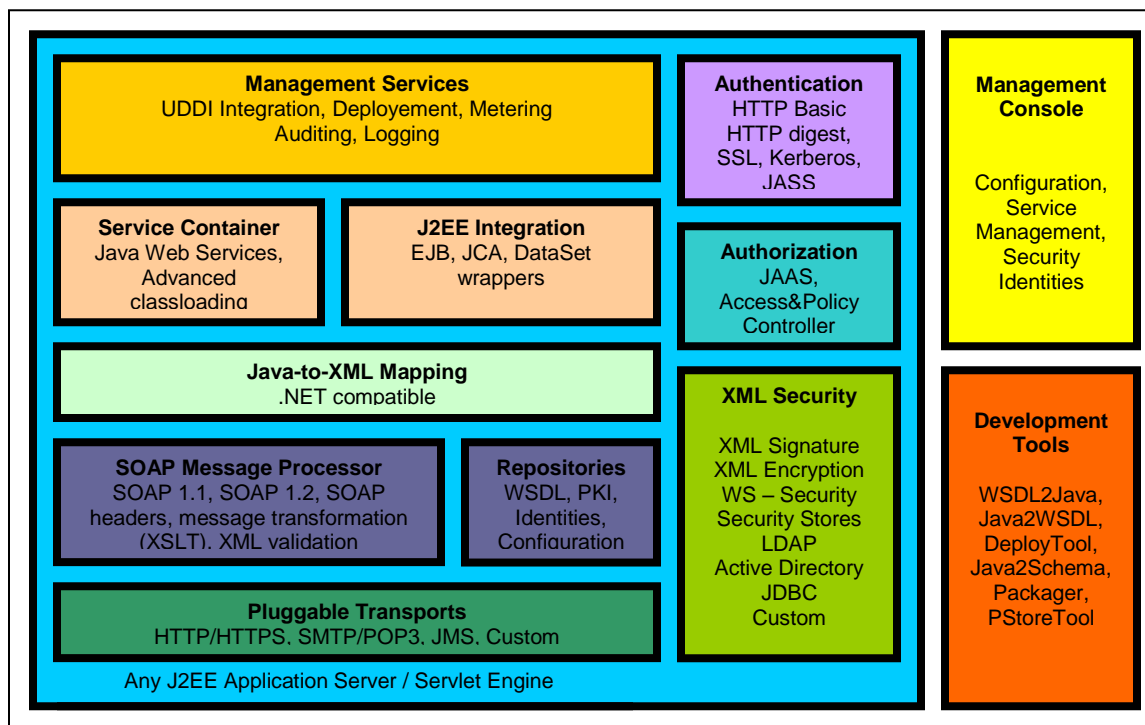


Figura 2.0 Arquitectura WASP

### 1.6.1- Instalación del Software

REQUERIMIENTOS:

- Instalar una versión del SDK superior a la 1.3 en el directorio C:\j2sdk
- Instalar WASP 4.7.1 o superior C:\wasp

- Disponer de un navegador HTTP estándar
- Definir las siguientes variables de entorno

<b>Nombre Variable</b>	<b>Valor Variable</b>
JAVA_HOME	C:\j2sdk
ANT_HOME	C:\ant
WASP_HOME	C:\wasp
Path	%WASP_HOME%\bin;

Si eliges nombres diferentes de directorios, cambia las variables de entorno definidas en la tabla anterior para que apunten respectivamente al directorio de instalación de SDK Y WASP.

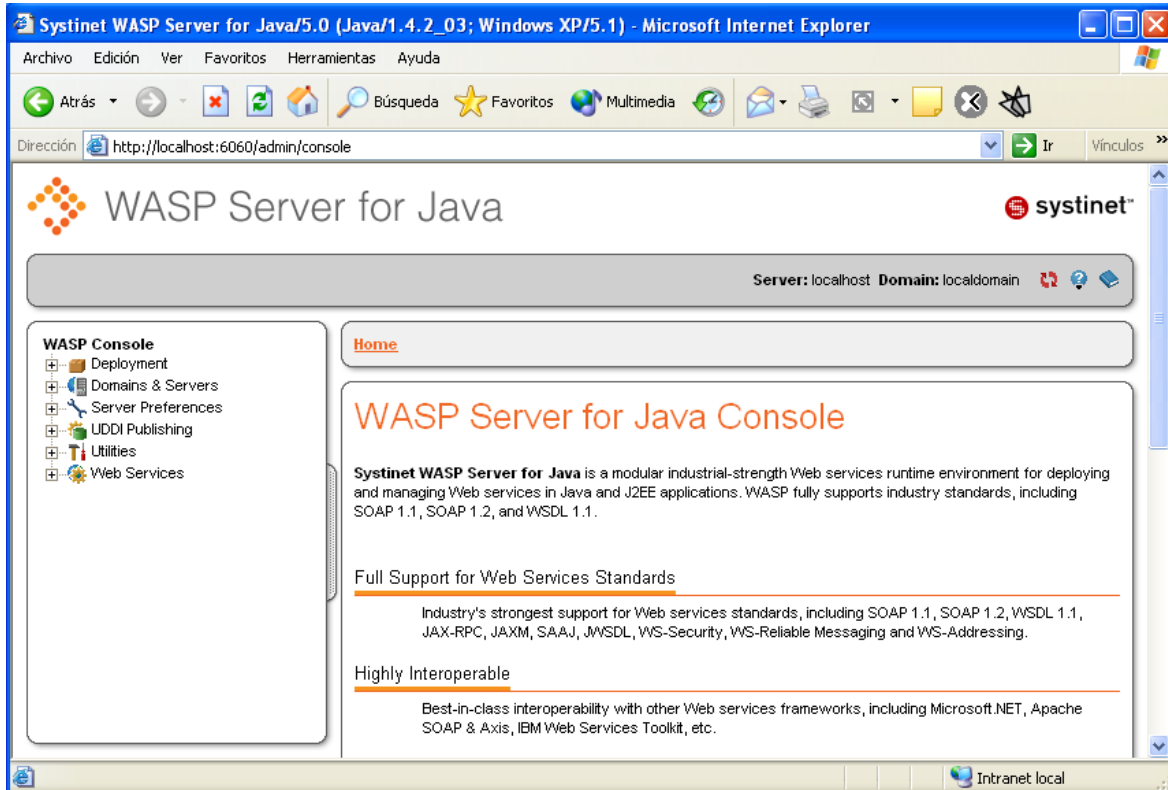
#### 1.6.1.1- **Funcionamiento de WASP**

Para poner en funcionamiento este servidor se puede hacer a través del menú Inicio-> Programas->Systinet Wasp 4.7.1 -> Start WASP Server.

Esto arrancará el proceso, que pondrá un servidor a escuchar en el puerto 6060 de TCP.

Para pararlo se irá al menú inicio-> Programas -> Systinet Wasp 4.7.1 -> Stop WASP Server.

Para comprobar que el servidor está funcionando, acceda con un navegador al UR: <http://localhost:6060/>, donde deberá salir una página como:



### 1.6.1.1- Paquete de Aplicaciones

WASP incluye:

- Java 1.3 or higher-enabled platform
- Tested on Windows NT/2000/XP, Linux, Sun Solaris 2.6 and HP-UX
- JCE 1.2.1 & Sun JCE providers
- JSSE 1.0.3 & Sun JSSE providers for SSL/HTTPS
- JAAS 1.0
- Tomcat 3.x/4.x, BEA WebLogic 6.x/7.x, IBM WebSphere 4.x, Orion 1.5.x, Oracle 9 iAS, Sun J2EE, JBoss 2.4.x, IBM MQSeries JMS Provider



# CAPITULO 2

## II.- METALENGUAJE DE MARCAS XML

Con la especificación de XML, el documento que lo define, se pretende asegurar el futuro de la información en Internet y crear un sistema que soporte el paso de los años, las modas y los monopolios.

A primera vista XML suena a algo que tuviera que ver con HTML, y así es en efecto. Ambos términos están relacionados, aunque no son lo mismo, ni tampoco es cierto que XML sea el sustituto de HTML. Para entender las diferencias hay que remontarse al pasado.

La diferencia fundamental entre HTML y XML es que mientras el primero es un lenguaje, XML es un metalenguaje. De hecho es un subconjunto de SGML, una versión reducida y totalmente compatible diseñada específicamente para la Web. Todos los documentos XML se ajustan a las normas de SGML.

Los documentos XML deben ser fáciles de crear, legibles por personas (no sólo por computadoras) y razonablemente claros. También debe ser fácil escribir programas de tratamiento de documentos XML y se debe reducir al mínimo el número de elementos opcionales.

Pero lo más importante es que, utilizando un metalenguaje, las empresas podrán desarrollar aplicaciones personalizadas para mostrar y recibir información, como páginas de comercio electrónico, sistemas de teletrabajo o gestión de bases de datos. Todas estas aplicaciones serán compatibles y fácilmente reutilizables, ya que compartirían la base común de XML.

### 2.1.- SGML: HTML

## **2.1.1- SGML**

### **2.1.1.1- Introducción a SGML**

SGML fue diseñado para permitir el intercambio de información entre distintas plataformas, soportes físicos, lógicos y diferentes sistemas de almacenamiento y presentación (bases de datos, edición electrónica, etc.), con independencia de su grado de complejidad.

### **2.1.1.2- Definición de Documentos SGML**

SGML es un acrónimo de Standard Generalized Markup Language o Lenguaje de Señalización General Normalizado, una norma anterior adoptada de IBM como parte de su sistema de procesamiento de textos. Se dice "generalizado" porque el etiquetado debe describir la estructura de un documento y otros atributos. SGML es un estándar internacional, no propietario y abierto, que provee un método para la descripción de la estructura de documentos basándose en la relación lógica de sus partes. Provee una codificación estándar para la transmisión de documentos entre sistemas de computadoras diferentes: distintas plataformas, soportes físicos, lógicos y diferentes sistemas de almacenamiento y presentación, con independencia de su grado de complejidad.

Para el etiquetado se utiliza un conjunto de caracteres basado en el estándar ASCII, reconocido de manera universal por cualquier tipo de plataforma y sistema. Los caracteres especiales no contemplados en ASCII se transforman en representaciones ASCII y se denominan referencias de entidad. En SGML todo el etiquetado es lógico, es decir, en lugar de utilizar códigos crípticos (^C, ^D, etc.) se utilizan nombres de elementos, delimitados por marcas que indican el comienzo y el final de los objetos lógicos. Estos delimitadores permiten que el software reconozca qué caracteres deben ser leídos en modo de etiqueta, y qué otros como contenido.

SGML debe utilizarse cuando existe alguna de las siguientes condiciones:

- Cuando existe la necesidad de intercambiar documentos entre diferentes sistemas de cómputo o de publicación.
- Cuando los documentos tendrán una vida de uso larga.
- Cuando la estructura de un documento es fundamental.

- Cuando se utiliza una base de datos para el almacenamiento y recuperación de elementos del documento.

Cuando se concibe un documento electrónico en SGML se debe tener en cuenta que:

- El material que constituye un documento se puede distribuir en diferentes archivos, tantos como sean necesarios. Estos archivos además pueden estar almacenados en un sistema distribuido. Un archivo puede contener la portada, otro la introducción, otro una parte de una hoja de cálculo, otro un gráfico, otro un organigrama, otro la bibliografía, etc.
- En SGML, cada uno de estos objetos recibe el nombre de entidad. Las entidades se conciben como objetos independientes.
- Las entidades pueden tener cualquier tamaño, haber sido creadas por cualquier programa de software o estar guardadas en cualquier computadora.
- Las entidades pueden ser compartidas por distintos documentos.
- Un documento estará definido en función de la estructura de las entidades que lo conforman.
- En el índice de materias de un documento no se encontrará ninguna referencia a los archivos que contienen las entidades.
- Las entidades se organizan en una estructura lógica de manera jerarquizada, en la que se definen conceptos como capítulos, tablas y párrafos y que configuran lo que se denomina estructura de los elementos del documento.
- Elementos y entidades pueden coincidir: un elemento lógico como tabla puede ser también una entidad en un archivo hoja de cálculo.

Un documento SGML se marca de tal modo que no dice nada respecto a su representación en la pantalla o en papel. Un programa de presentación (filtro) debe unir el documento marcado con un esquema de estilo, a fin de producir una representación impresa en la pantalla o en papel del contenido del documento.

### **2.1.1.3- Capacidades de SGML**

Con SGML se pueden definir varias características que dependan de las necesidades de cada documento:



- La ubicación de otros documentos dentro del sistema. En relación con el documento principal, otros documentos (sub-documentos) pueden ser fácilmente unidos mediante un comando conocido como referencia de entidad. Mediante la referencia de entidad, estos sub-documentos pueden referenciarse fácilmente.
- Caracteres especiales requeridos en un documento. SGML provee un mecanismo para la recuperación de caracteres de notación matemática o símbolos y acentos diacríticos de idiomas extranjeros.
- Simplificaciones. Como esfuerzo para reducir el número de palabras reservadas para producir un documento SGML, el lenguaje provee palabras simplificadas o reducidas que pueden utilizarse en diferentes formas. Cadenas de caracteres largas pueden invocarse con un comando sencillo. Los comandos de etiquetado de SGML pueden reducirse cuando el autor está escribiendo el manuscrito, a tal grado, que el autor no necesita preocuparse los numerosos comandos y sólo se concentrará en unos cuantos que le son necesarios. Esto es especialmente útil cuando varios autores trabajan en diferentes capítulos de un libro.

#### **2.1.1.4- Componentes de un Documento SGML**

Un documento SGML actualmente está compuesto de tres archivos que trabajan de manera conjunta: el DTD, la declaración SGML y la Muestra de Documento. Los tres archivos son necesarios para la creación de un documento SGML.

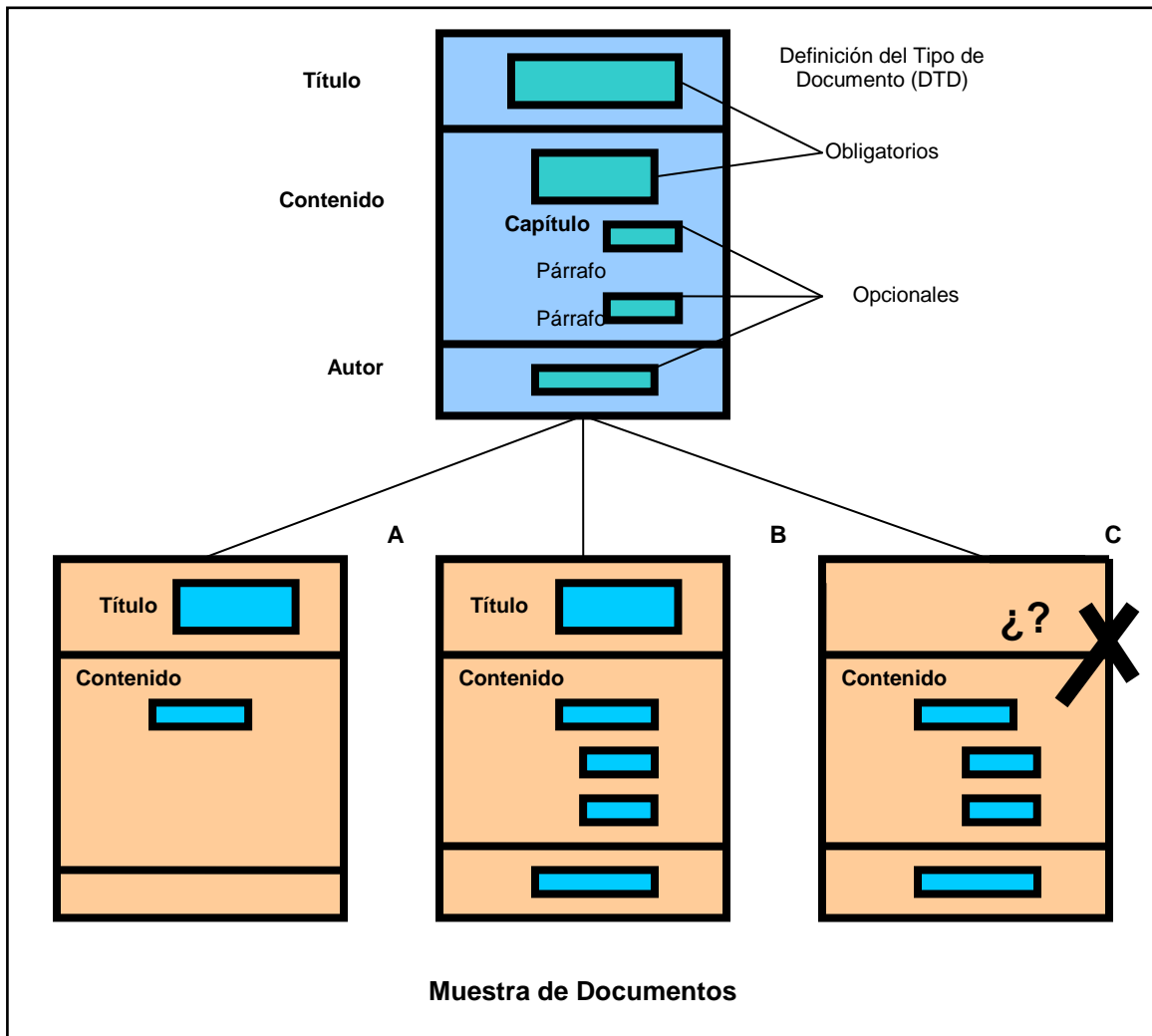


Figura 1: Concepto de un documento SGML

### 2.1.1.5- Introducción a DTD(Document Type Definition o Definición del Tipo de Documento)

En el DTD se identifica la estructura del documento, es decir, aquellos elementos que son necesarios en la elaboración de un documento o un grupo de documentos estructurados de manera similar. Contiene las reglas de dichos elementos: el nombre, su significado, dónde pueden ser utilizados y qué pueden contener. Si el DTD se almacena en un archivo separado, éste se puede referenciar sin residir dentro del documento SGML. Sin el DTD ( o una referencia a él), el documento SGML no será validado apropiadamente por el parser. El parser es una herramienta que asegura la adecuada conformación de SGML dentro de las múltiples especificaciones que un documento SGML puede tener. Por ejemplo, en la figura 7, el Documento Muestra C no

será validado por el parser ya que no cumple con las especificaciones del DTD, porque carece de título y éste es un elemento obligatorio.

En este ejemplo, la definición del DTD sería la siguiente:

```
<!ELEMENT DOCUMENTO - - (titulo, contenido, autor?) >
```

```
<!ELEMENT CONTENIDO - - (capitulo+) >
```

```
<!ELEMENT CAPITULO - (subtitulo, parrafo?)
```

Lo que está dentro de los paréntesis especifica qué puede o debe contener el elemento y es llamado el contenido del modelo. Es obligatorio que haya un título y un contenido, pero es opcional que tenga autor. Dentro de CONTENIDO puede haber uno o más capítulos, que a su vez también es otro elemento que contiene subtítulo y párrafos. Puede verse claramente que hay una estructura organizada y jerárquica.

Escribir el DTD es la tarea más compleja e importante. Se requiere paciencia y previsión, empezar con un análisis de los documentos y continuar con los requerimientos futuros que se pueden tener en cambios, revisiones y borrado de los documentos.

Gracias a las etiquetas definidas en el DTD las aplicaciones como editores, motores de búsqueda y visualizadores interpretan los documentos SGML.

Los documentos no tienen una estructura generalizada, es decir, una narración tiene una estructura muy diferente a la de un verso o a la de un reporte científico y algunos casos incluso hay variaciones entre documentos del mismo tipo, por tanto, es recomendable que un especialista en el tipo de documentos, acompañe al analista para definir la estructura de los documentos y poder jerarquizar y desmembrar adecuadamente las partes que contendrá el DTD.

Un fichero DTD define siempre una o más estructuras jerárquicas, con una marca principal, o padre, compuesta por otras marcas, o hijos. La figura muestra la estructura de un DTD simple, con un elemento principal <article>. Dentro del elemento principal pueden aparecer otros elementos: <arthead>, <sect1> y <index>, y estos a su vez se componen de otros elementos.

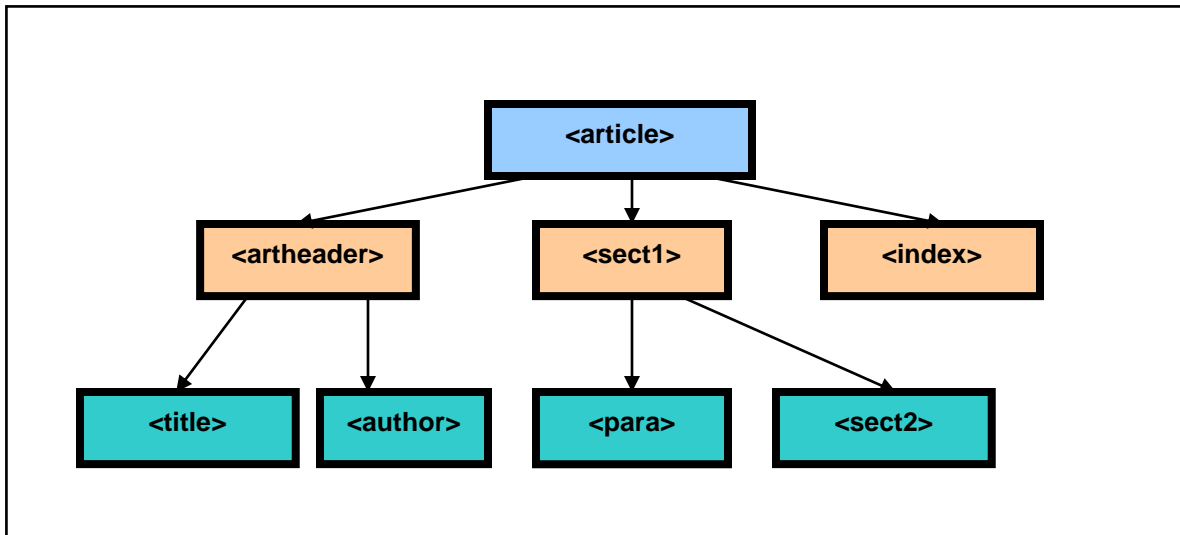


Fig. 2: Estructura jerárquica de un DTD.

El DTD puede ser diseñado de forma a hacer obligatorio el uso de algunos sub-elementos y limitar el número de veces que un elemento puede aparecer y el orden de los elementos. De esta forma el DTD puede ser bastante flexible o tan exigente como se desee, para forzar a los autores a ceñirse a un determinado estilo.

#### 2.1.1.6- Declaración SGML

La declaración SGML dice al usuario qué puede y qué no puede estar contenido en el documento SGML, qué caracteres serán usados, qué características específicas de SGML serán implementadas, y qué sintaxis se utilizará en el documento. La declaración SGML y el DTD trabajan juntos. Si la declaración SGML dice que una función en particular no podrá ser usada, el DTD debe respetarlo, de otra manera existirán errores en el documento. La declaración SGML puede ser parte de la Muestra de Documento.. La declaración SGML es necesaria para cada documento SGML que es transmitido y también puede utilizarse desde cualquier ubicación considerada no local. A través de la declaración SGML se identifican de manera inmediata los parámetros para la marca generalizada contenida en el DTD.

#### 2.1.1.7- Muestra de Documento

La Muestra de Documento contiene el marcado o etiquetado y el contenido del texto. El contenido del documento tiene objetos SGML que siguen la estructura del árbol definido en DTD. Los objetos SGML son principalmente dos: elementos, insertados como etiquetas y sus atributos locales; y entidades, para caracteres especiales, texto almacenado y archivos externos.

El siguiente ejemplo es una pequeña muestra de cómo se hace el marcado de texto del ejemplo que hemos manejado:

```
<titulo>CONCEPTOS GENERALES DE SGML</titulo>
```

```
<autor> CLARA LOPEZ </autor>
```

```
<contenido>
```

```
<capitulo>
```

```
<subtitulo><DEFINICION DE SGML</subtitulo>
```

```
<parrafo> SGML es un acrónimo de Standard Generalized Markup Lenguaje o  
Lenguaje de Señalización General Normalizado. </parrafo>
```

```
</capitulo>
```

```
</contenido>
```

Intuitivamente se puede distinguir las etiquetas del contenido. Las etiquetas están delimitadas por los caracteres < > para abrir y por </ > para cerrar. Claramente observamos que sólo se divide un documento en partes y no se dice cómo cada una de esas partes deberá de diseñarse. Para este efecto debe hacerse un filtro, u hoja de estilo, un programa que traducirá las etiquetas SGML en cualquiera de las aplicaciones con las cuales es compatible.

Para el caso de html existen diferentes aplicaciones que hacen la traducción, para algunos casos podremos utilizarlos, pero si tenemos requerimientos específicos tendremos que desarrollar nuestros propios filtros.

#### **2.1.1.8- Metadatos**

Metadato es un término utilizado para describir datos que dan el tipo y clase de la información, es decir, son datos acerca de datos, que proveen la información necesaria para que los datos puedan ser empleados ágilmente en diferentes aplicaciones. Los metadatos pueden ser almacenados dentro de una base de datos con una referencia al documento completo o ser incluidos en un encabezado dentro del propio texto. En el

contexto del Web, los metadatos como índices y URL's son formados y almacenados principalmente para la implementación de motores de búsqueda.

## **2.1.2- HTML**

### **2.1.2.1- Introducción a HTML**

El HTML no es más que una aplicación del SGML (Standard Generalized Markup Language), un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos. El término HTML se suele referir a ambas cosas, tanto al tipo de documento como al lenguaje de marcas.

A medida que nos afianzamos en el manejo de Internet cada uno de nosotros pasa por tres etapas diferentes: Al principio solamente conocemos unas pocas páginas, luego nos damos cuenta que existen buscadores lo cual lo hace más interesante y por último nos damos cuenta que en Internet no solamente se puede ver la información sino que también se puede publicar. ¿y qué otra manera más fácil y más sencilla? Si Internet tiene acceso a todos los rincones del mundo.

Para que varias personas se comuniquen es necesario que éstas hablen un mismo idioma. El lenguaje que utilizan las computadoras que están conectadas a Internet es HTML.

### **2.1.2.2- Que es el HTML**

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de

sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html

Estos documentos pueden ser mostrados por los visores o "browsers" de paginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

### **2.1.2.3- Orígenes del HTML**

1986. Publicación de la ISO 8879 que presenta el Standard General Markup Language, origen del HTML.

1989. Tim Berners-Lee, a la sazón en el Centro Europeo de Investigaciones Nucleares presenta su artículo Information Management: A Proposal dedicándose de lleno al desarrollo de un sistema que permitiera el acceso en línea de manera uniforme a la información disponible en muchos recursos distintos, y que pudiese funcionar en máquinas que conectadas por redes basadas en TCP/IP.

1990-1991. Tim Berners-Lee define el HTML como un subconjunto de SGML (Standard Generalized Markup Language), que más tarde se llamará nivel 0; soporta encabezados, listas y anclas. Se crea el nombre World Wide Web.

1991. Tim Berners-Lee introduce el primer visor de HTML, LineMode, que trabaja en modo texto y sólo en plataformas UNIX. El Centro Europeo de Investigaciones Nucleares realiza la apertura del primer sitio con acceso público de World Wide Web (<http://info.cern.ch>).

1992. Dan Connolly produce la primera Definición de Tipo de Documento (DTD) para el lenguaje, llamada HTML 1.0, agregando a la definición original atributos para modificar el estilo físico del texto. Se distribuye Viola, primer visor gráfico de Web y disponible sólo para X.11.

1993. Un nuevo visor que soporta un mayor nivel, Lynx, es producido por la Universidad de Kansas, si bien lee sólo texto. Aparece Mosaic, desarrollado por el Centro Nacional para Aplicaciones de Supercomputadoras, es el primer visor de Web en entorno gráfico que se hace disponible para computadoras personales, lo que lo hace inmediatamente popular. A fines de año, comienzan a aparecer los primeros artículos sobre WWW en diarios y revistas de circulación masiva. Tim Berners-Lee utiliza el trabajo del año anterior de Connolly para presentar el borrador de la primera norma (RFC -Recommendation for Comments) de HTML para Internet.

1994. La Universidad Técnica de Graz desarrolla un servidor y clientes con mayores prestaciones para HTML, Hyper-G, que no tiene gran éxito. Cello, primer visor de HTML que no requiere TCP/IP presentado por la Escuela de Leyes de la Universidad de Cornell. Dan Connolly y Karen Olson Muldrow redefinen el HTML para el nivel 2.0, que ahora soporta formularios. Un grupo de programadores que desarrollaran el Mosaic producen un nuevo visor de World Wide Web, Netscape (también conocido como Mozilla), que tiene una amplia aceptación entre los usuarios, pero que soporta elementos de programación que equivalen a una degeneración del HTML (tamaños de letra, fondos). Se define un equivalente para los modelos en tres dimensiones del HTML, el VRML (Virtual Reality Modeling Language), que permite moverse dentro de los ambientes definidos. En este mismo año se realizan la Primera y Segunda



conferencias internacionales de WWW, en Ginebra y Chicago, respectivamente. Se crea la W3 Organization.

1995. Dave S. Raggett (Hewlett-Packard, Inglaterra) comienza a compilar la normativa del nuevo nivel del lenguaje, el HTML 3.0, cuya principal novedad es el soporte de tablas. Microsoft produce su primer visor de Internet, el cual también utiliza elementos de HTML degenerados. Una nueva versión de Netscape, Navigator 2.0, agrega soporte de encuadres. Sun Microsystems produce el primer visor de World Wide Web con soporte de un lenguaje de programación, HotJava. Se celebran la Tercera y Cuarta conferencias internacionales de WWW, en Boston y Darmstadt respectivamente, y la conferencia de WWW para Asia y el Pacífico en Wagga-Wagga.

1996. Netscape Communications y Microsoft presentan las nuevas versiones de sus visores que soportan gran parte del nivel de HTML 3.0. Aparecen visores no comerciales que implementan la norma completa de HTML 3.0. Se formaliza un nuevo nivel para la modelación en tres dimensiones, VRML 3.0, que permite interactuar con los objetos definidos. Se celebra la Quinta conferencia internacional de WWW en Rocquencourt.

1997. D. Raggett presenta, en enero, la versión normalizada del 3.2. En julio, aparece la versión 4.0, experimental.

1998. HTML 4.0.

#### **2.1.2.4- Creación de Páginas Web con Lenguaje HTML**

Para crear una página web se pueden utilizar varios programas especializados en esto, como por ejemplo, el Microsoft Front Page o el Macromedia Dreamweaver 3. Otra forma de diseñar un archivo .html, es copiar todo en el Bloc de Notas del Windows, ya que este sencillo programa cumple con un requisito mínimo que es la posibilidad de trabajar con las etiquetas con las que trabaja este lenguaje.

### 2.1.2.5- Estructura de los Documentos de HTML

Si se tiene en cuenta el contenido del documento, todos los documentos de HTML bien escritos comparten una estructura en común. Un documento de HTML empieza con la etiqueta <HTML>, que es la que encerrará el documento actual. Contiene dos secciones primordiales: la cabecera y el cuerpo encerradas respectivamente por los elementos <HEAD> cabeza y <BODY> cuerpo.

La cabecera puede contener información y siempre contiene el título del documento encerrado por el elemento <TITLE>.

En el cuerpo se encuentra todo el contenido del documento, ya sea, texto, imágenes, sonidos, hipervínculos, etc.

Un documento escrito en HTML contiene las siguientes etiquetas en el siguiente orden:

Ejemplo:

```
<HTML>
<HEAD>
<TITLE> Título de mi página de Internet </TITLE>
</HEAD>
<BODY>
<H1> <CENTER> Primera pagina </CENTER> </H1>
<HR>
```

Esta es mi primera pagina, aunque todavía es muy sencilla. Como el lenguaje HTML no es difícil, pronto estaremos en condiciones de hacer cosas mas interesantes.

```
<P> Aquí va un segundo párrafo.
```

```
</BODY>
```

</HTML>

Para escribir títulos se usa la etiqueta <Hx></Hx> en donde x es un número.

Ejemplo:

<h1>Titulo principal</h1>

<h2>Titulo secundario</h2>

<h3>Titulo terciario</h3>

<h4>Titulo cuarto nivel</h4>

<h5>Titulo quinto</h5>

<h6>Titulo sexto</h6>

Quedaría mas o menos así:

Titulo principal

Titulo secundario

Titulo terciario

Titulo cuarto nivel

Titulo quinto

Titulo sexto

Es por ello que si deseamos que cualquier visor de páginas Web pueda visualizar las letras acentuadas de nuestro documento debemos utilizar sus correspondientes códigos para representarlas.

Recuerden que las etiquetas se pueden escribir en mayúsculas o en minúsculas, es lo mismo poner <HTML>, <html> o <HtmL>.

Una vez que terminamos de escribir todo nuestro documento, lo guardamos poniéndole un nombre, y necesariamente lo debemos guardar con la extensión .HTM o .HTML.

Para ir visualizando lo que hemos estado haciendo tendríamos que abrir nuestro browser (Netscape o Internet Explorer), hacemos click en la opción FILE o ARCHIVO, estando allí seleccionamos OPEN o ABRIR, le damos BROWSE o EXAMINAR,

seleccionamos el archivo y le damos OPEN. Luego le damos ACEPTAR, e inmediatamente se abrirá nuestra pagina.

Sin HTML no existiría el Internet que hoy conocemos, si no se hubiera inventado, seguiríamos con el aburrido FTP (Protocolo de Transferencia de Ficheros) mandándonos o bajando archivos que a veces ni sabíamos lo que eran, sin tener la posibilidad de visualizarlo antes. Ahora por medio de una página llena de colores y botones por la cual se puede navegar, se puede tener acceso a bastante información y archivos.

El lenguaje de HTML abrió una puerta al mundo permitiéndole a las personas expresar sus ideas por medio de páginas y mostrárselas a todas las personas de todos los países.

Con el HTML se logró un gran movimiento económico ya que muchísimas empresas publican, venden, y ofrecen sus productos, sus servicios y sus ofertas atrayendo a mayor cantidad de personas. También gracias al HTML nacieron muchas empresas que ofrecen diversos servicios como Yahoo, Altavista, HotMail, Terra, Yupi, Mercadolibre, De Remate, etc.

Infinitas son las posibilidades que te brindan las páginas WEB ya que no solo te dan la posibilidad de pasar el tiempo navegando, sino que también hasta se puede comprar un auto por Internet, solo basta con llenar un formulario con los datos personales y el número de tarjeta de crédito y en ocho días lo tenemos en nuestra casa.

Todo el universo de Internet se lo debemos al HTML, ya que todas las páginas con las que se compone la World Wide Web están hechas con el lenguaje de programación HTML.

#### **2.1.2.6- SGML - HTML**

Un lenguaje de marca o etiquetado no es lo mismo que un lenguaje de marca generalizado. Con un lenguaje de marca se describen las reglas para el procesamiento de un texto, para describir los diferentes caracteres y sus características de impresión. La marca generalizada, por otro lado, no especifica cómo deben verse las cosas, en su lugar, provee información del sistema en el que se está trabajando, con información sobre la estructura del documento únicamente.

HTML es un DTD de SGML utilizado para documentos en el Web. El HTML original no era completamente compatible con SGML, tenía algunas deficiencias sintácticas a cambio de simplicidad. Ahora, las ventajas en la compatibilidad, como la habilidad de validación, la utilización de herramientas basadas en SGML y la capacidad de interactuar, han sobrepasado a las desventajas. HTML2 es el nivel común en la mayoría de los navegadores. La siguiente versión, HTML3, será completamente compatible con ISO 8879, con ampliaciones para soportar notaciones matemáticas y tablas, lo cual hará a SGML la columna vertebral del sistema de hipertexto distribuido más grande del mundo.

Es importante entender que la compatibilidad con el Web reside en la estandarización de los datos, no en las aplicaciones. En la medida en que el lenguaje de marca utilizado en el Web permanezca abierto, no propietario y uniforme, cualquiera puede hacer aplicaciones acordes a cualquier plataforma y los datos podrán ser intercambiados y visualizados sin conversiones

## **2.2.- INTRODUCCION A XML(Extensible Markup Language)**

Cada documento XML tiene una estructura lógica y otra física. Físicamente el documento está compuesto por entidades, las cuales se agrupan de forma jerárquica, existiendo una sola entidad raíz. Lógicamente el documento se compone de declaraciones, elementos, comentarios, referencias a carácter e instrucciones de proceso

XML (eXtended Markup Language) es un lenguaje para la definición de lenguajes de etiquetas, lo que llamaríamos "metalenguaje". XML va a indicar como pueden ser creados otros lenguajes por marcas, define las gramáticas de otros lenguajes.

Este lenguaje es relativamente joven, eso teniendo en cuenta que la versión 1.0 es una recomendación del World Wide Web Consortium (W3C). Si bien las ideas no son tan nuevas, ya que XML se apoya en la definición del SGML que es un estándar ISO desde los años 80.

Como acabo de mencionar, el XML es un subconjunto del lenguaje SGML, el cual también es un metalenguaje. El problema del SGML es que es una definición demasiado extensa, engorrosa y que su uso podría generar lenguajes incompletos en su definición como sucede con el HTML.

### **2.1.1- El estándar XML**

XML publicó su recomendación "XML 1.0" en Febrero del 1998. En Enero de 1999 se publicó "NameSpaces in XML" y en Junio del mismo año la recomendación sobre el "Style Sheet Linking".

Actualmente XML es un estándar abierto del W3C que agrupa una serie de tecnologías:

- XML. Lenguaje que define la sintaxis del XML que nos ayudará a crear nuevos lenguajes de etiquetas.
- XLink. Define la forma estándar de añadir enlaces dentro de un documento XML.
- XPointer y XFragments. Que define como poder hacer referencias a partes dentro del documento XML. Es como las URL, pero haciendo referencia a partes dentro del documento XML.
- XSL (eXtensible StyleSheet Language). Define el estándar para las hojas de estilo de XML. Es la ampliación y modificación de las CSS. XSL está basado en XSLT.

- XSLT (XSL Transformations). Es un lenguaje de transformación que se usa para ordenar, añadir y eliminar etiquetas y atributos.
- XML Schemas. Ayuda a los desarrolladores a definir estructuras precisas basadas en XML

Otras tecnologías que van ligadas tanto al XML, como al HTML son:

- CSS (Cascading Style Sheets). Es la especificación sobre las Hojas de Estilo.
- DOM (Document Object Model). Es un conjunto estandar de funciones que nos van a permitir modificar los documentos XML y HTML.

## **2.3.- ENTORNO DE TRABAJO DE XML**

XML es un lenguaje de metamarcado que ofrece un formato para la descripción de datos estructurados. Esto facilita declaraciones de contenido más precisas y resultados de búsquedas más significativos en varias plataformas. Además, XML habilitará una nueva generación de aplicaciones para ver y manipular datos basadas en el Web.

### **2.1.3- Representación estructural de los Datos**

XML ofrece una representación estructural de los datos que se puede implementar ampliamente y es fácil de distribuir. XML es un subconjunto de SGML optimizado para el Web. Definido por el World Wide Web Consortium (W3C) (en inglés), XML garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes. La interoperabilidad resultante está creando rápidamente una nueva generación de aplicaciones de comercio electrónico en la Web.

XML, que proporciona un estándar de datos que puede codificar el contenido, la semántica y los esquemas de una gran variedad de casos, desde los más simples a los más complejos, sirve para marcar lo siguiente:

- Un documento normal.
- Un registro estructurado, como un registro de citas o un pedido de compra.
- Un objeto con datos y métodos, como el formulario permanente de un objeto Java o de un control ActiveX.
- Un registro de datos, como el conjunto de resultados de una consulta.
- Metacontenido sobre un sitio Web, como el formato de definición de canal (CDF).
- Representaciones gráficas, como la interfaz de usuario de una aplicación.
- Entidades y tipos de esquema estándar.

Todos los vínculos entre datos y personas que hay en el Web.

Cuando los datos llegan al escritorio del cliente, se pueden manipular, editar y presentar en varias vistas, sin tener que regresar al servidor. Ahora los servidores pueden ser más escalables, gracias a la reducción de las cargas de ancho de banda y computación. Además, dado que los datos se intercambian en el formato XML, se pueden combinar fácilmente desde distintas fuentes.

XML es muy valioso para Internet, así como para los entornos de intranets corporativas de gran tamaño, pues proporciona interoperabilidad mediante un formato basado en estándares flexible y abierto, con formas nuevas de acceso a las bases de datos existentes y de entregar datos a clientes de Web. Las aplicaciones se pueden generar más rápidamente, su mantenimiento es más sencillo y pueden ofrecer fácilmente varias vistas de los datos estructurados.



En un principio, no rivalizarán HTML y XML, estos se complementarán el uno al otro, uniéndose ambas gramáticas. Este Lenguaje de marcas extensible (XML) es una versión abreviada del SGML (Standard Generalized Markup Language).

Algunos de los objetivos planteados por el Grupo de Trabajo XML y el W3C son:

- XML debe ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El número de características opcionales en XML debe ser absolutamente mínimo, idealmente cero.
- Los documentos XML deben ser legibles por los usuarios de este lenguaje y razonablemente claros.
- El diseño de XML debe ser formal, conciso y preparado rápidamente.
- Los documentos XML deben ser fácilmente creables.
- La brevedad en las marcas XML es de mínima importancia.

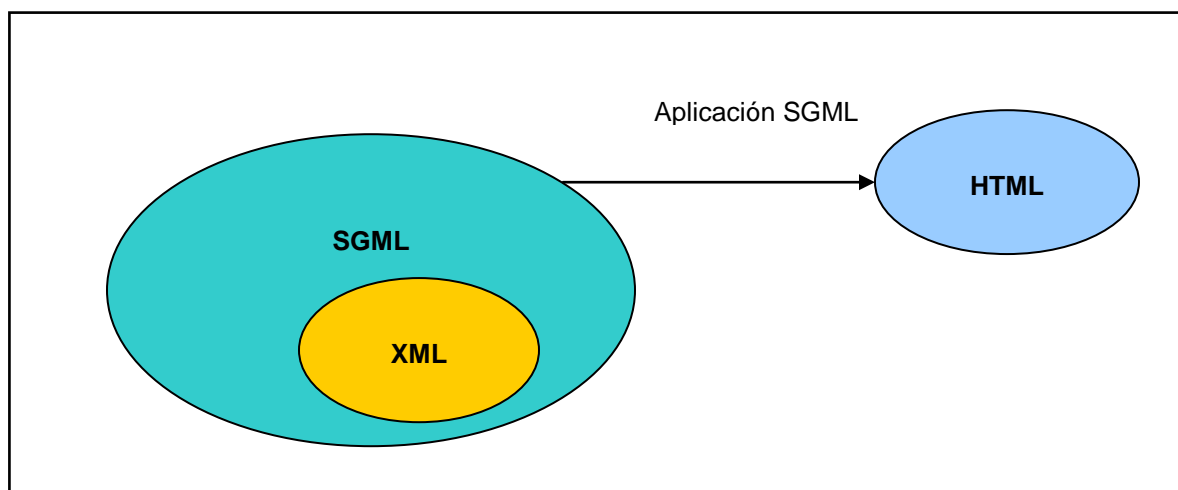
A estos fines se le une unos estándares como el Unicode e ISO/IEC 10646 para caracteres, el Internet RCF 1766 para identificación de lenguajes, ISO 639 para códigos de nombres de lenguajes y también el ISO 3166 para códigos de nombres de países, para la normal comprensión de esta versión de XML.

Tampoco tenemos que equivocarnos y pensar que el XML es un HTML++. Tanto el XML como el HTML tienen su base en el SGML. Es decir, es un metalenguaje que nos permite definir lenguajes para definir la estructura y el contenido de nuestros documentos. La definición de la estructura y el contenido de un tipo de documento se realiza en una DTD. En ella definimos los elementos que conformarán ese tipo de documentos y como tienen que estar organizados para que sea correcto.

Un ejemplo de DTD es por ejemplo la que define cómo tendrán que ser los documentos HTML. Por tanto, el HTML no es más que un tipo de documento SGML

que se utiliza en la Web, y esto es importante, ya que aquí radica su principal diferencia con el XML.

El XML no es ningún tipo de documento SGML, sino que es una versión abreviada de SGML optimizada para su utilización en Internet. Esto significa que con él vamos a poder definir nuestros propios tipos de documentos (podremos definir nuestras propias etiquetas) y, por tanto, ya no dependeremos de un único e inflexible tipo de documento HTML.



#### 2.1.4- Diferencia entre SGML, XML y HTML

El XML más que un HTML hay que considerarlo como un SGML optimizado para su utilización en Internet. Como escribió Richard Ligth en su libro Presentig XML, "XML ofrece el 80% de las ventajas del SGML con un 20% de su complejidad". Y es que los diseñadores de XML intentaron dejar fuera sólo aquellas partes que raramente se utilizan. Esta reducción resultó ser muy importante: la especificación XML ocupa aproximadamente 30 páginas, frente a las 500 del SGML.

#### 2.1.5- Estructura de XML

Un documento XML tiene dos estructuras, una lógica y otra física. Físicamente, el documento está compuesto por unidades llamadas entidades. Una entidad puede hacer referencia a otra entidad, causando que esta se incluya en el documento. Cada documento comienza con una entidad documento, también llamada raíz. Lógicamente, el documento está compuesto de declaraciones, elementos, comentarios, referencias a caracteres e instrucciones de procesamiento, todos los cuales están indicados por una marca explícita. Las estructuras lógica y física deben encajar de manera adecuada:

Los documentos XML se dividen en dos grupos, documentos bien formados y documentos válidos.

Un objeto textual o documento XML se dice que está bien formado si, considerándolo como conjunto, encaja con las especificaciones XML de producción, lo que implica:

- Contiene uno o más elementos.
- Hay exactamente un elemento, llamado raíz o elemento documento, de forma que ninguna parte del mismo aparece en el contenido de ningún otro elemento. Para todos los demás elementos, si la etiqueta inicial está en el contenido de otro elemento, la etiqueta final forma parte del contenido del mismo elemento. Dicho de forma más clara, los elementos, delimitados por una etiqueta de inicio y otra de final, se encuentran anidados correctamente.
- Cumple todas las restricciones que proporciona su especificación a través del DTD. Si no se utiliza DTD, el documento debe comenzar con una Declaración de Documento único (SDD, Standalone Document Declaration) que indique:

```
<?XML version="1.0" standalone="yes"?>
```

- Cada una de sus partes procesadas está bien formada.
- Todas las etiquetas deben estar balanceadas: esto es, todos los elementos que contengan datos de tipo carácter deben tener etiquetas de principio y fin (no está permitida la omisión excepto para los elementos vacíos.);

- Todos los valores de los atributos deben ir entrecomillados (el carácter comilla simple [el apóstrofe] puede utilizarse si el valor contiene caracteres comillas dobles, y viceversa): si necesitas ambos, utiliza &apos; y &quot;;
- Cualquier elemento VACÍO (p.e. aquellos que no tienen etiqueta final como <IMG>, <HR>, y <BR> y otros de HTML) deben terminar con '>' o debes hacerlos no VACÍOS añadiéndoles una etiqueta de fin;
- Ejemplo: <BR> se convertirá en <BR/> o en <BR></BR>.
- No debe haber etiquetas aisladas (< ó &) en el texto (p.e. debe darse como &lt; y &amp;), y la secuencia ]]> debe darse como ]]&gt;; si no ocurre esto como final de una sección marcada como CDATA;
- Los elementos deben anidar dentro de sí sus propiedades (no se deben sobreponer etiquetas, como en el resto de SGML);
- Los Archivos bien-formados sin-DTD pueden utilizar atributos en sus elementos, pero éstos deben ser todos del tipo CDATA, por defecto.

De un documento bien formado, se dice además que es válido, si tiene DTD como el resto de aplicaciones SGML. Un archivo XML válido comienza como cualquier otro archivo SGML, con una Declaración de Tipo de Documento:

```
<?xml version="1.0"?>
<!DOCTYPE anuncio SYSTEM "http://www.foo.org/ad.dtd">
<anuncio>
<titulo>...<foto/>...</titulo>
<texto>...</texto>
</anuncio>
```

Dado que XML está diseñado para ser un subconjunto de SGML, cualquier documento XML válido debe ser también un documento SGML válido.

### 2.1.5.1- Estructuras Lógicas

Cada documento XML contiene uno o más elementos, cuyos límites están delimitados por etiquetas de comienzo y de final o, en el caso de elementos vacíos, por una etiqueta de elemento vacío.

Cada elemento tiene un tipo, identificado por un nombre, denominado identificador genérico, y puede tener un conjunto de especificaciones de atributos.

Cada especificación de atributo tiene un nombre y un valor. Estas especificaciones no restringen la semántica, el uso o (más allá de la sintaxis) los nombres de los tipos de los elementos y los atributos, excepto de los nombres que comienzan por XML, que se reservan para estandarizar etiquetas o atributos en versiones posteriores del estándar.

### **Parte Lógica**

```
<! ELEMENT persona (nombre.apellido+.foto) >
```

```
<! ELEMENT nombre (#PCDATA) >
```

```
<! ELEMENT apellido (#PCDATA) >
```

```
<! ELEMENT foto (EMPTY) >
```

```
<! ATTLIST persona edad CDATA #IMPLIED>
```

```
<! ATTLIST foto origen CDATA #REQUIRED>
```

### **2.1.5.2- Estructuras Físicas**

Un documento XML puede consistir en una o más unidades de almacenamiento virtual, llamadas entidades. Todas estas unidades tienen contenido y todas ellas (excepto la entidad documento y el subconjunto externo del DTD) están identificadas por un nombre. Cada documento XML contiene una entidad, llamada entidad documento, que sirve como punto de comienzo para el procesador XML y puede contener el documento completo.

### **Parte Física**

```
<persona edad=24>
```

```
<nombre>Victor</nombre>
<apellido>López</apellido>
<foto origen='victor.jpg' />
</persona>
```

Como hemos comentado, las entidades pueden ser analizadas o sin analizar (también llamadas procesadas o sin procesar). El contenido de una entidad analizada se conoce también como texto de reemplazo, y es parte integrante del documento. Las entidades no analizadas son recursos (tales como enlaces) cuyo contenido puede o no ser texto, o en caso de que sea texto que no sea XML.

Cada entidad no asociada tiene una notación asociada, identificada por un nombre. Aparte de obligar al procesador XML que haga accesible a la aplicación el nombre de esta notación y sus identificadores asociados, XML no proporciona ninguna otra restricción sobre el contenido de estas entidades. La forma de invocar ambos tipos de entidades es a través de su nombre, en el caso de las analizadas a través de su referencia a entidad y en el de las no analizadas a través de sus atributos de entidad.

Las entidades generales, son entidades analizadas que se usan en el interior del documento. Las entidades parametrizadas son entidades analizadas que se usan en el ámbito del DTD. Estos dos tipos de entidades usan distintos tipos de referencias y se reconocen en contextos distintos.

XML consta de cuatro especificaciones (el propio XML sienta las bases sintácticas y el alcance de su implementación):

- DTD (Document Type Definition): Definición del tipo de documento. Es, en general, un archivo(s) que encierra una definición formal de un tipo de documento y, a la vez, especifica la estructura lógica de cada documento. Define tanto los elementos de una página como sus atributos. El DTD del XML es opcional. En tareas sencillas no es necesario construir una DTD, entonces se

trataría de un documento "bien formado" (well-formed) y si lleva DTD será un documento "validado" (valid).

- XSL (eXtensible Stylesheet Language): Define o implementa el lenguaje de estilo de los documentos escritos para XML. Desde el verano de 1997 varias empresas informáticas como Arbortext, Microsoft e Inso vienen trabajando en una propuesta de XSL (antes llamado "xml-style") que presentaron a W3C. Permite modificar el aspecto de un documento. Se puede lograr múltiple columnas, texto girado, orden de visualización de los datos de una tabla, múltiples tipos de letra con amplia variedad en los tamaños. Este estándar está basado en el lenguaje de semántica y especificación de estilo de documento (DSSSL, Document Style Semantics and Specification Language, ISO/IEC 10179) y, por otro lado, se considera más potente que las hojas de estilo en cascada (CSS, Cascading Style Sheets), usado en un principio con el lenguaje DHTML. "Se espera que el CSS sea usado para visualizar simples estructuras de documentos XML (actualmente se ha conseguido mayor integración en XML con el protocolo CSS2 (Cascading Style Sheets, level 2) ofreciendo nuevas formas de composición y una más rápida visualización) y, por otra parte, XSL pueda ser utilizado donde se requiera más potencia de diseño como documentos XML que encierran datos estructurados (tablas, organigramas, etc.)
- XLL (eXtensible Linking Language): Define el modo de enlace entre diferentes enlaces. Se considera que es un subconjunto de HyTime (Hipermedia/Timed-based structuring Language o Lenguaje de estructuración Hipermedia/basado en el tiempo, ISO 10744) y sigue algunas especificaciones del TEI (Text Encoding Initiative o Iniciativa de codificación de texto). Desde marzo de 1998 el W3C trabajo en los enlaces y direccionamientos del XML. Provisionalmente se le renombró como Xlink y a partir de junio se le nombra como XLL. Este lenguaje de enlaces extensible tiene dos importantes componentes: Xlink y el Xpointer. Va más allá de los enlaces simples que sólo soporta el HTML. Se podrá

implementar con enlaces extendidos. Los siguientes mecanismos hipertextuales que soportará esta especificación:

- Denominación independiente de la ubicación.
- Enlaces que pueden ser también bidireccionales.
- Enlaces que pueden especificarse y gestionarse desde fuera del documento a los que se apliquen (Esto permitirá crear en un entorno intranet/extranet un banco de datos de enlaces en los que se puede gestionar y actualizar automáticamente. No habrá más errores del tipo "404 Not Found").
- Hiperenlaces múltiples (anillos, múltiples ventanas, etc.).
- Enlaces agrupados (múltiples orígenes).
- Transclusión (el documento destino al que apunta el enlace aparece como parte integrante del documento origen del enlace).
- Se pueden aplicar atributos a los enlaces (tipos de enlaces).
- XUA (XML User Agent): Estandarización de navegadores XML. Todavía está en proceso de creación de borradores de trabajo. Se aplicará a los navegadores para que compartan todas las especificaciones XML.

### **2.1.6- Aplicaciones de XML**

Algunas de las aplicaciones de XML son:

- Ofrecer mecanismos más versátiles para mostrar datos. Actualmente, bajo el nombre de DOM (Document Object Model) se está desarrollando una API que sea soportada por todos los procesadores de XML y HTML. La idea detrás de esta API es que podamos representar (a través de javascripts o JavaApplets) documentos XML en los navegadores Web, pero de una forma más sofisticada que los documentos HTML, ya que XML no solo proporciona una sintaxis, sino también una semántica.



- Buscadores inteligentes. Debido a que la información en los documentos XML está etiquetada por su significado de forma precisa, se puede localizarla de forma mucho más clara que en documentos HTML. Con DTDs estandarizados para distintas aplicaciones (librerías, tiendas de deporte, catálogos de componentes,...) podríamos programar buscadores Web que recuperasen información sobre un producto de cualquier website en el mundo sabiendo que todos tendrán el mismo formato de datos (gracias al DTD), aunque no tengan necesariamente la misma representación gráfica (gracias al XML/XSL).
- Intercambio de información entre sistemas heterogéneos. El fundamento es el mismo que par los buscadores inteligentes. Debido a que el DTD proporciona un formato estándar para representar la información de un tema específico, puede usarse para simplificar el intercambio de información entre distintas fuentes (actualmente existen ya dos DTD estandarizados uno para fabricantes de chips y otro para industrias químicas, llamado CML).

## **2.4.- VENTAJAS DE XML**

- XML es un lenguaje independiente de la plataforma sobre la que se trabaje.
- Es UNICODE, lo que hace que pueda ser utilizado en múltiples lenguajes.
- Independencia de la información con respecto a la representación.
- Cuando se incluye un cambio en el documento no supone un problema para su interpretación, ya que siempre hay que leer el DTD.
- XML sigue un estándar.
- Los datos solo dependen de los datos en si, no de su formato.

- Para acceder a la información se puede utilizar una representación a alto nivel y no a bajo nivel como con el HTML.
- Los autores y proveedores pueden diseñar sus propios tipos de documentos usando XML, en vez de limitarse a HTML. Los tipos de documentos pueden ser explícitamente 'hechos a la medida de una audiencia', por lo que las difíciles manipulaciones que debes hacer con HTML para conseguir efectos especiales serán cosa del pasado: autores y diseñadores serán libres de inventar sus propias etiquetas;
- La información contenida puede ser más 'rica' y fácil de usar, porque las habilidades hipertextuales de XML son mayores que las de HTML.
- XML puede dar más y mejores facilidades para la representación en los visualizadores.
- Elimina muchas de las complejidades de SGML, en favor de la flexibilidad del modelo, con lo que la escritura de programas para manejar XML será más sencilla que haciendo el mismo trabajo en SGML.
- La información será más accesible y reutilizable, porque la flexibilidad de las etiquetas de XML pueden utilizarse sin tener que amoldarse a reglas específicas de un fabricante, como es el caso de HTML.
- Los archivos XML válidos son válidos también en SGML, luego pueden utilizarse también fuera de la Web, en un entorno SGML (una vez la especificación sea estable y el software SGML la adopte).

Se especificara algunas de las implementaciones/aplicaciones que se han desarrollado con XML. Suponemos una implementación para Bibliotecas y Centros de Documentación; y con ello se pueda extraer la información desde la misma red (no por métodos de infranet).

CDF (Channel Definition Format): Los canales creado por Microsoft en el explorador IE4 con tecnología push.

RDF (Resource Description Framework): Esquema de descripción de recursos. Una de las aplicaciones más importantes que permitirá describir los datos de cada documento y definir las relaciones que hay entre los datos XML. Tratará de los metadatos (metadata). Se les podría considerar como "los META del XML". Muchas compañías en Internet se están adhiriendo a esta aplicación. RDF Posee las siguientes virtudes:

- Mejores motores de búsqueda. Se han adherido a esta especificación Yahoo!, Altavista, Excite, Lycos, WebCrawler, Amazon, etc.
- La capacidad de describir los contenidos y sus relaciones en una biblioteca digital o sede Web. Permitirá el acceso a una parte concreta del documento y se facilitará el intercambio de los datos.
- Se pueden calificar los contenidos para establecer la protección infantil y de la propia intimidad, desarrollado a través de las marcas (tags) de PICS (Platform for Internet Content Selection).
- Establece los derechos de propiedad intelectual en las propias páginas Web.

OSD (Open Software Description Format): Formato abierto de descripción de software. Desarrollo de software en múltiples plataformas. Describe el reparto de software a través de la Red. Las etiquetas XML con las que está descrito definen los componentes, la versión que es, la plataforma en la que ha sido creado, la relación con otros componentes, etc. Esto hará que se simplifique el proceso de instalación para el usuario y permitir también un fácil uso de las actualizaciones.

CML (Chemical Markup Language): Lenguaje de marcas para química. Describe, entre otras formulas, las estructuras moleculares y cristalinas, los análisis de espectros y otros objetos de interés para los químicos.

MathML (Mathematical Markup Language): Lenguaje de marcas para matemáticas. Apto para codificar signos matemáticos, símbolos científicos, etc. El MathML es un lenguaje de bajo nivel que tiene en cuenta la comunicación maquina a maquina de datos estructurados como información de bases de datos. El lenguaje MathML utiliza dos series de códigos progresivos: el primero presenta los signos matemáticos en series crecientes, y el segundo transmite el significado semántico de las expresiones matemáticas, lo que posibilita la codificación de símbolos y signos tanto matemáticos como científicos.

EDI (Electronic Document Interchange): Intercambio electrónico de datos. Lleva un tiempo en proceso de estudio, pero finalmente dará un espaldarazo con el estándar XML. InfoVía Plus lo integra en uno de sus servicios (InfoEDI).

# **CAPITULO 3**

## **III.- CONFERENCIA WEB : CHAT**

### **3.1.- CONCEPTO DE LAS CONFERENCIAS WEB**

Conferencias Web o IRC (Internet Relay Chat) es un sistema de comunicación que permite conversar con múltiples usuarios de forma simultánea, en tiempo real.

Cuando comenzó a usar Internet como medio para comunicarse el sistema comenzó a popularizarse rápidamente y pasó a convertirse en una herramienta de comunicación casi indispensable para todos aquellos que necesitaban comunicarse de una manera más directa que con el correo electrónico

#### **3.1.1- Introducción a IRC**

El IRC, acrónimo de Internet Relay Chat, es un sistema de conversación en tiempo real para usuarios de Internet. Para poder participar solo es necesario disponer de un programa cliente de IRC y una conexión a Internet.

El IRC permite que múltiples usuarios se reúnan simultáneamente en debates, en los cuales cada uno va expresando sus opiniones de forma escrita y en tiempo real. Esta es la diferencia fundamental entre IRC y las USENET (las llamadas news o foros de debate) - estas últimas se asemejan más a una publicación en la que cada usuario lee lo que han escrito los demás, pudiendo insertar su artículo en espera de que sea leído mas tarde, cuando le llegue al resto de subscriptores.

Para participar en IRC solo es necesario conectarse a Internet en la forma habitual, ejecutar el programa cliente de IRC, y acceder a alguno de los muchos servidores de repartidos por todo el mundo. Gran cantidad de estos servidores se encuentran conectados entre si, formando lo que son conocidas como redes de IRC, que permiten que usuarios conectados a un servidor europeo hablen con usuarios conectados a un servidor australiano, por ejemplo.

IRC Manga Tiene como meta formar uno de los servidores IRC Mas Grandes sobre Anime, Videojuegos, Comics y Entretenimiento en Español IRC Manga es un proyecto en expansión que Tiene cada día un mayor número de usuarios repartidos en numerosos canales, muchos de ellos registrados y mantenidos por el servicio oficial de canales esta red.

Las Charlas de IRC se reúnen en canales (conocidos en otros entornos como grupos, rooms, etc) donde coinciden varios usuarios conversando simultáneamente sobre los más diversos temas. Existen canales muy específicos en los que se intenta debatir sobre un tema concreto, canales creados para agrupar simplemente a usuarios de un mismo idioma o cultura y charlando sobre cualquier tema, o simples reuniones de amigos. Cualquier usuario puede abrir un canal a su gusto, llamarlo como desee si en

esa red no existe ya otro con ese mismo nombre, y simplemente esperar a que lleguen otras personas.

El interés para crear un canal puede ser variado, incluso es posible hacer que el canal sea secreto y no aparezca en la lista de canales disponibles, siendo solo accesible a aquellas personas que ya conocen previamente de su existencia; también se puede hacer que un canal sea solo para invitados; se limite a un máximo de participantes; esté moderado y solo tengan la palabra las personas autorizadas a ello,... y muchas cosas más.

Además, siempre que estemos hablando en un canal podemos vernos interesados en hablar con alguno de los usuarios que están en él de forma privada. El IRC nos permite hablar con alguna persona de forma privada sin que otras personas sepan ni que estemos hablando ni que estamos diciendo.

El IRC es un sistema de comunicaciones multiplataforma, es decir, puede poner en contacto ordenadores que funcionen bajo las mas diversas arquitecturas y sistemas operativos (Unix, OS2, Windows...), solo es necesario que cada uno disponga del programa cliente apropiado a sus características, y conexión a un servidor. Los programas clientes más habituales y extendidos por su sencillez son los que están diseñados para el entorno del sistema operativo Windows, tanto en 16 como en 32 bits; los más potentes son los construidos para los distintos entornos Unix, si bien estos son mucho menos populares debido a la mayor profesionalidad y conocimientos que su manejo exige.

Cuando un usuario ejecuta su programa cliente de IRC, y antes de conectarse, debe de especificar algunos de los datos que lo identificarán dentro de la red de chats. El más característico es su "nick" o pseudónimo, el nombre bajo el cual será conocido en todos los canales en los que intervenga, o esté de simple "oyente". Debido a que sólo puede haber una persona por nick en un mismo momento, y debido a que en las redes de IRC con mayores características se puede "registrar" o reservar, encontraremos que los

primeros nicks que nos vengan a la cabeza pueden ya estar reservados por otras personas, sobre todo en redes internacionales donde se pueden reunir más de cincuenta mil usuarios simultáneamente.

### **3.1.2- Historia de los Chat**

Los inicios del IRC se remontan a 1988, cuando un finlandés llamado Jarkko Oikarinen escribió el código original. Fue por tanto en Finlandia donde se comenzó a usar esta tecnología, aunque en ese momento todavía no estaba en Internet, sino que J. Oikarinen la diseñó para usarla en su propia BBS6 como un sistema multichat en tiempo real.

Cuando comenzó a usar Internet, el sistema comenzó a popularizarse rápidamente y pasó a convertirse en una herramienta de comunicación casi indispensable para todos aquellos que necesitaban comunicarse de una manera más directa que con el correo electrónico.

Hay dos fechas clave que marcaron el impulso definitivo del IRC. La primera es 1991, con el estallido de La Guerra del Golfo; el uso de este sistema de comunicación que plasmaba la realidad segundo a segundo comenzó a tomarse en serio. Fue en este momento cuando comenzaron a florecer los programas de IRC.

La otra fecha es Septiembre de 1993, cuando gran número de usuarios (en tiempo real) informaban desde Moscú de la inestabilidad social y política por la que estaba pasando el país.

Actualmente, los canales de conversación del IRC abarcan todos los temas imaginables, pudiendo encontrar canales en los que se habla de los temas más simples, hasta canales en donde los temas de conversación son absolutamente serios y de gran acervo cultural.



### **3.1.2.- Evolución de Chats**

En verano de 1988, Jarkko Oikarinen trabajaba en el departamento de procesamiento de información de la Universidad de Oulu. Su función era administrar el servidor SUN del departamento, tarea que no requería excesivo tiempo, de manera que empezó a escribir un programa de comunicaciones que le permitiera hacer su BBS (OuluBoX, tolsun oulu.fi) un poco más fácil de usar por sus usuarios.

El objetivo inicial era permitir discusiones tipo Usenet además de discusiones a tiempo real. Poco tiempo atrás, Jyrki Kuoppala había creado el programa rmsg para enviar mensajes a usuarios de máquinas remotas, aunque no tenía aun el concepto de canales, de manera que era utilizado únicamente para comunicación de usuario a usuario.

De esta manera, el nacimiento del IRC fue en Agosto de 1988, y hasta la actualidad sigue optimizándose y realizando mejoras.

#### **3.1.2.1.- Mensajes Instantáneos**

La Red de Redes ha aportado al mundo de la telecomunicaciones grandes avances en la vida de la comunicación entre usuarios de Internet. Primero los correos electrónicos, después llegó el Chat y los últimos en sumarse a esta tendencia fueron los mensajeros instantáneos, una manera rápida de comunicarse con los demás.

El 15 de noviembre de 1996, cuatro meses después de su creación, la compañía Mirabilis Inc., compuesta por los israelíes Yair Goldfinger, Arik Vardi, Sefi Vigiser y Amnon Amir, presentó un software que revolucionó las comunicaciones en Internet. Medio año después, tenía un millón de usuarios. Y ya más de 127 millones de personas se registraron para usar el servicio. Estamos hablando, claro, del ICQ14

(derivado de la frase inglesa "I see you", te veo), el programa que definió el concepto de mensajería instantánea, mezcla de Chat y e-mail que da la posibilidad de comunicarse en forma inmediata con otros usuarios de la Red en cualquier parte del mundo. Permite saber cuál de sus amigos está conectado a la Web en cada momento, conectarse con él e iniciar una conversación tipo Chat, con respuesta instantánea.

Pero estos mensajeros permiten hacer más cosas que intercambiar textos. Las nuevas versiones pueden transmitir audio, video, datos, enviar mensajes a celulares, componer listas de tareas, tener páginas Web gratuitas, revisar el e-mail, etcétera. Y gratis: ninguno de los mensajeros instantáneos pide dinero para funcionar.

El último ICQ, por ejemplo, ofrece la opción de compartir archivos que estén en una carpeta designada (pero limitando el acceso al que el usuario quiera).

En mayo de 1997, AOL15 lanzó su propio mensajero. Y a mediados de 1998, la empresa compró a Mirabilis en 287 millones de dólares, poniendo en evidencia el valor del servicio. La acción fue seguida por los lanzamientos de Yahoo! en marzo de 1998 y Microsoft en julio del año siguiente.

Justamente, el MSN Messenger de la empresa de la ventanita tiene cada vez más seguidores, debido a su asociación al servicio de correo gratuito Hotmail, a la inclusión del software en el nuevo Windows XP (preinstalado con el nombre de Windows Messenger), y el peso de Microsoft como marca.

### **3.2.- CLASES DE CONFERENCIAS WEB**

El IRC está basado en una estructura de cliente-servidor. Esto quiere decir que para mantener una conversación vía IRC es necesario conectarse a un servidor que disponga de un software que canalice las conversaciones. Por otro lado se necesita un programa (cliente) para que el ordenador pueda enviar nuestras intervenciones a la conversación y recibir los datos de los demás usuarios.

### **3.2.1.- Chats: Tipología Básica**

Hay diversos tipos de chats. Los que funcionan sobre una aplicación de telnet son los más antiguos. Algunos mantienen todavía un alto grado de popularidad, pero su uso está claramente en descenso. De igual modo, los chats CGI se ven superados por la tecnología JAVA, y su uso es cada vez más marginal.

Los chats en JAVA están experimentando un crecimiento importantísimo. Requieren versiones de Netscape 2.0 o Explorer 3.0 y superiores, lo que, hoy en día no es una restricción muy considerable.

La entrada en juego de grandes campañas publicitarias y nuevas ofertas de conexión están potenciando este tipo de chats, al ser los más fácilmente accesibles desde los portales más concurridos. Por otro lado, los chats en JAVA son los que más se benefician de las constantes mejoras tecnológicas y, a la vez, los que más dependen de ellas. Un buen módem y un buen equipo, además de un poco de suerte hacen de un chat en JAVA algo casi tan rápido y efectivo como un chat en telnet o en IRC, algo imposible dos años atrás. Así pues, los chats en JAVA son los más comprometidos con una constante mejora del equipo informático de que se disponga, puesto que los portales parecen estar siempre dispuestos a aumentar el ancho de banda y las capacidades multimedia de sus servicios.

Sea como sea, el tipo de chat preferido por la mayoría de los chateros medianamente experimentados son los que se basan en Internet Relay Chat o IRC, que ya hemos mencionado previamente. En la actualidad, IRC es el medio que permite mayor control del entorno, el más rápido y el que da al usuario un mayor número de opciones (para otras consideraciones introductorias sobre IRC).

Actualmente se esta fusionando en cierta forma los chats de IRC con los JAVA, aunque manteniendo la riqueza y las mayores posibilidades que ofrecen los primeros. La

experiencia aún está en fase de asentamiento y las consecuencias de la fusión están aún por ver, aunque algunos usuarios habituales de IRC no se muestran muy ilusionados al respecto.

Teniendo en cuenta todo esto, no es extraño que los entornos donde se produce una mayor experimentación/corrupción de la lengua sea, precisamente, en los chats de IRC.

### **3.2.2- La Lengua del Chats: Tipología Básica**

Las restricciones del medio son cruciales en los chats. También, por supuesto, en IRC. No hay gestos, no hay voz, no hay miradas... Sólo hay un teclado. Sin embargo, los usuarios se han comprometido a sacarle al teclado el máximo partido. Y para ello, juegan con el medio. Aprovechan la expresividad de las letras, alargando vocales, usando mayúsculas, recortando palabras, imitando frases, etc. Aprovechan los recursos gráficos que pone a su disposición el juego de caracteres ASCII. Y, en el caso de IRC, aprovechan diversos recursos técnicos entre los que destaca, por encima de todo, el uso de la tercera persona narrativa (TPN). Veamos, en cierto detalle, cada una de estas posibilidades.

#### Jugar con las palabras

Dentro de las palabras existentes en el diccionario y confeccionables con el alfabeto corriente, los usuarios de IRC tienen su lenguaje propio.

- a. En primer lugar están las distorsiones más o menos involuntarias provocadas por la falta de atención y corrección en el tecleo y por la necesidad de intervenir con rapidez en las conversaciones. La velocidad de tecleo es una de las aptitudes más apreciadas en los chats.

- b. En segundo lugar encontramos distorsiones voluntarias que pretenden dar énfasis, tono, cadencias, volumen, etc. Para ello los usuarios recurren a alargar las vocales, a usar mayúsculas, a repetir sus intervenciones, a utilizar un número variable de símbolos de exclamación al final de una intervención, etc. También podemos contar en este epígrafe el uso de onomatopeyas, que en los chats es remarcablemente abundante. Aparte de los momentos en que, al parecer de los usuarios, se requieren estas alteraciones y énfasis, es frecuente ver que se utilizan para saludar la llegada de algún personaje más o menos conocido en su entorno. Al producirse esto, tal y como ocurriría en la vida real, se manifiestan las emociones. Con la particularidad de que en los chats las emociones sólo se vehiculan por teclado y pantalla. De ahí las alteraciones mencionadas.
- c. En tercer lugar están las distorsiones que podemos etiquetar como 'incorrecciones deliberadas'. Los usuarios normalmente con el tiempo tienden no sólo a descuidar su nivel de corrección gramatical general y a utilizar los recursos mencionados en segundo lugar, sino que una buena parte de ellos cometen multitud de 'faltas' de modo deliberado y exagerado. Con ello pretenden individualizar su forma de expresión y hacer del entorno algo mucho más coloquial. Así, se entra en un juego metalingüístico de alto contenido humorístico.
- d. Jugar con los símbolos

De todos es conocido lo que son los emoticonos o 'smilies'. Sea cual sea su origen (unos lo remontan a los años cincuenta, usándose emoticonos en postales y cartas; otros al mundo del cómic japonés y otros los más lo consideran algo propio de las comunicaciones mediadas por ordenador), los emoticonos juegan un papel expresivo muy importante en los chats. La teoría de los emoticonos consiste en utilizar los recursos gráficos de que dispone el código ASCII para expresar emociones y

sentimientos de forma directa, proporcionando una información pseudo-gestual y empática.

Existe una importancia expresiva que tienen los emoticonos para los usuarios de chats de IRC.

e. Jugar con el medio

Se pueden confeccionar mensajes de entrada, de salida o comentarios para las acciones que lleva a cabo un usuario, lo que enriquece el nivel comunicativo del chat. Existe una multitud de comandos y opciones con las que gestionar el comportamiento y las posibilidades del usuario en su entorno de charla IRC. Y, a nivel comunicativo y lingüístico, por encima de todo esto, está el comando "/me", al que podemos llamar 'tercera persona narrativa' (TPN). Mediante la TPN, se elaboran mensajes del tipo "Usuario se está durmiendo" o "Usuario se muere de risa", etc.

El resultado de la sencilla utilización de este comando es fascinante. Lo que en un principio sería un simple intercambio de 'cosas dichas', por medio de la TPN también encontramos que se comunican 'cosas hechas'. No se supe así la falta de 'carnalidad' del chat, pero proporcionan una dimensión diferente y dramática. Sirve para llamar la atención, para subrayar de modo especial un contenido y para jugar el papel de 'voz' en off descriptiva de lo que los participantes están haciendo -y no sólo diciendo. Permite imaginar los chats -IRC, en particular- como algo con una inusitada vehemencia teatral, enriquecen la creación dramática del personaje/usuario y proporciona una elaboración narrativa/performativa que mezcla lo espontáneo y lo próximo con un lenguaje en tercera persona, convencionalmente distante.

HTML (Hypertext Markup Language) se ha convertido en un lenguaje de marcas de inmensa popularidad durante estos últimos años. También debemos anotar que nos hemos encontrado con sus propias limitaciones, que algunas de ellas se han querido

subsano con la incrustación de scripts, javascripts, Active X, HTML dinámico, hojas de estilo en cascada (CSS). Todo esto es insuficiente para crear una arquitectura abierta de tipo cliente/servidor, con lo que el W3C (Word Wide Web Consortium), organismo que vela por el desarrollo de la Word Wide Web, se ha replanteado crear un nuevo estándar llamado XML (eXtensible Markup Language), que parte de las amplias especificaciones del SGML (Standard Generalized Markup Language ) es el estándar internacional para la definición de la estructura y el contenido de diferentes tipos de documentos electrónicos. XML fue desarrollándose por el Grupo de Trabajo XML desde 1996 (en estos primeros años llamado SGML Editorial Review Board). La especificación XML 1.0 ha sido ratificada por la W3C el 10 de febrero de 1998, e interpretado como "un sistema para definir, validar y compartir formatos de documentos en la Web".

En el futuro y actual presente de la red está en el comercio electrónico y en el uso de tecnologías de telefonía IP, WAP y UMTS. Ante este futuro y ya presente que puede también permanecer incierto, cuanto menos en sus momentos iniciales, están comenzando a surgir lenguajes alguno de ellos propios a determinada tecnología como el WML y WMLScript con la telefonía móvil WAP.

- CHATS CGI

Los chats CGI funcionan con cualquier navegador y son más lentos, aunque esto puede dar lugar a conversaciones más pensadas.

- TERTULIAS

Una tertulia no es lo mismo que un chat. Son tablones de anuncios donde la gente deja sus mensajes y espera a que le contesten, desde minutos hasta horas o días.

- El Club de la Amistad.

Es una forma muy eficaz de hacer amigos via e-mail. Tienen una enorme cantidad de amigos inscritos en el club, todos con ganas de conocer nuevas personas.

- CHATS EN JAVA

Requieren Netscape 2.0 o superior o Explorer 3.0 o superior. Son rápidos y con posibilidades

- CHATS TELNET

Estos chats no utilizan navegadores como Netscape, etc. sino que requieren el uso de un programa especial llamado cliente TELNET.

- CHATS RESTRINGIDOS

Estos son chats que presentan algún tipo de restricciones o limitación, tales como el horario, ser de pago, etc.

- CHATS EROTICOS

A estos chats se va principalmente a tener conversaciones eróticas. En el resto de los chats se considera de muy mal gusto tener estas conversaciones en público.

### 3.2.3- Mensajería Instantánea

La Mensajería Instantánea es un punto intermedio entre los sistemas de chat y los mensajes de corre electrónico, las herramientas de mensajería instantánea, son programas regularmente gratuitos y versátiles, residen en el escritorio y, mientras hay una conexión a Internet, siempre están activos.

El servicio de mensajería instantánea ofrece una ventana donde se escribe el mensaje, en texto plano o acompañado de iconos o "emoticons" (figura que representan estados



de ánimo), y se envían a uno o varios destinatarios quienes reciben los mensajes en tiempo real, el receptor lo lee y puede contestar en el acto.

A las últimas versiones se les han añadido una serie de aplicaciones extra como la posibilidad de entablar conversaciones telefónicas, utilizando la infraestructura de Internet, lo mismo que contar con sistemas de información financiera en tiempo real, y el compartir diferentes tipos de archivos y programas, incluidos juegos en línea.

La mensajería instantánea es una mezcla de correo electrónico y chat, por un lado nos permite enviar mensajes a amigos (conocidos como contactos) que estén en ese momento conectados a Internet y, ya en las últimas versiones de las distintas aplicaciones de mensajería, disponen de "salas" de chat en donde te encuentras con gente desconocida.

La mensajería instantánea nace en Israel gracias a la idea de un padre y su hijo que crean el primer programa llamado ICQ en el año 1996. Más tarde, en el año 1998, America Online compra Mirabili (empresa propietaria de ICQ). Según AOL, ICQ tenía 20 millones de usuarios activos en el año 2000 pero esta cifra ha ido decreciendo constantemente debido a la aparición de nuevos programas de mensajería.

En la actualidad hay una gran variedad de programas con los que podemos conectarnos y charlar con nuestros amigos o enviar ficheros entre ordenadores. Los más importantes son: MSN Messenger, AOL Instant Messenger o Yahoo! Messenger entre otros. El gigante de la mensajería instantánea es, hoy por hoy, AOL Instant Messenger que según algunas estadísticas ocupa el 80% de usuarios que utilizan dicho servicio, aunque en los últimos meses MSN Messenger y Yahoo! Messenger han cobrado importancia.

El gran problema es que todos estos clientes no son compatibles entre si, es decir si tenemos contactos que utilizan MSN y otros que usan Yahoo! no podremos verlos a

todos a no ser que tengamos instalados ambos clientes aunque en la actualidad han aparecido programas compatibles con mensajerías diferentes.

### **3.2.4- El Futuro de la Mensajería Instantánea**

En la mayoría de sistema de mensajería no existe compatibilidad alguna entre la herramienta que utiliza y el producto similar de la competencia, este es uno de los principales problemas de este tipo de servicios.

Los datos de Nielsen//NetRatings muestran que ante la incompatibilidad de los sistemas de mensajería, la combinación más común por parte de los usuarios es utilizar los productos de AOL y Microsoft ya que son las aplicaciones más populares.

Se espera que en breve America OnLine, el principal jugador en sistemas de mensajería instantánea permita a otros programas interactuar con sus herramientas, más por presiones de las autoridades antimonopolio, para evitar que sus competidores se unan e impacten su posición en el mercado.

"Las discusiones sobre compatibilidad y estándares han bañado a la industria de la mensajería instantánea en el pasado", la interoperabilidad entre las plataformas de mensajería será un punto clave para promover e incrementar el uso y desarrollo de aplicaciones en este sector".

Puede que por ahora los sistemas de mensajería instantánea, al ser en su mayor parte gratuitos, no muevan miles de millones de dólares, pero al igual que los navegadores de Internet su importancia es estratégica.

Las herramientas de mensajería son una interfaz hacia el mundo en línea y que, de encontrarse el modelo adecuado para interactuar con el usuario, se pueden convertir

en un recurso importante para que las compañías e instituciones ofrezcan nuevos esquemas de contenido y comercio apoyados por socios comerciales.

En este sentido los servicios de mensajería instantánea podrían convertirse en un medio eficiente por el cual, instituciones comerciales y de servicios públicos podrían ofrecer servicios personalizados a los usuarios, explotando las funciones de saber en todo momento si la persona se encuentra en línea y conociendo sus hábitos de consumo.

### **3.3.- VENTAJAS DE LA CONFERENCIAS WEB**

La tecnología de la CMC hace posible que un grupo de personas distantes físicamente, sin la posibilidad de verse el uno al otro puedan comunicarse de manera sincrónica, al igual que en los encuentros cara a cara, usando la palabra escrita. En esta forma de comunicación se combinan la permanencia de la palabra escrita y la fluidez del intercambio propia de las conversaciones presenciales.

Dentro de las características principales podemos menciona:

Abierto las 24 horas del día todos los días. Internet y la totalidad de sus aplicaciones están disponibles las 24 horas del día todos los días. Sólo un par de clicks separan a la persona del acceso al mundo virtual si cuenta con el software necesarios. Una vez ingresado (conectado) a la red, siempre habrá personas esperando alguien con quien conversar. Puede plantearse la posibilidad de que la persona frecuente un mismo chat room y que en éste, a las 7 de la mañana, no haya usuarios. Este pequeño problema se soluciona fácilmente: se puede entrar a otros canales de otros países (por ejemplo, al de otro país que remite a un lugar del mundo donde son las 11 de la mañana y probablemente haya mas usuarios en línea).

Control sobre la presentación de uno mismo y sobre lo que los otros ven del sí mismo. En IRC, el anonimato, facilita la creación de un personaje. Las máscaras esconden a la persona y permiten jugar un personaje cuyas características son fácilmente configuradas por la propia persona.

Control sobre la relación. Los programas de IRC ofrecen la posibilidad de elegir con quien hablar y con quien no. Es decir, que si al sujeto no le interesa comunicarse con una determinada persona, con sólo tipear un comando (/ignore) seguido por, por ejemplo, el nickname de ésta, logra su objetivo. }

### **3.4.- REQUISITOS FUNCIONALES Y TECNOLÓGICOS DE LAS CONFERENCIAS WEB**

#### **3.4.1.- REQUERIMIENTOS TECNOLÓGICOS**

Para la instalación del Chat o Conferencias Web necesitaremos los siguientes requisitos:

##### **3.4.1.1- Instalación JDK**

- Instalar SDK

##### **3.4.1.2- Instalación Java Web Services Developer Pack**

Necesitamos bajarnos el JWSDP de la Dirección:

<http://java.sun.com/products/jsp/download.html>.

##### **3.4.1.- REQUERIMIENTOS FUNCIONALES**

- Pentium 3 o superior

- 256 -512 RAM
- Sistema Operativo
  - Sun Solaris Operating Environment 8 and 9, and Sun Solaris 9 for X86
  - Windows 2000 Professional Edition
  - Windows XP Professional Edition
  - RedHat Linux 8.0

# CAPITULO 4

## APLICACIÓN DE LA METODOLOGÍA UML EN EL DESARROLLO DEL SOFTWARE

### *FASE DE INICIO*

### GESTIÓN DEL PROYECTO

#### *Plan de Desarrollo de Software*

#### Introducción

El Plan de Desarrollo del Software provee una visión global del proyecto de desarrollo propuesto basado en la metodología “UML”.

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores

Se incluirá el detalle para las fases de Análisis, Diseño, Implementación, Pruebas, adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo el proceso.

El enfoque de desarrollo propuesto constituye una configuración del proceso UML de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (Productos entregables) que serán generados.

### *Propósito*

El propósito del Plan de Desarrollo de Software es brindar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software. Los usuarios del Plan de Desarrollo del Software son:

El jefe del proyecto, que lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.  
Los miembros del equipo de desarrollo, que lo usan para entender lo que deben hacer, cuándo deben hacerlo y que otras actividades dependen de ello.

### *Alcance*

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “Chat para la Escuela Politécnica del Ejército de Latacunga”. Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones. Para el Plan de Desarrollo del Software, se utilizara el paradigma Iterning que es el concepto que actualmente se les da a las conferencias Web o Chats empresariales, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”.

## Vista General del Proyecto

### *Propósito, Alcance y Objetivos*

La Escuela Politécnica del Ejército al ser una de las mejores instituciones del País debe tener en su Pagina Principal zonas de esparcimiento para los alumnos.

Es importante señalar que una de las principales críticas que se le hace a Internet y a los estudiantes es aquella de que pasar mucho tiempo en el Internet desvincula y enajena a las personas convirtiéndolas en seres solitarios sentados frente a un ordenador. Además que la

página de la Escuela Politécnica del Ejército Sede Latacunga no cuenta con áreas de distracción pero que al mismo tiempo sean de gran ayuda como el Chat.

La información que a continuación se incluye ha sido extraída de distintos Chat con el objetivo de conocer la manera como se manejan.

Por todo lo señalado anteriormente se impulsa a la elaboración de nuevas formas de hacer más divertida la comunicación

Los subsistemas se pueden diferenciar en 3 bloques:

a) Gestión de Administración, incluyendo:

- Gestión de Salas
- Gestión de Moderadores
- Gestión de Perfiles de Usuarios
- Asignación de Salas a Moderadores

b) Gestión de Usuarios:

- Ingresar al Chat

d) Gestión Mensajes:

- Envío Mensajes Generales y Privados

### *Suposiciones y Restricciones*

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de la información recolectada son:

- El sistema maneja las gestiones de administración, mensajes, usuarios y al ser tipo empresarial tienen un concepto diferente a la mayoría de chats existentes en el mercado razón por la cual no dispone de todas las características de dichos chats.



- El sistema lleva y elabora información fiable, pero esto no significa que no deba existir protección de información o mal manejo del sistema por parte de los usuarios.
- El sistema estará implementado hacia el ambiente Internet para en una segunda fase implementar algunos servicios como clases a través de Internet.

### *Entregables del proyecto*

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables.

Es preciso destacar que de acuerdo a la filosofía de UML todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, solo al término del proceso se podrá tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos.

- Plan de Desarrollo del Software
- Modelo de Casos de Uso del Negocio  
Es un modelo de las funciones del negocio vistas desde la perspectiva de los actores externos (solicitantes finales, otros sistemas etc.). Permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso.
- Modelo de Objetos del Negocio  
Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo asociados al caso de uso del negocio.
- Glosario  
Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.
- Modelo de Casos de Uso  
El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.
- Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo a los requisitos del sistema.

➤ Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos.

➤ Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Solo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

➤ Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases, pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño, de acuerdo al avance del proyecto.

➤ Modelo de Datos

Previendo que la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos, de acuerdo con el enfoque para modelado relacional de datos.

➤ Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.

➤ Modelo de Despliegue

Este modelo muestra el despliegue, la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

➤ Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba.

➤ **Manual de Instalación**

Este documento incluye las instrucciones para realizar la instalación del producto.

➤ **Material de Apoyo al Usuario Final**

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Manual de Usuario y Sistema de Ayuda en Línea

➤ **Producto**

Los ficheros del producto empaquetados y almacenadas en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva versión al final de cada iteración.

### *Evolución del Plan de Desarrollo del Software*

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

### **Organización del Proyecto**

#### *Participantes en el Proyecto*

El personal del proyecto (por la parte del la empresa adjudicataria), considerando las fases de Inicio, Elaboración y Construcción, estará formado por los siguientes puestos de trabajo y personal asociado:

- **Jefe de Proyecto.-** Ing. Raúl Cajas, Ingeniero en Sistemas e Informática, director de los Laboratorios de Sistemas de la ESPEL.

- Analista de Sistemas.- El perfil establecido es: Ingeniero en Informática con conocimientos de UML y experiencia en sistemas afines a la línea del proyecto, labor que llevará a cabo Cristian Molina.
- 1 Analistas - Programadores. Con experiencia en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser lo más cercanos posibles al producto final. Este trabajo ha sido encomendado a Cristian Molina.
- 1 Ingenieros de Software. El perfil establecido es: Ingeniero en Informática que realizará labores de gestión de requisitos, gestión de configuración, documentación y diseño de datos; además es el encargado de las pruebas funcionales del sistema. Esta labor la realizarán Cristian Molina.

### *Roles y Responsabilidades*

A continuación se describen las principales responsabilidades de cada uno de los roles en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan.

<b>Roles</b>	<b>Responsabilidades</b>
Jefe de Proyecto	<ul style="list-style-type: none"> <li>• Asignar recursos y gestionar prioridades</li> <li>• Coordina las interacciones con los clientes y usuarios.</li> <li>• Establecer un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto.</li> <li>• Supervisar el establecimiento de la arquitectura del sistema y planificación y control del proyecto.</li> </ul>
Analista de Sistemas	<ul style="list-style-type: none"> <li>• Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios.</li> <li>• Elaboración del Modelo de Análisis y Diseño.</li> <li>• Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.</li> </ul>
Programador	<ul style="list-style-type: none"> <li>• Construcción de prototipos.</li> <li>• Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario</li> </ul>

Ingeniero de Software	<ul style="list-style-type: none"> <li>• Gestión de requisitos.</li> <li>• Gestión de configuración y cambios.</li> <li>• Elaboración del modelo de datos.</li> <li>• Preparación de las pruebas funcionales.</li> <li>• Elaborar modelos de implementación y despliegue.</li> </ul>
-----------------------	--

**Tabla 0.1 Roles y Responsabilidades del Equipo de Desarrollo del Proyecto**

## Gestión del Proyecto

### *Estimaciones del Proyecto*

A continuación se muestran los recursos y costos directos e indirectos que se han generado:

#### 1. Recurso Humano

Nombre	Disponibilidad para el trabajo	Tiempo	Conocimiento de las herramientas	Remuneración
Cristian Molina	30%	12 meses	15%	\$150
<b>TOTAL:</b>				<b>\$ 1800</b>

**Tabla 0.2 Costos de Recurso Humano para el Proyecto.**

#### 2. Entorno de Ingeniería de Software

##### *Software*

Descripción	Precio	Cantidad	Tiempo	Costo
Base de Datos (XML)	\$0	1	6 meses	\$0
Servidor Web Apache Tomcat	\$0	1	12 meses	\$0
<b>TOTAL:</b>				<b>\$0</b>

**Tabla 0.3 Costos de Software para el Proyecto**

##### *Hardware*

Descripción	Precio	Cantidad	Tiempo	Costo
Computadora	40	1	12 meses	\$ 480
Impresora	10	1	12 meses	\$ 120
Cartuchos	5	1	12 meses	\$ 60
<b>TOTAL:</b>				<b>\$ 660</b>

**Tabla 0.4 Costos de Hardware para el Proyecto**

### 3. Varios

Descripción	Precio	Cantidad	Tiempo	Costo
Luz	16	1	12 meses	\$ 192
Internet	24	1	6 meses	\$ 144
Disquetes	0.50	12	12 meses	\$ 72
CR-RW	1.10	10	12 meses	\$ 132
<b>TOTAL:</b>				<b>\$ 540</b>

**Tabla 0.5 Costos Varios para el Proyecto**

### 4. Costo Total del Proyecto

Descripción	Costo
Recurso Humano	\$ 1800
Software	\$ 0
Hardware	\$ 660
Varios	\$ 540

<b>TOTAL:</b>	<b>\$3000</b>
---------------	---------------

**Tabla 0.6 Costo Total del Proyecto**

*Plan del Proyecto*

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

*Plan de las Fases*

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es solo una aproximación).

<b>Fase</b>	<b>Nro. Iteraciones</b>	<b>Duración</b>
Inicio	1	4 semanas
Elaboración	1	4 semanas
Construcción	2	16 semanas
Transición	-	-

**Tabla 0.7 Distribución de tiempos e iteraciones para cada Fase**

Los hitos que marcan el final de cada fase se describen en la siguiente tabla:

<b>Descripción</b>	<b>Hito</b>
Fase de Inicio	<ul style="list-style-type: none"> <li>La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.</li> </ul>
Fase de Elaboración	<ul style="list-style-type: none"> <li>La revisión y aceptación del diseño lógico y físico del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito.</li> </ul>
	<ul style="list-style-type: none"> <li>El hito que marca el fin de esta fase se establece con la</li> </ul>

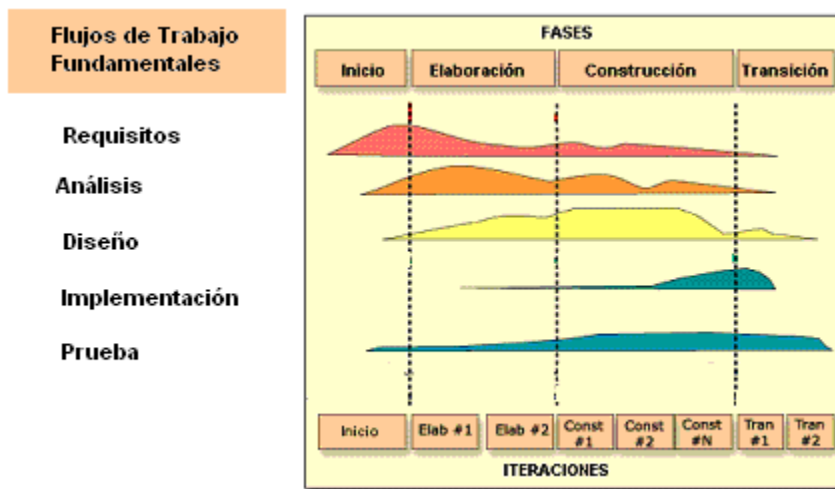
Fase de Construcción	capacidad operacional parcial del producto, lista para ser entregada a los usuarios para pruebas.
Fase de Transición	<ul style="list-style-type: none"> <li>El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios.</li> </ul>

**Tabla 0.8 Hitos que marcan el final de cada fase**

### *Calendario del Proyecto*

El Calendario presenta las principales tareas del proyecto incluyendo solo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto.

La siguiente figura ilustra este enfoque, en ella se marca el énfasis de cada disciplina (flujo de trabajo) en un momento determinado del desarrollo.





**Figura 0.1 Evolución de los flujos de trabajo en cada una de las fases**

Para este proyecto se ha establecido el siguiente calendario: la fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación.

<b>Disciplinas / Artefactos generados o modificados durante la Fase de Inicio</b>	<b>Comienzo</b>	<b>Aprobación</b>
<b><i>Modelado del Negocio</i></b>		
Modelo de Casos de Uso del Negocio Modelo de Objetos del Negocio	Semana 1 01/07/2004	
<b><i>Requisitos</i></b>		
Glosario	Semana 1 01/07/2004	
Visión	Semana 2 08/07/2004	
Modelo de Casos de Uso	Semana 3 15/07/2004	
Especificación de Casos de Uso	Semana 3 15/07/2004	
Especificaciones Adicionales	Semana 3 15/07/2004	
<b><i>Análisis / Diseño</i></b>		
Modelo de Análisis / Diseño	Semana 2 08/07/2004	
Modelo de Datos	Semana 2 08/07/2004	
<b><i>Implementación</i></b>		
Prototipos de Interfaces de Usuario	Semana 3 15/07/2004	
Modelo de Implementación	Semana 3 15/07/2004	
<b><i>Pruebas</i></b>		
Casos de Pruebas Funcionales y de Integración	Semana 3 15/07/2004	
<b><i>Despliegue</i></b>		
Modelo de Despliegue	Semana 3 15/07/2004	siguiente fase
<b>Gestión de Cambios y Configuración</b>	Durante todo el proyecto	
<b><i>Gestión del proyecto</i></b>		
Plan de Desarrollo del Software	Semana 1 01/07/2004	

<b>Ambiente</b>	Durante todo el proyecto
-----------------	--------------------------

**Tabla 0.9 Calendario del Proyecto para la Fase de Inicio**

<b>Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración</b>	<b>Comienzo</b>	<b>Aprobación</b>
<b><i>Modelado del Negocio</i></b>		
Modelo de Casos de Uso del Negocio Modelo de Objetos del Negocio	Semana 1 01/07/2004	Aprobado
<b><i>Requisitos</i></b>		
Glosario	Semana 1 01/07/2004	Revisar en cada iteración
Visión	Semana 2 08/07/2004	Revisar en cada iteración
Modelo de Casos de Uso	Semana 3 15/07/2004	Revisar en cada iteración
Especificación de Casos de Uso	Semana 3 15/07/2004	Revisar en cada iteración
<b><i>Análisis / Diseño</i></b>		
Modelo de Análisis / Diseño	Semana 2 08/07/2004	Revisar en cada iteración
Modelo de Datos	Semana 2 08/07/2004	Revisar en cada iteración
<b><i>Implementación</i></b>		
Prototipos de Interfaces de Usuario	Semana 3 15/07/2004	Revisar en cada iteración
Modelo de Implementación	Semana 3 15/07/2004	Revisar en cada iteración
<b><i>Pruebas</i></b>		
Casos de Pruebas Funcionales y de Integración	Semana 3 15/07/2004	Revisar en cada iteración
<b><i>Despliegue</i></b>		
Modelo de Despliegue	Semana 3 15/07/2004	Revisar en cada iteración
<b><i>Gestión de Cambios y Configuración</i></b>		
Durante todo el proyecto		
<b><i>Gestión del proyecto</i></b>		
Plan de Desarrollo del Software	Semana 4 22/07/2004	Revisar en cada iteración
<b>Ambiente</b>	Durante todo el proyecto	

**Tabla 0.10 Calendario del Proyecto para la Fase de Elaboración**

## *Seguimiento y Control del Proyecto*

### ***Gestión de Requisitos***

Los requisitos del sistema son especificados en el artefacto Visión. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

### ***Control de Plazos***

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto.

### ***Control de Calidad***

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias.

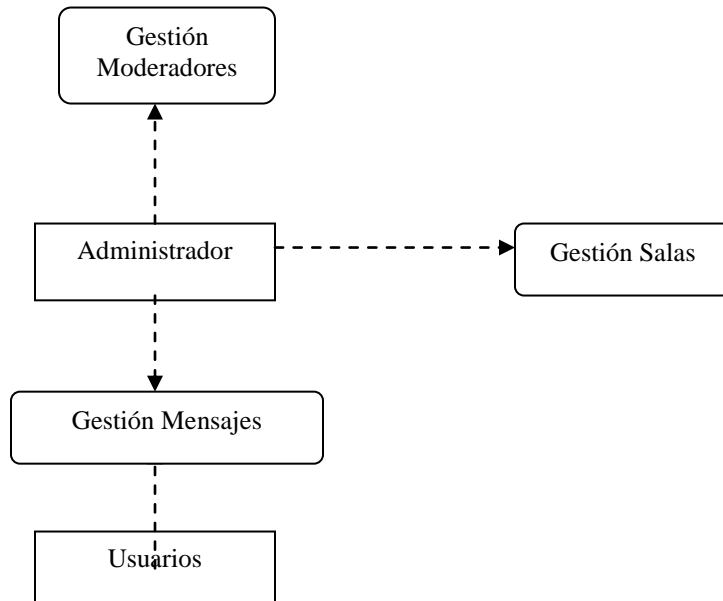
### ***Gestión de Configuración***

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que estas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá una línea base, la cual podrá ser modificada solo por una Solicitud de Cambio aprobada.

## **MODELADO DEL NEGOCIO**

El Chat tomado como referencia para el proyecto de desarrollo software consta de varios departamentos que contribuyen con cada una de sus funciones a la meta de la Institución Educativa que se resume en la siguiente frase.

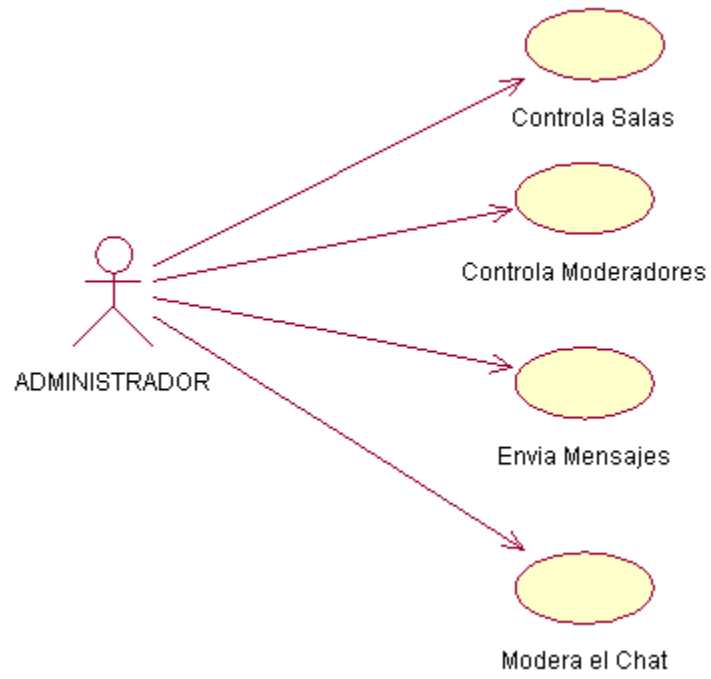
En el siguiente diagrama se representan los diferentes subsistemas en los que se ha dividido la Institución a nivel de abstracción:



### *Modelo de Casos de Uso del Negocio*

La Institución Educativa interactúa con distintos elementos externos, entre los que se identifican:

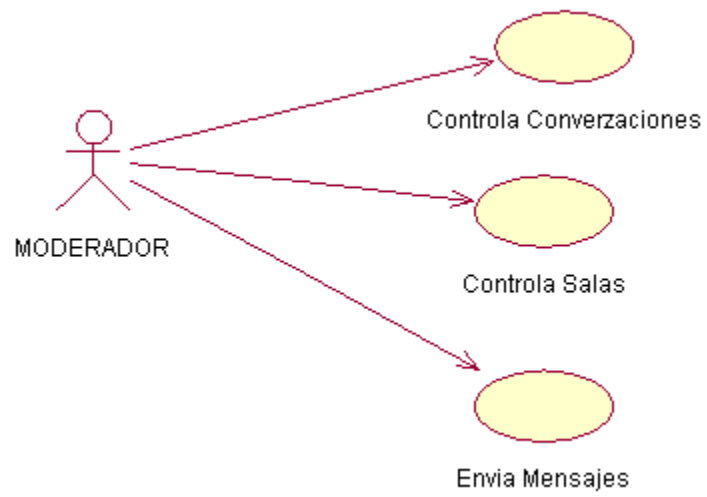
- Moderador.- Persona o profesor que se encarga de dirigir las charlas dependiendo de sus asignaturas respectivas.
- Alumno.- Persona que toma clases o ingresa al chat.



**Caso de Uso del Negocio 0.1**

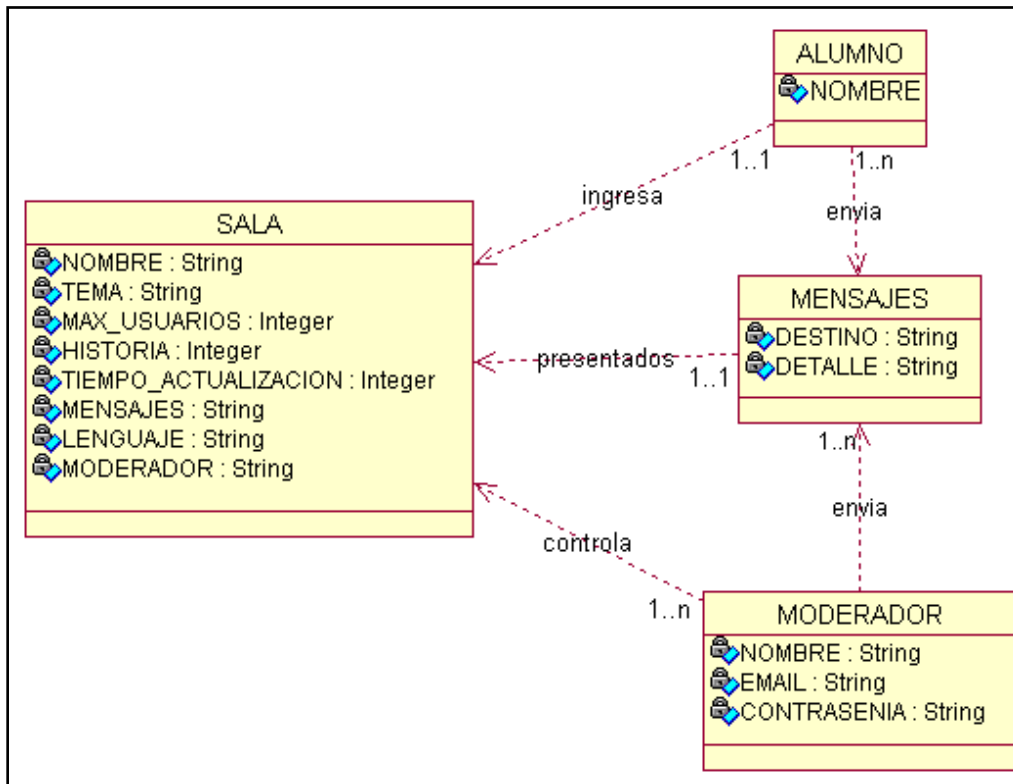


**Caso de Uso del Negocio 0.1**



**Caso de Uso del Negocio 0.2**

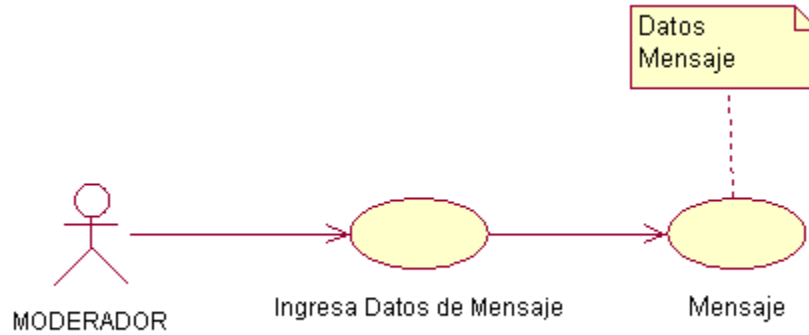
*Modelo del Dominio*



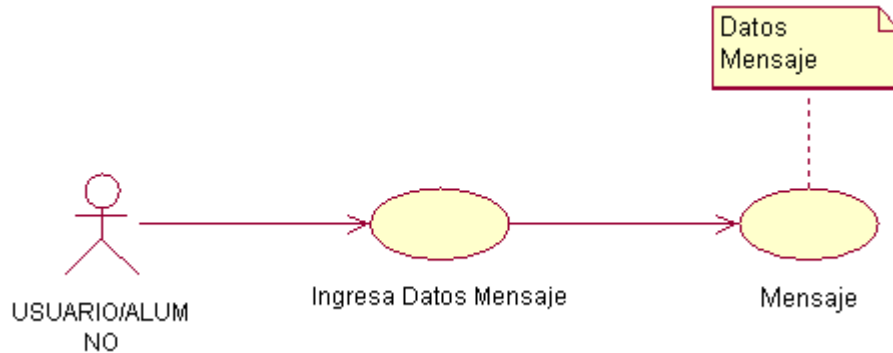
**Modelo del Dominio del Negocio 0.1**

### *Modelo de Objetos del Negocio*

Los modelos de objetos del negocio están asociados a los casos de uso del negocio que serán implementados en el proyecto.



**Modelo de Objetos 0.1 Caso de Uso Enviar Mensajes Moderador**



**Modelo de Objetos 0.2 Caso de Uso Enviar Mensajes Alumno**

## REQUISITOS

### *Glosario*

### Introducción

El Glosario recoge todos y cada uno de los términos manejados a lo largo de todo el proyecto de desarrollo del Chat. Se trata de un diccionario informal de datos y



definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para todo el proyecto.

### *Propósito*

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo de un Chat. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto.

### *Alcance*

El alcance del presente documento pretende que todos los usuarios manejen la misma terminología con claridad.

### Definiciones

A continuación se presentan todos los términos manejados a lo largo de todo el proyecto de desarrollo de un Sistema de Chat.

#### **Administrador**

El Administrador es la persona del departamento de Unidad y Sistemas que se encarga de la función de administración del Web Site de la Escuela Politécnica del Ejército Sede Latacunga y seguridades en cuanto a claves y usuarios se refiere.

#### **Moderador**

Se refiere a la característica especial de un usuario que le permite tener acceso a las Salas con el objetivo de imponer y controlar las reglas de las conversaciones y realizar actividades determinadas, se utiliza para darle seguridad a las Salas.

## **Usuario**

Se refiere a las personas pertenecientes a la ESPEL que hacen uso del sistema.

## **Salas**

Se refiere a un nombre que identifica la conversación o clase que se está dictando en el Chat.

## **Parámetros**

Los Parámetros son los datos relacionados con la Sala y dependen de cada Profesor o Moderador para definir el número de usuario, tiempo de actualización del chat, etc.

## **Gestión Administración**

Este caso de uso incluye las siguientes gestiones: Gestión Moderadores, Gestión Salas, Asignar Moderadores a Salas, Enviar Mensajes, Sacar e Ingresar Usuarios a Chat, Consultar esta actual de Salas con Usuarios.

## **Gestión Profesores/Moderadores**

Gestión Profesores es un caso de uso definido dentro del Departamento de Unidad y Sistemas, y cuya funcionalidad está definida por el administrador. La gestión Profesores trata todos aquellos aspectos que conciernen al tratamiento de datos relacionados con los Profesores de la Institución, ya sea el alta de un nuevo Profesor, dar de baja un Profesor, modificar de los datos de un Profesor y la búsqueda de los datos relacionados a él. Este caso de uso se puede invocar a través de la interfaz de Administración.

## **Gestión Alumnos**

Gestión Alumnos que permite el ingreso a las diferentes Salas del Chat. Este caso de uso se puede invocar a través de la interfaz de Usuarios.

## *Visión*

### Introducción

#### *Propósito*

El propósito de éste documento es recoger, analizar y definir las necesidades de alto nivel y las características del Chat. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales.

Esta funcionalidad se basa principalmente en el concepto de Chat Empresariales.

Los detalles de cómo el sistema cubre los requerimientos de los usuarios se pueden observar en la especificación de los casos de uso.

#### *Alcance*

El documento Visión se refiere, como ya se ha indicado, al Chat. Dicho sistema será realizado por el grupo de desarrollo de software.

El sistema permitirá al personal Docente de la Escuela dictar clases en Línea a través del Chat, y a los alumnos poder comunicarse.

#### *Definiciones, Acrónimos, y Abreviaciones*

**UML:** (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos

**Stakeholders:** Son las personas que participan directamente en el proyecto.

### Posicionamiento

#### *Oportunidad de Negocio*

Este sistema permitirá que los Alumnos de la Escuela Politécnica del Ejército puedan informatizar sus clases, lo cual supondrá un acceso rápido y sencillo a las discusiones en la Salas, gracias a interfaces sencillas y amigables.

*Sentencia que define el problema*

El problema de:

Informatizar y controlar todo lo relativo a las discusiones y foros de manera que se puedan comunicar entre alumnos y profesores de manera rápida y eficaz.

Afecta a:

- Profesores, Tutores o secretarios académicos encargados del manejo de alumnos
- Alumnos.

El impacto asociado es:

Automatizar la comunicación de Alumnos y Profesores.

Una solución adecuada sería:

Implementar un Chat que permita informatizar una parte de la comunicación.

*Sentencia que define la posición del Producto*

Para:

- Profesores, Tutores.
- Alumnos de la Escuela Politécnica del Ejército

Quienes:

- Estan encargados de dictar clases, realizar discusiones con los Alumnos.
- El nombre del producto
- Es un producto software.

Que:

Permite la comunación entre Profesores y/o Alumnos.

Nuestro producto:

Permite una comunicación entre el Personal Docente y alumnado mediante una interfaz sencilla y amigable.

### *Resumen de Usuarios*

<b>Nombre</b>	<b>Descripción</b>	<b>Responsabilidades</b>	<b>Stakeholder</b>
Administrador y/o Secretario	Encargado de la función de registrar y/o actualizar la información necesaria para que el sistema software (Creación de Salas , Moderadores ,etc)	Estructuración de información de salas y moderadores	Organización y Sistemas
Docente	Persona que modera el chat a los alumnos.	Modera discusiones	Docentes
Alumno	Persona que utiliza .el producto software	Utilización del Chat	Alumno

**Tabla 0.11 Resumen de Usuarios**

### *Entorno de usuario*

No se puede estimar el número exacto de personas que se involucren con el sistema, ya que al ser de una institución educativa los alumnos se van incrementando o decrementando. El tiempo que lleva el realizar una tarea es en segundos, aunque esto puede fluctuar de acuerdo al número de procesos concurrentes en el servidor.

Este sistema es similar a cualquier aplicación Windows y por tanto los usuarios estarán familiarizados con su entorno.

Los usuarios entrarán al sistema sobre un ordenador con un sistema operativo Windows 9X o Xp.

Si se desea que el sistema funcione en el Web la aplicación se integra con este servicio.

## Perfil de los Stakeholders

<b>Representante</b>	<b>Varios</b>
Descripción	No existen nombres ya que los datos fueron recabados de diferentes servicios de chat disponibles en la web.
Tipo	Son personas con nivel medio de conocimiento en la rama de conferencias web.
Responsabilidades	Encargado de mostrar las necesidades de cada usuario para el desarrollo correcto del sistema. Además, lleva a cabo un seguimiento del desarrollo del proyecto y aprobación de los requisitos y funcionalidades del sistema
Criterio de Éxito	Cuando se confronta los requisitos contra el sistema y el resultado es el 100%.
Grado de participación	En la fase de inicio y pruebas 100%.
Comentarios	Ninguno

**Tabla 0.12 Perfil de Stakeholders**

## Perfiles de Usuario

### Administrador

<b>Representante</b>	<b>Departamento Secretaria</b>
Descripción	Personal del área académica/administrativa.
Tipo	Usuario importante del sistema.
Responsabilidades	Encargado de la función de registrar y/o actualizar la información necesaria para que el sistema software funcione correctamente.
Criterio de Éxito	A definir por el cliente
Grado de participación	A definir por el cliente
Comentarios	Ninguno.

**Tabla 0.13 Perfil de Usuario Administrador**

### Docente / Moderador

<b>Representante</b>	<b>Departamento Docentes</b>
Descripción	Representante de las actividades del docente.
Tipo	Usuario importante del sistema.
Responsabilidades	Persona que modera las conerezaciones.
Criterio de Éxito	A definir por el cliente
Grado de participación	A definir por el cliente

Comentarios	Ninguno.
-------------	----------

**Tabla 0.14 Perfil de Usuario Docente**

Alumno

<b>Representante</b>	<b>Alumnado</b>
Descripción	Persona que se involucra directamente con la institución.
Tipo	Usuario no habitual.
Responsabilidades	Persona base de la Institución Educativa.
Criterio de Éxito	A definir por el cliente
Grado de participación	A definir por el cliente
Comentarios	Ninguno.

**Tabla 0.15 Perfil de Usuario Alumno**

Apreciación Global del Producto

*Perspectiva del producto*

El producto a desarrollar es un sistema de Chat, con la intención de agilizar la comunicación Alumnos Profesores. Las áreas que son tratadas por el sistema son: académica, docente.

*Resumen de características*

A continuación se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

<b>Beneficio del cliente</b>	<b>Características que lo apoyan</b>
Mayor facilidad para la comunicación entre alumnos y profesores desde cualquier lugar del Mundo.	A travez de la Pagina Web de la Escuela.

**Tabla 0.16 Resumen de Características**

### *Suposiciones y dependencias*

La versión del sistema operativo puede variar de acuerdo al hardware designado para el producto del software.

Los Navegadores de Internet pueden ser: Microsoft Internet Explorer 5, Netscape 5 o superiores, esto es por los efectos que se utilizan en la aplicación.

### *Costo y precio*

El costo o precio de la siguiente lista de productos es independiente del costo del sistema; es decir, la distribución cuesta:

<b>Cantidad</b>	<b>Descripción</b>	<b>Costo</b>
3	CD-ROM	2.50
2	Manual de Usuario	200.00
1	Logística	5.00

**Tabla 0.17 Costo y Precio de la Distribución del Producto**

### Descripción Global del Producto

#### *Departamento Docente*

Departamento que dirige y gestiona las salas de los alumnos, es decir; el docente tiene la opción de enviar mensajes. Habilitar/deshabilitar usuarios y consultar usuarios activos.

### Restricciones

Debe contemplarse las siguientes restricciones:



- El sistema tiene algunas opciones de la mayoría de Chats pero no dispone de algunas funciones por seguridad en la comunicación y para que sea más manejable las Salas.
- El sistema esta creado en entorno Windows pero si las personas encargadas de administrarlo requieran cargarlo en Linux, no se requiere cambiar el producto software.
- El sistema estará implementado hacia el ambiente Internet.

## Precedencia y Prioridad

La prioridad de todos los procesos que a continuación se detalla es de alto nivel, ya que los procesos se relacionan entre sí y en un ejemplo crítico si no se implementa un proceso no puede existir el otro.

La precedencia de los procesos es la siguiente:

- Moderadores
- Salas
- Usuarios

## Otros Requisitos del Producto

Para la correcta operación del Sistema es fundamental conocer los requisitos que el sistema puede utilizar:

### *Estándares Aplicables*

- Normas de comunicación: TCP/IP

- Plataforma: Múltiples plataformas

### *Requisitos del Sistema*

#### *Requisitos de Hardware:*

Un servidor WEB, la configuración de hardware recomendada para el servidor mencionado es la siguiente:

- CD-ROM drive
- 512MB – 1 GB de memoria
- 1 GB de espacio libre en disco para instalación
- 1 tarjetas de red Ethernet y soporte TCP/IP instalado

#### *Requerimientos de Software:*

- Instalar J2SD versión 1.4.2 o superior
- Se recomienda la instalación de Java Web Services Developer Pack versión 1.3. o superior, un servidor WEB que permite presentar los elementos, objetos y componentes del sistema tanto a nivel local como en cada una de las computadoras personales de los miembros del portal de Cliente.
- Navegadores de Internet: (Microsoft Internet Explorer 5, Netscape 5 o superiores) en las computadoras personales de los miembros del portal de Cliente.

### *Requisitos de Desempeño*

A continuación se detalla los requisitos de actuación con respecto al sistema:

<b>Requisito</b>	<b>Valor</b>
Capacidad de comunicación	El sistema puede funcionar en uno o varios terminales, esto no influye en la calidad del producto.
Exactitud	100%
Fiabilidad	100%
Velocidad	Es importante pero no es fundamental en el sistema.

**Tabla 0.18 Requisitos de Desempeño**

### *Requisitos de Entorno*

- Temperatura: Ambiente
- Humedad: Normal

### Requisitos de Documentación

#### *Manual de Usuario*

El manual de usuario tendrá la longitud de no menos de 10 hojas, el nivel de detalle de los temas que se trataran en el manual varía de acuerdo al tópico que se este tratando, el manual de usuario estará diseñado de la siguiente manera:

- Información del manual
- Índice
- Contenido
- Glosario

#### *Ayuda en Línea*

El sistema no cuenta con una ayuda en línea al internet, pero dentro del sistema se tiene esta opción, la misma que tiene como objetivo guiar al usuario en el uso del sistema.

### *Guías de Instalación, Configuración*

El sistema contará con una guía de instalación y configuración que al final del proyecto se le entregará al usuario en el que se detallará todos los pasos que debe seguir.

### *Atributos de Características*

<b>Número y nombre de la característica</b>	<b>Estado</b>	<b>Beneficio</b>	<b>Esfuerzo</b>	<b>Riesgo</b>	<b>Estabilidad</b>	<b>Asignación</b>
5.1.3 Seguridades	Propuesta: Sí Aprobada: Sí Incorporada: No	Importante	medio	bajo	estable	Cristina Molina
5.2 Departamento de docente	Propuesta: Sí Aprobada: Sí Incorporada: No	Importante	medio	medio	no estable	Cristina Molina
5.3.2 Usuarios	Propuesta: Sí Aprobada: Sí Incorporada: Sí	Crítico	Alto	medio	no estable	Cristina Molina

**Tabla 0.19 Atributos de Características**

### *Modelo de Casos de Uso*

#### **Casos de Uso**

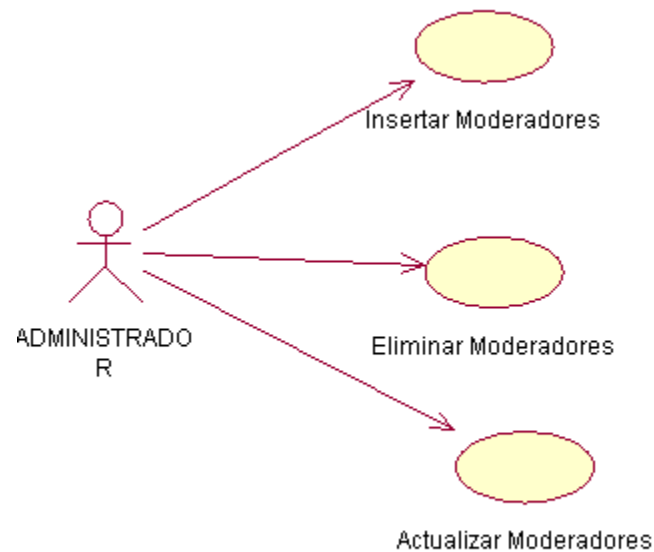
A continuación se presentan los diagramas de casos de uso planteados para cada uno de los subsistemas definidos para la Institución Educativa.

#### **Especificación de casos de uso: Gestión de Profesores/Moderadores**

#### **Descripción**

Este caso de uso resume la utilidad de alta, baja y modificación de los datos registrados en el documento XML de la plantilla de moderadores. El administrador

podrá acceder a los datos de profesores para realizar las siguientes operaciones: crear, eliminar, modificar

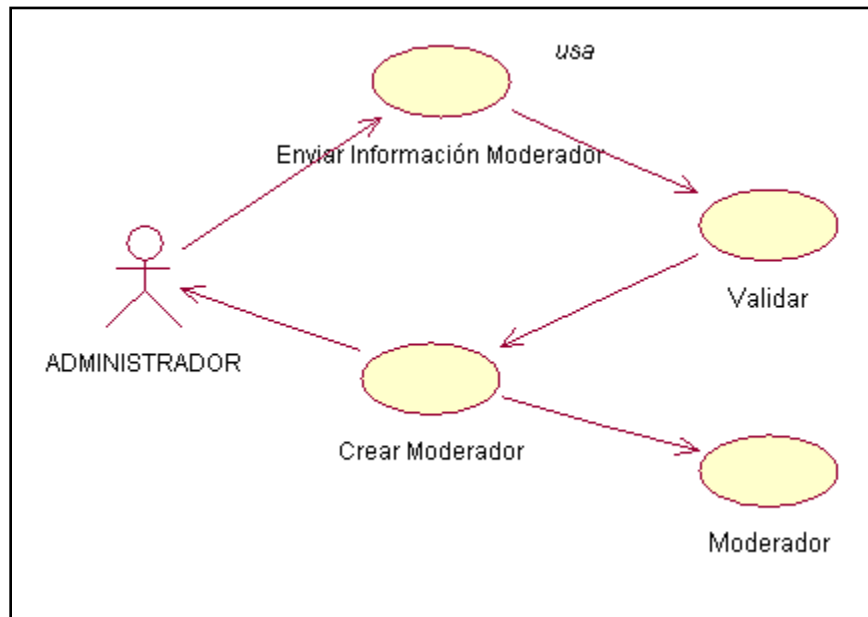


### Caso de Uso 0.2 Gestión Profesores

#### Flujo de Eventos

##### Flujo Básico

1. El Administrador puede seleccionar dar de alta un nuevo moderador, pasar al punto 2; dar de baja un profesor, pasar al punto 3; modificar datos de un profesor.
2. El Administrador solicita el alta de un nuevo moderador presionando la opción Añadir Moderador.



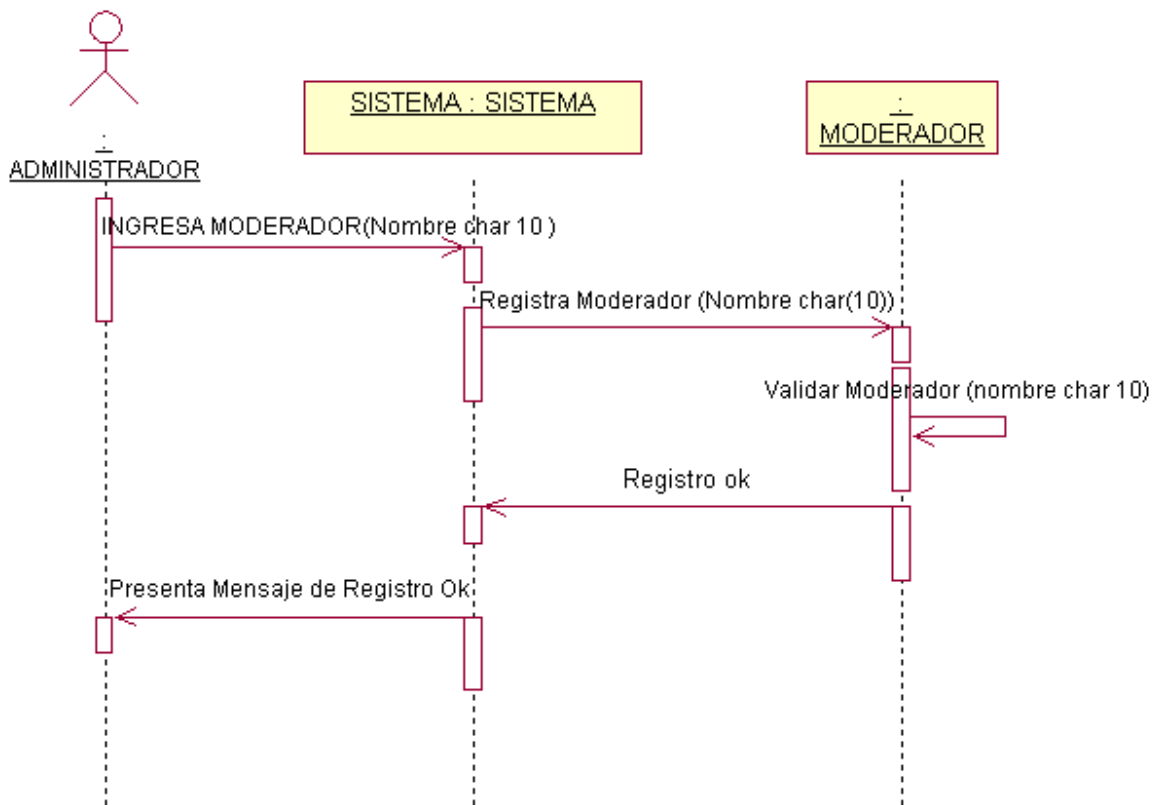
### **Caso de Uso 0.3 Insertar Moderadores**

El sistema presenta los siguientes campos de datos a introducir: nombre y mail.

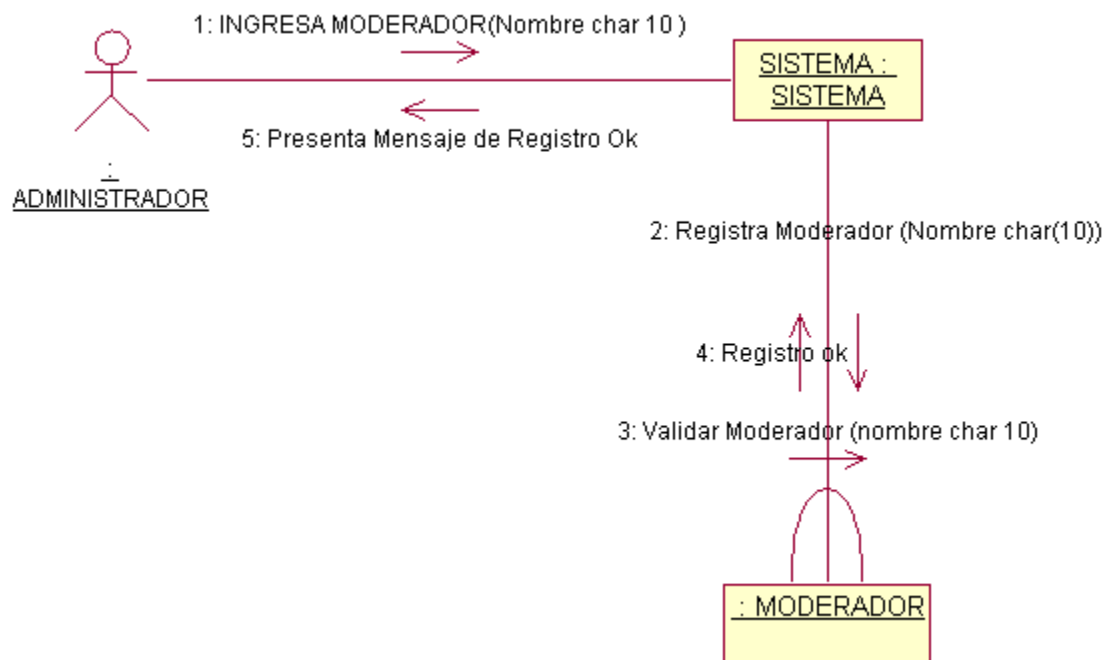
El campo nombre debere ser de tipo texto

El campo mail debere ser un mail valido

El Administrador pulsa el icono “guardar”. Ver punto 4.1.



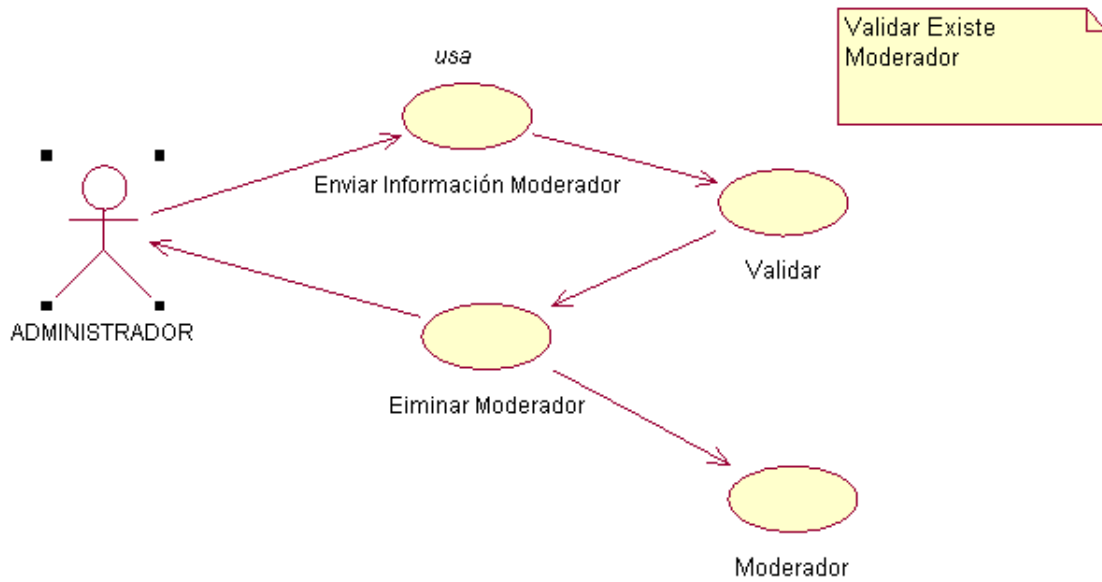
**Diagrama de Secuencia 0.1 Insertar Profesores**



**Diagrama de Colaboración 0.1 Insertar Profesores**

3. El Administrador solicita la baja de un moderador.



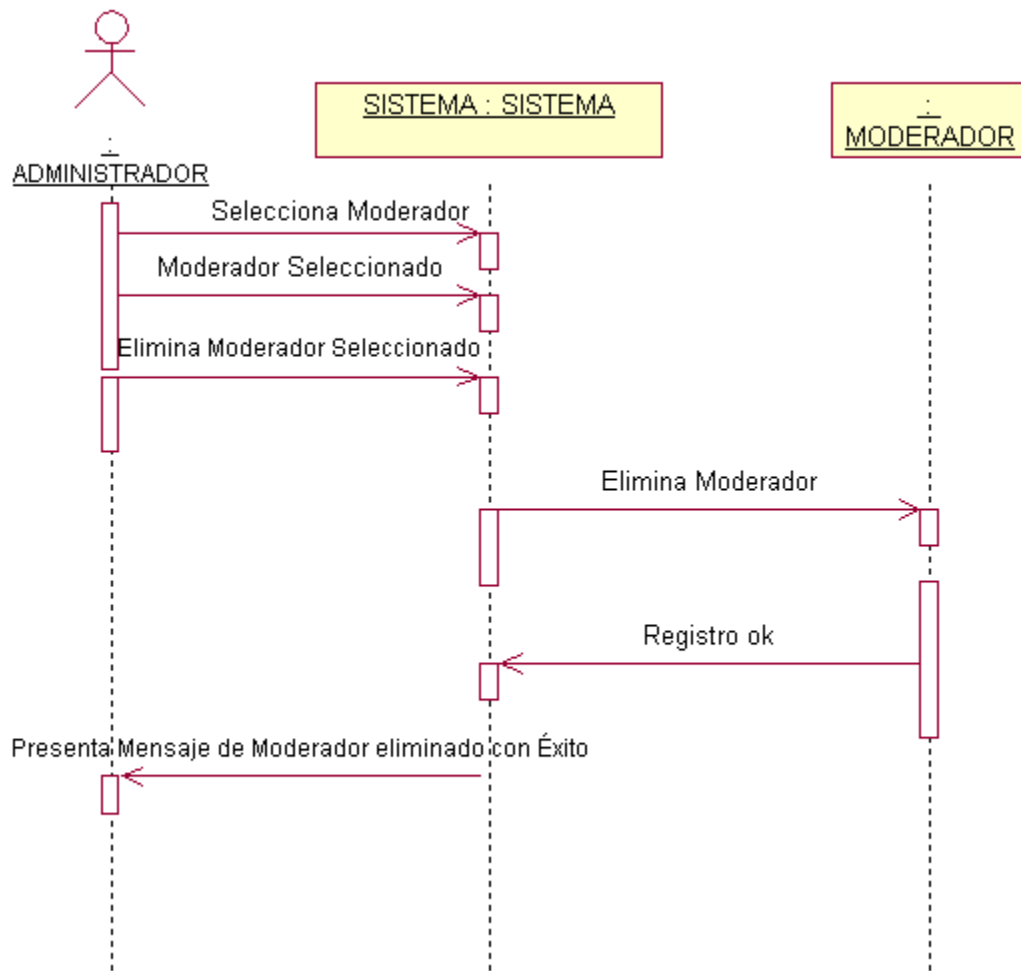


#### Caso de Uso 0.4 Eliminar Moderador

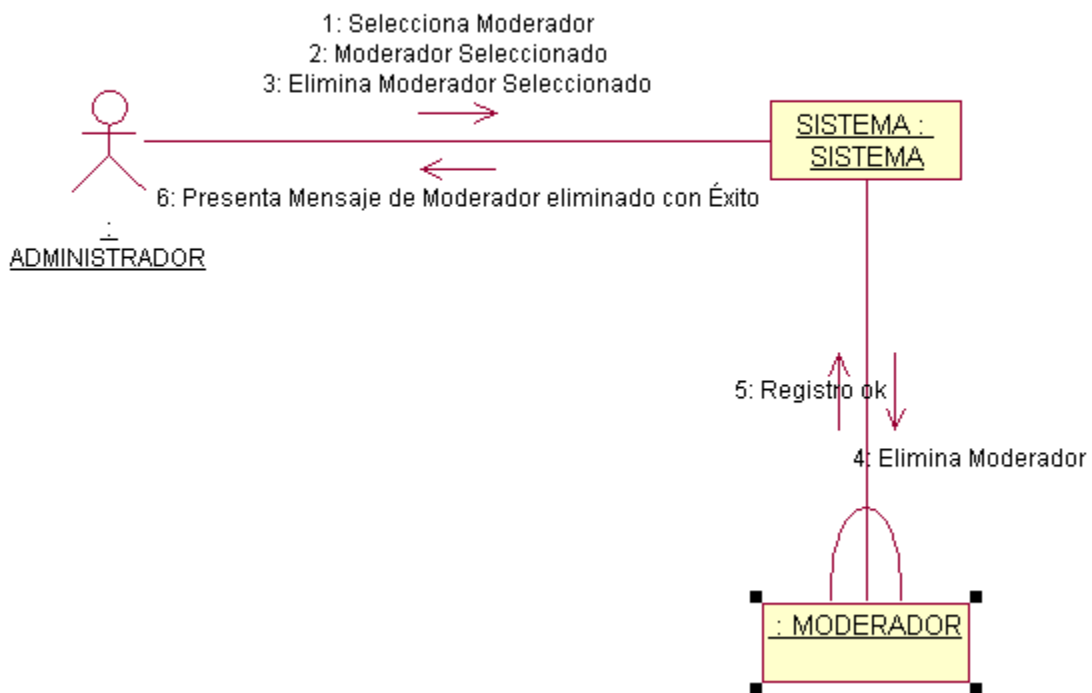
El Administrador selecciona el moderador a ser eliminado del listado de moderadores activos de la Institución.

El Administrador presiona el boton Eliminar

El sistema genera un mensaje de aviso de borrado.

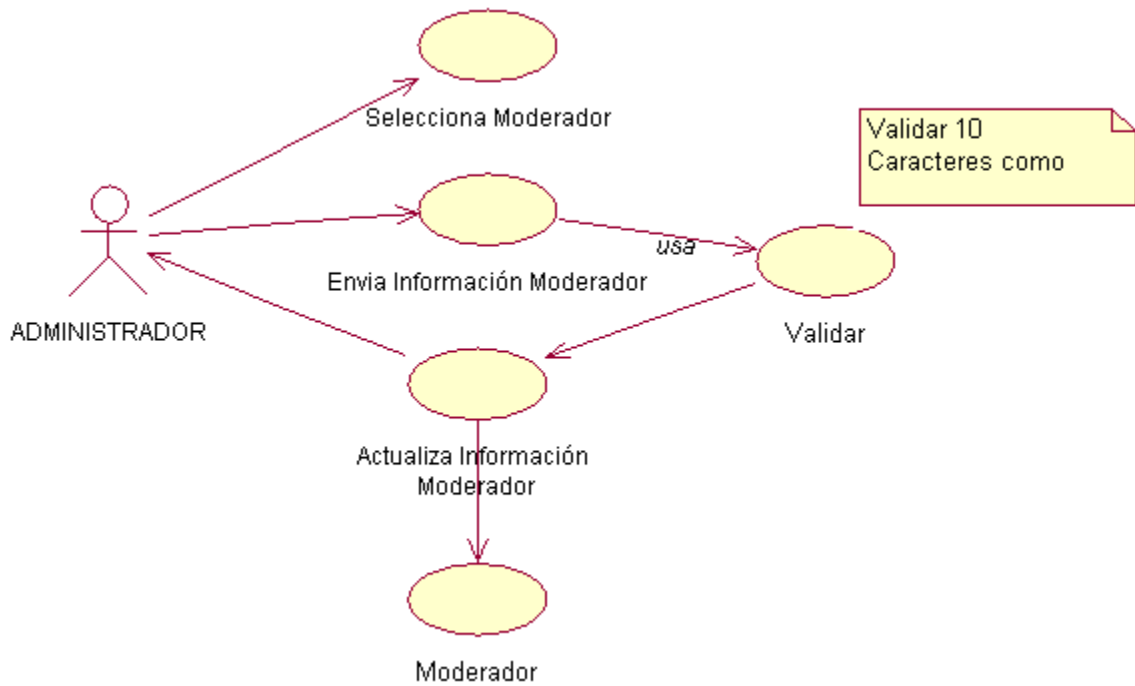


**Diagrama de Secuencia 0.2 Eliminar Moderador**



**Diagrama de Colaboración.0.2 Eliminar Moderador**

4. El Administrador solicita la modificación de los datos de un moderador previa selección del registro a ser modificado.



### Caso de Uso 0.5 Modificar Moderador

El Administrador selecciona el registro del moderador a ser modificado del listado de moderadores activos de la Institución.

El sistema muestra los datos del moderador que se ha solicitado para la modificación.

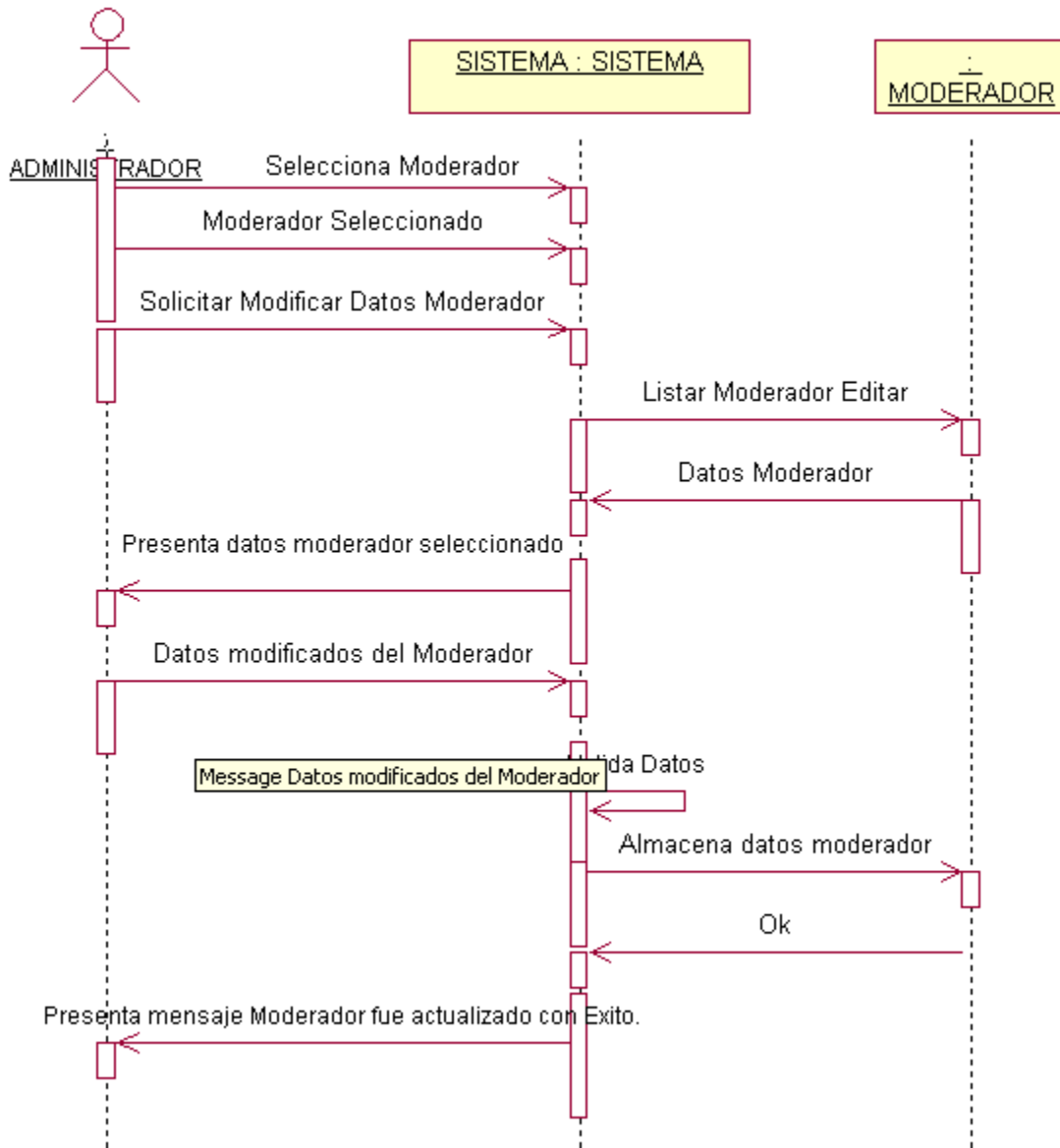
El Administrador puede modificar los siguientes campos: nombre, mail.

El campo nombre debe ser carácter de 10

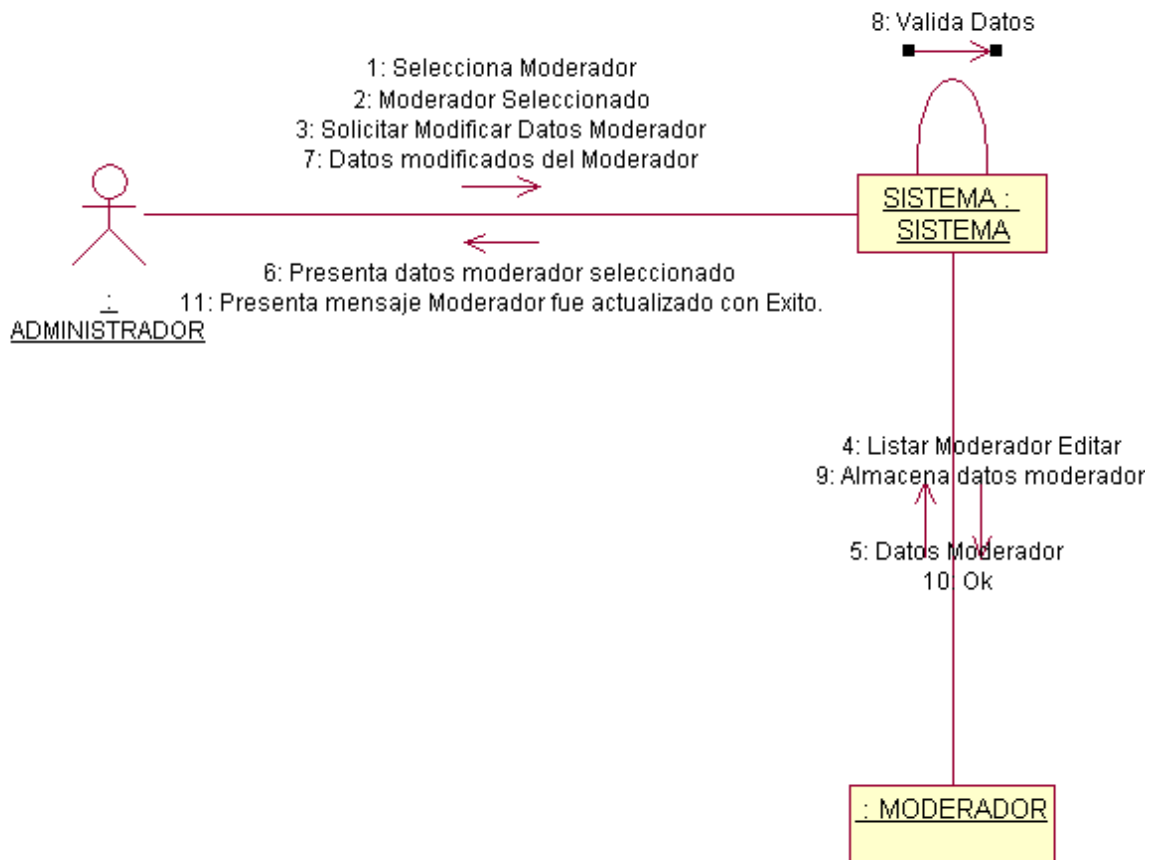
El campo mail debe ser de tipo correo

El campo password puede ser alfanumérico.

El Administrador puede solicitar almacenar los datos modificados pulsando el botón “Actualizar” de la interfaz gráfica. Ver punto 4.3.



**Diagrama de Secuencia 0.3 Modificar Moderadores**



**Diagrama de Colaboración.0.3 Modificar Profesores**

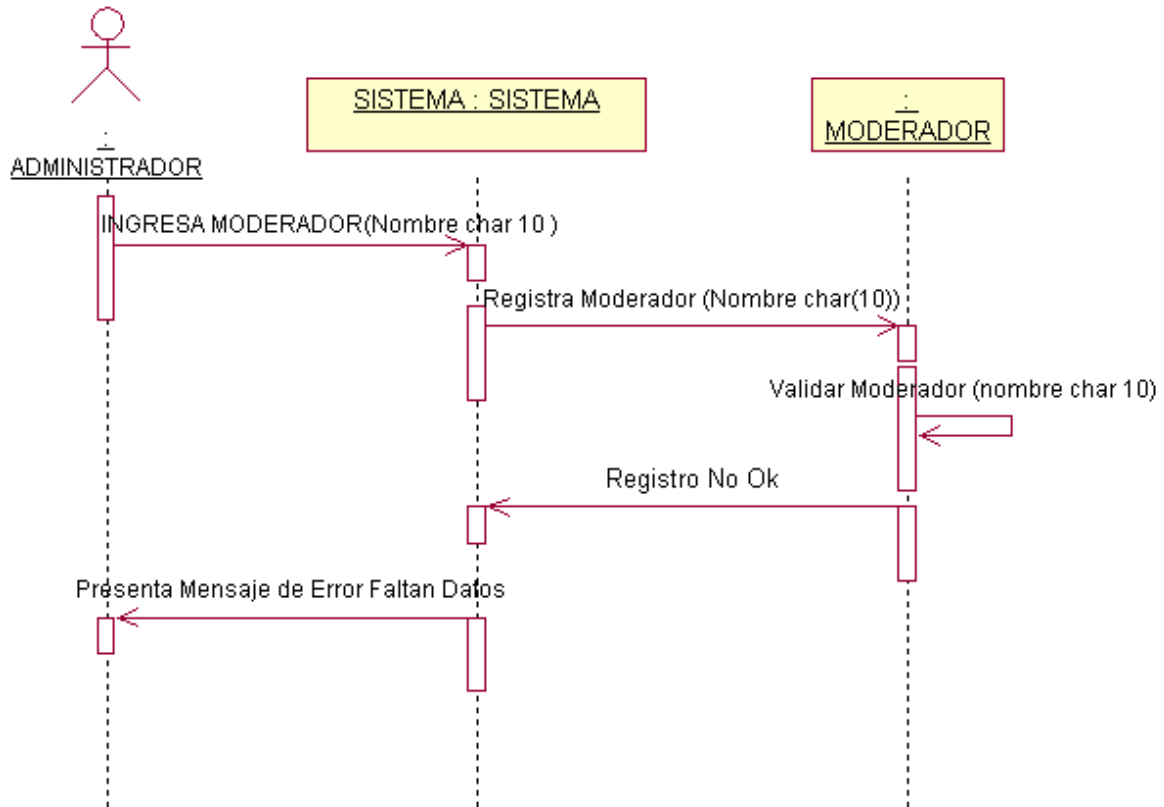
### Flujos Alternativos

*En el punto 2.2*

El sistema comprueba que el nombre del moderador no corresponda con ningún otro registro en el documento XML. En caso afirmativo, generará un mensaje comunicando que dicho moderador ya está dado de alta en el documento XML.

*En el punto 2.2*

Mensaje de Error



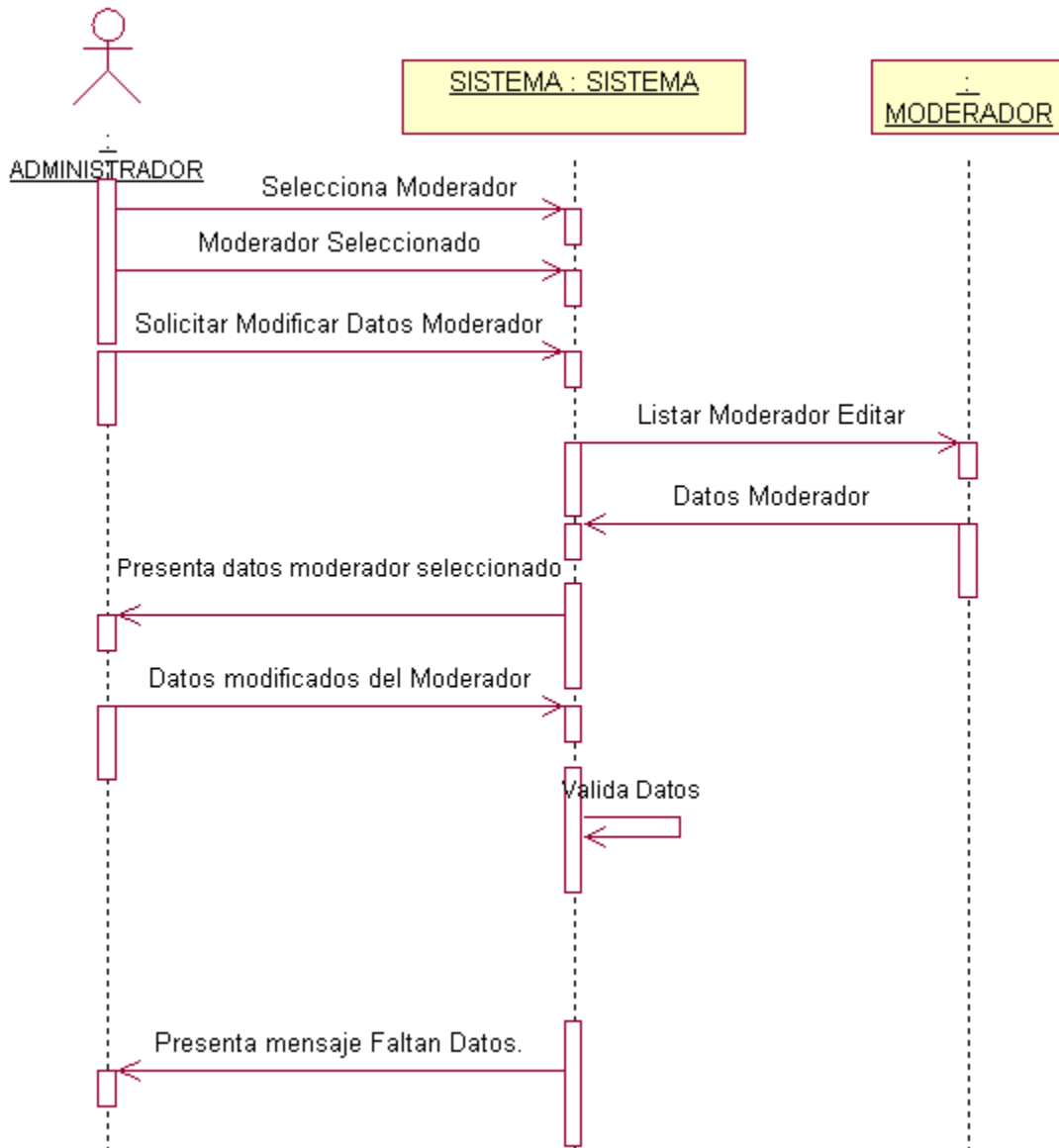
**Diagrama de Secuencia Alternativo 0.1 Insertar Moderadores**

*En el punto 4.6*

El sistema comprueba que todos los datos obligatorios no estén vacíos, caso contrario el sistema generará un mensaje de error comunicando que faltan datos necesarios del moderador.

*En el punto 4.6*

Si se ha generado mensaje de error.



**Diagrama de Secuencia Alternativo 0.2 Modificar Moderadores**

**Precondiciones**

- 3.1 El Administrador debe ingresar correctamente su login y password que le acrediten como administrador sus permisos correspondientes en el sistema.
- 3.2 El Administrador debe haber seleccionado el link de “Moderadores” de su interfaz gráfica.

**Postcondiciones**

- 4.1 En caso de haberse dado de alta un nuevo moderador, los datos del moderador quedan almacenados en el documento XML.

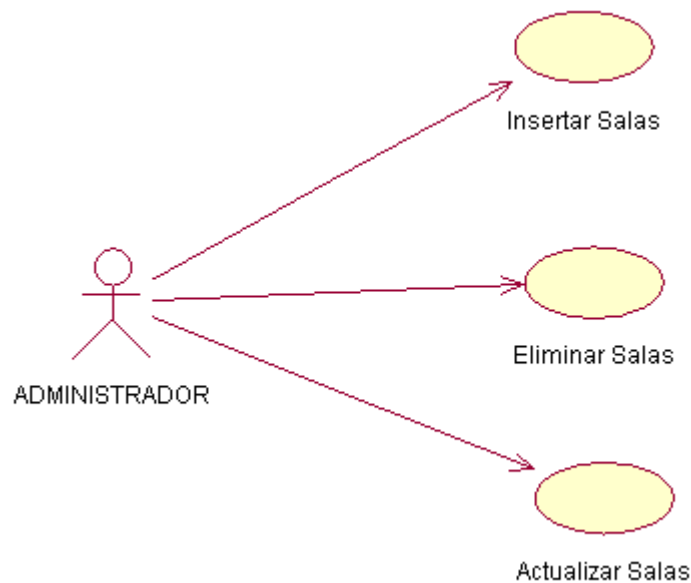


- 4.2 En caso de haberse realizado la eliminación de un moderador, el moderador queda eliminado completamente del documento XML.
- 4.3 En caso de haberse realizado una modificación de los datos de un moderador, quedan almacenados en el documento XML.

### **Especificación de casos de uso: Gestión de Salas**

#### **Descripción**

Este caso de uso resume la utilidad de alta, baja y modificación de los datos registrados de los Salas o Cuartos de Chat. El administrador podrá acceder a los datos de las Salas para realizar las siguientes operaciones: dar de alta, eliminar o modificar datos.

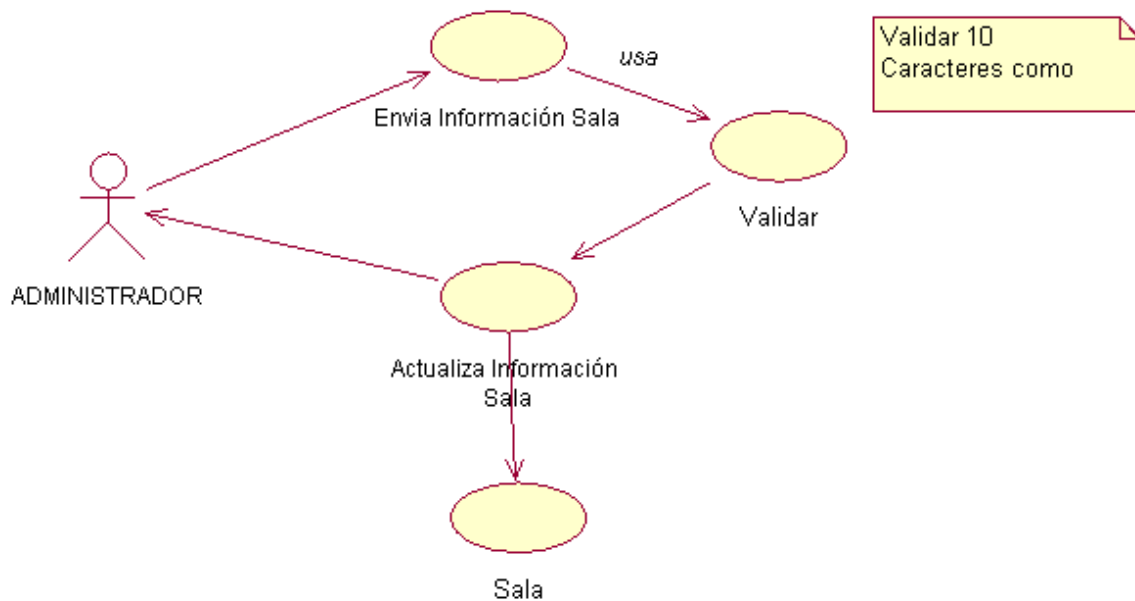


**Caso de Uso 0.6 Gestión Salas**

## Flujo de Eventos

### Flujo Básico

1. El Administrador puede seleccionar dar de alta una nueva sala, pasar al punto 2; dar de baja una sala, pasar al punto 3 y modificar datos de una sala, pasar al punto 4.
2. El Administrador solicita el alta de una nueva sala presionando el icono “Abrir una Sala”.



### Caso de Uso 0.7 Insertar Salas

El sistema muestra los campos de datos necesarios a introducir; los campos son: nombre, tema y debe seleccionar los siguientes campos: máximo de usuarios, historia, tiempo de actualización, mensajes privados, lenguaje, moderador.

- 2.1.1 El campo máximo de usuarios se selecciona de la lista que contiene las siguientes opciones:

- 1.- 5
- 2.- 10
- 3.- ...
- 4.- 200

Indica la cantidad de usuarios que serán permitidos en la Sala de Chat.

2.1.2 El campo Historia se selecciona de la lista que contiene las siguientes opciones:

- 1.- 20
- 2.- 30
- 3.- ...
- 4.- 150

Indica el tiempo que permanecerá activo los mensajes del Chat.

2.1.3 El campo Tiempo de Actualización se selecciona de la lista que contiene las siguientes opciones:

- 1.- 1 segundo
- 2.- ...
- 3.- 50 segundos

Indica el tiempo de refresco de la Página.

2.1.4 El campo Mensajes Privados se selecciona de la lista que contiene las siguientes opciones:

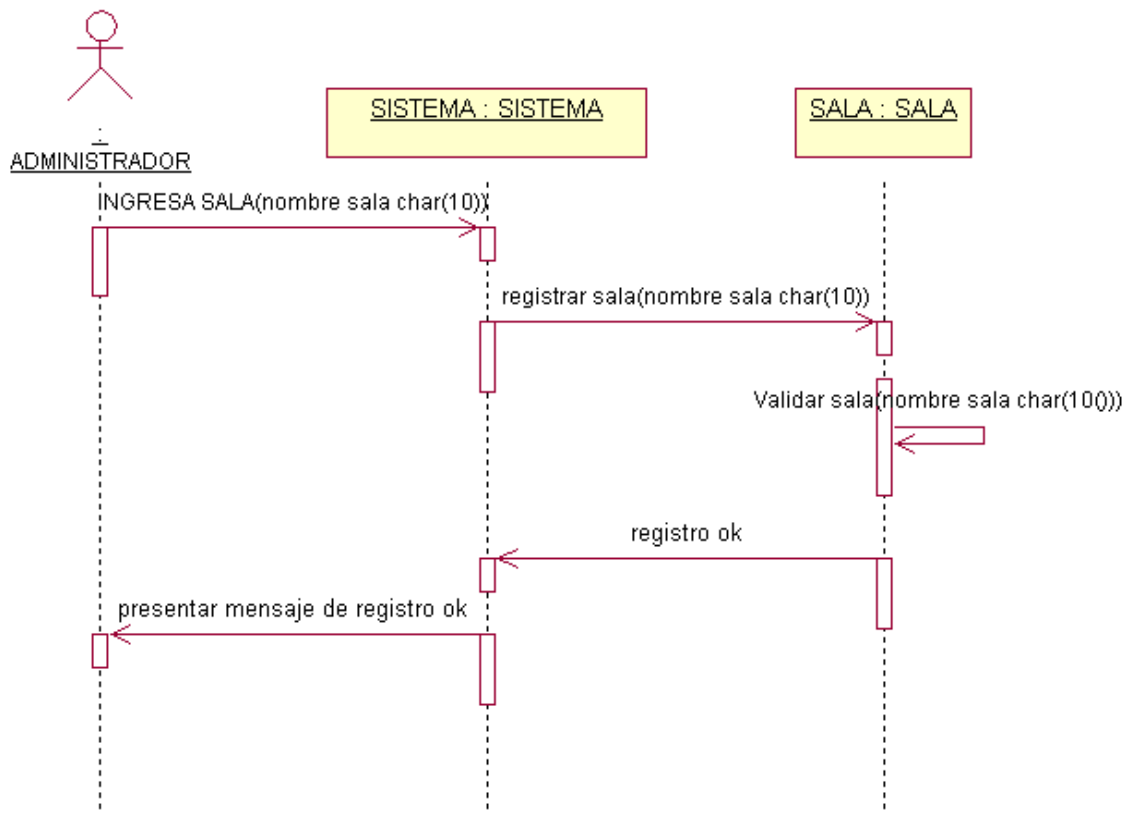
- 1.- si.- en el caso que el moderador decida habilitar esta opción
- 2.- no.- en el caso que el moderador no habilite esta opción

2.1.5 El campo Lenguaje se selecciona de la lista que contiene las siguientes opciones:

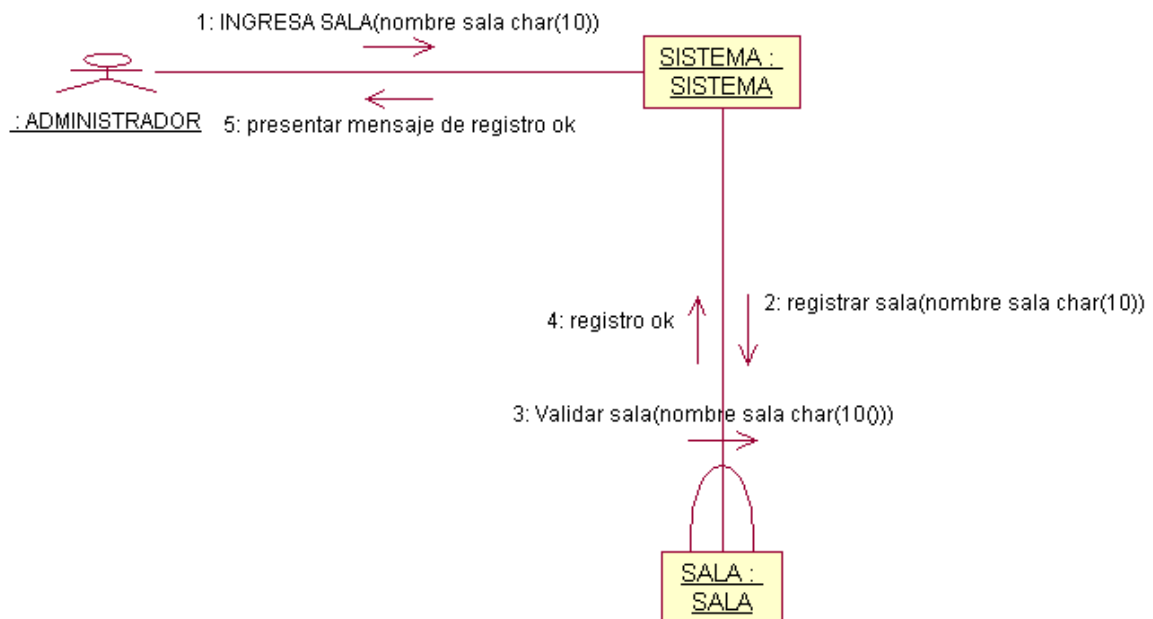
- 1.- Ingles.- solo debería tener esta opción

2.1.6 El campo moderador se selecciona de una lista que contiene los moderadores registradas en el sistema.

2.2 El Administrador pulsa el icono “guardar”. Ver punto 4.1.

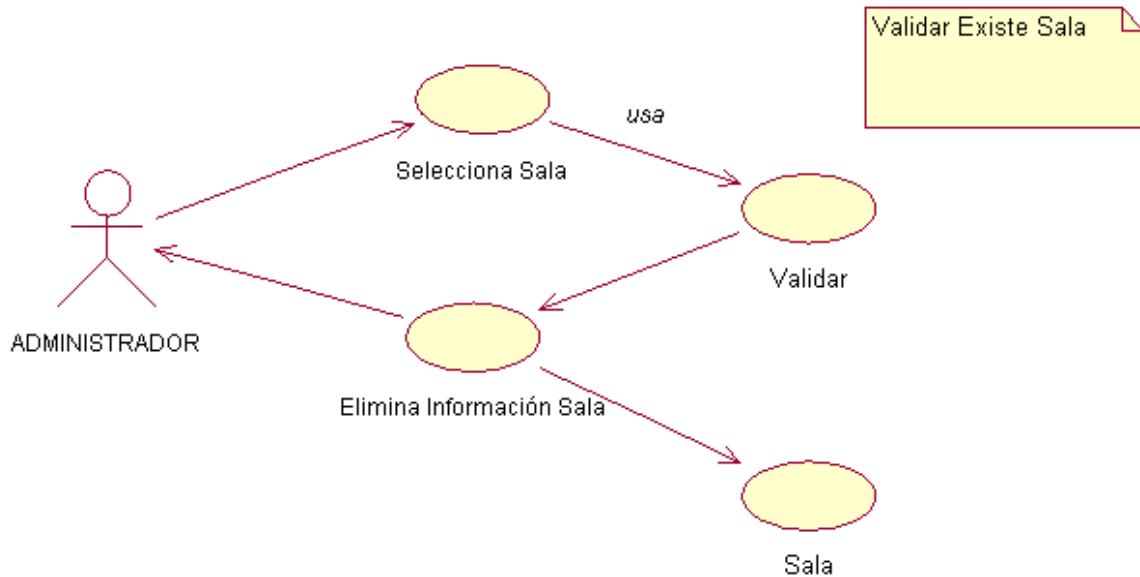


**Diagrama de Secuencia 0.4 Insertar Salas**



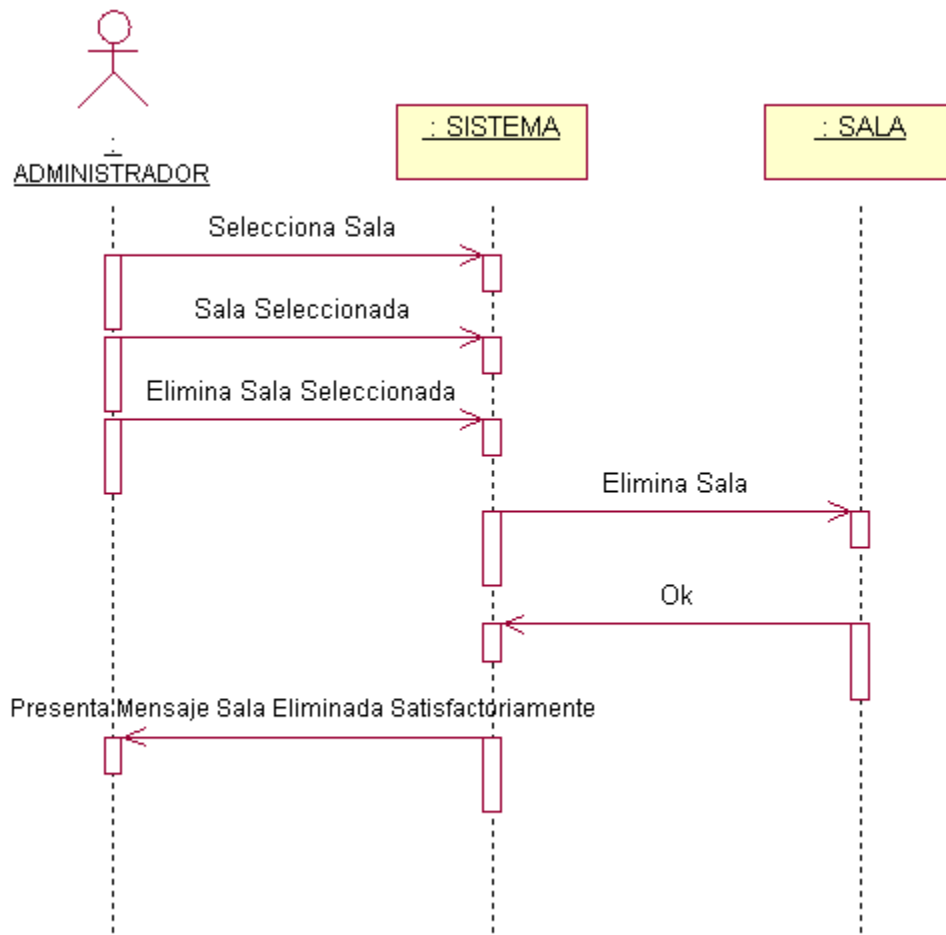
**Diagrama de Colaboración 0.4 Insertar Salas**

3 El Administrador solicita la eliminar una sala.

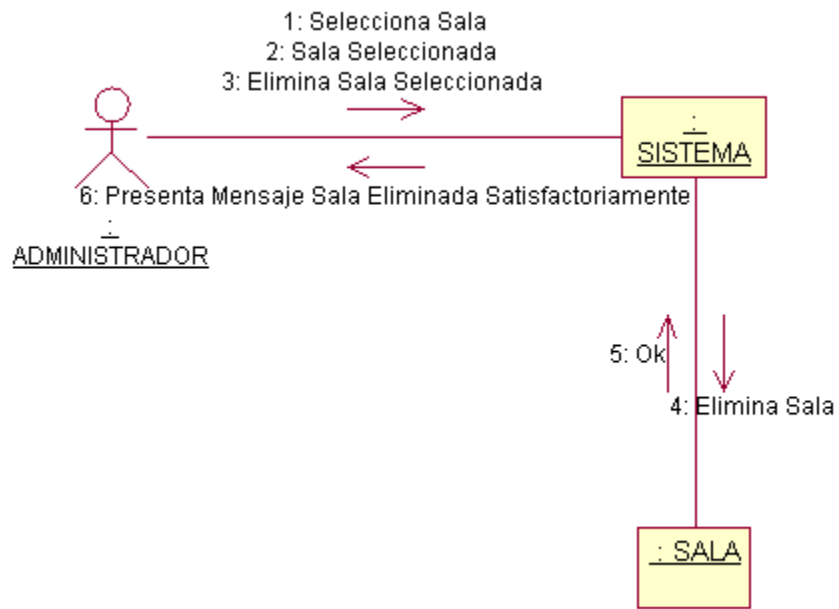


### Caso de Uso 0.8 Eliminar Sala

- 3.1 El Administrador selecciona la sala a ser eliminada del listado de salas que se presenta en la interfaz gráfica de Gestión de Salas.
- 3.2 El sistema presenta el sala seleccionada.
- 3.3 El Administrador pulsa el icono “Cerrar” de su interfaz gráfica.
- 3.4 El sistema presenta el mensaje de Sala Fue Removida Satisfactoriamente. Ver punto 4.2.

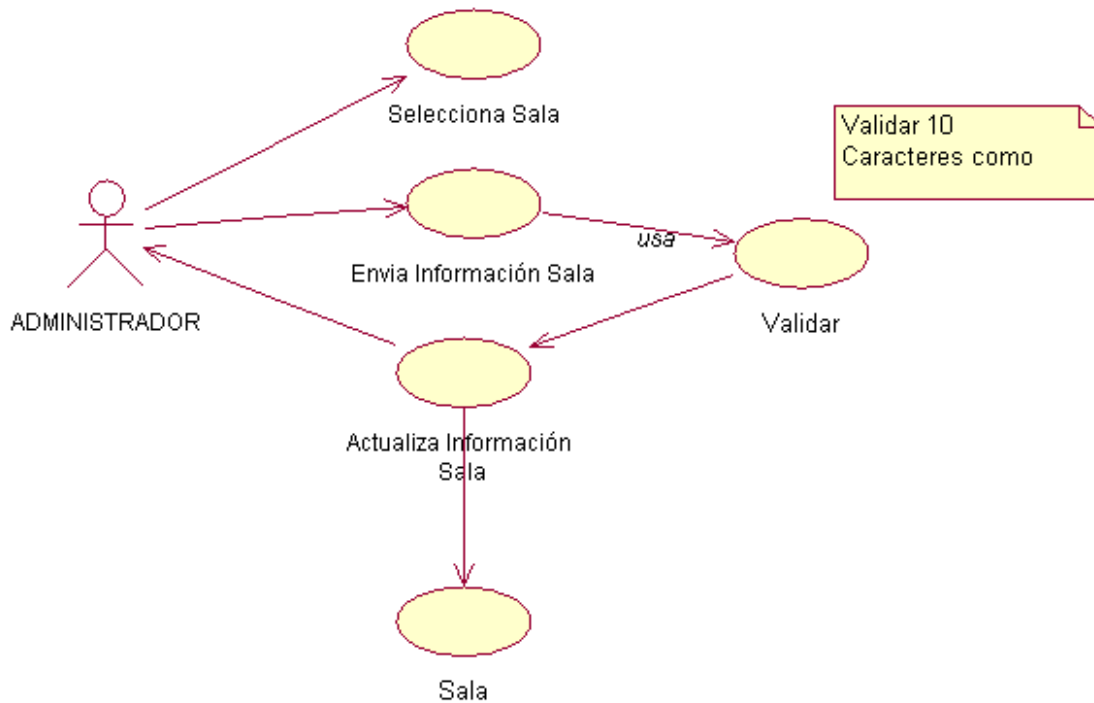


**Diagrama de Secuencia 0.5 Eliminar Salas**



**Diagrama de Colaboración.0.5 Eliminar Salas**

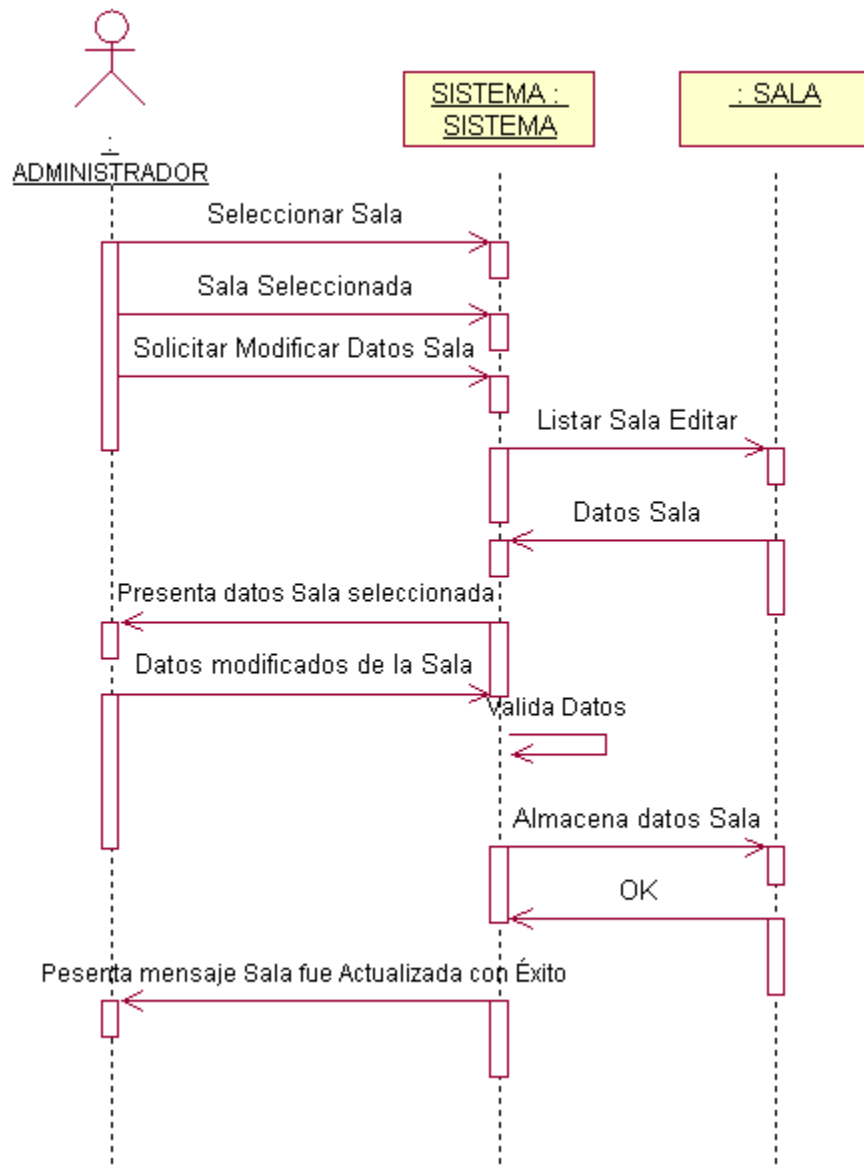
4 El Administrador solicita la modificación de los datos de una sala.



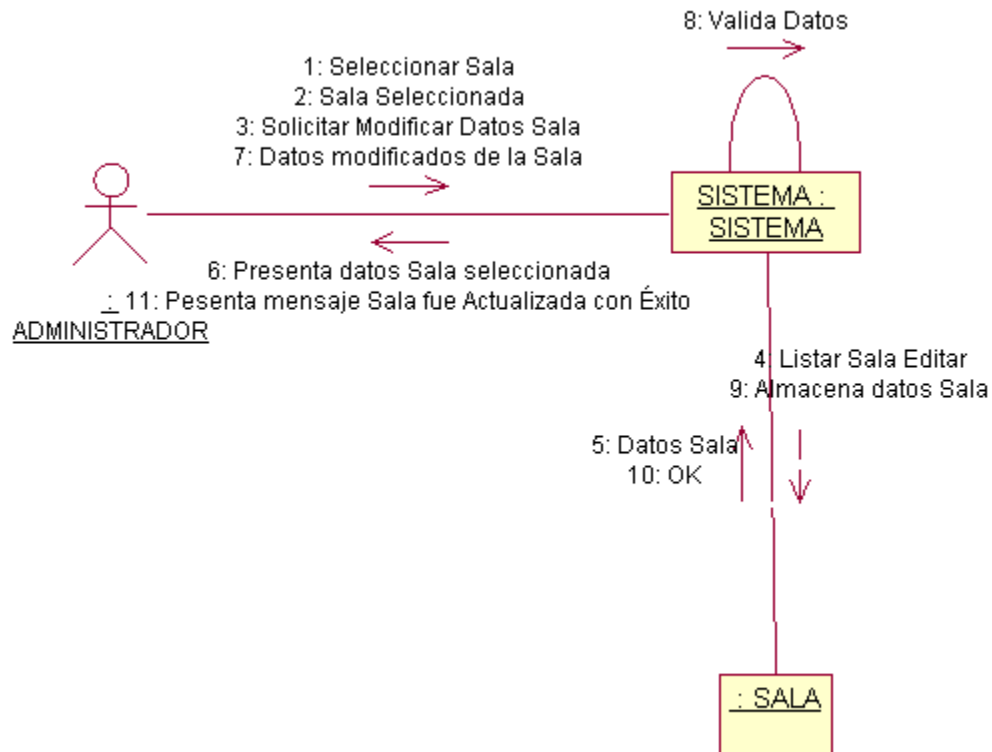
**Caso de Uso 0.9 Modificar Usuarios**



- 4.1 El Administrador selecciona la sala a ser modificada del listado de salas que se presenta en la interfaz gráfica de Gestión de Salas.
- 4.2 El sistema presenta la sala seleccionada.
- 4.3 El sistema muestra los datos de la sala que se ha solicitado para la modificación.
- 4.4 El Administrador puede modificar los siguientes campos: descripción, máximo de usuarios, historia, tiempo actualización, mensajes privados, lenguaje, moderador.
- 4.5 El Administrador puede solicitar almacenar los datos modificados pulsando el botón “Actualizar” de la interfaz gráfica. Ver punto 4.3.



**Diagrama de Secuencia 0.6 Modificar Salas**



**Diagrama de Colaboración.0.6 Modificar Salas**

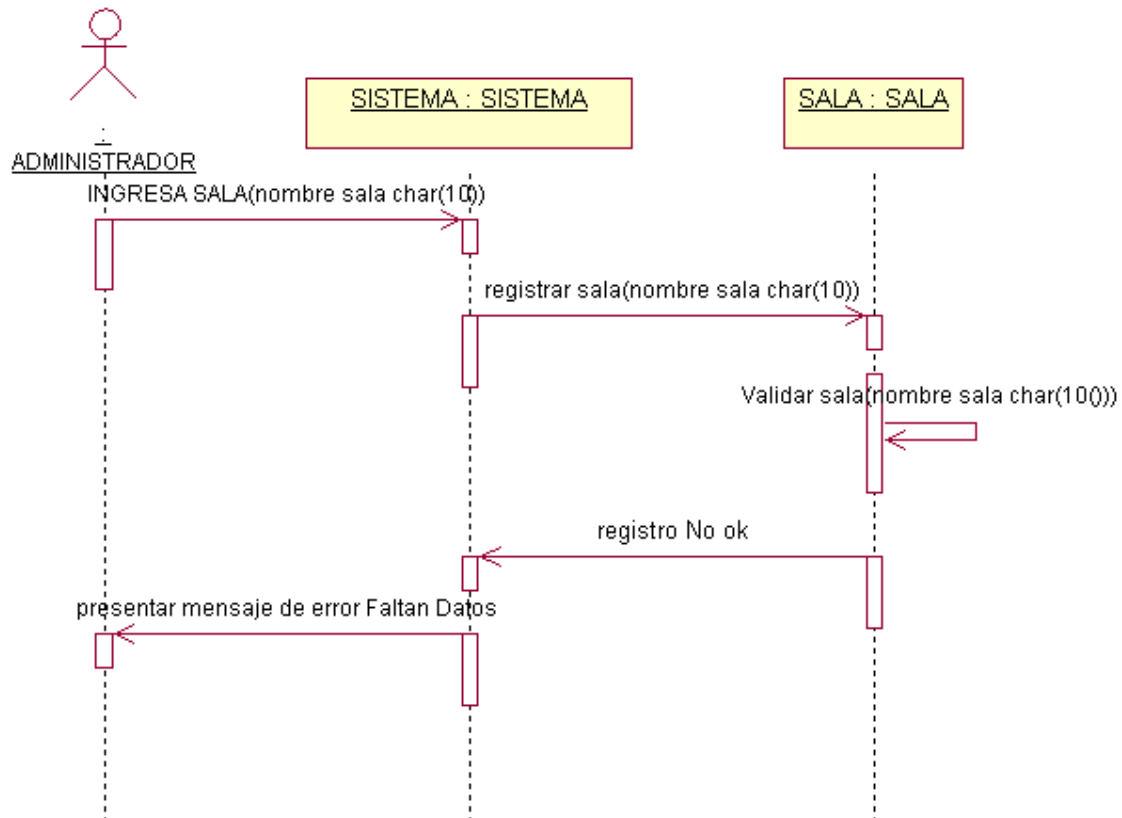
### Flujos Alternativos

*En el punto 2.2*

El sistema comprueba que se han introducido todos los datos obligatorios y caso contrario el sistema generará un mensaje de error comunicando que faltan.

*En el punto 2.2*

Si se ha generado mensaje de error, el sistema vuelve a mostrar la interfaz gráfica de alta de salas.



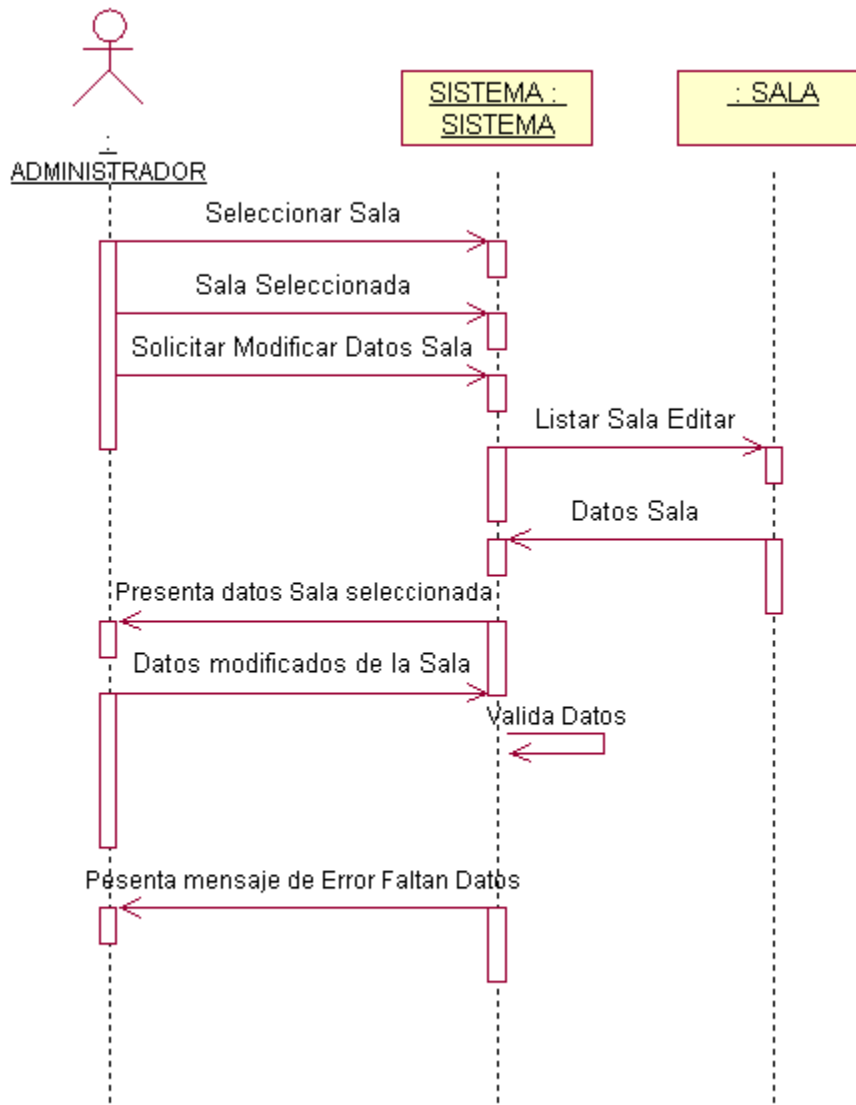
**Diagrama de Secuencia Alternativo 0.3 Insertar Usuarios**

*En el punto 4.6*

El sistema comprueba que todos los campos de datos obligatorios no estén vacíos, caso contrario el sistema generará un mensaje de error comunicando que faltan datos necesarios de la sala.

*En el punto 4.6*

Si se ha generado mensaje de error, el sistema muestra la interfaz gráfica de modificar salas.



**Diagrama de Secuencia Alternativo 0.4 Modificar Salas**

**Precondiciones**

- 3.1 El Administrador debe ingresar correctamente su login y password que le acrediten como administrador sus permisos correspondientes en el sistema.
- 3.2 El Administrador ha seleccionado el link de “Salas” de su interfaz gráfica

**Postcondiciones**

- 4.1 En caso de haberse dado de alta un nuevo sala, los datos de la sala quedan almacenados en el docuemntno XML.
- 4.2 En caso de haberse realizado la eliminación de una sala, la sala queda eliminado del docuemntno XML.
- 4.3 En caso de haberse realizado una modificación de los datos de una sala, quedan almacenados en el docuemnto XML.

## ***FASE DE ELABORACION***

### **MODELO DE ANALISIS Y DISEÑO**

#### ***Construcción del Modelo de Datos del Sistema***

##### Introducción

En este punto se recogen todas las entidades a ser procesadas o consultadas por el sistema y todas las relaciones existentes entre las entidades, de manera que puedan acceder a la información proporcionada por el sistema.

##### Información del modelo de Datos

**Nombre del Proyecto:** SIRC CHAT  
**Código del Proyecto:** SIRC CHAT  
**Nombre:** Modelo Lógico de SIRC CHAT  
**Descripción:** Modelo Lógico de SIRC CHAT  
**Autor:** Integrantes Proyecto SIRC CHAT  
**Versión:** 1.0

## Diagrama del Modelo Lógico

## Descripción de Datos de las entidades

### Lista de Datos

Nombre	Código	Dominio	Tip.
nombre_mod	NOMBRE_MOD	nombre	A20
email_mod	EMAIL_MOD	email	A20
contraseña_mod	CONTRASENIA_MOD	password	A10
nombre_sala	NOMBRE_SALA	nombre	A20
tema_sala	TEMA_SALA	observaciones	A40
maxusr_sala	MAXUSR_SALA	descripcion	I
historia_sala	HISTORIA_SALA	descripcion	I
tiempoact_sala	TIEMPOACT_SALA	descripcion	I
mensajespri_sala	MENSAJESPRI_SALA	descripcion	A4
lenguaje_sala	LENGUAJE_SALA	descripcion	A10
nombre_mod	NOMBRE_MOD	nombre	A20

## Lista de Datos por Entidades

### ENTIDAD USUARIOSPERFIL

#### Dato serial\_usrper

**Nombre:** serial\_usrper  
**Código:** SERIAL\_USRPER  
**Descripción:** Identificador de la entidad usuariosperfil, la información que tiene este atributo es autoincremental es decir es asignado automáticamente por la base de datos.  
**Dominio:** serial  
**Tipo de dato:** Entero Autoincremental

## Descripción de Dominios

### Lista de dominios

Nombre	Código	Tipo de dato
nombre	NOMBRE	A20
descripcion	DESCRIPCION	A70
observaciones	OBSERVACIONES	I
email	EMAIL	A20
login	LOGIN	A10
password	PASSWORD	A10

## Descripción de las entidades

### Lista de Entidades

Nombre	Código
alumno	ALUMNO
moderador	MODERADOR
salas	SALAS

### Entidad alumno



**Nombre:** alumno  
**Código:** ALUMNO  
**Descripción:** En esta entidad se almacena la información correspondiente a los alumnos de la Institución

#### Lista de atributos

Nombre	Código	Tipo	I	M
nombre_alu	NOMBRE_ALU	A70	No	Si

#### Lista de relaciones

Ninguna

#### Entidad moderador

**Nombre:** moderador  
**Código:** MODERADOR  
**Descripción:** En esta entidad se almacena la información correspondiente a los profesores o moderadores que laboran en la Institución

#### Lista de atributos

Nombre	Código	Tipo	I	M
nombre_mod	NOMBRE_MOD	A20	Si	Si
email_mod	EMAIL_MOD	A20	No	Si
contrasenia_mod	CONTRASENIA_MOD	A15	No	Si

#### Lista de relaciones

Relación	Entidad 2	Entidad 1	Ent1- Ent 2 Card.	Ent2- Ent 1 Card.
nombre_mod	profesor	Sala	0,n	0,1

## *Construcción del Modelo Físico del Sistema*

### Introducción

En este modelo se recogen todas las tablas, índices, claves primarias y foráneas y todas las relaciones existentes entre las tablas.

### Información del Modelo Físico del Sistema

<b>Nombre del Proyecto:</b>	SEDNA
<b>Código del Proyecto:</b>	SEDNA
<b>Nombre:</b>	Modelo Físico de SEDNA
<b>Descripción:</b>	Modelo Físico de SEDNA
<b>Autor:</b>	Integrantes Proyecto Sedna
<b>Versión:</b>	1.0

### Diagrama del Modelo Físico

## Lista de Dominios

Nombre	Código	Tipo de dato
nombre	NOMBRE	A20
descripcion	DESCRIPCION	A20
observaciones	OBSERVACIONES	I
login	LOGIN	A20
password	PASSWORD	A20

## Dominio nombre

**Nombre:** nombre  
**Código:** NOMBRE  
**Tipo de dato:** caracter  
**Longitud:** 20

Nombre	Código	Tabla
nombre_sala	NOMBRE_SALA	sala
nombre_mod	NOMBRE_MOD	moderador
nombre_pro	NOMBRE_PRO	profesor
nombre_alu	NOMBRE_ALU	alumno

## Descripción de las Tablas

## Lista de Tablas

Nombre	Código
--------	--------

Profesores	PROFESORES
Alumnos	ALUMNOS
Salas	SALAS

### Tabla alumno

**Nombre:** alumno  
**Código:** ALUMNO  
**Descripción:** En esta entidad se almacena la información correspondiente a los alumnos de la Institución.

### Lista de índices y claves primarias y foráneas

Nombre	Código	Unico	Clave Primaria	Clave Foránea
nombre_alu_fx	ALUMNO_PK	X		

### Tabla moderador

**Nombre:** moderador  
**Código:** MODERADOR  
**Descripción:** En esta entidad se almacena la información correspondiente a los moderadores que laboran en la Institución.

### Lista de índices y claves primarias y foráneas

Ninguna

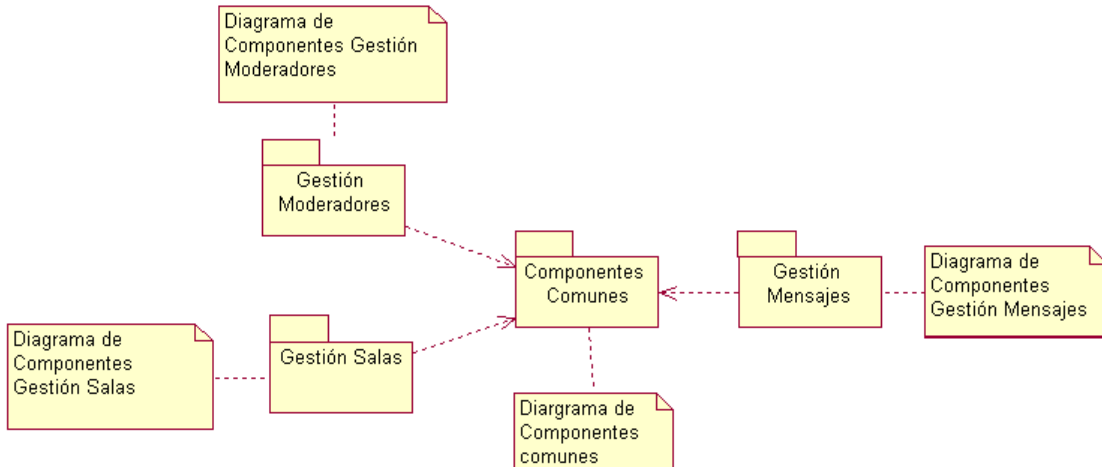
## **FASE DE CONSTRUCCION**

### **IMPLEMENTACION**

#### *Modelo de Implementación*

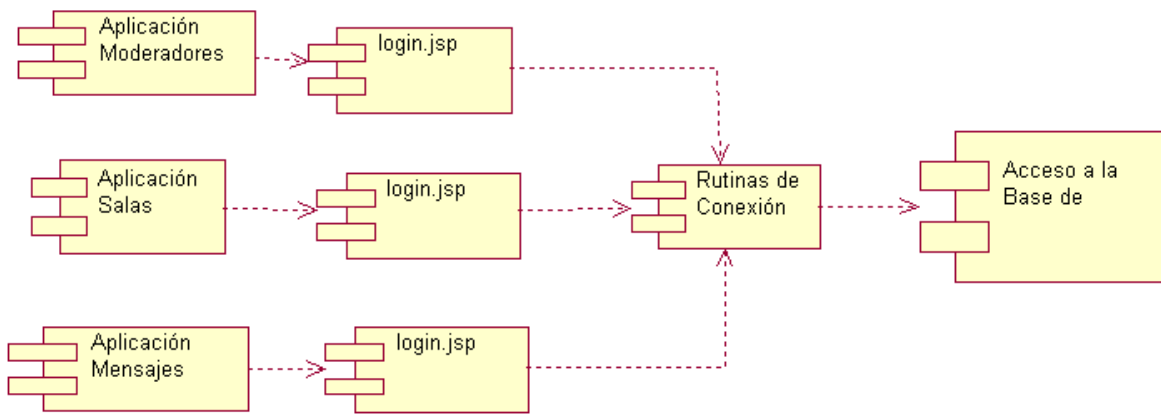
A continuación se presentan los diagramas de componentes y diagrama de despliegue del proyecto. En el primero de ellos se muestra la disposición de las partes integrantes de la aplicación y las dependencias entre los distintos módulos de la aplicación. En el segundo se muestra la representación de los distintos nodos repartidos en la institución que forman parte del sistema completo.

#### Diagrama Global de Paquetes



**Diagrama Global de Paquetes 0.1**

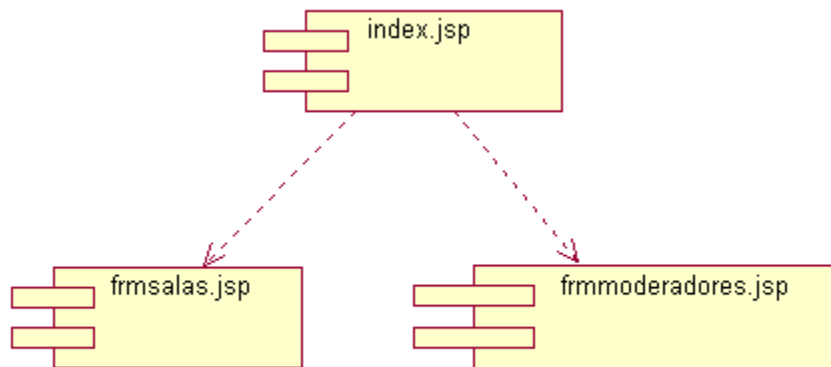
#### Diagrama de Componentes Comunes



**Diagrama de Componentes 0.1**

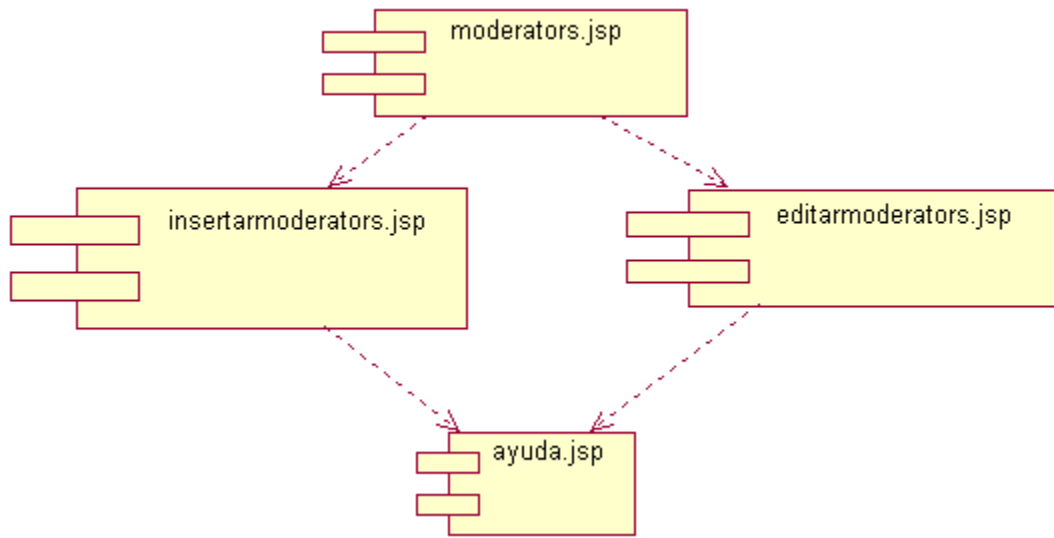
## Diagramas de Componentes Específicos

### *Diagrama de Componentes Administración*



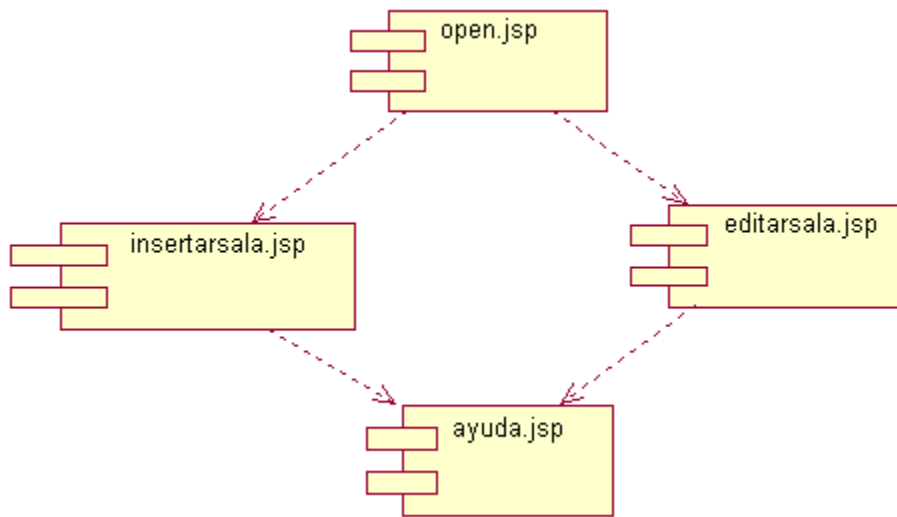
**Diagrama de Componentes 0.2 Gestión Administración**

### *Diagrama de Componentes frmprofesores*



**Diagrama de Componentes 0.3 moderaqdores**

*Diagrama de Componentes salas*



**Diagrama de Componentes 0.4 salas**

**PRUEBAS**

## *Plan de Pruebas*

### Introducción

#### *Propósito*

El propósito del Plan de Pruebas es recoger toda la información necesaria para planear y controlar el esfuerzo de las pruebas dadas.

Este Plan de Pruebas para el Chat tiene los siguientes objetivos:

- Identificar las pruebas que se realizarán en el sistema.
- Identificar problemas en el funcionamiento del sistema.
- Establecer recursos requeridos para la realización de cada una de las pruebas.

#### *Alcance*

El Plan de Pruebas describe los niveles de comprobación del sistema; es decir, las pruebas de unidad e integración y los tipos de comprobación como la funcionalidad, utilidad, fiabilidad las mismas que serán dirigidas por este plan de prueba.

#### *Personas al que se dirige el plan*

Este Plan de Pruebas esta dirigido exclusivamente para la o las personas encargadas de la verificación funcional del sistema o para aquellas personas que vean en este documento una ayuda al uso impropio del sistema.

#### *Preparación del Plan de Pruebas*

La siguiente tabla que se presenta a continuación, permitirá determinar para cada requisito la característica a ser probada y los tipos de prueba que se emplearán.

<b>Requisito</b>	<b>Característica a probar</b>	<b>Tipos de prueba</b>
Gestión Moderadores	Crear moderador y almacenar datos. Crear moderador con campos obligatorios vacíos. Eliminar un moderador.	Pruebas de caja negra. Valores típicos de error Valores imposibles



	Modificar los datos de un moderador y actualizarlos. Modificar los datos de un moderador con campos obligatorios vacíos.	
Gestión Salas	Crear sala y almacenar datos. Crear sala con campos obligatorios vacíos. Eliminar una sala. Modificar los datos de una sala y actualizarlos. Modificar los datos de una sala con campos obligatorios vacíos.	Pruebas de caja negra. Valores típicos de error Valores imposibles
Gestión Alumno	Crear alumno y almacenar datos. Crear alumno con campos obligatorios vacíos.	Pruebas de caja negra. Valores típicos de error

**Tabla 0.20 Preparación del Plan de Pruebas**

## Pruebas planeadas

Se ha diseñado un conjunto de pruebas para comprobar el cumplimiento de las especificaciones de requisitos.

Se van a desarrollar las siguientes pruebas:

### *Pruebas Unitarias*

El objetivo de esta prueba es verificar la lógica y las funciones de cada uno de los módulos, comprobando la integridad de los datos como también de la de la información almacenada en el documento XML.

Las Pruebas Unitarias del plan de pruebas debe enfocarse en cualquier requisito para probar y puede remontarse en los casos de uso o funciones de negocio y reglas del negocio. Las metas de estas pruebas son verificar la aceptación de los datos apropiados, el procesamiento, recuperación, y la aplicación apropiada de las reglas del negocio. Este tipo de comprobación está basado en las técnicas de caja negra; que verifican la aplicación y sus procesos interiores actuando recíprocamente con la aplicación a través la Interfaz Gráfica de Usuario (GUI) y analizan el rendimiento o resultado.

La siguiente tabla identifica un contorno de la comprobación recomendada para cada aplicación.

Objetivo de la técnica:

La funcionalidad de la Comprobación del funcionamiento, incluye la navegación, la entrada de los datos, procesamientos, y recuperación para observar las conductas entre ellos.

Técnica:

- Ejecutar cada caso de uso en su propia interfaz , de manera individual cada flujo de eventos de cada caso de uso así como cada función., usando datos válidos e inválidos, para verificar que:
  - Los resultados esperados ocurren cuando se usan datos válidos
  - Los mensaje de error o alerta apropiados se despliegan cuando se usan datos inválidos.
- Que cada regla del negocio se aplica propiamente

### *Prueba de Integración de Componentes*

El objetivo de esta prueba es comprobar el correcto funcionamiento de la relación que existe entre las interfaces de cada uno de los componentes.

- Comprobación del Ciclo del Negocio

La comprobación del Ciclo del Negocio deben emular las actividades realizadas en el Chat, en el tiempo actual. Debe por ejemplo, ejecutarse las transacciones y actividades que ocurrirían durante una conversación normal en una sala.

Objetivo de la técnica:

El objetivo es probar y respaldar que los procesos se realizan según el modelo del negocio.

Técnica:

Se simularán varios ciclos del negocio.

Criterios de Exito:

La técnica apoya la comprobación de todos los ciclos del negocio.

### *Prueba de Integración de Componentes*

Las pruebas de seguridades y control de acceso enfocan dos áreas importantes de seguridad:

- Seguridad a nivel de aplicación, incluso acceso a los Datos o Funciones de Negocio
- Seguridad a nivel del sistema, incluyendo logoneo o accesos remotos al sistema.

Basados en la seguridad deseada, los niveles de seguridad en la aplicación-nivelada asegura la restricción de actores a funciones específicas o casos de uso, y la limitación a los datos disponibles a ellos.

La seguridad a nivel del sistema se asegura cuando los usuarios que acceden al sistema son capaces de acceder a las aplicaciones solo a través de las entradas apropiadas.

Objetivo de la técnica:

Bajo las siguientes condiciones se pueden observar:

- La Seguridad a nivel de aplicación: un actor puede acceder solo a las funciones o datos a los que su tipo de usuario tiene permiso.
- La Seguridad a nivel de sistema: solo los actores con acceso al sistema pueden acceder a la aplicación .El objetivo es probar y respaldar que los procesos se realizan según el modelo del negocio.

Técnica:

La Seguridad a nivel de Aplicación: Identificar y listar cada tipo de usuario y las funciones o datos al que cada tipo tiene permiso. Para:

- Crear las pruebas para cada tipo del usuario.

Criterios de Exito:

La técnica apoya la comprobación para cada tipo del actor y pueden probarse las funciones apropiadas o datos afectados por escenas de seguridad.

Consideraciones Especiales:

El Acceso al sistema debe ser revisado y discutido por el administrador de la red del sistema.

## Especificación de la Plantilla para los Casos de Uso

### *Descripción*

Resumen de la plantilla para el caso de prueba.

### *Condiciones de Ejecución*

Especifica los usuarios que pueden realizar el caso de prueba.

### *Crterios de Entrada*

Especifica el criterio que se usará para determinar si la ejecución de la prueba puede empezar.

### *Crterios de Salida*

Especifica el criterio que se usará para determinar si la ejecución de la prueba está completa o su ejecución no proporciona beneficio.

### *Resultado Esperado*

Proporciona un contorno breve de la forma y contenido de los resúmenes de evaluación de la prueba.

### *Evaluación de la Prueba*

Proporciona un contorno breve de la forma y contenido de los informes que miden la magnitud de la prueba.

### Recursos Requeridos

#### *Hardware Base del Sistema*

La siguiente tabla muestra los recursos del sistema para realizar el Plan de Pruebas.

<b>Recurso</b>	<b>Cantidad</b>	<b>Nombre y Tipo</b>
Servidor de Base de Datos	1	
- Red o Subred		192.168.1.101
- Nombre del Servidor		Server chat
- Nombre del documento XML		jchatboxusers.dtd
PCs clientes	0-n	
- Requerimientos especiales de Configuración		Requerimiento: Internet Explorer 5.5 o superior

**Tabla 0.21 Requisitos de Hardware**

## *Software Base del Sistema*

La siguiente tabla muestra los recursos del sistema para realizar el Plan de Pruebas.

<b>Nombre del Elemento Software</b>	<b>Tipo y otras Notas</b>
Windows XP	Sistema Operativo
Internet Explorer 5.5 o superior	Explorador de Internet

**Tabla 0.22 Requisitos de Software**

## ***FASE DE TRANSICION***

### **MANUAL DE INSTALACION**

Este documento incluye los pasos para la instalación del producto software.  
Debido a la extensión de este documento referirse a los Anexos.

### **MANUAL DE USUARIO**

Este documento explica el correcto manejo del sistema.  
Debido a la extensión de este documento referirse a los Anexos.

# **CAPITULO 5**

## **CONCLUSIONES Y RECOMENDACIONES**

### ***CONCLUSIONES***

- Desde el inicio de este proyecto de investigación, y hasta la actualidad lo que mas se ha notado ha notado en la tecnología de Web Services, es la falta de madurez de las herramientas libres disponibles actualmente, en procesos de automatización de tareas para el desarrollo de Web Services. La mayoría tiene un número reducido de funcionalidades, por ejemplo: el IBM WSDK 3.1, que incluye un generador de archivos WSDL, sólo permite generarlos a partir de un JavaBean que utilice tipos básicos, lo que restringe fuertemente el tipo de clases que pueden ser publicadas en forma automática como Web Services con esta herramienta.
- Durante todo el proceso se trabajo con algunas herramientas como la herramienta WASP con Netbeans, Axis SOAP, Apache SOAP v2.3.1 y WSIF stands for the Web Services Invocation Framework resultaron ser inmaduras al no disponer de información y la poca que existe no funciona
- La documentación en muchos casos es escasa o nula, por ejemplo acerca de JWSP, WASP los demos y ejemplos que acompañan al software normalmente son básicos, y no ilustran problemas reales para soluciones corporativas, ocasionando retrasos no planificados en el desarrollo del proyecto.
- Los Manuales en la mayoría de casos traen captura de pantallas demostrando que funcionan pero ya en la practica siguiendo todos los pasos se presenta problemas cosa que demuestra que las herramientas estan todavía inmaduras.
- Cada servicio de Web tiene su casa dónde vive. Ésta es su identidad. Su dirección se define por un URI.

- Java es un lenguaje de programación ideal para chats, permite a tus páginas Web contener un código que se ejecuta desde el browser, es independiente del sistema operativo. Tardan un poco en cargar pero tienen una gran ventaja, no necesitas más que tu explorador para usarlos. Funcionan con los navegadores Netscape 2.x o superior o Explorer 3.x o superior.
- Para el desarrollo de aplicaciones o servicios web usando la tecnología JWSDP es necesario tener en cuenta que ahora las aplicaciones publicadas en un servidor web deben tener una jerarquía de directorios que corresponde a la estructura que conforma una aplicación web.

## **RECOMENDACIONES**

- Antes de comenzar una investigación estudiar bien la herramienta JAVA a utilizar por ejemplo: en el caso que no le permita cumplir con algún requisito le fuerza a cambiar de herramienta ocasionando pérdida de tiempo.
- Realizar pruebas al producto software en cada una de las iteraciones.
- Utilizar páginas jsp en lo posible porque son más entendibles y permiten la modificación rápida



## **REFERENCIAS BIBLIOGRAFICAS**

- **Java XML Programing with Servlets and JSP**  
Alexander Nakhimovsky and Tom Myers.
- **El Proceso Unificado de Desarrollo de Software**  
Ivar Jacobson, Grady Booch, James Rumbaugh.
- **Rational Unified Process,**  
Best Practices for Software Development Teams  
A Rational Software Corporation White Paper

## **DIRECCIONES WEB**

- <http://iaaa.cps.unizar.es/docencia/SW.html>  
Web Services
- <http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>  
Wasp
- <http://www.cica.es/formacion/JavaTut/>  
Manual Java Básico
- <http://www.adictosaltrabajo.com>  
Java Web Services y otros.
- <http://www.fisica.uson.mx/carlos/WebServices/WSRevolution.htm>  
Web Services
- [http://glud.udistrital.edu.co/glud/areas/doc/articulos/1\\_articulo\\_ws/servicios-web.html](http://glud.udistrital.edu.co/glud/areas/doc/articulos/1_articulo_ws/servicios-web.html)  
Servicios Web
- <http://revista.robotiker.com/articulos/articulo62/pagina1.jsp>  
Web Services
- <http://web-services.bankhacker.com/>  
Web Services
- [http://www.uco.es/ccs/sistemas/doc\\_soft/java/jwsdp-1\\_0-  
ea2/docs/tutorial/doc/JAXRPC5.html](http://www.uco.es/ccs/sistemas/doc_soft/java/jwsdp-1_0-<br/>ea2/docs/tutorial/doc/JAXRPC5.html)  
Manual de JWSP
- [http://www.programacion.com/java/tutorial/servic\\_web/5/](http://www.programacion.com/java/tutorial/servic_web/5/)  
Web Services
- <http://www.revista.unam.mx/vol.3/num1/art3/>  
Servicios Web
- <http://www.programacion.com/java/articulo/jms>  
Mensajería en Java
- [http://www.programacion.com/tutorial/servic\\_web/3/](http://www.programacion.com/tutorial/servic_web/3/)  
Web Services

- <http://www.educacioninformatica.com/modules/tutorials/tutoriales/chtml/texto20.html>  
Manual de Html
- [http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec\\_36.htm](http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec_36.htm)  
Manual de Html
- <http://www.oasis-open.org/cover/xml.html>  
Aplicaciones XML
- <http://face.el.uma.es/imasd/xml/xml.html>  
Aplicaciones XML
- <http://weezer.dragon-lance.net/articulos/xmlintro.html>  
Aplicaciones XML
- <http://www.desarrolloweb.com/articulos/831.php?manual=15>  
Manual de Jsp
- <http://www.todochat.com/intro.html>  
Manual de Chats

# ANEXOS

**Anexo**

**A**

# **MANUAL DE INSTALACIÓN**

Información CD #1

# **Chat Web Service en Java**

# **Manual de Instalación**

# Tabla de Contenidos

<b>1. Introducción</b> .....	<b>4</b>
<b>2. Recursos Requeridos</b> .....	<b>4</b>
2.1 Software Base del Sistema .....	4
2.2 Lenguajes Desarrollados .....	4
<b>3. Instalación</b> .....	<b>4</b>
3.1 Variable de Entorno .....	4
3.2 Instación Software .....	9
3.3 Instación Chat.....	15
3.4 Levantar Servidor .....	16
3.5 Parar el Servidor .....	17

---

**ESPEL**

---



# **MANUAL DE INSTALACION**

## **1. Introducción**

Para la instalación del Chat y al estar desarrollado en Java se necesitan ciertas herramientas utilizadas que debe instalarse en el servidor.

La funcionalidad del Chat dependerá de la instalación de las mismas razón por la que se debe seguir bien los pasos para no tener problemas luego con la funcionalidad del Chat

## **2. Recursos Requeridos**

### **2.1 .- Software Requerido**

Se necesita las siguientes herramientas:

- J2SDK (Maquina Virtual Java)
- JWSDP (Será nuestro servidor ya que incluye como una herramienta Tomcat)
- Internet Explorer 5.0 o Superior

### **2.2 .- Lenguajes Desarrollados**

Se ha desarrollado en los siguientes lenguajes

- XML
- Java

## **3. Instalación**

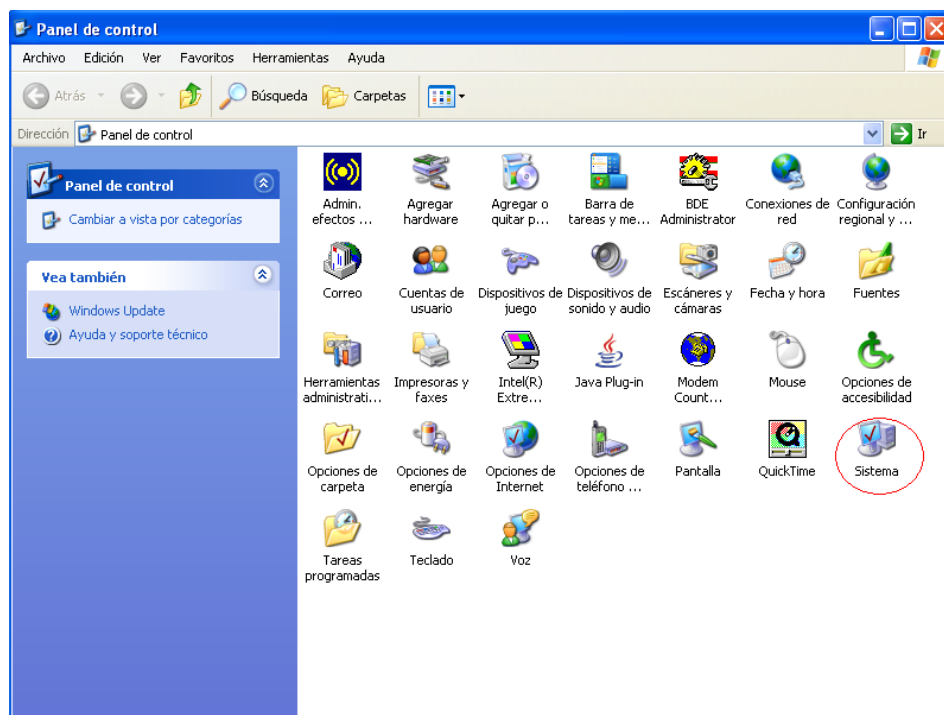
### **3.1 .- Variables de Entorno**

Para definir las variables de entorno seguimos los siguientes pasos:

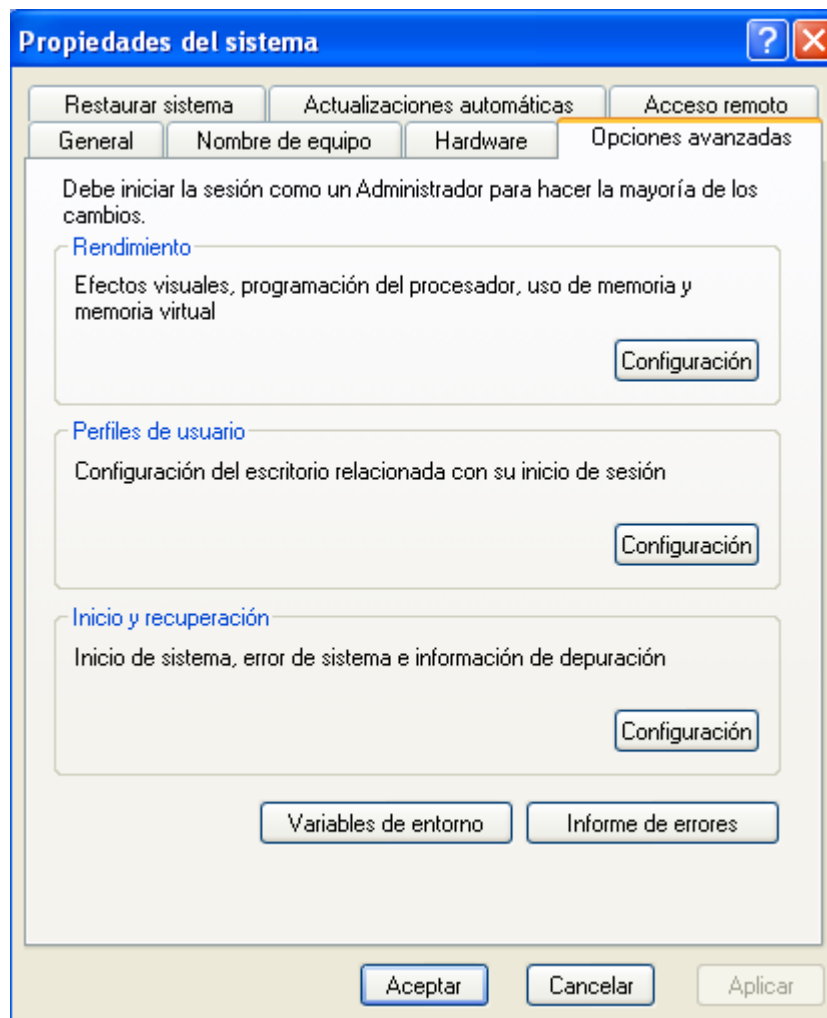
- En el Menú Inicio damos Click el la Opción Panel de Control



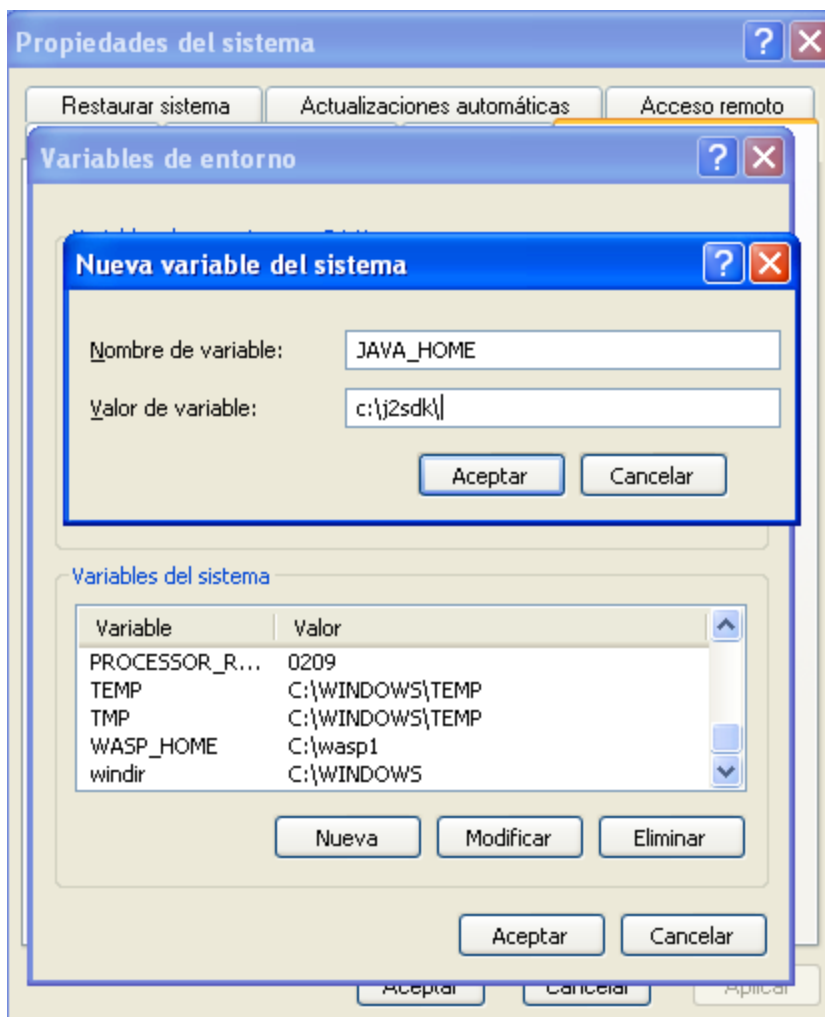
- Luego Damos Click en el icono de Sistema



- En opciones Avanzadas damos Click en variables de entorno

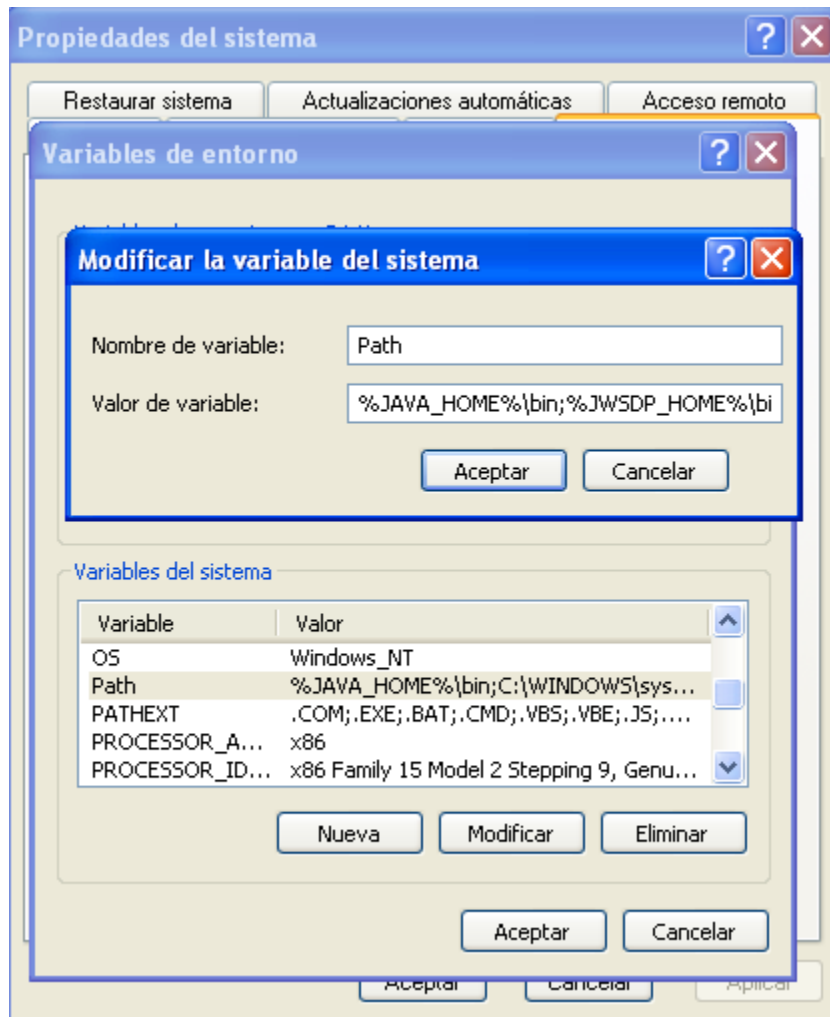


- Ingresamos las variables JAVA\_HOME y JWSDP\_HOME



- Añadimos a la variable de sistema Path lo siguiente y presionamos aceptar

`%JAVA_HOME%\bin;%JWSDP_HOME%\bin;`



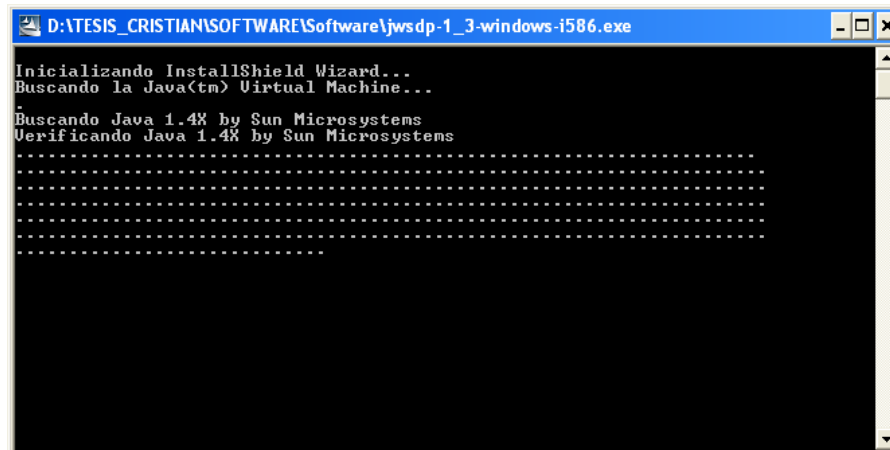
### 3.2 .- Instalación Software

#### 3.1 .- J2SDK

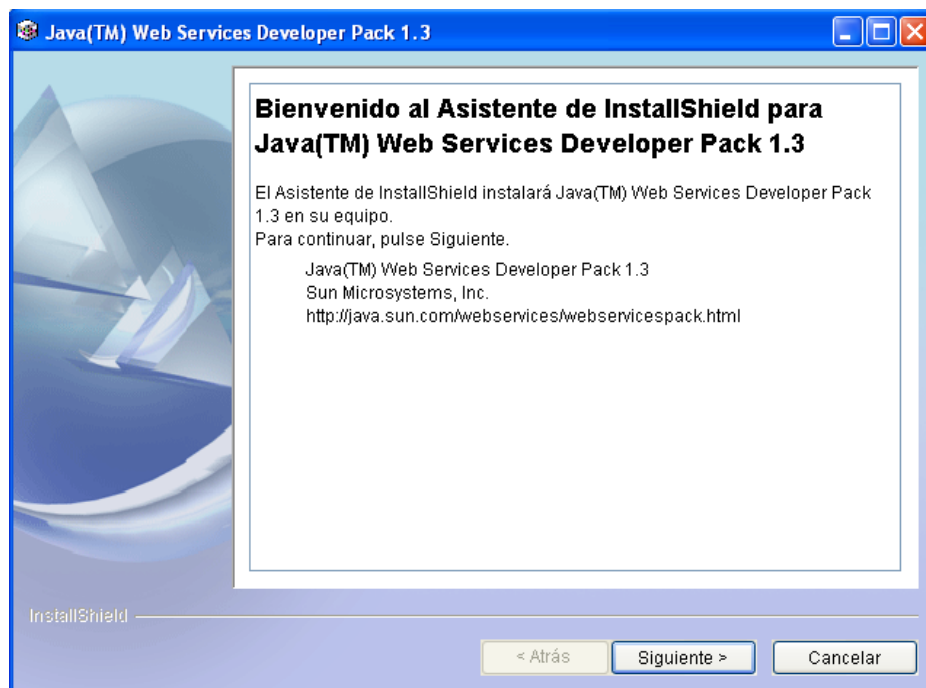
- Instalamos J2SDK

#### 3.2 JWSDP

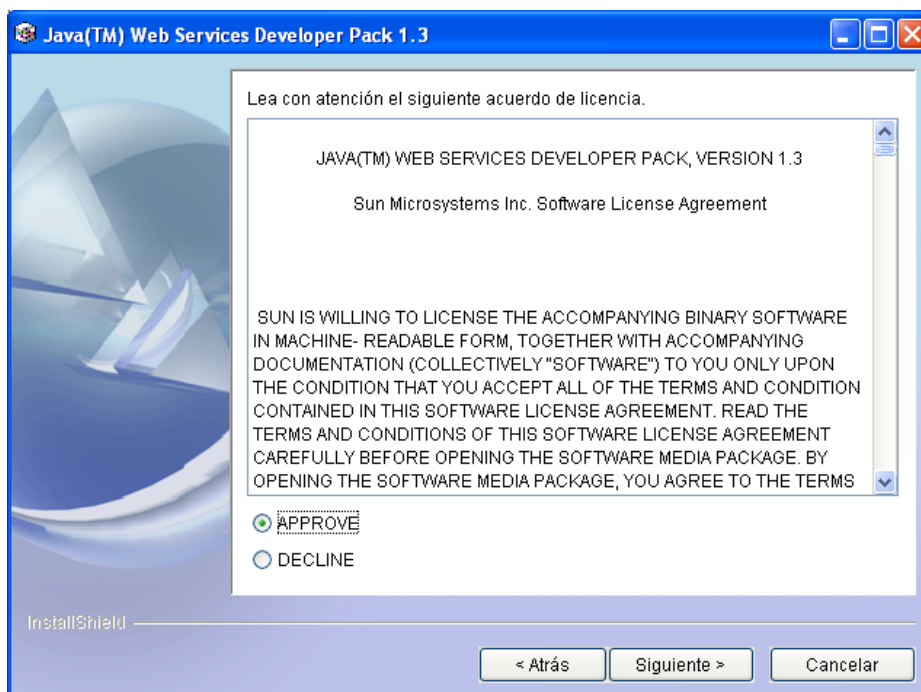
- Ejecutamos el instalador
- Verifica si esta instalado la maquina Virtual Java



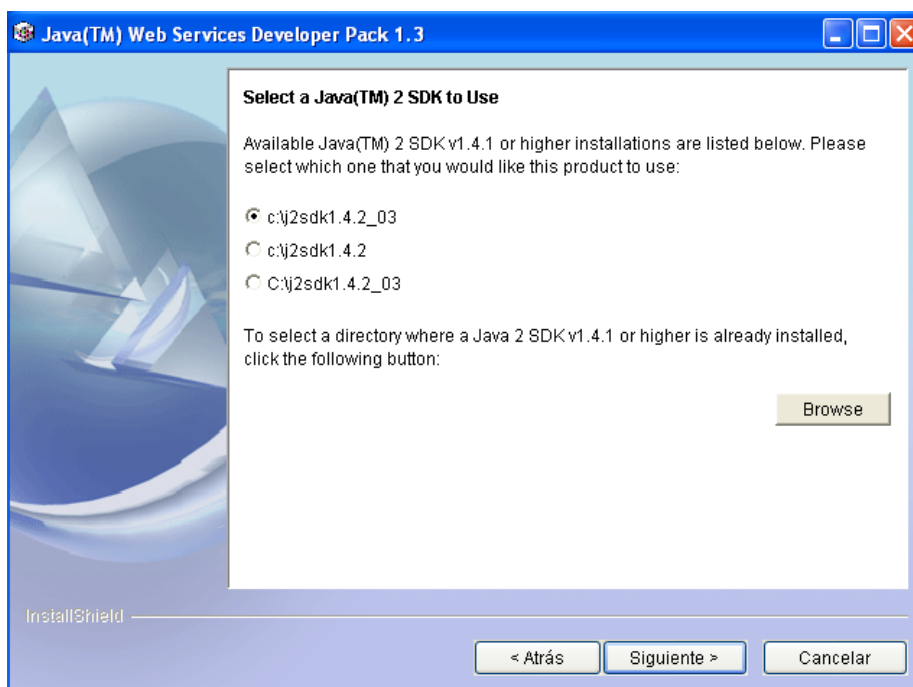
- Como ya tenemos instalado j2sdk se presenta la siguiente ventana



- Luego



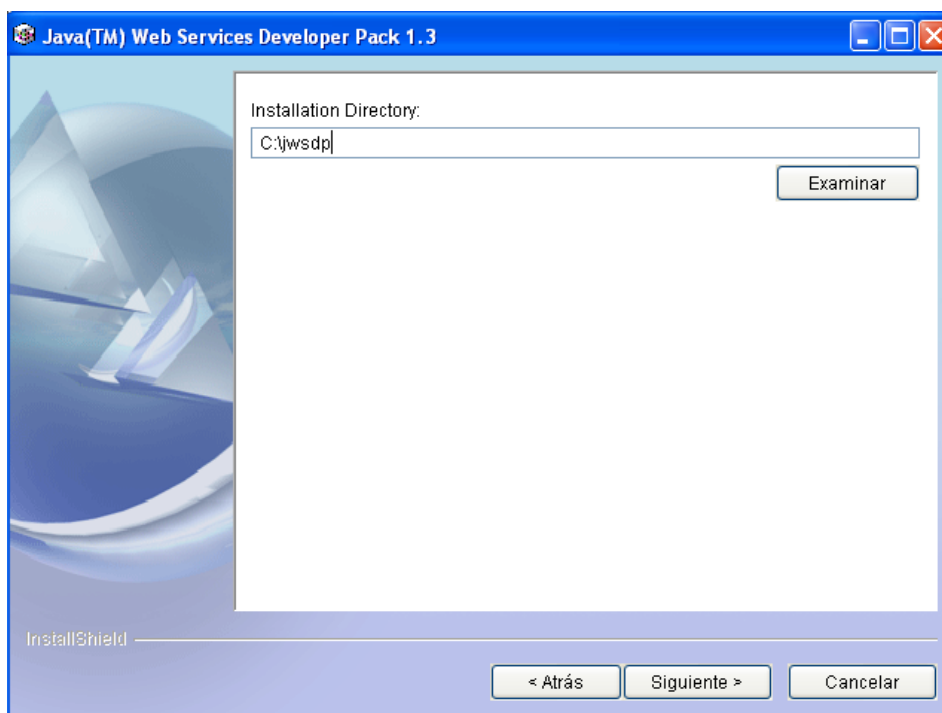
- Aprobamos presionamos siguiente



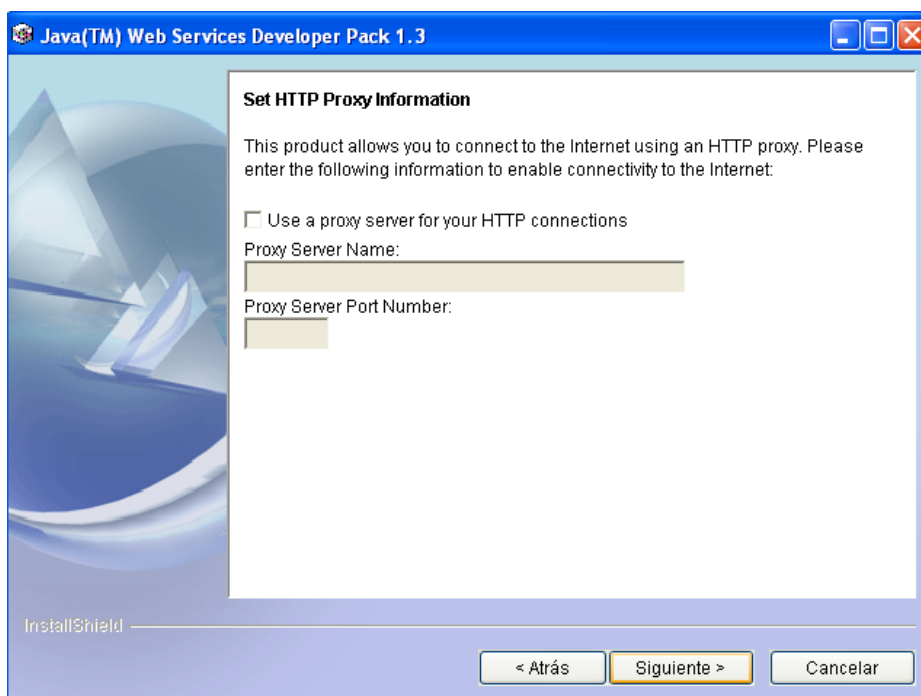
- Seleccionamos la versión de j2sdk a utilizar en caso de tener mas de una y presionamos siguiente



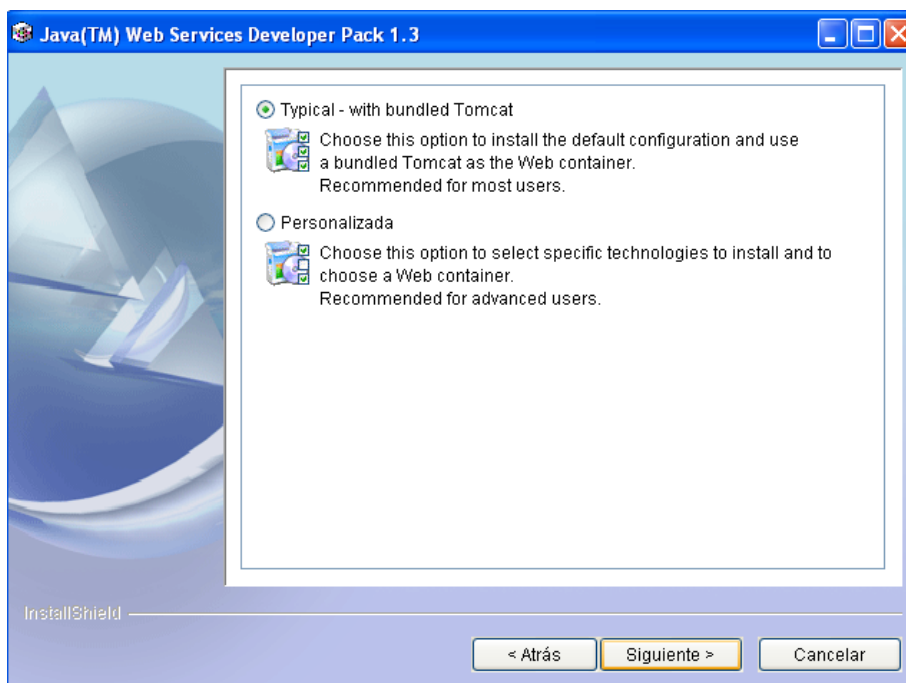
- Ingresamos el directorio de JWSDP



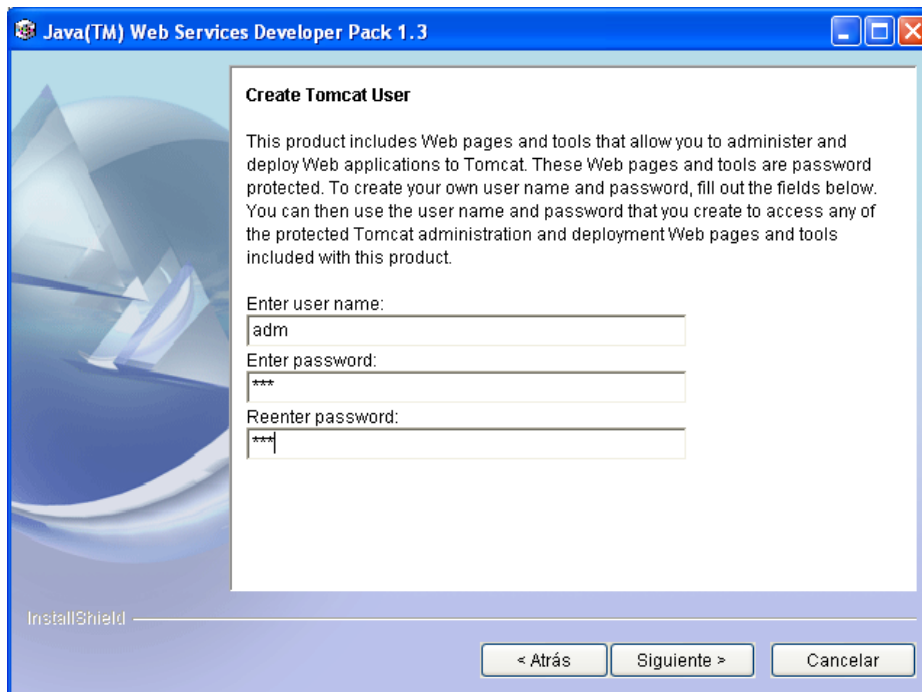
- Ingresamos información del proxy para nuestro caso solo pulsamos siguiente



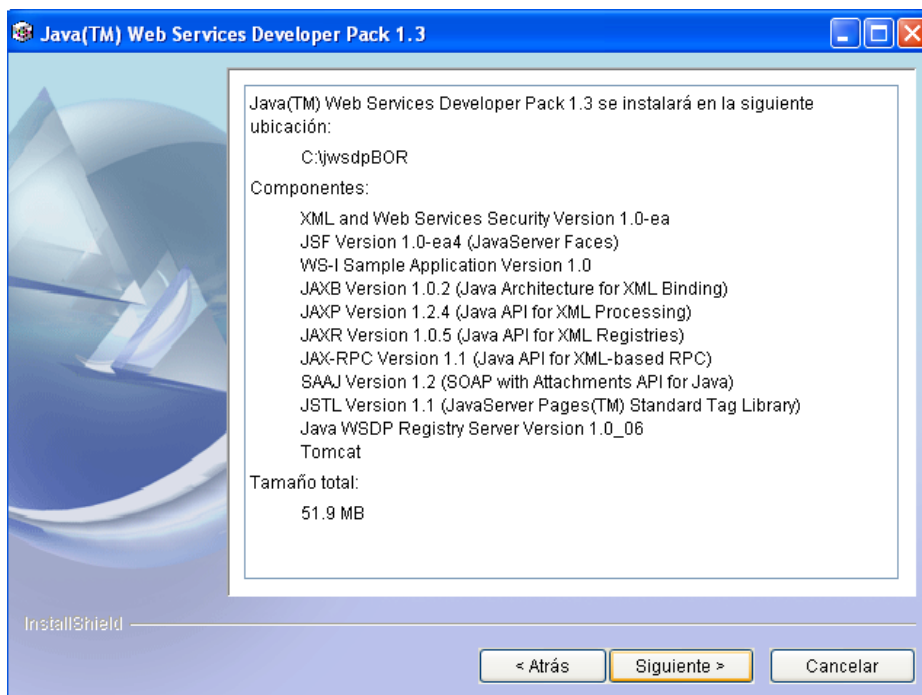
- Escojemos Instalación Típica pulsamos siguiente



- Ingresamos el usuario y la contraseña del administrador



- A continuación aparecerán las componentes de JWSDP a instalar presionamos siguiente



- A continuación comienza a instalarse

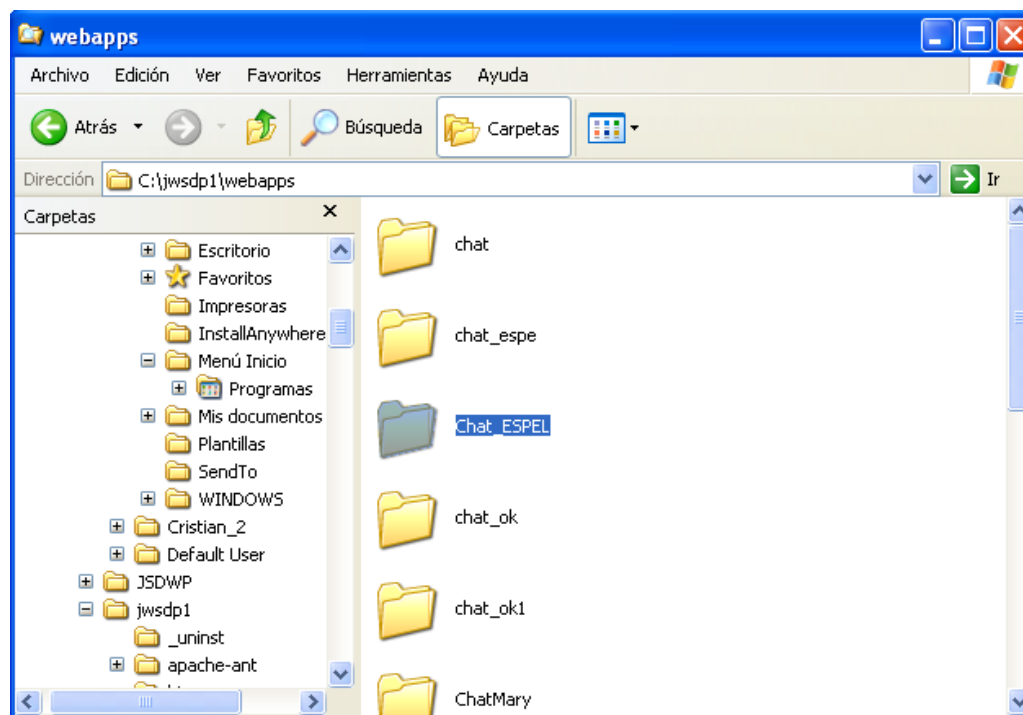


- Esperamos y al final presionamos finalizar

Para definir las variables de entorno seguimos los siguientes pasos:

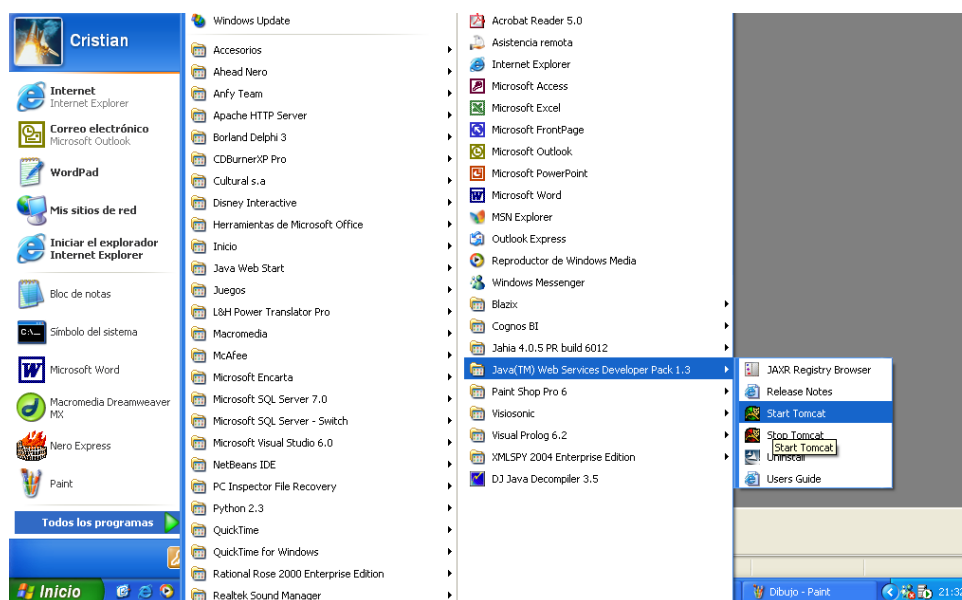
### **3.3 .- Instalación Chat**

Para instalar el Chat lo único que debemos hacer es copiar la carpeta Chat\_Espel dentro de la carpeta donde instamos el JWSDP es decir C:\jwsdp\webapps



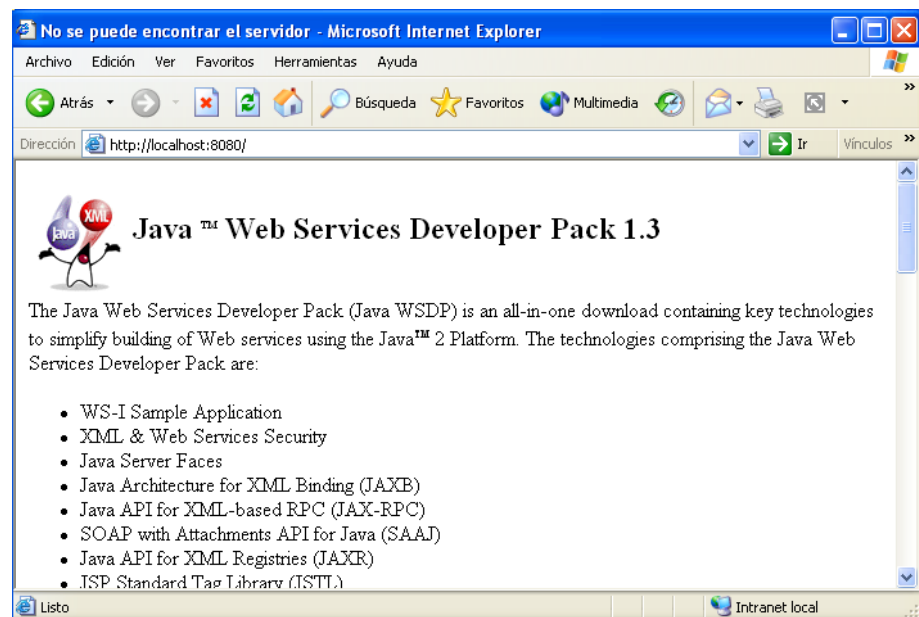
### 3.4 .- Levantar el Servidor

- Para levantar el servidor damos Click en Start Tomcat



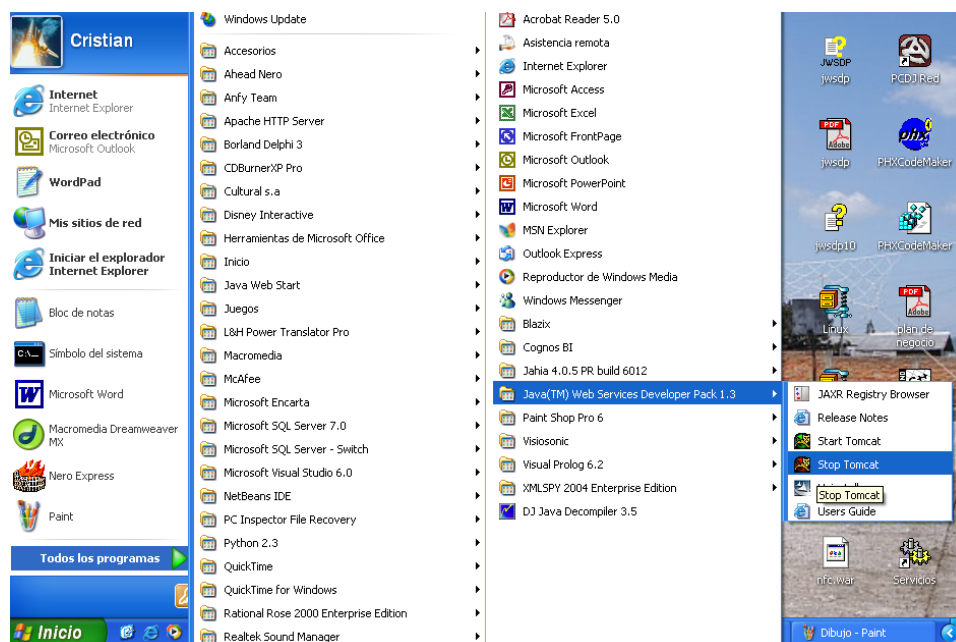
- Para probar q se levanto el servidor digitamos la siguiente dirección en el browser

<http://localhost:8080/>



### 3.5 .- Parar el Servidor

Para parar el servidor damos en Click en Stop Tomcat



**Anexo**

**B**

**MANUAL DE USUARIO**

Información CD #1



# **Chat Web Service en Java Manual de Usuario**

# Tabla de Contenidos

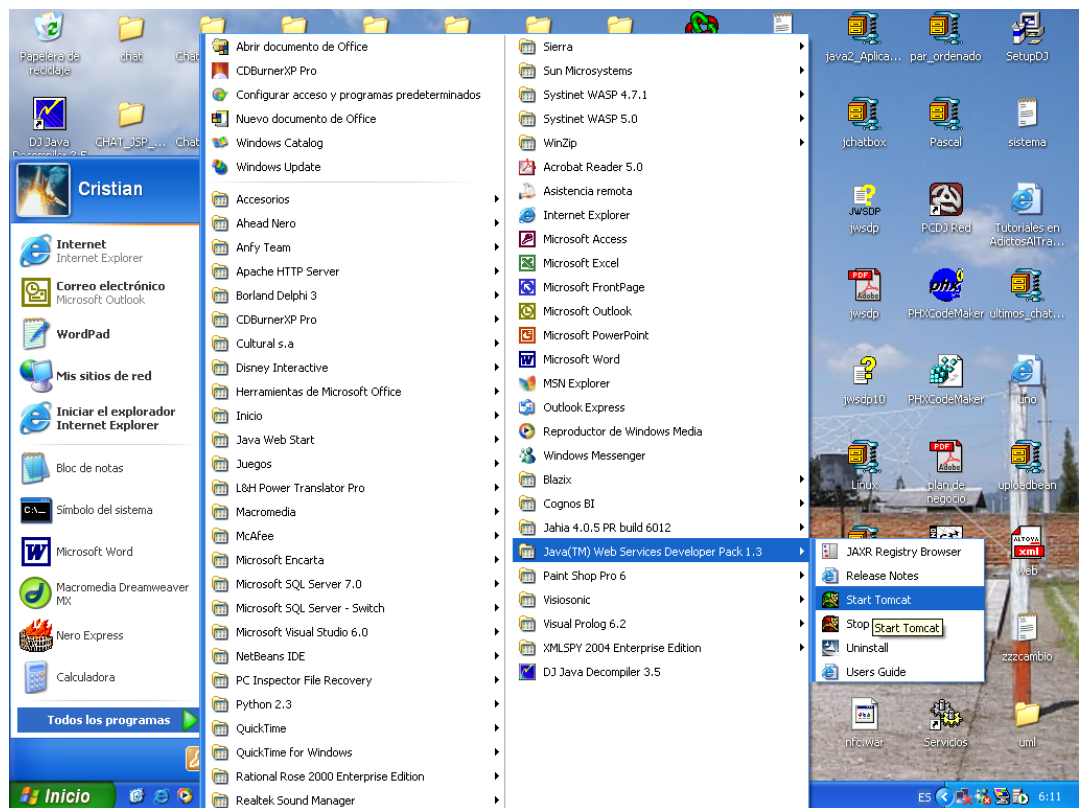
<b>1. Módulo Usuarios.....</b>	<b>195</b>
1.1    Gestión Servidor .....	195
1.1.1    Levantar el Servidor Web	195
1.2    Gestión Administrador.....	197
1.2.1    Ingresar al Chat Como Administrador	197
1.3    Gestión Cliente/Alumno .....	213
1.3.1    Ingreso al Chat	213
<b>2. Módulo Mensajes.....</b>	<b>215</b>
2.1    Gestión Mensajes Generales.....	215
2.1.1    Envío de Mensajes	215
2.2    Gestión Mensajes Privados .....	215
2.2.1    Envío de Mensajes	215

## **Módulo Usuarios**

### ***Gestión Servidor***

#### **Levantar el Servidor Web**

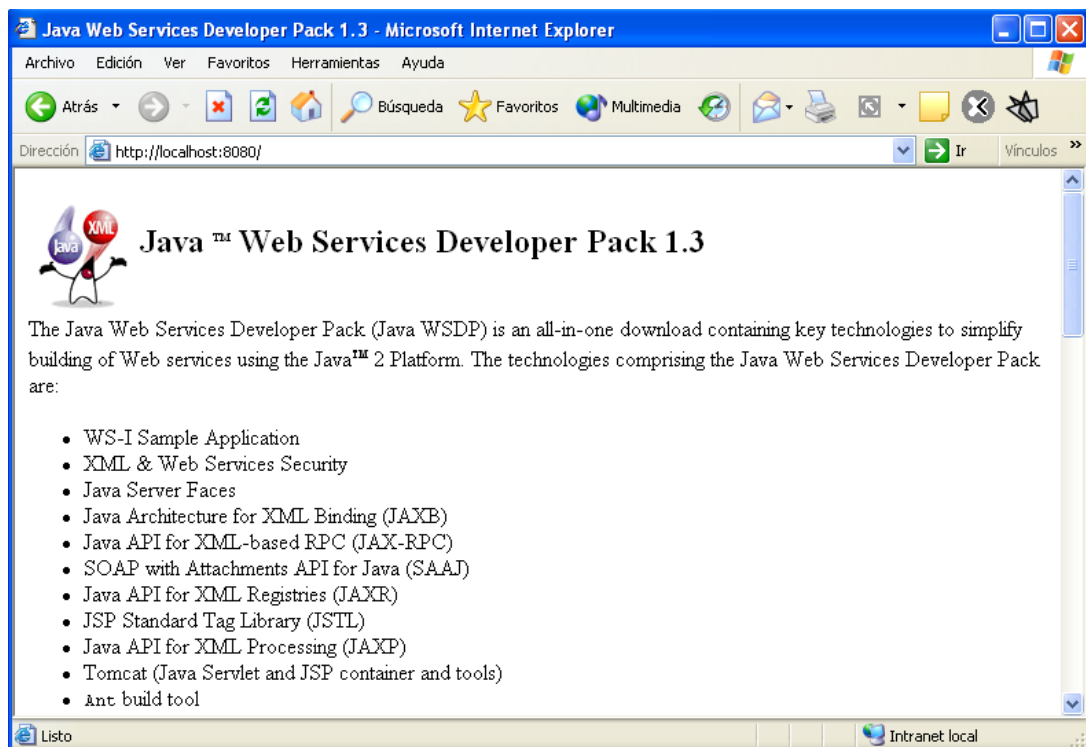
- Desde los programas de botón inicio en la opción Java Web Services Developer Pack 1.3 presionamos Start Tomcat



- O en una ventana de DOS ingresamos a la Carpeta c:\jwsdp\bin\
- Ejecutamos el comando startup
- Luego se abre una ventanita como la siguiente



- Para probar que se levanto el JWSDP en el browser ingresamos la dirección:  
<http://localhost:8080/>  
<http://127.0.0.1:8080/>
- Aparecerá una ventana como la siguiente



## ***Gestión Administrador***

### **Ingresar al Chat Como Administrador**

- En el browser ingresamos la dirección [http://localhost:8080/Chat\\_ESPEL/admin/](http://localhost:8080/Chat_ESPEL/admin/)



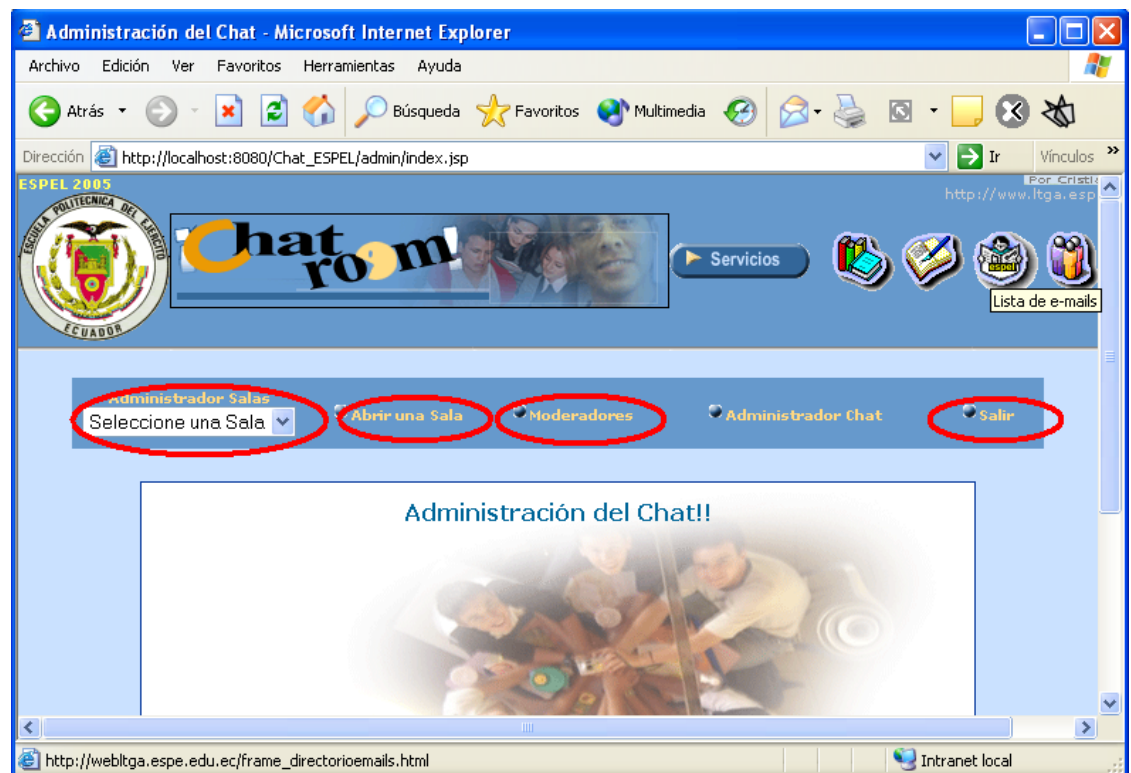
- Ahora ingresamos los siguientes datos :  
Nombre : cris  
Contraseña : cris
- La Pantalla q se presenta una vez ingresado la contraseña es :



- En el caso que no se ingreso la clave correcta la ventana a presentar es

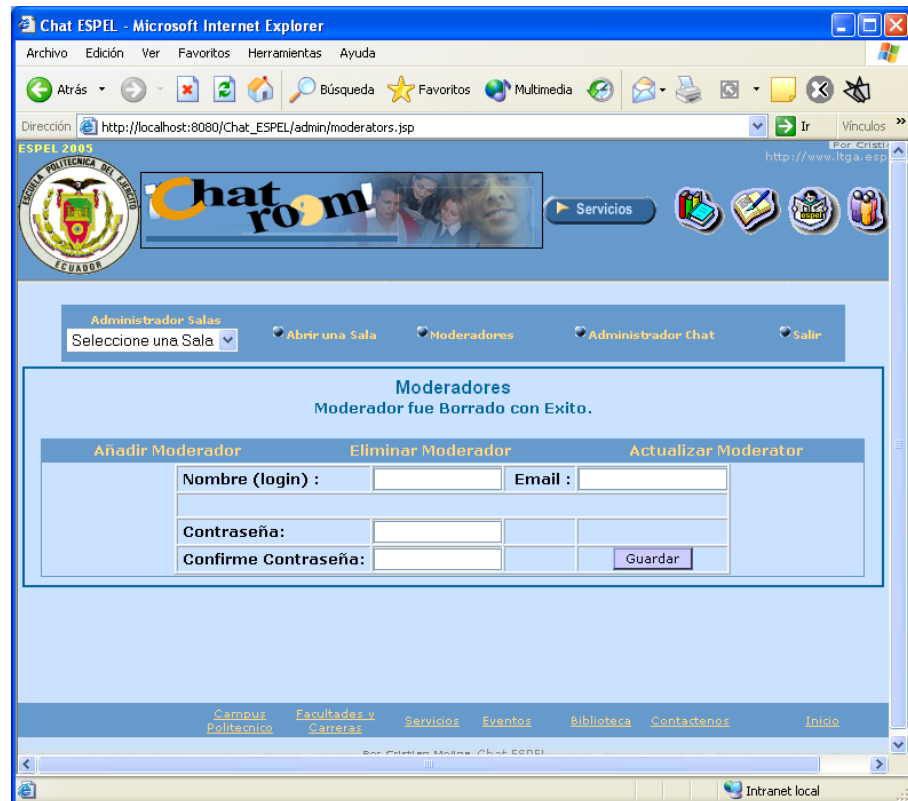


- Una vez ingresado al Chat tenemos las siguientes opciones:

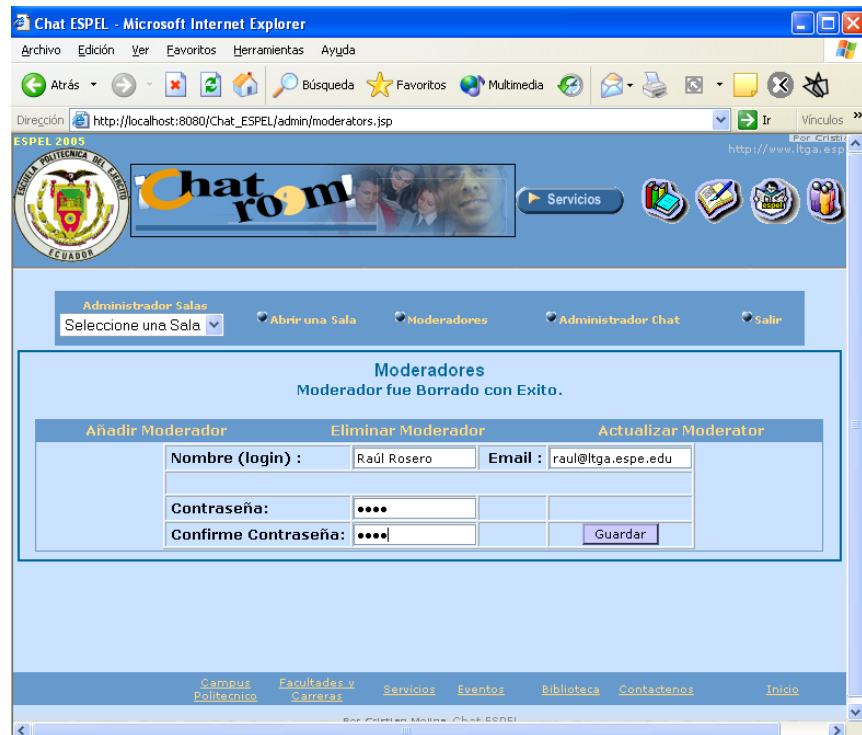




- Opción Crear Moderadores
  1. Se presenta la siguiente ventana

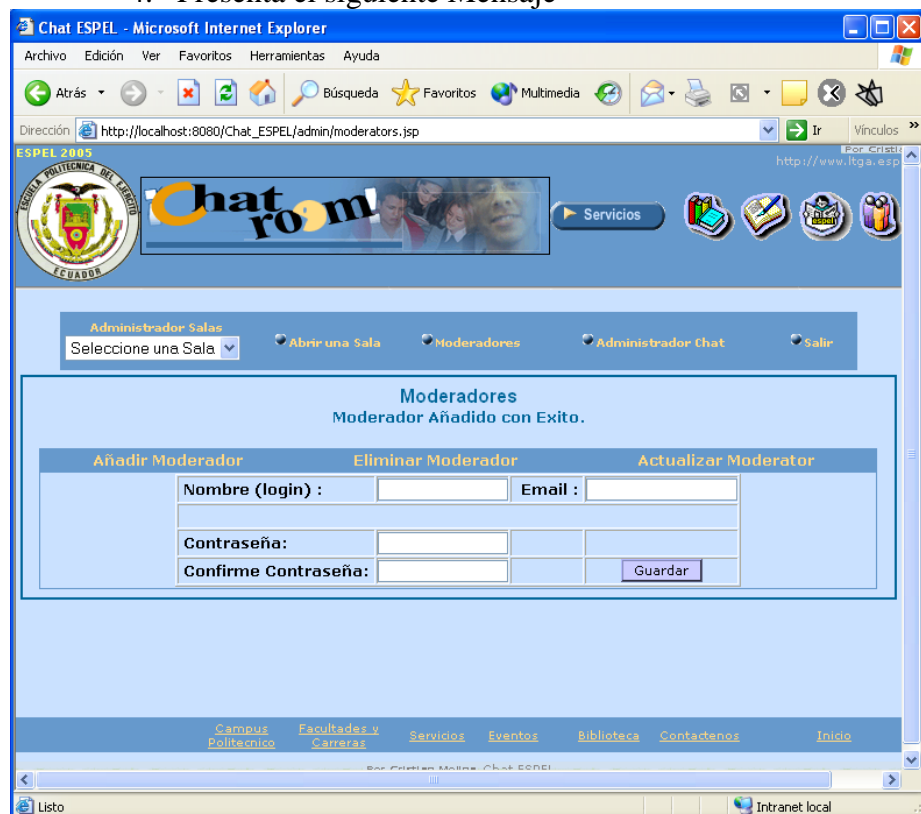


2. Ingresamos la información:

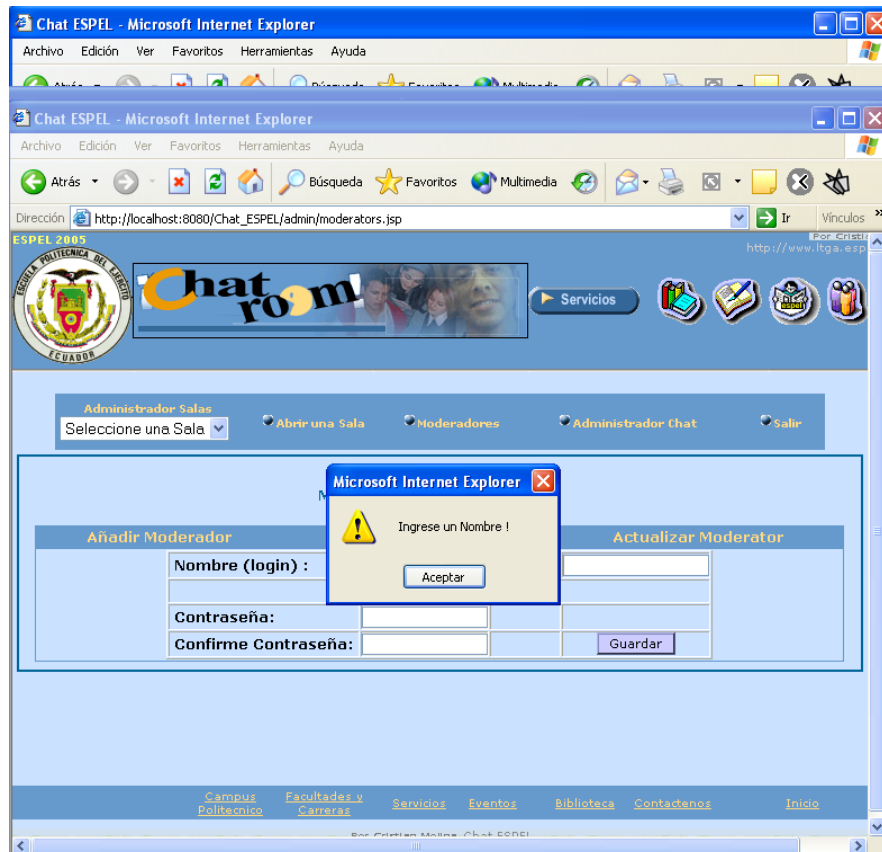


3. Presionamos Guardar

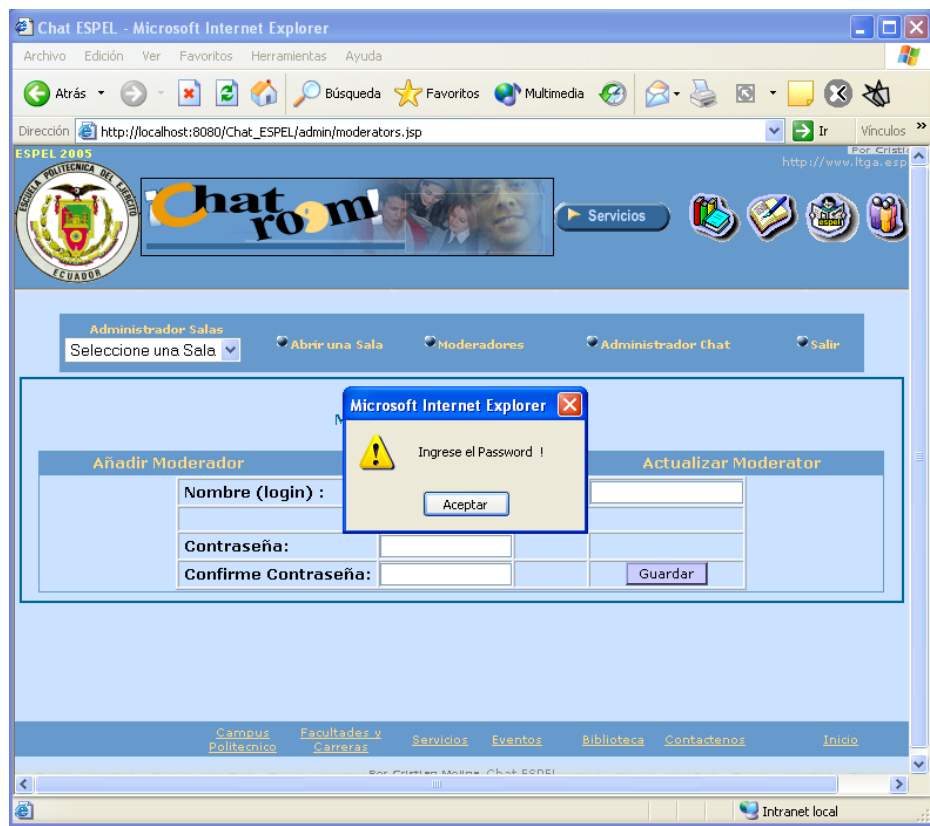
4. Presenta el siguiente Mensaje



5. En el Caso que presiono Guardar con los campos en blanco el mensaje a presionar es:



6. Una vez ingresado el nombre también se debe ingresar una contraseña caso contrario se le presentara el siguiente mensaje al grabar

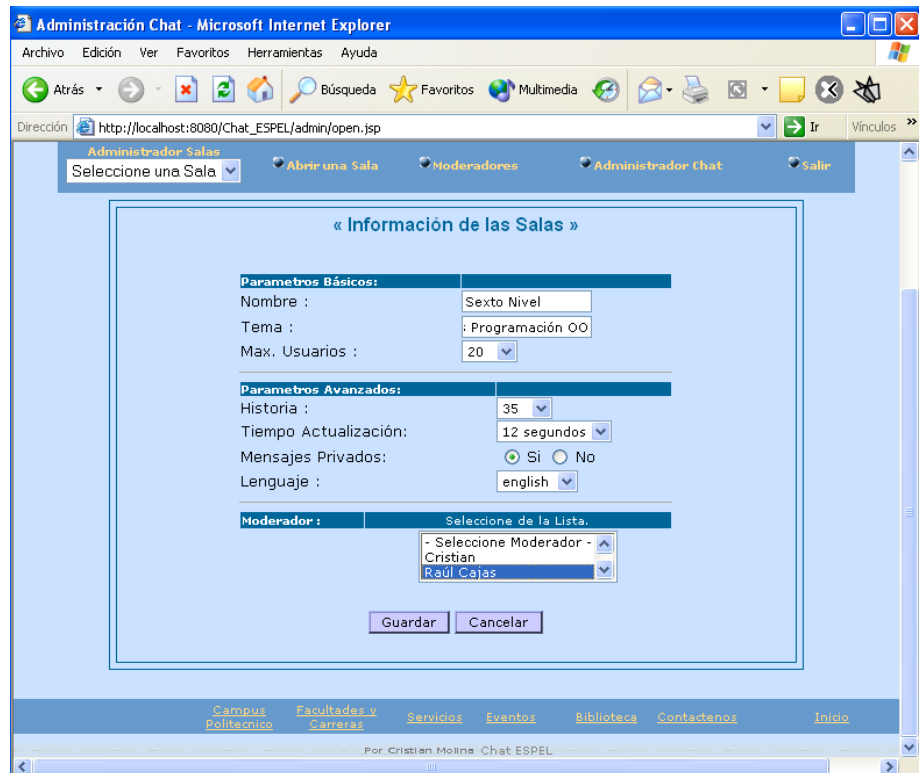


## 7. Luego Presione Guardar y se actualizan los Datos

- Opción Crear Salas
  1. Damos Click en la Opción Ingresar Salas
  2. Aparecerá la siguiente ventana



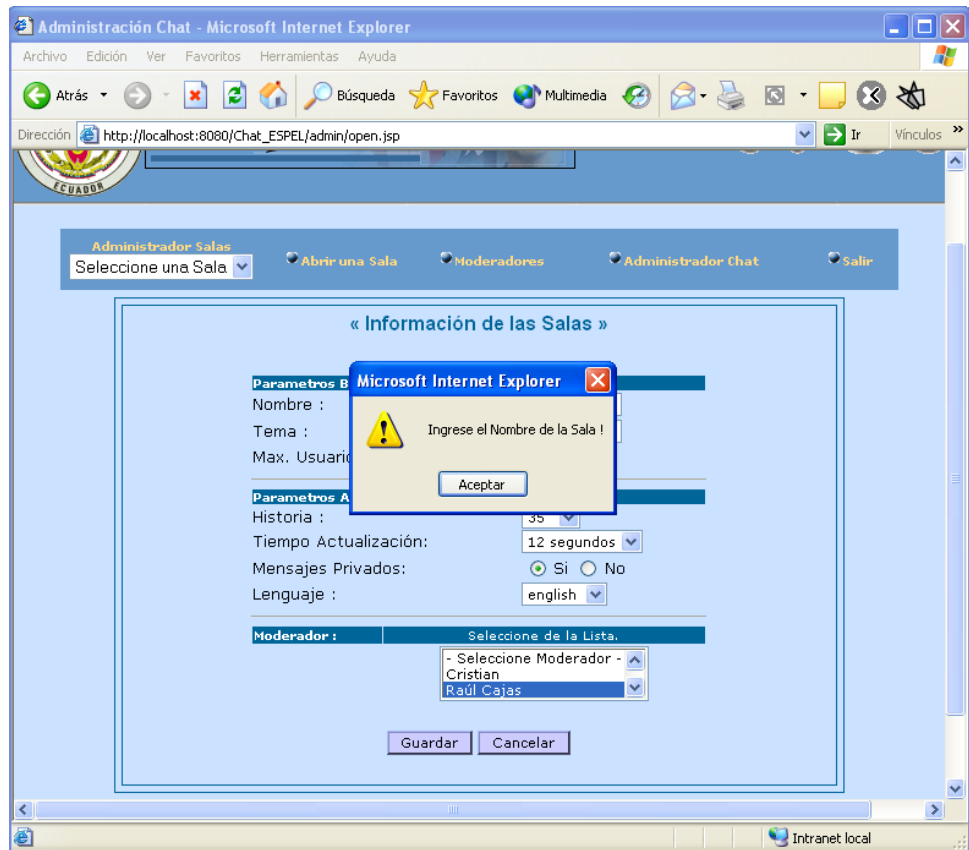
3. Ingresamos la información y Presionamos el Botón Guardar



4. Una vez ingresada toda la información se almacenara la información y se presenta el mensaje Sala Creada con Éxito

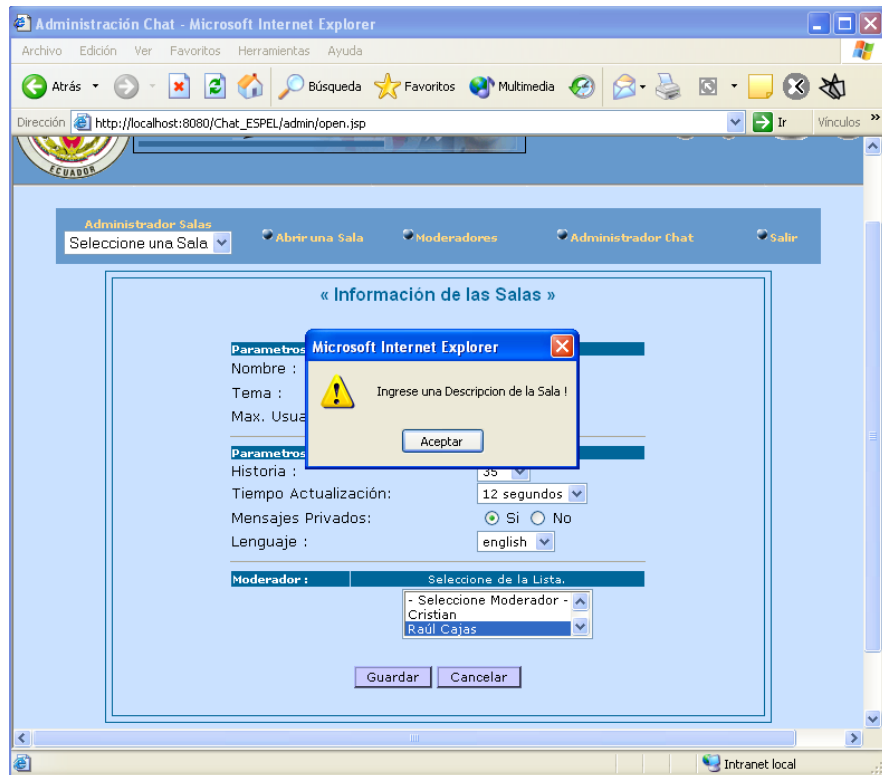


5. En el caso de no haber ingresado toda la información se van presentando mensajes de advertencia que falta llenar campos
6. Si no ingresamos el nombre de la sala la pantalla a presentar es :



7. Si no se ingresa la descripción de sala la ventana a presentar es

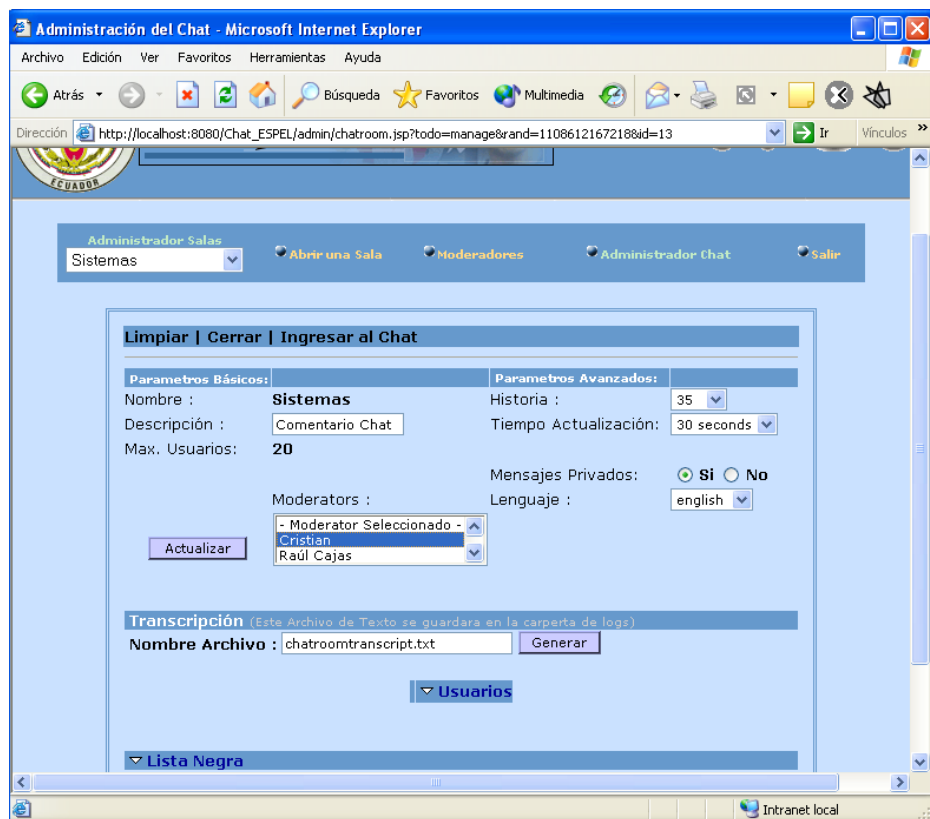




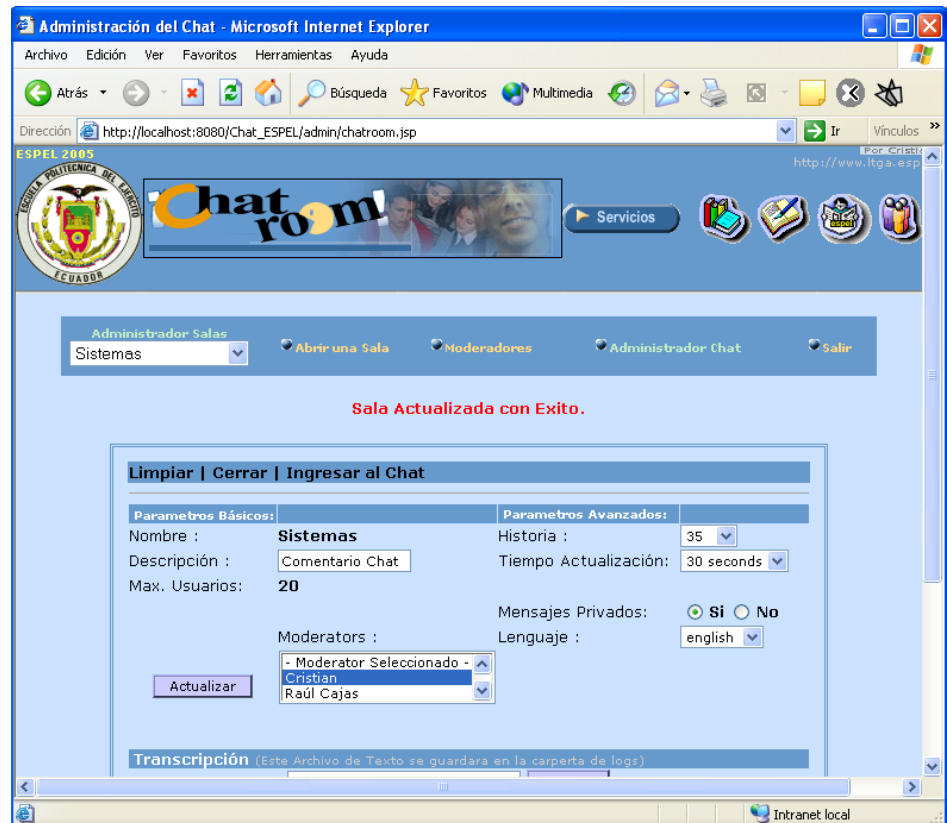
- Opción Actualizar Salas
8. Seleccionamos la sala a modificar



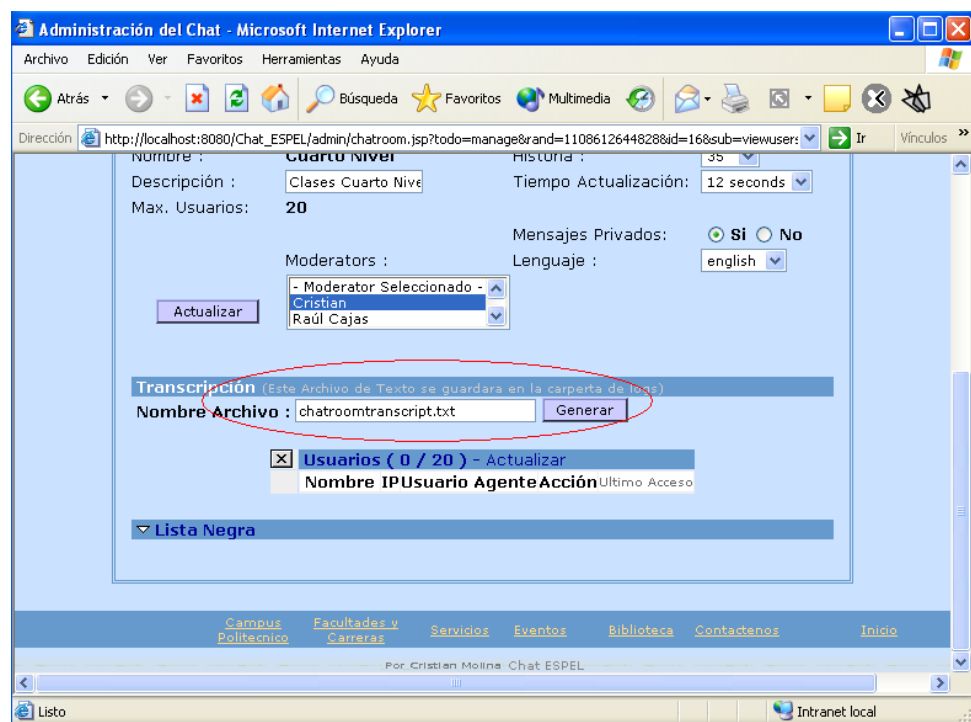
9. Aparecerá la siguiente ventana con los datos de la Sala



10. Una vez modificados los Datos Presionamos el Botón Actualizar
11. Presenta el Mensaje Sala Actualizada con Éxito



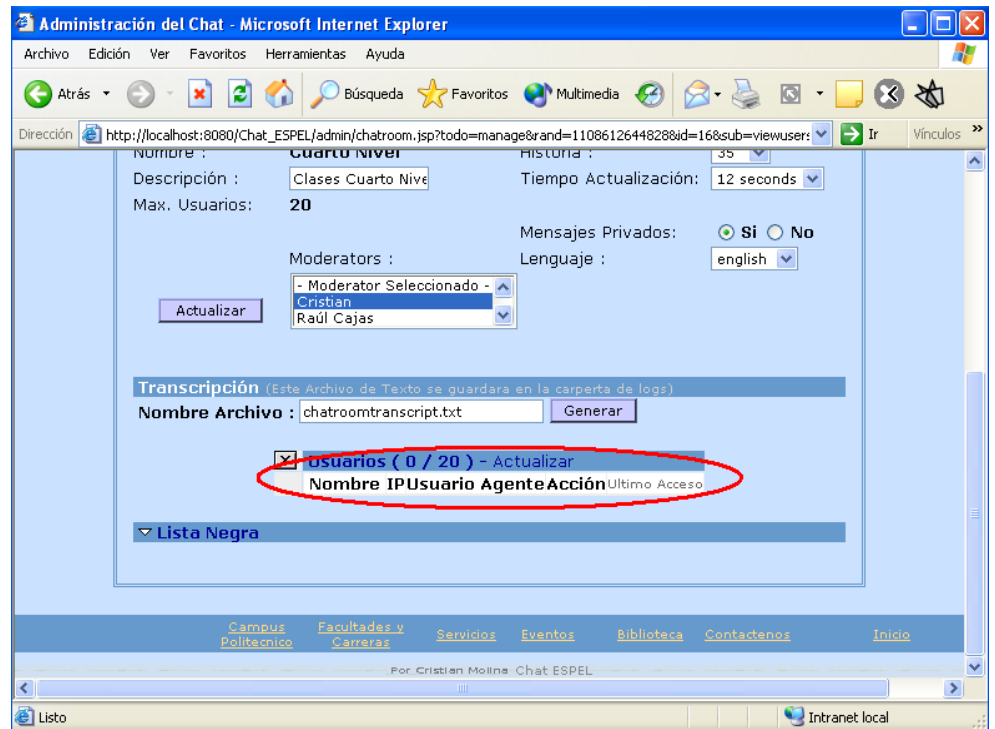
12. Para sacar respaldo de la las conversaciones ingresamos el nombre de archivo y presionamos el botón Generar



13. El Archivo se Almacena dentro de la siguiente carpeta

C:\jwsdp\webapps\Chat\_ESPEL\WEB-INF\conf\

14. También podemos observar los usuarios conectados a las salas

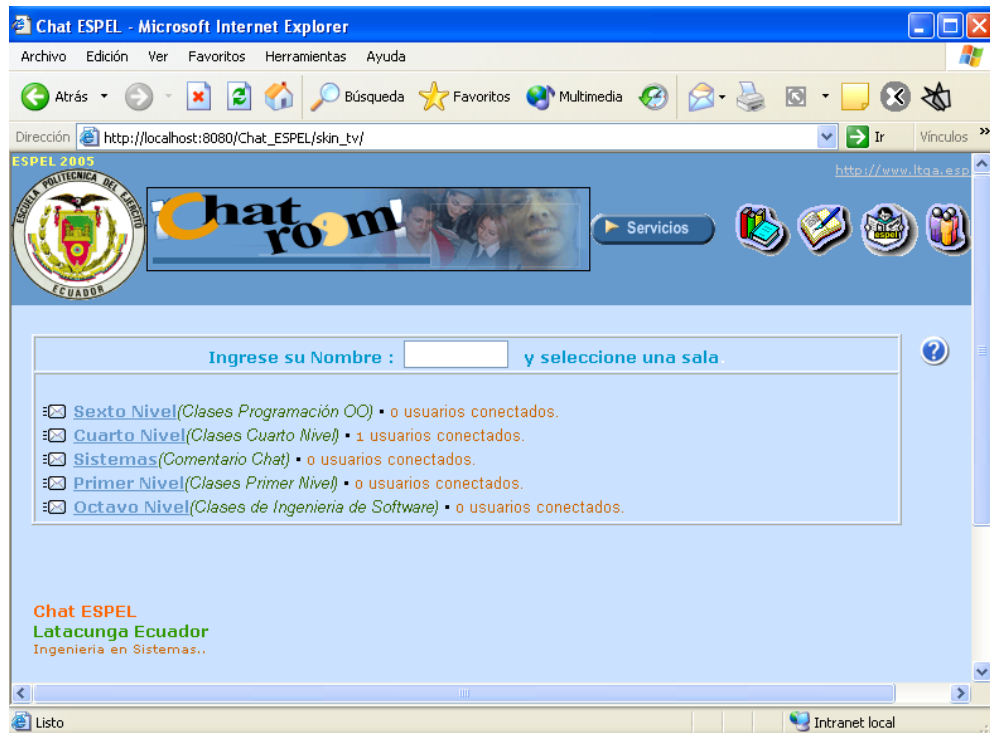


15. En esta opción podemos sacar a los usuario de las salas

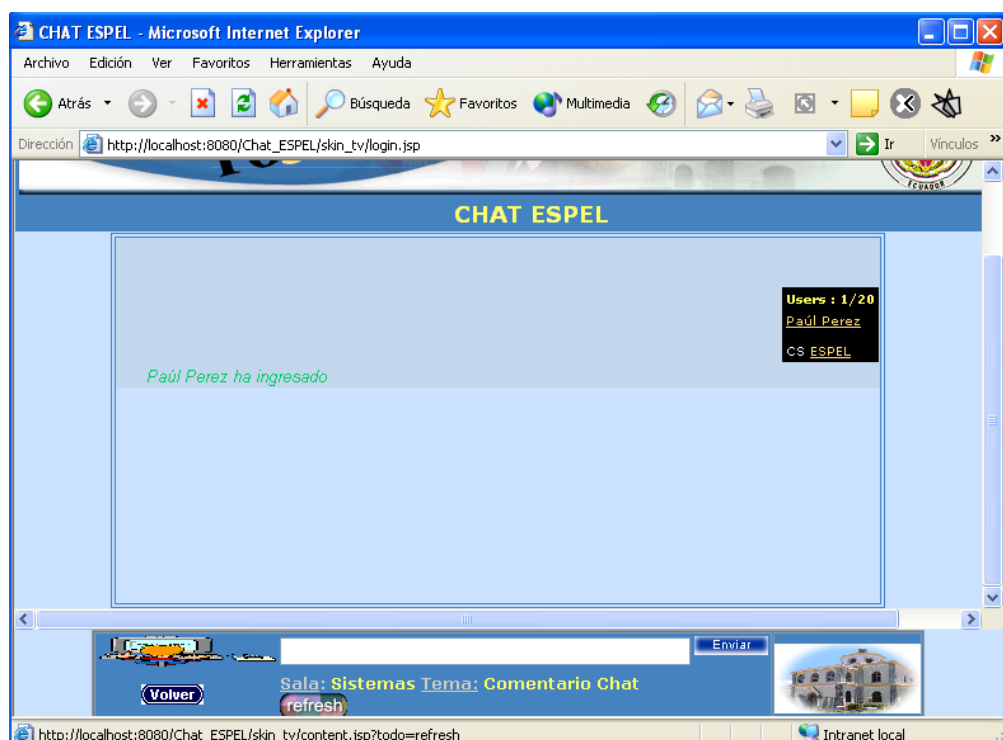
## **Gestión Cliente/Alumno**

### **Ingreso al Chat**

- En el browser ingresamos la dirección
- [http://localhost:8080/Chat\\_ESPEL/skin\\_tv/](http://localhost:8080/Chat_ESPEL/skin_tv/)



- Ahora si ingresamos un nombre y damos un Click en la sala que queremos ingresar



## Módulo Mensajes

### *Gestión Mensajes Generales*

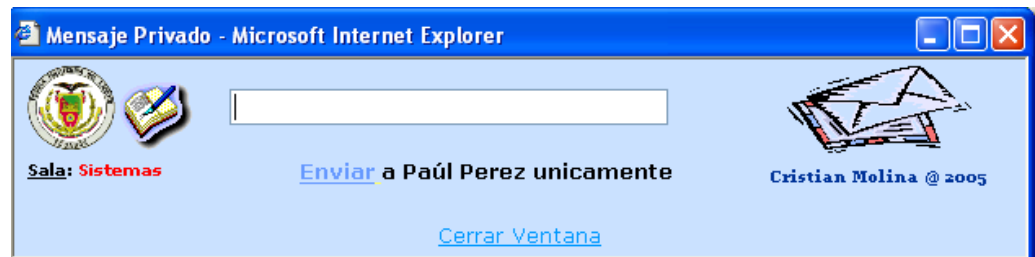
#### Envió de Mensajes

- Ingresamos en la Parte Inferior el Mensaje
- Presionamos el Botón Enviar

### *Gestión Mensajes Privados*

#### Envió de Mensajes

- Seleccionamos un usuario
- Damos doble Click y nos aparece una ventana como la siguiente



- Ingresamos el Mensaje en la Ventana
- Presionamos el Botón Enviar

**Anexo**  
**C**

**GLOSARIO**



**APACHE TOMCAT:** SW utilizado en los servidores Web de tecnología UNIX.

**JWS DP.-** Servidor Web para el desarrollo de aplicaciones empresariales multicapa.

**JAR (Java ARchive).**- formato de archivo que combina múltiples archivos en uno  
Javac: compilador del lenguaje java.

**JSP.-** Página HTML con código JSP entremezclado en el documento.

**JVM.-** entorno de ejecución de instrucciones que genera un compilador java.

**Middleware:** Es el estrato de software que provee una abstracción de programación, así como un enmascaramiento de la heterogeneidad subyacente de las redes, hardware, sistemas operativos y lenguajes de programación.

**Parser.-** Objeto encargado de construir e interpretar XML.

**RPC.-** Permite a los desarrolladores de software hacer funciones o llamadas a métodos a través de la red.

**Servlet.-** Objeto Java basado en solicitudes/respuestas que corre dentro del entorno del contenedor administrado.

**SOAP.-** Protocolo de formato de mensajes extensible.

**UDDI.-** Estándar para registrar y descubrir los Web Services.

**Web Service.-** Llamada a procedimientos remotos sobre la Web, construidas sobre estándares de Internet y tecnología XML.

**EbXML.-** Protocolo de negocios electrónicos embebido.

**WSDL.-** Formato XML que describe los Web Services.

**DFD's.-** Diagrama de Flujo de datos.

**POO.-** Paradigma Orientado a Objetos.

**AOO.-** Análisis Orientado a Objetos.

**UML.-** Lenguaje Modeling Unified, Lenguaje de Modelado Unificado.