



Facultad 9

Diseño de interfaces de interacción externa para la Plataforma de Televisión Informativa PRIMICIA

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS**

AUTOR:

José Ernesto Rodríguez Zúñigas

TUTOR:

Ing. José Luís Leyva Ramírez

Ciudad de la Habana, Junio 2010

“Año 52 de la Revolución”

Dedicatoria

A mis padres.

A mi familia en general.

A mis amigos.

Agradecimientos

A mi mamá por ser la luz de mi vida, por apoyarme siempre, por confiar en mí y por estar en los momentos malos y en los buenos.

A mi papá por los consejos.

A mi hermana Gretel por ser mi segunda madre y por haberme ayudado tanto.

A Cuesta por ser mi segundo padre y consejero.

A mis hermanos.

A mi tía Mery, Mirita, Mario.

A toda mi familia.

A la comunidad de la UCI, ellos saben quiénes son.

A todos mis amigos.

A mi tutor, y todos los que de una forma u otra hicieron posible la realización de este trabajo.

Muchas Gracias.

Ciudad de la habana, diciembre 2009

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Señales Digitales de la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

José Ernesto Rodríguez Zúñigas

Ing. José Luís Leyva Ramírez

Resumen

La humanidad ha encontrado diversas formas de difundir la información desde tiempos ancestrales, con el surgimiento de nuevas tecnologías y de la Internet, la gran red de redes se ha hecho más fácil esta operación. La creación de plataformas web informativas ha sido el paso más importante en este proceso. En la Universidad de la Ciencias Informáticas (UCI) se ha desarrollado un producto informático llamado PRIMICIA (Plataforma de Televisión Informativa) que brinda la posibilidad de transmitir informaciones en formato de Texto, Texto – Imagen, Imagen y Video, la cual para brindar un mejor servicio a sus usuarios debería poder importar información de otras plataformas para así enriquecer su oferta. Esta investigación tiene como objetivo diseñar interfaces para la comunicación de PRIMICIA, funcionalidades que harán de esta un mejor producto.

Palabras Claves: Aplicación Web, Primicia, Interacción, Información, Televisión Informativa.

Contenido

Dedicatoria.....	II
Agradecimientos.....	III
Resumen.....	V
Palabras Claves: Aplicación Web, Primicia, Interacción, Información, Televisión Informativa.....	V
Introducción	1
Fundamentación Teórica.....	5
1. Introducción.....	5
1.1. Conceptos asociados al dominio del problema	5
1.1.1. Interacción	5
1.1.2. La televisión.....	6
1.1.3. Plataforma de Transmisión	6
1.1.4. Plataforma Web	7
1.1.5. Aplicaciones Web	7
1.1.5.1. Ventajas de las aplicaciones Web.....	8
1.1.6. PRIMICIA.....	9
1.1.7. Arquitectura Cliente Servidor.....	10
1.1.7.1. Características principales de la tecnología cliente-servidor	11
1.1.7.2. Principales componentes	11
1.1.7.3. Tipos de servidores.....	12
1.1.8. Servidor WEB.....	13
1.1.9. Servicios Web	13
1.2. Conceptos asociados al intercambio de información en internet.....	15
1.2.1. Transmisión de Datos en Internet	15
1.2.2. El modelo OSI	17

1.2.3.	El Protocolo TCP/IP	17
1.2.4.	TCP (Transmission Control Protocol)	19
1.2.5.	Servicios de Internet: el nivel de aplicación.....	19
1.2.6.	Transferencia de ficheros.....	20
1.3.	Interacción entre sistemas informáticos	21
1.3.1.	Proceso de interacción entre sistemas informáticos	21
1.3.2.	Descripción actual del proceso de interacción entre sistemas informáticos	23
1.4.	Situación problemática	23
1.5.	Análisis de otras Soluciones.....	25
1.6.	Conclusiones.....	26
	Modelos de intercambio de información	27
2.	Introducción.....	27
2.1.	Descripción de los modelos de Intercambio de Información entre Aplicaciones de escritorio.	27
2.1.1.	Sockets	27
2.1.2.	RPC (Llamada a Procedimientos Remotos).....	29
2.1.2.1.	Transporte	30
2.2.	Intercambio de Información entre Plataformas Web	30
2.2.1.	Web Services (Servicios Web)	30
2.2.1.1.	¿Cómo funciona un Web Services?	31
2.2.1.2.	Requerimientos de los Web Services	31
2.2.1.3.	Tecnologías que se utilizan en Web Services.....	31
	XML	32
	SOAP	32
	WSDL.....	32
	UDDI.....	33

2.2.1.4.	Especificación en los Web Services.....	33
2.2.2.	Really Simple Syndication RSS (Sindicación Realmente Simple)	35
2.2.2.1.	Ventajas del RSS	36
2.3.	Intercambio de Información entre una Aplicación de escritorio y una Plataforma Web	37
2.3.1.	Internet Communication Engine ICE (Motor de Comunicaciones por Internet) ...	37
2.3.1.1.	El Protocolo de ICE.....	39
2.3.1.2.	Servicios de ICE.....	40
2.3.1.3.	Beneficios de la arquitectura de ICE	40
2.3.1.4.	Paquetes de ICE	41
2.3.2.	CORBA (Common Object Request Broker Architecture).....	43
2.3.2.1.	Componentes de CORBA	44
2.3.2.2.	EL ORB (Object Request Broker).....	45
2.3.2.3.	Qué es un objeto CORBA	46
2.3.2.4.	El IDL (Interface Definition Language)	46
2.3.2.5.	Facilidades de CORBA.....	46
2.3.2.6.	Paquetes de CORBA.....	47
2.3.3.	DCOM	49
2.3.3.1.	La Arquitectura DCOM	50
2.4.	Conclusiones.....	52
	Solución Propuesta.....	53
3.	Introducción.....	53
3.1	Solución propuesta para el intercambio de información entre Aplicaciones de Escritorio	53
3.1.1	RPC.....	53
3.2	Solución propuesta para el intercambio de información entre Plataformas Web	55

3.2.1	Servicios Web	55
3.2.1.1	Requisitos de un Web Services.....	56
3.3	Solución propuesta para el intercambio de información entre una Aplicación de escritorio y una Plataforma Web.....	57
3.4	Validación de la Propuesta	58
	Características del Método Delphi.....	59
3.4.1	Fase Preliminar	59
	Objetivo a alcanzar	60
	Selección del tipo de Delphi a utilizar.....	60
	Selección de expertos	60
	Confirmación de la participación de los expertos.....	61
3.4.2	Fase Exploratoria.....	62
	Elaboración de la Encuesta.....	62
3.4.3	Fase Final.....	62
3.5	Conclusiones Parciales.....	66
	Conclusiones Generales	67
	Recomendaciones	68
	Bibliografía	69
	Bibliografía Consultada.....	71
	Glosario.....	72
	Índice de Anexos.....	75

Introducción

Desde el surgimiento de la informática se vio la necesidad de contar con software capaces de resolver las necesidades cada vez más crecientes en todos los campos de la economía, a su vez con la interconexión de varias computadoras se dio un mejor uso de la información presente en cada una de estas, así surgió Internet la gran red de redes. La Web surgió con el objetivo de hacer más fácil para los usuarios la obtención de la información que se hacía difícil, para esto se construyó sobre un concepto de lectura universal, esta información podía ser accedida desde cualquier ordenador en cualquier parte del mundo, por cualquier persona autorizada, esta se basa en el protocolo de transmisión de hipertexto (HTTP), el lenguaje de marcas de hipertexto (HTML) y el localizador uniforme de recursos (URL). La Web consiste en ofrecer una interfaz simple para acceder a los recursos de Internet, a través de páginas electrónicas.

En las áreas relacionadas con computadoras, las comunicaciones implican transferencia de datos de una computadora a otra a través de un medio de comunicación, como por ejemplo: el teléfono, las microondas, el enlace de satélite o el cable físico. Existen dos métodos primarios de comunicaciones entre computadoras: conexión temporal de dos computadoras a través de una red conmutada, como por ejemplo: el sistema de teléfono público; y el enlace permanente o semipermanente de múltiples estaciones de trabajo o computadoras en una red.(1)

Las comunicaciones que pueden existir entre dos o más computadoras son: asíncronas, síncronas o remotas.

Las comunicaciones asíncronas son aquellas comunicaciones entre computadoras (computadora a computadora) en las que el envío y recepción no se basa en el tiempo, lo que significa que se debe determinar explícitamente cuando empiezan y terminan las transmisiones. Las comunicaciones síncronas son aquellas comunicaciones computadora a computadora en las que las transmisiones están sincronizadas mediante temporización entre la máquina emisora y receptora. En las comunicaciones remotas una computadora interactúa con otra computadora remota a través de una conexión telefónica y otra línea de comunicaciones. (1)

Las comunicaciones entre sistemas pueden estar dadas: por la necesidad de comunicación que pueda existir entre sus componentes, y la interacción con otros sistemas de información.

Hoy en día no se puede hablar de educación, cultura y desarrollo, si no se integran eficientemente y en todas las esferas de la sociedad las Nuevas Tecnologías de la Información y Comunicaciones (TICs), es por esta razón que en los últimos años en Cuba, se ha promovido el uso masivo de las TICs, insertándolas en dichas esferas. Para darle cumplimiento a este propósito se insertaron televisores, computadoras, reproductores de video en cada uno de los sectores de la sociedad, para incrementar el uso de las tecnologías. Por otro lado el gobierno cubano creó en septiembre del 2002 la Universidad de las Ciencias Informáticas (UCI), la cual estaría encaminada a promover la producción del software en el país.

La Universidad de las Ciencias Informáticas (UCI) que surge con la intención de formar profesionales altamente calificados con el objetivo de contribuir al desarrollo científico y tecnológico, y a su vez desarrollar la industria del software nacional. Esta cuenta con numerosas aplicaciones web que necesitan tener cierta interoperabilidad entre ellas, para lograr el intercambio de datos y servicios.

La Facultad 9 la cual dentro de sus vertientes de producción cuenta con el Departamento de Señales Digitales, el cual presenta entre sus proyectos con la Plataforma de Televisión Informativa PRIMICIA.

Este producto está destinado a facilitar la transmisión automática y eficiente de las informaciones generadas por un canal de televisión interno utilizando software libre. En estos momentos cuenta con varias soluciones entre ellas ACN, solución destinada a llevar la noticias desde Cuba hacia los colaboradores internacionalistas que prestan servicios en diferentes partes del mundo.

Esta solución también es utilizada para llegar a las zonas de silencio del país, zonas donde las señales no llegan vía terrestre y se envían por satélite.

Actualmente PRIMICIA está en su segunda etapa de desarrollo, dentro de sus deficiencias se encuentra la necesidad de contar con interfaces de interacción externa con otros sistemas que permita a los usuarios la actualización de la información.

Se propone poder determinar las líneas de conexión con diversos servicios web posibilitando que no tengan que ser escritos directamente en el código fuente y en tanto estos se almacenen en una Base de Datos para su posterior gestión.

De acuerdo a la situación problemática anterior el **problema científico** planteado es el siguiente: Necesidad de interacción externa de la Plataforma de Televisión Informativa PRIMICIA con servicios ofrecidos por la web que permita recibir y exportar información desde y hacia otros sistemas.

Como **objeto de estudio** en el cual se enmarca el problema está: El proceso de Interacción entre sistemas informáticos.

El **campo de acción** sobre el que se va a trabajar es el proceso de interacción de la Plataforma de Televisión Informativa PRIMICIA con servicios externos.

Para dar solución al problema científico se traza el siguiente **objetivo general**: Diseñar interfaces de comunicación externa entre la Plataforma de Televisión Informativa PRIMICIA con servicios externos.

Por lo que se puede plantear como **idea a defender** que contar con interfaces de comunicación para la Plataforma de Televisión Informativa PRIMICIA con servicios externos, facilitará a los usuarios la actualización de la Información.

Para el desarrollo de esta investigación se trazaron las siguientes tareas investigativas:

1. Establecer las tendencias actuales para el intercambio de información entre sistemas informáticos.
2. Establecer la situación actual de comunicación de la Plataforma de Televisión Informativa PRIMICIA.
3. Analizar otras soluciones.
4. Describir los modelos de intercambio de información.
5. Obtener el diseño de comunicación para el intercambio de información con el exterior de la Plataforma de Televisión Informativa PRIMICIA.

6. Valorar la factibilidad y aportes de las mejoras propuestas.

Para dar cumplimiento a las tareas antes mencionadas se emplearon los siguientes métodos de investigación:

Métodos teóricos:

- **Análisis y síntesis:** La investigación se dividió en distintas partes para llevarla a cabo, donde posteriormente, delimitando las relaciones que existían entre ellas, se unieron como un todo.
- **Histórico lógico:** Utilizado para describir las tendencias actuales de creación de interacción entre sistemas informáticos y de los sistemas enfocados a la transmisión de Televisión Informativa a nivel mundial, nacional y en la UCI, de manera que se puede deducir cual es la más idónea para PRIMICIA.

Fundamentación Teórica

1. Introducción

El funcionamiento de las sociedades humanas es posible gracias a la comunicación. Esta consiste en el intercambio de mensajes entre los individuos. Comunicarse es transmitir a otras personas información y conocimiento. Para que exista comunicación debe existir un emisor, un receptor, un mensaje y un canal a través del cual se envía el mensaje. Si alguno de estos elementos falla, se dice que se ha producido una interferencia y no podrá establecerse la comunicación. En este capítulo se ofrecen una serie de definiciones relacionadas con el dominio del problema.

1.1. Conceptos asociados al dominio del problema

1.1.1. Interacción

La interacción es un dialogo de comportamiento entre 2 entidades, el accionar de una condiciona la respuesta de la otra. De acuerdo al diccionario de la Real Academia Española (RAE): (2)

Interacción: acción que se ejerce recíprocamente entre 2 o más objetos, agentes, fuerzas, funciones, etc.

La interacción entre un sistema y su usuario se canaliza a través de una interfaz o punto de encuentro. La interfaz hace tangibles las posibilidades del sistema y permite al usuario comunicar sus comandos al sistema. Cuando se habla de interacción en la Web se piensa en intercambio, trato entre un usuario y el sistema, en fin, en un proceso de intercambio entre dos integrantes del proceso comunicativo.

La interacción persona-computadora es una disciplina relativamente joven. Por eso, es habitual que se generalice, se mezclen conceptos y se caiga en errores conceptuales.

Interacción es en definitiva, hacer que la página o sitio responda a las acciones de los usuarios.

La interacción es la posibilidad de que el usuario interactue con la información que le presente el cibermedio. Por tanto el grado de interactividad de un cibermedio, y de cualquier sitio Web en general, puede medirse según el número de cambios que el usuario pueda efectuar en él. (3)

Cuando un sitio Web, o un portal digital, es interactivo se puede modificar la forma en que el usuario se enfrenta a la información o también puede enviar información al sistema y recibir información de acuerdo a los datos que le suministró a este. Es simplemente aprovechar herramientas del sistema para acopiar información requerida según las necesidades particulares de cada usuario.

1.1.2.La televisión

Es un sistema para la transmisión y recepción de imágenes en movimiento y sonido a distancia. Esta transmisión puede ser efectuada mediante ondas de radio o por redes especializadas de televisión por cable. El receptor de las señales es el televisor. La palabra “televisión” es un híbrido de la voz griega “tele” (distancia) y la latina “visio” (visión). El término televisión se refiere a todos los aspectos de transmisión y programación de televisión. A veces se abrevia como TV.

1.1.3.Plataforma de Transmisión

En Telecomunicación, una plataforma es un conjunto de elementos interconectados que se utilizan para transmitir una señal de un lugar a otro. La señal transmitida puede ser eléctrica, óptica o de radiofrecuencia. En Informática, una plataforma es precisamente el basamento, ya sea de hardware o software, sobre el cual pueden realizarse varias tareas.

Se define como plataforma de transmisión, al conjunto de elementos interconectados a partir de la base de intercambios y de conectividad, gestionado por ciertas reglas, ciertas capacidades y ciertas limitaciones en un ambiente determinado, permitiendo la realización

de varias tareas referentes a la acción de transmitir contenidos por un medio; integra la posibilidad de controlar el proceso de transmisión.

Los elementos básicos de cualquier plataforma de transmisión son la pareja multiplexor/demultiplexor que pueden ser analógicos o digitales, los equipos terminales de línea, los repetidores o regeneradores y el sistema de gestión de contenidos.

1.1.4. Plataforma Web

Una plataforma Web es un sistema formado por un conjunto de componentes hardware y software que proporcionan capacidades (servicios) sobre las que se deberá apoyar cualquier aplicación software y cuyo funcionamiento es a través de internet.

El resultado final debe ser un conjunto (plataforma) formado por componentes horizontales o herramientas de colaboración (gestor de usuarios, gestores de contenido y gestores documentales) que pueden reutilizarse en las distintas aplicaciones específicas que se introduzcan.

Según el punto de vista del usuario final, el funcionamiento de una plataforma debe ser muy parecido al de una "gran aplicación" que cubre todas las funcionalidades necesarias.

1.1.5. Aplicaciones Web

El World Wide Web Consortium (W3C) define el término "Web" como: "el universo de información accesible a través de la red". Una aplicación Web es un sistema que permite a un usuario final acceder a una parcela de información contenida en el universo al que hace referencia la anterior definición de W3C. Las aplicaciones Web están basadas en el paradigma "cliente/servidor". Este paradigma consiste en un servidor que sabe cómo proporcionar un servicio y un cliente que desea acceder al servicio. (4)

El incremento del uso de Internet ha aportado ventajas sustanciales en el desarrollo de las aplicaciones Web, los usuarios para poder utilizarlas sólo tienen que acceder a un servidor Web a través de Internet o Intranet. Una característica importante es la facilidad de actualizar y mantener aplicaciones Web sin distribuir o instalar software en miles de

clientes potenciales. El acceso y manejo de la información en forma de aplicación Web asegura una manera rápida, sencilla y concisa de realizarse, donde el usuario no necesariamente debe conocer de procedimientos ni protocolos para ello.

Es vital destacar que no siempre fue así, en los primeros años de inicio de la computación “cliente-servidor”, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados por separado en cada estación de trabajo de los usuarios. Una mejora en el servidor requería una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un coste de soporte técnico y disminuyendo la eficiencia del personal.

1.1.5.1. Ventajas de las aplicaciones Web

La creciente popularidad de las aplicaciones Web se debe a sus múltiples ventajas, entre las cuales se pueden citar:

- ✓ **Multiplataforma:** Con un solo programa, un único ejecutable, las aplicaciones pueden ser utilizadas a través de múltiples plataformas, tanto de hardware como software.
- ✓ **Actualización instantánea:** Debido que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.
- ✓ **Suave curva de aprendizaje:** Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.
- ✓ **Fácil de integrar con otros sistemas:** Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- ✓ **Acceso móvil:** El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una PC portátil o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.

¿Qué se logra con las ventajas de las aplicaciones Web en un sistema como la Plataforma de Transmisión Abierta para Radio y Televisión? Estas ventajas le proporcionan a la plataforma mecanismos de comunicación estándares entre los diferentes módulos dentro de ella, que interactúan entre sí para presentar información dinámica al usuario. Además para proporcionar interoperabilidad y extensibilidad entre estos módulos, para que al mismo tiempo sea posible su combinación para realizar operaciones complejas.

1.1.6.PRIMICIA

PRIMICIA se ha conceptualizado para ser desarrollado y soportado completamente en software libre, contribuyendo de esta manera a la soberanía tecnológica del país. Se encuentra estructurado en dos subsistemas que se relacionan entre sí y actúan como un todo para brindar un resultado final eficiente y acorde a las necesidades de los usuarios. En el Subsistema de Administración, se realiza la administración del canal y toda la gestión de las informaciones y recursos multimedia que se transmiten, y el Subsistema de Transmisión es el encargado de la visualización de las noticias y materiales publicados. Su arquitectura física puede ser adaptada en dependencia del tipo de red de televisión que se desee utilizar para la transmisión del canal y las condiciones del entorno donde se instale, las que pueden ser más sofisticadas o en algún caso estar restringidos por un bajo presupuesto. El canal informativo que soporta la plataforma muestra de forma automática ciclos de noticias constantes y repetitivos condicionados por las informaciones publicadas para determinados períodos de tiempo, logrando integrar en sus transmisiones textos combinados con imágenes y video. Durante las emisiones permite la reproducción de un fondo musical que puede ser personalizado según la noticia que se muestra, además es posible la utilización de cintillos informativos o infocintas que permitan el adelanto de breves asuntos de carácter relevante o promocional.

Como elemento adicional es posible mostrar la fecha y hora, tiempo restante de la noticia o pantalla y titular de la próxima. El sistema ofrece además la transmisión de señales de video externas, tales como filmaciones en vivo o transmisiones de otras televisoras. Es importante señalar que el desarrollo de PRIMICIA se ha enfocado hacia la obtención de funcionalidades genéricas fácilmente escalables, que no dependan de un entorno dado y

no atadas a un diseño gráfico específico, pudiendo acordar con los clientes la personalización o eliminación específica de alguna funcionalidad.

1.1.7.Arquitectura Cliente Servidor

La arquitectura cliente-servidor es una forma específica de diseño de aplicaciones, aunque también se conoce con este nombre a los ordenadores en los que se estas aplicaciones son ejecutadas. Por un lado, el cliente es el ordenador que se encarga de efectuar una petición o solicitar un servicio. El cliente no posee control sobre los recursos, sino que es el servidor el encargado de manejarlos. Por otro lado, el ordenador remoto que actúa como servidor evalúa la petición del cliente y decide aceptarla o rechazarla consecuentemente. Una vez que el servidor acepta el pedido la información requerida es suministrada al cliente que efectuó la petición, siendo este último el responsable de proporcionar los datos al usuario con el formato adecuado. Finalmente se debe precisar que cliente y servidor no tienen que estar necesariamente en ordenadores separados, sino que pueden ser programas diferentes que se ejecuten en el mismo ordenador.

Se hace necesario en el uso de aplicaciones Web tecnologías para el acceso y distribución de los servicios, tecnologías que permitan ejecutar y responder de forma segura y óptima las peticiones de los usuarios por muy complejas que estas puedan ser. Teniendo en cuenta los múltiples servicios que brindan las aplicaciones Web, no solo internamente sino también para el servicio público se recurre a la tecnología cliente-servidor.

Esta tecnología es utilizada por la mayoría de las aplicaciones que hoy coexisten en Internet. La misma permite a una aplicación de usuario (comúnmente llamado cliente) desde un ordenador, hacer una petición de algún servicio suministrado por otro ordenador dedicado (servidor) a esta función mediante una red ya sea local o externa. Estos servicios pueden ser peticiones de datos a una base de datos de información contenida en archivos o los archivos en sí mismos o peticiones de imprimir datos en una impresora asociada. (5)

Un solo servidor puede atender a varios clientes y brindarles la información o pedidos requeridos por los mismos de manera rápida y simultánea, así se ahorra el problema de tener la información instalada y almacenada localmente.

1.1.7.1. Características principales de la tecnología cliente-servidor

La tecnología cliente-servidor refleja diversas características, entre las que cabe señalar más importantes:

- ✓ El Cliente y el Servidor pueden funcionar como una única entidad y también actuar como entidades separadas, haciendo actividades o tareas de manera independiente.
- ✓ Sus funciones pueden estar en plataformas apartadas o en la misma.
- ✓ Un servidor ofrece servicio a variados clientes de forma concurrente.
- ✓ Toda plataforma pudiera ser escalable absolutamente. Las modificaciones realizadas en las plataformas de los clientes o de los servidores, ya sean por modernización o por sustitución tecnológica, se efectúan de manera clara para el usuario final. (5)

1.1.7.2. Principales componentes

Los Clientes son los que interactúan con los usuarios, de forma gráfica usualmente. Se comunican con procesos auxiliares encargados de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad

Los Servidores son los encargados de proporcionar un servicio al cliente y devolver los resultados. En determinados casos se hallan procesos subalternos encomendados a recibir solicitudes del cliente, verificar protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además manejan los interbloqueos, la recuperación ante fallas y otros aspectos afines. Por los elementos antes expuestos la plataforma computacional asociada a los servidores es más eficaz que la de los clientes. Igualmente deben manejar servicios como la administración de la red, la gestión de la entrada al sistema "login", auditorías, recuperación y contabilidad.

Para lograr que los clientes y servidores consigan comunicarse se requiere de una infraestructura de comunicaciones, la cual es la responsable de proporcionar los

mecanismos básicos de direccionamiento y transporte. Los sistemas cliente-servidor actuales, en su mayoría, se centran en las redes locales, por lo cual emplean protocolos no orientados a conexión, esto implica que las aplicaciones deben hacer las verificaciones. La red debe presentar características ajustadas al desempeño, confiabilidad, transparencia y administración.

1.1.7.3. Tipos de servidores

Los sistemas cliente-servidor pueden ser de varios tipos, esto depende principalmente de las aplicaciones que el servidor instala a disposición de los clientes. Entre los que se pueden encontrar:

- ✓ **Servidores de Impresión**, mediante los cuales los usuarios comparten impresoras.
- ✓ **Servidores de Archivos**, con los que los clientes pueden compartir discos duros.
- ✓ **Servidores de Bases de Datos**, donde se puede encontrar una base de datos única.
- ✓ **Servidores de Lotus Notes**, que permiten el trabajo paralelo de distintos clientes con datos semejantes, documentos o modelos.
- ✓ **Servidor de Aplicaciones:** Windows NT, Novell.

Los servidores Web igualmente se emplean en la tecnología cliente-servidor, aunque incorporan aspectos nuevos y propios a la misma. Por esta razón y teniendo en cuenta que la Plataforma de Transmisión Abierta para Radio y Televisión puede ser accedida simultáneamente por un grupo de usuarios solicitando un servicio o recurso ofrecido por un servidor central, se justifica la utilización de esta tecnología para el desarrollo del sistema.

Con su uso se produce un aumento de la interactividad y se facilita el mantenimiento de las aplicaciones. Además que implicaría un mejor manejo de errores e implementación de estrategias para garantizar la integridad de los datos, si se quisiera modificar la infraestructura del sistema podría resultar sencillo pues integrar más ordenadores o sistemas externos no constituiría un problema.

1.1.8. Servidor WEB

Un servidor Web es un programa que implementa protocolos como HTTP (Hypertext Transfer Protocol) y HTTPS (Hypertext Transfer Protocol Secure). HTTP está diseñado para transferir los llamados hipertextos, páginas Web o páginas HTML (Hypertext Markup Language). Está disponible por medio de Internet, emplea normas en XML y sistema de mensajería. Además no está vinculado a ningún sistema operativo o lenguaje de programación. Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que se suele conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al hacer una solicitud mediante el navegador, éste realiza una petición HTTP al servidor de dicha dirección, el servidor responde al cliente enviando el código HTML de la página; el navegador (cliente), una vez recibido el código, lo interpreta y lo muestra en pantalla.

1.1.9. Servicios Web

Existen múltiples definiciones alrededor de los Web Services, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todas sus características y lo que implica su uso. Se podrían ver como un conjunto de aplicaciones o tecnologías que interactúan entre sí en la Web. Estas aplicaciones intercambian datos entre sí con el objetivo principal de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a éstos procedimientos a través de la Web.

La World Wide Web Consortium (*W3C*), define Web Services como:

“...un sistema de software diseñado para soportar la interoperabilidad máquina a máquina sobre una red. Este tiene una interfaz descrita en un formato procesable por una máquina (específicamente WSDL). Otros sistemas interactúan con los servicios web en una manera prescrita por su descripción usando mensajes SOAP, típicamente enviados usando HTTP con una serialización XML...”(4)

Los Web Services son desarrollos de software que permiten el intercambio de datos y funcionalidad de aplicaciones sobre una red. Está soportado en diferentes estándares que garantizan la interoperabilidad entre servicios. Los Web Services utilizan como su gran insumo el lenguaje extensible de marcado XML y se basa en una arquitectura en la que se define el servicio web. A través de uno de los lenguajes se publica en un directorio donde se halla la descripción anteriormente hecha, y se utiliza de acuerdo a las expectativas de resolver una necesidad de acuerdo con la descripción provista. Los Web Services presentan una evolución desde sistemas orientados a objetos en sistemas basados en servicios. Un Servicio Web es una aplicación modular, contenida en sí misma y accesible a través de protocolos web. Es capaz de recibir una petición, activar unos procesos y devolver los resultados. Todo esto en internet, y a través de protocolos de red (HTTP, FTP, SMTP). La comunicación entre los diferentes entornos del Web Services se realiza mediante XML. Los Web Services combinan lo mejor de los desarrollos basados en componentes y la web. Permiten la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como http) y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo. Un Web Services es un contenedor que encapsula funciones específicas y hace que estas funciones puedan ser utilizadas en otros servidores.

Algunas ventajas que presentan los Web Services son:

- Son programables.
- Están basados en XML, que es un lenguaje abierto.
- Son autos descriptivos.
- Pueden buscar registros de otros Web Services.

En los últimos años se ha popularizado un estilo de arquitectura Software conocido como REST (**RE**presentational **S**tate **T**ransfer). Este nuevo estilo ha supuesto una nueva opción de estilo de uso de los servicios web. Los tres estilos de usos más comunes:

- Remote Procedure Calls (RPC, Llamadas a Procedimientos Remotos): Los servicios web basados en RPC presentan una interfaz de llamada de

procedimientos y funciones distribuidas. Típicamente, la unidad básica de este tipo de servicios es la operación WSDL.

- Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA): Los servicios web pueden también ser implementados siguiendo los conceptos de la arquitectura SOA, donde la unidad básica de comunicación es el mensaje, más que la operación. Esto es típicamente referenciado como servicios orientados a mensajes.
- REST (**RE**presentation **St**ate **T**ransfer): fue ganando amplia adopción en toda la web como una alternativa más simple a SOAP y a los servicios web basados en WSDL. REST define principios arquitectónicos por los cuales se diseñan servicios web haciendo énfasis en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes. Una implementación concreta de un servicio web REST sigue algunos principios de diseño como los siguientes:
 - Utiliza los métodos HTTP de manera explícita.
 - Expone URIs con forma de directorios.
 - Transfiere XML, JavaScript Object Notation (JSON), o ambos inclusive. (6)

1.2. Conceptos asociados al intercambio de información en internet

1.2.1. Transmisión de Datos en Internet

Una red de ordenadores permite conectar a los ordenadores que la forman con la finalidad de compartir información, como documentos o bases de datos, o recursos físicos, como impresoras o unidades de disco. Las redes suelen clasificarse según su extensión en:

- LAN (Local Area Network): Son las redes de área local. La extensión de este tipo de redes suele estar restringida a una sala edificio, aunque también podría utilizarse para conectar dos más edificios próximos.

- WAN (Wide Area Network): Son redes que cubren un espacio muy amplio, conectando a ordenadores de una ciudad o un país completo. Para ello se utilizan las líneas de teléfono y otros medios de transmisión más sofisticados, como pueden ser las microondas. La velocidad de transmisión suele ser inferior que en las redes locales.

Varias redes pueden conectarse entre sí formando una red lógica de área mayor. Para que la transmisión entre todas ellas sea posible se emplean los routers, que son los sistemas que conectando físicamente varias redes se encargan de dirigir la información por el camino adecuado. Cuando las redes que se conectan son de diferente tipo y con protocolos distintos se hace necesario el uso de los gateways, los cuales además de encaminar la información también son capaces de convertir los datos de un protocolo a otro. Generalmente los términos router y gateway se emplean indistintamente para referirse de forma general a los sistemas encargados del encaminamiento de datos en Internet.

Lo que se conoce como Internet es en realidad una red de redes, la interconexión de otras redes independientes de manera que puedan compartir información entre ellas a lo largo de todo el planeta. Para ello es necesario el uso de un protocolo de comunicaciones común. El protocolo que proporciona la compatibilidad necesaria para la comunicación en Internet es el TCP/IP.

Una interfaz, sin embargo, es el encargado de la conexión física entre los equipos, definiendo las normas para las características eléctricas y mecánicas de la conexión.

Exceptuando a los routers cualquier ordenador conectado a Internet y, por tanto, capaz de compartir información con otro ordenador se conoce con el nombre de host (anfitrión). Un host debe identificarse de alguna manera que lo distinga de los demás para poder recibir o enviar datos. Para ello todos los ordenadores conectados a Internet disponen de una dirección única y exclusiva. Esta dirección, conocida como dirección de Internet o dirección IP, es un número de 32 bit que generalmente se representa en cuatro grupos de 8 bit cada uno separados por puntos y en base decimal (esto es así en la versión número 4 del protocolo IP, pero no en la 6). Un ejemplo de dirección IP es el siguiente: 205.198.48.1.

1.2.2.El modelo OSI

El modelo OSI (Open System Interconnection) fue creado por el ISO (Organización Internacional de Normalización), y consiste en siete niveles o capas donde cada una de ellas define las funciones que deben proporcionar los protocolos con el propósito de intercambiar información entre varios sistemas. Esta clasificación permite que cada protocolo se desarrolle con una finalidad determinada, lo cual simplifica el proceso de desarrollo e implementación. Cada nivel depende de los que están por debajo de él, y a su vez proporciona alguna funcionalidad a los niveles superiores. Los siete niveles del modelo OSI son los siguientes:

- **Aplicación:** El nivel de aplicación es el destino final de los datos donde se proporcionan los servicios al usuario.
- **Presentación:** Se convierten e interpretan los datos que se utilizarán en el nivel de aplicación.
- **Sesión:** Encargado de ciertos aspectos de la comunicación como el control de los tiempos.
- **Transporte:** Transporta la información de una manera fiable para que llegue correctamente a su destino.
- **Red:** Nivel encargado de encaminar los datos hacia su destino eligiendo la ruta más efectiva.
- **Enlace:** Enlace de datos. Controla el flujo de los mismos, la sincronización y los errores que puedan producirse.
- **Físico:** Se encarga de los aspectos físicos de la conexión, tales como el medio de transmisión o el hardware.

1.2.3.El Protocolo TCP/IP

TCP/IP es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre sí. Hay que tener en cuenta que en Internet se encuentran conectados ordenadores de clases muy diferentes y con hardware y software incompatibles en muchos casos, además de todos los medios y formas posibles de conexión. Aquí se encuentra una de las grandes ventajas del TCP/IP, pues

este protocolo se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware.

TCP/IP no es un único protocolo, sino que es en realidad lo que se conoce con este nombre es un conjunto de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP (Transmission Control Protocol) y el IP (Internet Protocol), que son los que dan nombre al conjunto. En Internet se diferencian cuatro niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

- **Aplicación:** Se corresponde con los niveles OSI de aplicación, presentación y sesión. el protocolo HTTP (Hypertext Transfer Protocol).
- **Transporte:** Coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como TCP y UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos.
- **Internet:** Es el nivel de red del modelo OSI. Incluye al protocolo IP, que se encarga de enviar los paquetes de información a sus destinos correspondientes. Es utilizado con esta finalidad por los protocolos del nivel de transporte.
- **Enlace:** Los niveles OSI correspondientes son el de enlace y el nivel físico. Los protocolos que pertenecen a este nivel son los encargados de la transmisión a través del medio físico al que se encuentra conectado cada host, como puede ser una línea punto a punto o una red Ethernet.

El TCP/IP necesita funcionar sobre algún tipo de red o de medio físico que proporcione sus propios protocolos para el nivel de enlace de Internet. Por este motivo hay que tener en cuenta que los protocolos utilizados en este nivel pueden ser muy diversos y no forman parte del conjunto TCP/IP. Sin embargo, esto no debe ser problemático puesto que una de las funciones y ventajas principales del TCP/IP es proporcionar una abstracción del medio de forma que sea posible el intercambio de información entre medios diferentes y tecnologías que inicialmente son incompatibles.

Para transmitir información a través de TCP/IP, ésta debe ser dividida en unidades de menor tamaño. Esto proporciona grandes ventajas en el manejo de los datos que se transfieren y, por otro lado, esto es algo común en cualquier protocolo de comunicaciones.

En TCP/IP cada una de estas unidades de información recibe el nombre de "datagrama" y son conjuntos de datos que se envían como mensajes independientes.

1.2.4.TCP (Transmission Control Protocol)

El protocolo de control de transmisión (TCP) pertenece al nivel de transporte, siendo el encargado de dividir el mensaje original en datagramas de menor tamaño, y por lo tanto, mucho más manejables. Los datagramas serán dirigidos a través del protocolo IP de forma individual. El protocolo TCP se encarga además de añadir cierta información necesaria a cada uno de los datagramas. Esta información se añade al inicio de los datos que componen el datagrama en forma de cabecera.

La cabecera de un datagrama contiene al menos 160 bit que se encuentran repartidos en varios campos con diferente significado. Cuando la información se divide en datagramas para ser enviados, el orden en que éstos lleguen a su destino no tiene que ser el correcto. Cada uno de ellos puede llegar en cualquier momento y con cualquier orden, e incluso puede que algunos no lleguen a su destino o lleguen con información errónea. Para evitar todos estos problemas el TCP numera los datagramas antes de ser enviados, de manera que sea posible volver a unirlos en el orden adecuado. Esto permite también solicitar de nuevo el envío de los datagramas individuales que no hayan llegado o que contengan errores, sin que sea necesario volver a enviar el mensaje completo.

1.2.5.Servicios de Internet: el nivel de aplicación

Los diferentes servicios a los que se puede tener acceso en Internet son proporcionados por los protocolos que pertenecen al nivel de aplicación. Estos protocolos forman parte del TCP/IP y deben aportar entre otras cosas una forma normalizada para interpretar la información, ya que todas las máquinas no utilizan los mismos juegos de caracteres ni los mismos estándares. Los protocolos de los otros niveles sólo se encargan de la transmisión de información como un bloque de bits, sin definir las normas que indiquen la manera en que tienen que interpretarse esos bits. Los protocolos del nivel de aplicación

están destinados a tareas específicas, algunos de los cuales se consideran como tradicionales de Internet por utilizarse desde los inicios de la red, como son por ejemplo:

- Transferencia de ficheros (File Transfer).
- Correo electrónico (e-mail).
- Conexión remota (remote login).

1.2.6. Transferencia de ficheros

El protocolo FTP (File Transfer Protocol) se incluye como parte del TCP/IP, siendo éste el protocolo de nivel de aplicación destinado a proporcionar el servicio de transferencia de ficheros en Internet. El FTP depende del protocolo TCP para las funciones de transporte, y guarda alguna relación con TELNET (protocolo para la conexión remota).

El protocolo FTP permite acceder a algún servidor que disponga de este servicio y realizar tareas como moverse a través de su estructura de directorios, ver y descargar ficheros al ordenador local, enviar ficheros al servidor o copiar archivos directamente de un servidor a otro de la red. Lógicamente y por motivos de seguridad se hace necesario contar con el permiso previo para poder realizar todas estas operaciones. El servidor FTP pedirá el nombre de usuario y clave de acceso al iniciar la sesión, que debe ser suministrado correctamente para utilizar el servicio.

La manera de utilizar FTP es por medio de una serie de comandos, los cuales suelen variar dependiendo del sistema en que se esté ejecutando el programa, pero básicamente con la misma funcionalidad. Existen aplicaciones de FTP para prácticamente todos los sistemas operativos más utilizados, aunque hay que tener en cuenta que los protocolos TCP/IP están generalmente muy relacionados con sistemas UNIX. Por este motivo y, ya que la forma en que son listados los ficheros de cada directorio depende del sistema operativo del servidor, es muy frecuente que esta información se muestre con el formato propio del UNIX. También hay que mencionar que en algunos sistemas se han desarrollado clientes de FTP que cuentan con un interfaz gráfico de usuario, lo que facilita notablemente su utilización, aunque en algunos casos se pierde algo de funcionalidad.

Existe una forma muy utilizada para acceder a fuentes de archivos de carácter público por medio de FTP. Es el acceso FTP anónimo, mediante el cual se pueden copiar ficheros de los hosts que lo permitan, actuando estos host como enormes almacenes de información y de todo tipo de ficheros para uso público. Generalmente el acceso anónimo tendrá algunas limitaciones en los permisos, siendo normal en estos casos que no se permita realizar acciones tales como añadir ficheros o modificar los existentes. Para tener acceso anónimo a un servidor de FTP hay que identificarse con la palabra "anonymous" como el nombre de usuario, tras lo cual se pedirá la contraseña o clave correspondiente. Normalmente se aceptará cualquier cadena de caracteres como clave de usuario, pero lo usual es que aquí se indique la dirección de correo electrónico propia, o bien la palabra "guest". Utilizar la dirección de correo electrónico como clave de acceso es una regla de cortesía que permite a los operadores y administradores hacerse una idea de los usuarios que están interesados en el servicio, aunque en algunos lugares puede que se solicite esta información rechazando el uso de la palabra "guest".

El FTP proporciona dos modos de transferencia de ficheros: ASCII y binario. El modo de transferencia ASCII se utiliza cuando se quiere transmitir archivos de texto, ya que cada sistema puede utilizar un formato distinto para la representación de texto. En este caso se realiza una conversión en el formato del fichero original, de manera que el fichero recibido pueda utilizarse normalmente. El modo de transferencia binario se debe utilizar en cualquier otro caso, es decir, cuando el fichero que vamos a recibir contiene datos que no son texto. Aquí no se debe realizar ninguna conversión porque quedarían inservibles los datos del fichero.

1.3. Interacción entre sistemas informáticos

1.3.1. Proceso de interacción entre sistemas informáticos

La interacción entre sistemas de información o plataformas web es una necesidad creciente en la mayoría de los sitios especializados en brindar información fresca a los usuarios, poder determinar las líneas de conexión con diversos servicios web posibilitaría que no tengan que ser escritos directamente en el código fuente y en cambio estos se almacenen en una base de datos para su posterior gestión.

Un sistema informático como todo sistema, es el conjunto de partes interrelacionadas, hardware, software y de recurso humano. Un sistema informático típico emplea una computadora que usa dispositivos programables para capturar, almacenar y procesar datos. La computadora personal o PC, junto con la persona que lo maneja y los periféricos que los envuelven, resultan de por sí un ejemplo de un sistema informático. (5)

Los diseñadores de sistemas informáticos no necesariamente esperan que sus sistemas se puedan interconectar con otros sistemas. Por otro lado, los técnicamente eruditos a menudo pueden configurar sistemas diferentes para que se puedan comunicar entre sí usando un conjunto de reglas y restricciones conocidas como protocolos. Los protocolos tratan precisamente de definir la comunicación dentro de y entre sistemas informáticos distintos pero conectados entre sí. Si dos sistemas informáticos usan el mismo protocolo, entonces podrán ser capaces de interconectarse y formar parte de un sistema mayor.

La posibilidad de contar con que un sistema tenga información actualizada de otros sistemas informáticos, proveerá de actualidad y frescura a la plataforma, esta información será encontrada y almacenada en una base de datos para su posterior gestión por parte de los encargados de publicarla. Configurar las conexiones con diferentes plataformas con las cuales se conectará el sistema permitirá tener una gama de información adicional para beneficio del cliente.

La interacción también incluye la posibilidad de que otras plataformas puedan hacerse eco de información publicada en PRIMICIA favoreciendo el intercambio y logrando que el objetivo de masificar la información sea una realidad.

En la UCI se han desarrollado sistemas informativos automatizados que han sido de gran utilidad para la propia universidad (*Señal 3*) y para otras entidades como el Sistema Automatizado de Teletexto para la Plataforma de Televisión Digital Satelital Cubana, *Señal ACN*, administrado por la Agencia de Cubana de Noticias (ACN). Estos sistemas han solucionado problemas a la medida de estas entidades posibilitando la transmisión de noticias televisivas de una forma organizada y automatizada. A partir de estas experiencias se ha logrado conceptualizar y desarrollar el producto informático PRIMICIA, solución integral que permite proveer un canal de televisión informativa completamente automatizado.

1.3.2. Descripción actual del proceso de interacción entre sistemas informáticos

PRIMICIA cuenta con numerosas funcionalidades genéricas basadas en la experiencia alcanzada por el grupo de desarrolladores durante varios años, sin embargo a causa de la escasa variedad práctica y problemática en la que están basadas las mismas, muchas de ellas carecen de una total generalidad para todos los entornos en los cuales se proyecta el uso del sistema. Uno de los puntos más débiles que presenta el sistema es la no presencia de interacción de la plataforma con otros sistemas WEB o sistemas de televisión. Por tal motivo se debe repensar un mecanismo que logre incorporar funcionalidades y características necesarias para que se logre una interacción con otros sistemas.

Contar con una interacción entre sistemas informáticos permitiría poder ofrecer un mayor cúmulo de información del interés de los clientes, además de la consabida actualización de la información. Con el objetivo de configurar la cantidad de televisoras con las cuales se podrían conectar la plataforma para lograr una interactividad

La situación actual de la Plataforma de Televisión Informativa PRIMICIA está muy lejos de lo necesitado realmente pues no se cuenta con interfaces de interacción con plataformas externas, por lo que el intercambio de información y servicios en ambas direcciones no se hace posible, funcionalidad que dotaría a la plataforma de cierto dinamismo con el cual se actualizaría la información que se muestra a los usuarios y esta sería de una amplia gama de intereses.

1.4. Situación problemática

La necesidad de los seres humanos de comunicarse e intercambiar información de forma continua ha llevado a desarrollar medios de comunicación como la radio y la televisión según el avance de la tecnología de la época. Hay situaciones especiales en que es necesario llevar a una gran masa de personas un conjunto de información en diferentes formatos: texto, imagen, audio o video. Este es el caso de un entorno cerrado donde no

es posible emplear los medios tradicionales de comunicación, la situación se enmarca en sedes ministeriales o empresariales, universidades, hoteles, aeropuertos, entre otros.

El avance continuo de las Tecnologías de la Información y la Comunicación (TIC) ha contribuido al desarrollo de sistemas de producción basado en el tratamiento informático de la señal de televisión. Los sistemas de almacenamiento, como los magnetoscopios, pasaron a ser sustituidos por servidores informáticos de video y los archivos pasaron a guardar sus informaciones en discos duros y cintas de datos. La existencia de los servidores de video posibilita la automatización de las emisiones y de los programas informativos mediante la realización de listas de emisión.

De acuerdo al profundo y novedoso proceso de transformaciones educacionales y sociales que surgieron como parte del programa de la Batalla de Ideas que se ha estado llevando a cabo en todo el país, surgen en marzo de 2002 las primeras ideas sobre lo que se denominó inicialmente Proyecto Futuro y que más tarde se vería concretado en la creación de la UCI. En la Universidad, y prácticamente desde sus inicios, se hizo necesaria la creación de un proyecto que se dedicara a los temas referentes a la televisión moderna y que ofreciera servicios básicos que permitieran satisfacer las necesidades del departamento de tele-formación. Con estos objetivos surge el proyecto UCITeVe, que en muy poco tiempo se hace de una buena reputación en toda la comunidad universitaria en cuanto a la generación y procesamiento de medias, y que constituye una cantera para la vinculación de estudiantes a otros proyectos de este tipo.

Durante varios años de trabajo se han obtenido resultados satisfactorios con soluciones a la medida como Señal 3 y Señal ACN, desarrollando además la Plataforma de Televisión Informativa, PRIMICIA, una solución integral para estos fines.

PRIMICIA se ha conceptualizado como una solución integral capaz de proveer un canal de televisión informativa que integra informaciones en formato texto, imagen, audio y video, posibilitando la transmisión automática de las informaciones de acuerdo a las publicaciones realizadas a través de la administración del sistema. Esta solución la cual se encuentra en su segunda fase de desarrollo tiene entre sus prioridades la necesidad de interacción externa con servicios ofrecidos por la web que permita recibir y exportar información desde y hacia otros sistemas. De acuerdo a esta necesidad se ha planteado que se debe contar con un diseño que permita esta funcionalidad y que no necesariamente tenga que ser escrito en el código fuente sino que esta sería guardada en

una base de datos para su posterior gestión. Esta información sería utilizada para mantener actualizada la plataforma y así prestar un mejor servicio a los usuarios con información de otra índole a la que se trata asiduamente en la plataforma.

1.5. Análisis de otras Soluciones

EL portal MHP (Multimedia Home Platform) de la TDT (Televisión Digital Terrestre) de Navarra provincia de España ofrece funcionalidades al Gobierno de la Provincia en cuanto a los Servicios de Hacienda para el pago de la renta el cual ofrece varios servicios como:

- Servicio "Cuando": servicio por el cual se notificaba al contribuyente cuando se produciría la devolución de Hacienda, si ese era el caso, y de cuanto sería el importe. Por descarte, este mismo servicio podía indicar, en función del valor positivo o negativo del importe, si no se producía ninguna devolución, es decir, a pagar por el contribuyente a Hacienda.
- Servicio "Enviada": este otro servicio indicaba si se había enviado el borrador de la declaración por correo postal al contribuyente, y si no se había realizado, las causas.

Los cuales se realizarían mediante el uso de Servicios Web. Se debía identificar las comunicaciones entre la aplicación MHP en TDT y los servicios web en el back-office del Gobierno de Navarra. En dicha comunicación debían establecerse protocolos (SOAP = Simple Object Access Protocol, protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML), parámetros y formato de la información (DNI, fechas, cadenas de texto, etc.). Era obvio que muchos de estos aspectos venían determinados por los propios servicios web ya existentes. Para este análisis se contaba con las especificaciones WSDL (Web Services Description Language, un formato XML que se utiliza para describir servicios Web) de dichos servicios web. En esta Plataforma de televisión se usan los Servicios Web para implementar un servicio al cliente en cual este interactúa con los Servicios de Hacienda para el pago de la renta. Los servicios web determinarían la conexión entre la MHP y los servicios de la oficina de Hacienda.

1.6. Conclusiones

En este capítulo se han sistematizado los elementos teóricos que sustentan el problema científico y los objetivos de este trabajo. Mediante un estudio crítico y valorativo se han sentado las bases teóricas del estudio realizado. Se abordaron conceptos básicos como plataforma web e interacción, se analizó la necesidad que presenta la Plataforma de Televisión Informativa PRIMICIA por lo cual se hace inminente una solución.

Los temas relacionados en este capítulo permitirán un mejor entendimiento de la situación que llevó a la realización del siguiente trabajo, siendo muy útil para el entendimiento de los capítulos siguientes.

Modelos de intercambio de información

2. Introducción

La gran rapidez con la que Internet se ha expandido y popularizado en los últimos años ha supuesto una revolución muy importante en el mundo de las comunicaciones, llegando a causar cambios en muchos aspectos de la sociedad. Lo que se conoce hoy como Internet es en realidad un conjunto de redes independientes (de área local y área extensa) que se encuentran conectadas entre sí, permitiendo el intercambio de datos y constituyendo por lo tanto una red mundial que resulta el medio idóneo para el intercambio de información, distribución de datos de todo tipo e interacción.

2.1. Descripción de los modelos de Intercambio de Información entre Aplicaciones de escritorio.

2.1.1.Sockets

El origen de los socket tuvo lugar en una variante del sistema operativo Unix conocida como BSD Unix. En la universidad de Berkeley, en los inicios del Internet, pronto se hizo evidente que los programadores necesitarían un medio sencillo y eficaz para escribir programas capaces de intercomunicarse entre sí. Esta necesidad dio origen a la primera especificación e implementación de sockets. (7)

Una forma de conseguir que dos programas se transmitan datos, basada en el protocolo TCP/IP, es la programación de sockets. Un socket no es más que un "canal de comunicación" entre dos programas que corren sobre ordenadores distintos o incluso en el mismo ordenador. (8)

Desde el punto de vista de programación, un socket no es más que un "fichero" que se abre de una manera especial. Una vez abierto se pueden escribir y leer datos de él con las habituales funciones de `read ()` y `write ()` del lenguaje C.

Existen básicamente dos tipos de "canales de comunicación" o sockets, los orientados a conexión y los no orientados a conexión.

En el primer caso ambos programas deben conectarse entre ellos con un socket y hasta que no esté establecida correctamente la conexión, ninguno de los dos puede transmitir datos. Esta es la parte TCP del protocolo TCP/IP, y garantiza que todos los datos van a llegar de un programa al otro correctamente. Se utiliza cuando la información a transmitir es importante, no se puede perder ningún dato y no importa que los programas se queden "bloqueados" esperando o transmitiendo datos. Si uno de los programas está atareado en otra cosa y no atiende la comunicación, el otro quedará bloqueado hasta que el primero lea o escriba los datos.

En el segundo caso, no es necesario que los programas se conecten. Cualquiera de ellos puede transmitir datos en cualquier momento, independientemente de que el otro programa esté "escuchando" o no. Es el llamado protocolo UDP, y garantiza que los datos que lleguen son correctos, pero no garantiza que lleguen todos. Se utiliza cuando es muy importante que el programa no se quede bloqueado y no importa que se pierdan datos. Por ejemplo, un programa que está controlando la temperatura de un horno y envía dicha temperatura a un ordenador en una sala de control para que éste presente unos gráficos de temperatura. Obviamente es más importante el control del horno que el perfecto refresco de los gráficos. El programa no se puede quedar bloqueado sin atender al horno simplemente porque el ordenador que muestra los gráficos esté ocupado en otra cosa.

Definición de Socket: designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada.

Los sockets proporcionan una comunicación de dos vías, punto a punto entre dos procesos. Los sockets son muy versátiles y son un componente básico de comunicación interprocesos e intersistemas. Un socket es un punto final de comunicación al cual se puede asociar un nombre.

2.1.2.RPC (Llamada a Procedimientos Remotos)

RPC es un estándar desarrollado por Sun Microsystems y usado por muchos distribuidores de sistemas UNIX. La especificación actual de UNIX se halla en el RFC 1057 - RPC ("Remote Procure Call"): especificación de protocolo de la versión 2. El RPC es una interfaz de programación de aplicación (API) disponible para el desarrollo de aplicaciones distribuidas. Permite que los programas llamen a subrutinas que se ejecutan en un sistema remoto. El programa llamador, denominado client envía un mensaje de llamada al proceso servidor y espera por un mensaje de respuesta. La llamada incluye los parámetros del procedimiento y la respuesta los resultados. (9)

El RPC de Sun consta de las siguientes partes:

- *RPCGEN*: Un compilador que toma la definición de la interfaz de un procedimiento remoto, y genera los "stubs" del cliente y del servidor.
- *XDR ("eXternal Data Representation")*: Una forma estándar de codificar datos de modo para que sean transportables entre distintos sistemas. Impone una ordenación big - endian de los bytes y el tamaño mínimo de cualquier campo ha de ser 32 bits. Esto significa que tanto el cliente como el servidor han de realizar algún tipo de traducción.
- Una librería "run-time".

El concepto de RPC se puede simplificar del modo siguiente:

- El proceso llamador envía un mensaje de llamada y espera por la respuesta.
- En el lado del servidor un proceso permanece dormido a la espera de mensajes de llamada. Cuando llega una llamada, el proceso servidor extrae los parámetros del procedimiento, calcula los resultados y los devuelve en un mensaje de respuesta.

Las llamadas a procedimientos remotos difieren de las llamadas a procedimientos locales en los siguientes aspectos: (10)

- Uso de variables globales ya que el servidor no tiene acceso al espacio de memoria del llamador.
- El rendimiento puede verse afectado por los tiempos de transmisión.

- Puede ser necesaria la autenticación del usuario.
- Se debe conocer la dirección del servidor.

2.1.2.1. Transporte

El protocolo RPC se puede implementar sobre cualquier protocolo de transporte. En el caso de TCP/IP, puede usar tanto TCP como UDP como capa de transporte. El tipo de transporte es un parámetro del comando RPCGEN. En caso de que se use UDP, recuérdese que no proporciona fiabilidad, por lo que dependerá del programa llamador garantizarla (usando tiempos límite y retransmisiones, implementadas normalmente en rutinas de librería e RPC). Cabe señalar que incluso con TCP, el programa llamador sigue necesitando una rutina para el tiempo límite con el fin de tratar situaciones excepcionales, como por ejemplo la caída del servidor.

Los mensajes de llamada y respuesta se formatean al estándar XDR.(10)

2.2. Intercambio de Información entre Plataformas Web

2.2.1. Web Services (Servicios Web)

Un Servicio Web es un componente de software que se comunica con otras aplicaciones codificando los mensaje en XML (eXtensible Markup Language) y enviando estos mensaje a través de protocolos estándares de Internet tales como el Hypertext Transfer Protocol (HTTP). Intuitivamente un Web Services es similar a un sitio web que no cuenta con un interfaz de usuario y que da servicio a las aplicaciones en vez de a las personas. Un Web Services, en vez de obtener solicitudes desde el navegador y retornar páginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado en XML desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en XML. (11)

Microsoft y otras empresas líderes están promocionando SOAP como estándar de los mensajes para los Web Services. Un mensaje SOAP se parece mucho a una carta: es un

sobre que contiene una cabecera con la dirección del receptor del mensaje, un conjunto de opciones de entrega (tal como la información de encriptación), y un cuerpo o body con la información o data del mensaje.

2.2.1.1. ¿Cómo funciona un Web Services?

Para establecer un diálogo coherente entre el Web Services Cliente (WSC), que envía la petición y recibe la respuesta, y el Web Services Servidor (WSS), el que ejecuta el proceso y envía la respuesta, se utiliza Simple Object Access Protocol (SOAP), que es una codificación basada en XML.

Un Web Services en vez de obtener peticiones desde un navegador y devolver páginas web como respuesta, recibe peticiones mediante un mensaje formateado con SOAP, desde otras aplicaciones realiza la labor que le han pedido y devuelve un mensaje de respuesta también con formato SOAP.

2.2.1.2. Requerimientos de los Web Services

Independientemente de la arquitectura y lenguajes de programación utilizados tanto en el Web Services Cliente como en el Web Services Servidor, los dos tendrán que ser capaces de enviar, recibir e interpretar SOAP (Simple Object Access Protocol). Esto se consigue gracias a las diferentes librerías u objetos dedicados a los Web Services y basados en tecnologías como .NET, JAVA, PHP o PERL. Además, se requiere obtener y ofrecer la descripción de los Web Services (WSDL, Web Services Description Language). Y finalmente, se necesita obtener y dar información sobre qué hace y dónde está el Web Services (UDDI, Universal Description Discovery and Integration).

2.2.1.3. Tecnologías que se utilizan en Web Services

El modelo de Servicios Web está basado en el uso de ciertas tecnologías emergentes que son el resultado del trabajo de varias compañías y organizaciones entre las cuales se

destacan IBM y Microsoft. Estas tecnologías son: SOAP, WSDL y UDDI, usando lenguaje XML.

XML

Acrónimo de **eXtensible Markup Language**, una forma condensada de SGML (Standard Generalized Markup Language), XML permite a los desarrolladores Web y a los diseñadores crear etiquetas personalizadas que ofrecen gran flexibilidad para organizar y presentar información que antes era imposible de realizar con antiguo sistema de codificación de documentos HTML. La especificación XML se publicó en forma de borrador por un grupo de trabajo del consorcio del World Wide Web y es soportada por varias compañías líderes en la industria de computadoras. (12)

SOAP

(Simple Object Access Protocol, Protocolo de Acceso Simple a Objetos). Es un protocolo que proporciona un mecanismo estándar de empaquetar mensajes. Este protocolo está pensado para el intercambio de información en entornos descentralizados y distribuidos. Usa las tecnologías relacionadas con XML a fin de definir un marco de trabajo extensible para los mensajes. Provee una estructura de mensajes capaz de ser intercambiada sobre una gran cantidad de protocolos de soporte. Este marco ha sido diseñado con el fin de que fuera independiente de cualquier modelo de programación y otras implementaciones semánticas. (12)

WSDL

(Web Services Definition Language, Lenguaje de descripción de servicios web). Especificación XML para la formación del documento de descripción de un servicio web. Identifica los métodos, funciones y parámetros necesarios para invocar un determinado servicio. Así, un usuario puede crear una aplicación cliente que comunica con el servicio web. (12)

UDDI

Universal Description, Discovery and Integration (UDDI), traducido al español Descripción, Descubrimiento e Integración Universal, es un elemento básico sobre el que se asientan los Web Services, hace posible que empresas puedan tanto publicar como encontrar servicios web. UDDI provee un mecanismo para que los negocios se "describan" a sí mismos y los tipos de servicios que proporcionan y luego se pueden registrar y publicarse en un Registro UDDI. Tales negocios publicados pueden ser buscados, consultados o "descubiertos" por otros negocios utilizando mensajes con SOAP. (13)

Los datos manejados por UDDI se dividen en tres categorías:

- **Páginas Blancas:** Con información general sobre una empresa (nombre, descripción, información de contacto, dirección y teléfono).
- **Páginas Amarillas:** 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.
- **Páginas Verdes:** Con información técnica sobre un servicio web. Generalmente esto incluye un apuntador a la especificación externa y una dirección en la que invocar el servicio.

2.2.1.4. Especificación en los Web Services

Las especificaciones en los Web Services en sus siglas (WS-*) definen un conjunto de características para hacer a los software "seguros, confiables y transaccionales". Son el conjunto de especificaciones que se han publicado entorno a los servicios web. Tienen como propósito establecer una serie de estándares para Servicios Web Corporativos.

Algunas de estas especificaciones son:

WS-Security: Cubre un conjunto estándar de extensiones SOAP que permiten hacer más seguro un servicio web para garantizar la integridad y la confidencialidad.

WS-Security aborda el tema de la seguridad haciendo uso de las especificaciones y los estándares existentes.

Brinda mecanismos para lograr:

- Integridad del mensaje: usando firmas digitales para asegurar que el mensaje no fue cambiado.
- Confidencialidad: Encriptación XML para asegurar que sólo quien debe recibir el mensaje puede entenderlo.
- Autenticación y Autorización: Identificación de quien envía el mensaje y que derechos tiene para realizar operaciones.

WS-Addressing: provee elementos para identificar Web Services y asegurar la transmisión de mensajes de extremo a extremo de la comunicación. Habilita a los sistemas de mensajería para soportar transmisiones de mensajes a través de redes que incluyen nodos de procesamiento como firewalls y gateways de una forma neutra con respecto a la tecnología de transporte que se utilice.

WS-I (Web Services Interoperability): Se enfoca en la interoperabilidad de los servicios web entre diferentes plataformas, sistemas operativos y lenguajes.

WS-I Basic Profile: Define buenas prácticas en torno a SOAP, WSDL o UDDI para garantizar interoperabilidad y evitar implementaciones deficientes.

WS-ReliableMessaging: Esta especificación tiene como fin asegurar la entrega de mensajes independientemente de que existan fallas en la comunicación. **WS-RM** agrega confiabilidad al intercambio de mensajes entre Web Services.

WS-Policy: Su objetivo es poder definir en un servicio políticas y restricciones para que éste pueda ser invocado. Por citar un ejemplo de política, es muy común que al publicar un servicio se exijan ciertas políticas de seguridad para su invocación, como puede ser: exigir que los mensajes recibidos lleguen firmados y/o encriptados, en caso de que no sea así, el Web Services no responderá al requerimiento.

WS-Transaction: Es un estándar OASIS en el cual se define una familia de estándares con el objetivo principal de lograr transaccionalidad a nivel de Web Services. Entre ellos están los siguientes:

- **WS-AtomicTransaction:** En este estándar se definen protocolos para actividades de corta duración. El modelo es muy similar a lo que son las transacciones en los manejadores de bases de datos, pero en un ambiente de Web Services. Una secuencia de invocaciones a diferentes Web Services podrán ejecutar en un

contexto transaccional, y si en alguna invocación se produce alguna anomalía, se deshacen el resto de las invocaciones.

- **WS-BusinessActivity:** Su objetivo se enfoca en actividades de negocio de larga duración, a diferencia del estándar anterior, estas actividades pueden extenderse durante días y hasta semanas. Dada la naturaleza temporal de estas actividades, las transacciones en un contexto así no son manejadas de la misma forma que WS-AtomicTransaction, por lo tanto el protocolo provee mecanismos de compensación ante situaciones anómalas.
- **WS-Coordination:** Define un framework extensible para proveer protocolos de coordinación. Permite a un servicio poder crear un contexto en el cual se puedan propagar actividades a otros servicios. Se definen protocolos para que los servicios puedan cooperar y trabajar conjuntamente bajo un cierto contexto. Los estándares WS-AtomicTransaction y WS-BusinessActivity se apoyan en WS-Coordination a la hora de definir protocolos y contextos de ejecución.

2.2.2. Really Simple Syndication RSS (Sindicación Realmente Simple)

RSS es una forma muy sencilla para poder recibir, directamente en un ordenador o en una página web online (a través de un lector RSS) información actualizada sobre páginas web, sin necesidad de tener que visitarlas una a una. Esta información se actualiza automáticamente, sin tener que hacer nada. Para recibir las noticias RSS la página deberá tener disponible el servicio RSS y debe tener un lector RSS.

Si existen varias páginas web que interesan que van actualizando sus contenidos, un lector RSS ahorrará mucho tiempo en esta tarea. Gracias al RSS, no se tendrá que visitar cada una de las páginas web que interesan para ver si han añadido o no algún artículo que pueda interesar. Estas páginas informarán (a través del lector RSS). Cuando ingreses al Lector RSS (o RSS Reader), se está automáticamente informado sobre todas las novedades que se han producido en todas las páginas web que se han dado de alta.(14)

Los sistemas RSS tienen muchas ventajas que se pueden resumir. Gracias al RSS, se puede reunir en un mismo lugar y a un solo clic de distancia, toda la información actualizada de las páginas web (Fuentes o canales RSS) que más interesa.

El RSS facilita la gestión y publicación de información y noticias webs. RSS es una forma estandarizada de distribución de la información de las páginas web a los lectores de las páginas. Esta información se distribuye a través de las fuentes RSS o Canales RSS. Gracias al RSS, los lectores pasan a tener una herramienta útil para mantenerse informado sobre las noticias y webs que le resultan de interés, conservando y almacenando toda la información en un solo lugar que se actualiza de manera automática.(15)

Además, el RSS brinda al lector un importante ahorro en tiempo en la lectura de información y noticias, ya que sólo con abrir el lector RSS (ya sea un programa, el navegador o un lector web online), el usuario podrá ver cuáles son las últimas actualizaciones y noticias que han publicado las diferentes páginas web a las que está suscrito.

2.2.2.1. Ventajas del RSS

Las ventajas que ofrecen los Sistemas RSS son muchas.

Se pueden destacar las siguientes:(14)

- ✓ Las páginas web y blogs distribuyen a través de los canales RSS las últimas actualizaciones de aquellas páginas web que son de su interés.
- ✓ La decisión está del lado del usuario ya que él es quien elige a qué páginas web suscribirse y cuando darse de baja de estas páginas web.
- ✓ El RSS supone un importante ahorro en el tiempo de navegación y búsqueda de información. En el lector RSS, el usuario tendrá un resumen de los artículos para poder decidir qué información quiere leer.

- ✓ El RSS está libre de SPAM, porque no es necesario brindar el correo electrónico. Esto no ocurre con suscripciones por correo electrónico, en las que además de recibir noticias, podrías recibir también SPAM u otra información no deseada. Cuando se suscribe a las fuentes RSS de una página web, no se recibirá otra información que la que se publique en las páginas web que son del interés.
- ✓ La cancelación de la suscripción a la página web será rápida y sencilla. En las suscripciones vía correo electrónico, a menudo el suscriptor tiene que especificar las razones por las que quiere darse de baja y luego debe confirmar su petición. En cambio, con el RSS sólo se debe eliminar la página web del lector de RSS. Es así de sencillo.

Todas estas ventajas se resumen en: los sistemas RSS facilitan enormemente el acceso a aquella información de internet que más interesa y permite estar permanentemente informado.

2.3. Intercambio de Información entre una Aplicación de escritorio y una Plataforma Web

2.3.1. Internet Communication Engine ICE (Motor de Comunicaciones por Internet)

El aumento de middleware orientado a objetos a mediados de los noventa fue un importante paso adelante hacia la fabricación de computación distribuida a disposición de los desarrolladores de aplicaciones. Por primera vez, fue posible la construcción de aplicaciones distribuidas, la plataforma de middleware se encarga de la mayoría de las tareas en red, tales como el cálculo de referencias y unmarshaling (codificación y decodificación de datos para la transmisión), el mapeo de direcciones lógicas de objetos a los extremos de transporte físico, el cambio de la representación de los datos conforme a la arquitectura de la máquina nativa de cliente a servidor, y automáticamente el inicio de servidores de la demanda.(16)

A pesar de estos avances, las plataformas de middleware orientadas a objeto sufrían una serie de graves limitaciones prácticas que llevaron a ZeroC a desarrollar el motor de comunicaciones por Internet, o ICE.

Los objetivos de diseño principales de ICE son:

- Proporcionar una plataforma de middleware orientado a objeto adecuado para su uso en entornos heterogéneos. (16)
- Proporcionar un conjunto completo de características que apoyan el desarrollo de aplicaciones distribuidas realista para una amplia variedad de dominios. (16)
- Evite la complejidad innecesaria, haciendo que la plataforma fácil de aprender y de usar. (16)
- Proporcionar una implementación que sea eficiente en el ancho de banda de red, el uso de memoria y CPU. (16)
- Proporcionar una aplicación que se ha construido en seguridad, lo que es adecuado para su uso en las redes públicas inseguras. (16)

Para ser más simplista, los objetivos de diseño de ICE podrían ser declarado como "Vamos a construir una plataforma de middleware más poderosa que haga la vida del desarrollador más fácil y evite los errores de sus predecesores".

ICE es una plataforma de middleware orientada a objeto. Fundamentalmente, esto significa que ICE proporciona herramientas, APIs, y el apoyo de bibliotecas para la construcción de aplicaciones cliente-servidor orientadas a objeto. Aplicaciones de ICE son adecuados para uso en entornos heterogéneos: cliente y el servidor pueden estar escritos en diferentes lenguajes de programación, puede ejecutarse en diferentes sistemas operativos y arquitecturas de máquina, y puede comunicarse con una variedad de tecnologías de red. El código fuente para estas aplicaciones es portátil, independientemente del entorno de despliegue.(17)

Los términos cliente y servidor no son firmes designaciones para partes particulares de una aplicación; sin embargo, ellas denotan roles que son tomados por partes de una aplicación durante la duración de un pedido:

- Los clientes son entidades activas. Estos envían pedidos de servicio a los servidores.
- Los servidores son entidades pasivas. Estos proveen servicios en respuesta a las peticiones de los clientes.

2.3.1.1. El Protocolo de ICE

ICE proporciona un protocolo RPC que puede utilizar TCP / IP o UDP como el transporte subyacente. Además, ICE también le permite utilizar SSL como transporte, por lo que todos los comunicación entre el cliente y el servidor es encriptado.

El Protocolo ICE define:

- Una serie de tipos de mensajes, tales como la solicitud y la respuesta de los tipos de mensajes.
- Una máquina de estado de protocolo que determina en qué tipos de secuencias diferentes de mensajes se intercambian por el cliente y el servidor, junto con el establecimiento asociado de conexiones y desmontaje de semántica para TCP / IP.
- Reglas de codificación que determinan cómo cada tipo de datos se representa en el alambre.
- Un encabezado para cada tipo de mensaje que contiene detalles como el tipo de mensaje, el tamaño del mensaje, y el protocolo y la codificación de la versión en uso.

ICE también soporta la compresión en el alambre: mediante el establecimiento de un parámetro de configuración, usted puede organizar para todo el tráfico de red para ser comprimido para ahorrar ancho de banda. Esto es útil si su aplicación intercambios de grandes cantidades de datos entre el cliente y el servidor.(16)

El protocolo de ICE es muy adecuado para la creación de mecanismos de transmisión eficiente de eventos, ya que permite el reenvío de un mensaje sin el conocimiento de los detalles de la información dentro de un mensaje. Esto significa que los conmutadores de

mensajes no es necesario hacer ningún cambio en los mensajes, se puede reenviar un mensaje, simplemente tratarlo como un buffer de opaco de bytes.

El protocolo de ICE también apoya la operación bidireccional: si un servidor quiere enviar un mensaje a un objeto de devolución de llamada proporcionada por el cliente, la devolución de llamada se pueden hacer por la conexión que fue creado originalmente por el cliente. Esta característica es especialmente importante cuando el cliente está detrás de un firewall que permite conexiones de salida, pero no de entrada.

2.3.1.2. Servicios de ICE

El núcleo de ICE ofrece al cliente una sofisticada plataforma cliente-servidor, para el desarrollo de aplicaciones distribuidas. Sin embargo, las aplicaciones realistas por lo general requieren más que sólo una capacidad de interacción remota, también se necesita una forma de iniciar los servidores de la demanda, distribución de proxies a los clientes, distribuir eventos asincrónicos, configurar su aplicación, distribuir los parches para una aplicación, y así sucesivamente.(16)

Los servicios se implementan como servidores de ICE en que su aplicación actúa como un cliente. Ninguno de los servicios usa características internas de ICE que se ocultan de los desarrolladores de aplicaciones que, en teoría, se podría desarrollar servicios equivalentes por sí mismos. Sin embargo, tener estos servicios disponibles como parte de la plataforma le permite centrarse en el desarrollo de aplicaciones en lugar de tener que construir una gran cantidad de infraestructura primero. Por otra parte, la creación de estos servicios no es un esfuerzo trivial, por eso vale la pena conocer lo que está disponible y usarlo en vez de reinventar tu propia rueda.

2.3.1.3. Beneficios de la arquitectura de ICE

La arquitectura de ICE proporciona una serie de beneficios para los desarrolladores de aplicaciones:

- Semántica Orientada a Objetos:

Ice respeta en su totalidad el paradigma orientado a objetos "a través del alambre." Todas las operaciones de invocación utilizan el enlace, así que la ejecución de una operación se elige en función de la carrera en tiempo real (no estático) tipo de un objeto.

- Apoyo para la mensajería síncrona y asíncrona:

ICE proporciona invocación de la operación síncrona y asíncrona y expedición, así como de publicación-suscripción de mensajería a través de IceStorm. Esto le permite elegir un modelo de comunicación de acuerdo a las necesidades de la aplicación en lugar de tener que calzar la aplicación para ajustarse a un modelo único.

- Soporte para interfaces múltiples

Con facetas, los objetos pueden proporcionar múltiples interfaces independientes, manteniendo una sola identidad a través de estas interfaces. Esto proporciona una gran flexibilidad, en particular como una aplicación evoluciona, pero debe seguir siendo compatible con los antiguos clientes ya desplegados.

- Independencia del Transporte

Ice ofrece actualmente TCP / IP y UDP como protocolos de transporte. Ni el cliente ni el código del servidor son conscientes del transporte subyacente. (El transporte deseado puede ser elegido por un parámetro de configuración.)

- Independencia de la Implementación

Los clientes no son conscientes de cómo los servidores implementan sus objetos. Esto significa que la aplicación de un servidor puede ser cambiado después de los clientes se han desplegado, por ejemplo, utilizar un mecanismo de persistencia de diferencias, o incluso un lenguaje de programación diferentes.

2.3.1.4. Paquetes de ICE

- Slice: El lenguaje de especificación para ICE. Establece el contacto entre servidores y clientes y además es utilizado para describir datos persistentes.(18)
- Compiladores Slice: Las especificaciones en Slice se pueden compilar en varios lenguajes de programación: C++, Java, Python, PHP, C# y Visual Basic. Los

servidores y clientes ICE trabajan juntos, a pesar del lenguaje de programación. (18)

- ICE: El núcleo de librerías ICE, entre otras características este paquete gestiona todas las tareas de comunicación utilizando un protocolo altamente eficiente, proporciona soporte de hilos para servidores multihilo y funcionalidades adicionales para soportar escalabilidad extrema.
- IceUtil: una colección de utilidades como el manejo de Unicode y programación con hilos (C++ solo). (18)
- IceBox: Un servidor de aplicación específico para aplicaciones Ice. IceBox puede ejecutar y administrar servicios ICE que son cargados dinámicamente como una DLL, librería compartida o clase Java. (18)
- Freeze: Con unas pocas líneas de código, una aplicación puede incorporar un vector altamente escalable que gestiona eficientemente objetos persistentes. (18)
- FreezeScript: es necesario para cambiar tipos de datos persistentes, especialmente en proyectos de software grandes. Para minimizar el impacto de estos cambios, FreezeScript proporciona herramientas de inspección y migración para bases de datos Freeze. La herramienta soporta scripts basados en XML ya que es potente y fácil de usar. (18)
- IceSSL: Un plug-in de transporte SSL (Security Socket Layer) dinámico para el núcleo Ice. Proporciona autenticación, encriptación e integridad de mensaje, utilizando un protocolo SSL estándar. (18)
- Glaciar: Algunos retos de los sistemas middleware son la seguridad y los firewalls. Glaciar es una solución firewall de Ice, que simplifica el despliegue de aplicaciones seguras. Autentica y filtra las solicitudes de los clientes y permite callbacks al cliente de una manera segura. Utilizado en combinación con IceSSL, proporciona una solución segura que no permite intrusos y es fácil de configurar. (18)
- IceStorm: un servicio de mensajería que soporta la federación. En comparación con otros servicios de mensajería o eventos, IceStorm soporta tipado de eventos, entendiendo que la difusión de un mensaje en una federación es tan fácil como invocar un método sobre un interface. (18)
- IcePath: es un servicio de parcheado para distribuciones software. Mantener el software actualizado es una tarea tediosa. IcePath automatiza las actualizaciones tanto de archivos individuales y de jerarquías de directorios. Solo los archivos

modificados son descargados en las máquinas cliente, utilizando eficientemente algoritmos de compresión.(18)

2.3.2.CORBA (Common Object Request Broker Architecture)

La Arquitectura común de intermediarios en peticiones a objetos, es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. CORBA fue definido y está controlado por el Object Management Group (OMG) que define las APIs, el protocolo de comunicaciones y los mecanismos necesarios para permitir la interoperabilidad entre diferentes aplicaciones escritas en diferentes lenguajes y ejecutadas en diferentes plataformas, lo que es fundamental en computación distribuida.(19)

En un sentido general, CORBA "envuelve" el código escrito en otro lenguaje, en un paquete que contiene información adicional sobre las capacidades del código que contiene y sobre cómo llamar a sus métodos. Los objetos que resultan, pueden entonces ser invocados desde otro programa (u objeto CORBA) desde la red. En este sentido CORBA se puede considerar como un formato de documentación legible por la máquina, similar a un archivo de cabeceras, pero con más información. CORBA utiliza un lenguaje de definición de interfaces (IDL) para especificar las interfaces con los servicios que los objetos ofrecerán. CORBA puede especificar a partir de este IDL, la interfaz a un lenguaje determinado, describiendo cómo los tipos de dato CORBA deben ser utilizados en las implementaciones del cliente y del servidor. Implementaciones estándar existen para Ada, C, C++, SmallTalk, Java, Python y Perl.(20)

Al compilar una interfaz en IDL se genera código para el cliente y el servidor (el implementador del objeto). El código del cliente sirve para poder realizar las llamadas a métodos remotos. Es el conocido como stub, el cual incluye un proxy (representante) del objeto remoto en el lado del cliente. El código generado para el servidor consiste en unos skeletons (esqueletos) que el desarrollador tiene que rellenar para implementar los métodos del objeto.

CORBA es más que una especificación multiplataforma, también define servicios habitualmente necesarios como seguridad y transacciones. Y así este no es un sistema operativo en sí, en realidad es un middleware.

Dos aspectos de la arquitectura de CORBA sobresalen:

- Las implementaciones tanto del cliente como del objeto están aisladas del Object Request Broker (ORB) por una interfaz IDL.
- Las solicitudes no pasan directamente del cliente al objeto, siempre son administradas por un ORB. Lo que hace que los detalles de la distribución sean transparentes para los clientes y para los objetos.

La arquitectura de administración de objetos propuesta por OMG define una arquitectura común que hace posible que exista un sistema unificado de computo basado en componentes heterogéneos que están interoperando entre sí. Los servicios proporcionados por CORBA proporcionan servicios necesarios por casi todo sistema basado en objetos, mientras que las facilidades de CORBA permiten un acceso estándar a tipos de datos comunes y la funcionalidad necesaria para grupos de aplicaciones corporativas y específicos para las aplicaciones industriales.

2.3.2.1. Componentes de CORBA

- *Object Request Broker (agente de petición de objetos)* - (u ORB) un ORB es una pieza de "*middleware*", como se le llama, a lo que se sitúa en medio de los clientes y servidores y hace posible la fácil comunicación entre ellos. El ORB es un ente conceptual, que algunas veces toma la forma de una librería compartida y en otras ocasiones toma la forma de un programa externo. El ORB es el responsable de establecer y destruir las sesiones entre clientes y servidores, dirigir, y transportar mensajes entre ellos durante una sesión.
- *Object Adaptor (adaptador de objetos)*- (u OA) un *Object Adaptor* proporciona el canal por el cual un *object server* se comunica con el *Object Request Broker* (ORBit).
- GIOP/IIOP (*General Inter-ORB Protocol - Internet Inter-Orb Protocol*) CORBA toma todos estos lenguajes isms y handles orientados a objetos llevándolos entre

componentes software orientados a objetos. Si las dos piezas de software están en diferentes lugares, entonces se usará el protocolo GIOP para mover la información entre ellos. IIOP es una especificación de GIOP para el conjunto de protocolos de Internet; teóricamente se podría hacer funcionar GIOP sobre otra cosa distinta al IP, y en este caso se le llamaría de otra manera distinta.

2.3.2.2. EL ORB (Object Request Broker)

Una parte fundamental de la arquitectura CORBA es el ORB, componente software cuyo fin es facilitar la comunicación entre objetos. El ORB se encarga de enviar las peticiones a los objetos y retornar las respuestas a los clientes que invocan las peticiones. Se le conoce como el núcleo de las comunicaciones entre las aplicaciones CORBA o como el mensajero de las solicitudes entre los objetos y los clientes en una red remota. El ORB está configurado para acceder a los servicios de CORBA.

La principal característica del ORB es la transparencia, cómo facilita la comunicación cliente/servidor. Generalmente, el ORB oculta lo siguiente:

- Ubicación de los objetos. El cliente no sabe dónde se encuentra el objeto destino. Puede residir en un proceso diferente en otra máquina a través de la red, o dentro del mismo proceso.
- Implementación de los objetos. El cliente no sabe cómo está implementado el objeto remoto, en qué lenguaje de programación o de scripts está escrito, o el sistema operativo y el hardware, sobre el que se ejecuta.
- Estado de ejecución del objeto. Cuando el cliente lanza una petición sobre un objeto remoto, no necesita saber si el objeto está en ese momento en ejecución, y listo para aceptar peticiones. El ORB de forma transparente inicializa el objeto en caso de ser necesario, antes de enviarle la petición.
- Mecanismos de comunicación de los objetos. El cliente no sabe qué mecanismos de comunicación (por ejemplo, TCP/IP (Protocolo de Control de Transmisiones / Protocolo de Internet), memoria compartida, llamada a método local) utiliza el ORB para enviar la petición al objeto y retorna la respuesta al cliente.

2.3.2.3. Qué es un objeto CORBA

Un objeto CORBA cumple con:

- Encapsulamiento: Un módulo de software encapsulado consiste de dos partes distintas; su interfaz que es lo que el módulo representa para el mundo exterior, y su implementación que es mantenida privada. La encapsulación también le permite a CORBA proporcionar la transparencia en la localización, ya que los clientes envían sus solicitudes a su ORB local, y no al objeto destino.
- Herencia: Es posible crear un nuevo objeto adaptando uno existente a crear uno partiendo de cero.
- Polimorfismo: Las operaciones pueden pertenecer a más de un clase de objeto.

2.3.2.4. El IDL (Interface Definition Language)

En CORBA la interfaz de un objeto es definida en OMG IDL (Interface Definition Language). La definición de la interfaz especifica los métodos que el objeto está preparado para realizar, sus parámetros de entrada, su resultado y cualquier excepción que pueda generarse durante la ejecución.

En el momento de construir un objeto CORBA el primer paso es definir cuál va a ser la funcionalidad que va a proporcionar, para de esta manera poder escribir la interfaz en IDL. Toda la información necesaria para construir un cliente del objeto es proporcionada por la interfaz. Se debe escoger un lenguaje de programación que facilite la implementación de la interfaz de cada objeto.

2.3.2.5. Facilidades de CORBA

- Facilidades horizontales que son las que consisten en interfaces de usuarios, administración de información, administración de sistemas y administración de tareas.
- Facilidades verticales son las especificaciones IDL utilizadas en sectores específicos del mercado, como la salud y el comercio electrónico.

2.3.2.6. Paquetes de CORBA

- AdaBrokeres un conjunto de herramientas y librerías para desarrollar aplicaciones en lenguaje de programación Ada. Proporciona un analizador de IDL, generador de código Ada. AdaBroker es liberado bajo la GPL (*General Public License*).
- ORBit es el ORB para el proyecto GNOME para lenguajes de programación C y Ada que ofrece los servicios de *Naming* y *Events*.
- OmniORB tiene un puerto explícito para Linux. Características de C++ IDL, utiliza IIOIP (*Internet Inter-Orb Protocol*) y la IIOIP ha sido probado con otros ORB. OmniORB está licenciado bajo GPL.
- MICO: Como resultado de su origen académico en la Universidad de Frankfurt, en 1996, la aplicación MICO tiene un diseño claro y modular, incluso para implementaciones internas que permite asegurar el fácil uso de su extensibilidad. Sólo se basa en C++, la API estándar de Unix y bibliotecas no-propietario. Compatible con los SO Linux, Solaris y Windows NT.(20)
- Orbix es un ORB desarrollado para crear aplicaciones en lenguajes C++, Java y Cobol. Compatible con los SO Linux, Solaris y Windows NT.
- VisiBrokeres un ORB desarrollado para crear aplicaciones en lenguajes C++, Java. Compatible con los SO Linux, Solaris y Windows NT.
- OpenORB es un ORB desarrollado para crear aplicaciones en Java. Compatible con los SO Linux, Solaris y Windows NT.
- ORBacus (OmniBroker) es una implementación de CORBA de las más flexibles y mejor implementadas. Además, aunque no es gratis, su código está disponible para ser usado en universidades sin cargo(21). Es un ORB utilizado para desarrollar aplicaciones en lenguajes de programación C++ y Java. Compatible con los SO Linux, Solaris y Windows NT. Ofrece los servicios de *Naming*, *Events*, *Time*, *Property* y *Security*.
- OAK (Java) incluye extensiones para tolerancia a fallos.
- COOL ORB (*Contract Net Protocol Agent Talk*), es un lenguaje para la coordinación, que modela la conversación entre los agentes mediante una máquina de estados finitos. Cada estado representa el punto de la conversación en el que se encuentren. El paso de un estado a otro viene determinado por la

llegada del mensaje correcto. De este modo se puede establecer la conducta de los agentes frente a la llegada de los mensajes.(22)

- ILU (*Inter-Language Unification*) Incluye compiladores IDL para C + +, ANSI C, Python, Java, *Common Lisp* y Modula-3. ILU también incluye un equipo autónomo de implementación de la ONC (*Open Network Computing*) RPC, lo que hace posible el uso de los servicios existentes de RPC como objetos ILU. ILU también incluye un equipo autónomo de aplicación de HTTP (*Hyper Text Transfer Protocol*), lo que permite crear aplicaciones CORBA basados en servidores Web.
- Arachnesurge de un esfuerzo para desarrollar los componentes reutilizables para el ámbito médico. Arachne se ejecuta en una variedad de plataformas, incluyendo 95/NT, Mac, SunOS, HP / UX, pero Linux es la plataforma primaria de desarrollo.
- TAO está ampliamente desplegada en sistemas de misión crítica, pero pueden fácilmente ser aplicado a los escenarios de uso general para los usuarios de desarrollo en C++. Publicado bajo licencia GNU es compatible con los SO Linux, Solaris y Windows NT.(23)
- Electra Este ORB sobre Isis define un mecanismo básico de transferencia de estado, en la que el BOA (*Basic Object Adapter*) con el que trabaja (el POA, *Portable Object Adapter*, es una estandarización bastante posterior a Electra)(24).
- COPE es un ORB escrito completamente en Perl. No proporciona un IR (*Interface Repository*), pero puede utilizar las herramientas de terceros para ese fin (como Orbacus (OmniBroker)).
- JacORB es un CORBA ORB escrito completamente en Java. Dispone de un compilador de Java IDL, soporta hilos, IIOp, IR. Se conoce que funciona bien en Linux y es más compatible que algunos productos reconocidos de CORBA. Libre, bajo licencia GPL.
- Jorba es una implementación de CORBA escrito completamente en Java. Incluye el ORB, y un compilador de IDL. Se suministra bajo licencia GPL.
- CORBAPlus de ExpertSoft ofrece una aplicación CORBAv2 escrito en Java. Incluye soporte para un compilador de Java IDL, IIOp. Los términos de licencia no están del todo claras, pero parece ser libremente disponible para cualquier uso. No está claro si se ha probado en Linux.
- JavaIDL es una instantánea de compilador de Java SunSoft IDL. Incluye un núcleo de ORB genéricos.

- Fnorb es un ORB experimental escrito en Python. No proporciona un IR (*Interface Repository*), pero puede utilizar las herramientas de terceros para ese fin (como Orbacus (OmniBroker)).
- ROBIN (RPC y *Object Broker Interface*) implementa un subconjunto de la especificación CORBAv2.1. Apoya un IDL, no es compatible con IIOP. Se centra en la adquisición de datos y sistemas de control, incluyendo los ordenadores integrados.
- DOME de Object Oriented Technologies ofrece callback, operación asincrónica, la intermediación de localización, así como monitores de red y herramientas. Puntos fuertes: se ejecuta en una gran variedad de plataformas, y es compatible con muchos protocolos de red diferentes. Ofrece una versión libre para Linux.

2.3.3.DCOM

DCOM (Distributed Component Object Model) en español Modelo de Objetos de Componentes Distribuidos, es el resultado de la evolución y convergencia de dos tecnologías: la comunicación inter-procesos en los ambientes Windows de Microsoft y los esfuerzos de la OSF (Open Software Foundation) para desarrollar un ambiente de computación distribuido (DCE, Distributed Computing Environment), en particular un protocolo para la invocación de procesos remotos (RPC, Remote Procedure Calls).(25)

Es una tecnología propietaria de Microsoft para desarrollar componentes de software distribuidos sobre varios ordenadores y que se comunican entre sí. Extiende el modelo COM de Microsoft y proporciona el sustrato de comunicación entre la infraestructura del servidor de aplicaciones COM+ de Microsoft. Ha sido abandonada en favor del framework .NET

DCOM fue uno de los mayores competidores de CORBA. Los defensores de ambas tecnologías sostenían que algún día serían el modelo de código y servicios sobre Internet. Sin embargo, las dificultades que suponía conseguir que estas tecnologías funcionasen a través de cortafuegos y sobre máquinas inseguras o desconocidas, significó que las peticiones HTTP normales, combinadas con los navegadores web les ganasen la partida. Microsoft, en su momento intentó anticiparse a esto añadiendo un transporte extra HTTP a DCE/RPC denominado "ncacn_http" (Connection-based, over HTTP) y fracasó.

Normalmente llamado “COM on the wire”, dispone de la misma forma que CORBA de un lenguaje de definición de interfaces llamado en este caso MIDL (Microsoft Interface Definition Language) y de un protocolo que le permite realizar invocaciones remotas llamado ORPC (Object Remote Procedure Call). Cada servidor DCOM puede soportar múltiples interfaces cada uno de ellos representando el comportamiento de un objeto. Como la especificación DCOM se refiere a nivel binario permite que los componentes DCOM puedan ser implementados en diversos lenguajes de programación como C++, Java, Object Pascal (Delphi), Visual Basic e incluso COBOL. Y siempre que la plataforma soporte los servicios COM estos pueden ser utilizados en esa plataforma.

Microsoft Distributed COM (DCOM) extiende COM (Component Object Model) para soportar comunicación entre objetos en ordenadores distintos, en una LAN, WAN, o incluso en INTERNET. Con DCOM una aplicación puede ser distribuida en lugares que dan más sentido al cliente y a la aplicación.

2.3.3.1. La Arquitectura DCOM

DCOM es una extensión de COM, y éste define como los componentes y sus clientes interactúan entre sí. Esta interacción es definida de tal manera que el cliente y el componente se pueden conectar sin la necesidad de un sistema intermedio. El cliente llama a los métodos del componente sin tener que preocuparse de niveles más complejos.

En los actuales sistemas operativos, los procesos están separados unos de otros. Un cliente que necesita comunicarse con un componente en otro proceso no puede llamarlo directamente, y tendrá que utilizar alguna forma de comunicación entre procesos que proporcione el sistema operativo. COM proporciona este tipo de comunicación de una forma transparente: intercepta las llamadas del cliente y las reenvía al componente que está en otro proceso.

Cuando el cliente y el componente residen en distintas máquinas, DCOM simplemente reemplaza la comunicación entre procesos locales por un protocolo de red. Ni el cliente ni el componente se enteran de que la unión que los conecta es ahora un poco más grande.

Las librerías de COM proporcionan servicios orientados a objetos a los clientes y componentes, y utilizan RPC y un proveedor de seguridad para generar paquetes de red estándar que entienda el protocolo estándar de DCOM.

2.4. Conclusiones

En este capítulo se hizo referencia a un conjunto de middlewares para la comunicación en entornos heterogéneos, se dio un acercamiento a cada una de las tecnologías relacionadas con el intercambio de información entre aplicaciones, entre plataformas web y entre una aplicación web y una plataforma web. Una serie de candidatos para realizar el intercambio de información entre PRIMICIA y otras plataformas web se tendrá para elegir cuales y son los más idóneos para cada tipo de comunicación.

Solución Propuesta

3. Introducción

Teniendo en cuenta toda la serie de tecnologías existentes se procede a definir las más adecuadas para el intercambio de información con el exterior de la Plataforma de Televisión Informativa Primicia. Posteriormente se procede a la validación y aceptación del diseño propuesto, lo cual se efectúa utilizando el método Delphi, el cual es una técnica de investigación social que tiene como objetivo la obtención de una opinión grupal a partir del criterio de varios expertos. Por último se procesarán las encuestas aplicadas para obtener los argumentos necesarios para acreditar o rechazar el diseño de interfaces de comunicación para PRIMICIA.

3.1 Solución propuesta para el intercambio de información entre Aplicaciones de Escritorio

Para el intercambio de información entre aplicaciones se propone el uso de RPC (Llamada a Procedimientos Remotos).

3.1.1 RPC

La idea con RPC es hacer ver a una llamada remota como si fuera local, por esto la invocación debe ser transparente para el que la utiliza.

Presenta ventajas como:

- Se ocultan detalles de la comunicación
- No hay diferencias entre una llamada local y una remota
- Tiene una alta intuitividad.

Funcionamiento:

- Un proceso en una maquina A llama a un proceso en la maquina B, el proceso en A se bloquea mientras que se ejecuta el proceso en B.
- La información que se intercambia son parámetros y se espera respuesta.

Problemas:

- Espacio de direccionamiento diferente
- Diferentes tipos de datos
- Fallas

La transparencia en RPC se logra agregando un stub o proxy tanto al cliente como al servidor.

Localización de Servidores:

Como un cliente localiza un servidor

- podría ser que el cliente tuviera la dirección del servidor. Esto es muy ineficiente y estático.
- realizar un proceso dinámico de asociación o binding con el servidor.

Se parte de una especificación formal del servidor:

- nombre del servidor
- número de versión
- lista de procedimientos
- para cada procedimiento se especifica los parámetros

La especificación formal es muy útil para los generadores de código.

Cuando un servidor comienza a ejecutar envía un mensaje a un programa (local o remoto) llamado el "binder" o servidor de registro.

El cliente conoce la dirección del binder a quien interroga por la existencia o no de los servidores que desea contactar.

Este binder se conoce como un Localizador de Recursos en la red, un broker o un "corredor" de procedimientos.

En toda máquina que ofrezca servicios en RPC, debe tener un proceso llamado "Portmapper" el cual es responsable de mapear procedimientos a puertos UDP.

El portmapper utiliza en sí mismo un puerto bien conocido (111) por el cual cualquier cliente puede contactarlo.

3.2 Solución propuesta para el intercambio de información entre Plataformas Web

Para el intercambio de información entre Plataformas Web se propone el uso de Servicios Web.

3.2.1 Servicios Web

Los servicios web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente en las cuales el software está distribuido en diferentes servidores.

La informática se inició con programas monousuarios implantados en grandes ordenadores. Posteriormente estas primeras aplicaciones alcanzaron la capacidad de atender a diferentes usuarios. Pasaron los años y llegó la arquitectura cliente-servidor, que gracias a este modelo de desarrollo, la aplicación se dividía en una parte que interaccionaba con el usuario y otra parte destinada al procesamiento de información. En este acercamiento se consiguió que cada una de las partes que constituían la aplicación pudiera residir en computadoras distintas. Con el paso del tiempo, la computación aumentó y llegó la era de las aplicaciones distribuidas en las cuales los procesos se realizaban en diferentes unidades. De este paso surgió la tecnología Internet para solventar las problemáticas asociadas a fallo de aplicación centralizado.

Como punto final a esta cronología, los web services son un paso adelante en la computación ya que de esta forma un ordenador ya no se considerara como un núcleo de cómputo sino como un repositorio de servicios de n aplicaciones distribuidas por internet.

3.2.1.1 Requisitos de un Web Services

- Interoperabilidad: Un servicio remoto debe permitir su utilización por clientes de otras plataformas.
- Amigabilidad con Internet: La solución debe poder funcionar para soportar clientes que accedan a los servicios remotos desde internet.
- Interfaces fuertemente tipadas: No debería haber ambigüedad acerca del tipo de dato enviado y recibido desde un servicio remoto. Más aún, los tipos de datos definidos en el servicio remoto deben poderse corresponder razonablemente bien con los tipos de datos de la mayoría de los lenguaje de programación procedimentales.
- Posibilidad de aprovechar los estándares de Internet existentes: La implementación del servicio remoto debería aprovechar estándares de Internet existentes tanto como sea posible y evitar reinventar soluciones de problemas que ya se han resuelto. Una solución construida sobre un estándar de Internet ampliamente adoptado puede aprovechar conjuntos de herramientas y productos existentes creados para dicha tecnología.
- Soporte para cualquier lenguaje: La solución no debería ligarse a un lenguaje de programación particular Java RMI, por ejemplo, está ligada completamente a lenguaje Java. Sería muy difícil invocar funcionalidad de un objeto Java remoto desde Visual Basic o PERL. Un cliente debería ser capaz de implementar un nuevo servicio Web existente independientemente del lenguaje de programación en el que se halla escrito el cliente
- Soporte para cualquier infraestructura de componente distribuida: La solución no debe estar fuertemente ligada a una infraestructura de componentes en particular. De hecho, no se debería requerir el comprar, instalar o mantener una infraestructura de objetos distribuidos, solo construir un nuevo servicio remoto utilizar un servicio existente. Los protocolos subyacentes deberían proporcionar un nivel base de comunicación entre infraestructura de objeto distribuidos existentes tales como DCOM y CORBA.

3.3 Solución propuesta para el intercambio de información entre una Aplicación de escritorio y una Plataforma Web

Para el intercambio de Información entre una aplicación de escritorio y una plataforma web se propone el del middleware: ICE (Motor de comunicaciones por Internet).

El middleware ICE es una alternativa más moderna a CORBA, COM/DCOM/COM+, la cual da soporte a una gran variedad de lenguajes de programación como C++, C#, Java, Python, Ruby, Php y Visual Basic. Es libre (GPL), multiplataforma da soporte a Linux, Windows Mac Os X y plataformas móviles. ICE es una “versión simplificada” de CORBA desarrollada por algunos de sus principales desarrolladores. No está diseñado por un comité, por lo que esta implementado por los mismos que lo especifican, Slice no es un mero IDL, también puede ser utilizado para describir el estado de los objetos y hacerlos persistentes.

El middleware se define como un software distribuido constituido por un conjunto de recursos y servicios que permiten a múltiples procesos, corriendo en una o más máquinas, interactuar a través de una red. Es una colección no estructurada de recursos y servicios, ubicados entre la capa de transporte y la capa de aplicación, que pueden ser utilizados individualmente o conjuntamente por los procesos de la aplicación. Provee una interfaz abstracta que ofrece al programador de aplicaciones una visión uniforme de elementos heterogéneos de bajo nivel, como sistemas operativos y redes subyacentes.(26)

El middleware debe presentar las siguientes características(27):

- Interoperabilidad: es la habilidad de dos o más sistemas o componentes de intercambiar información y utilizarla. El middleware debe permitir el intercambio de información entre hardware y sistemas operativos distintos e inclusive, entre implementaciones de distintos fabricantes (siempre que adhieran a un mismo estándar).
- Transparencia: se deben ocultar las diferencias de las capas subyacentes de forma tal que el middleware sea visto en forma consistente por las aplicaciones y

procesos que se construyen sobre él. Por ejemplo, debe ser posible ubicar servicios y recursos distribuidos sin recurrir a direcciones de red.

- **Confiabilidad y Disponibilidad:** un middleware es confiable si es capaz de cumplir con la funcionalidad que se le requiere bajo ciertas condiciones durante un período específico de tiempo. La disponibilidad es la capacidad de prestar un servicio correcto. Para lograr middlewares confiables y de alta disponibilidad, es conveniente la distribución de recursos y servicios evitando componentes centralizados. No puede suceder que por fallar una máquina o un proceso, todo el middleware quede inutilizado.
- **Escalabilidad:** es una medida de la facilidad con la que un sistema o componente puede ser modificado para acomodarse a la envergadura del problema a resolver. A medida que crece la cantidad de procesos de aplicación, el middleware no debe perder funcionalidad ni performance en forma excesiva.
- **Abstracción:** un middleware tiene que facilitar la construcción de aplicaciones distribuidas. Debe proveer un mayor nivel de abstracción que las capas inferiores de forma que el desarrollador de aplicaciones, en lugar de manejar entidades como conexiones, puertos o semáforos, trabaje con abstracciones de mayor nivel como pueden ser objetos, eventos y transacciones.(27)

3.4 Validación de la Propuesta

Para la validación y aceptación de la propuesta del diseño de interfaces de interacción externa para la plataforma de televisión informática PRIMICIA, que se presentó en este capítulo, se utilizó el método Delphi, que es una técnica de investigación social que tiene como objetivo la obtención de una opinión grupal a partir del criterio de varios expertos.

En esencia, el método consiste en la realización de un diálogo anónimo entre los expertos, que son consultados individualmente mediante cuestionarios en aras de obtener un consenso general o, en caso contrario, conocer al menos los motivos de la discrepancia. La confrontación de opiniones se lleva a cabo mediante una serie de

interrogantes sucesivas que permiten que la información obtenida pase por un proceso estadístico-matemático que permita la evaluación cuantitativa de los criterios.

Características del Método Delphi

El método Delphi es una técnica grupal de análisis de opinión, parte de una suposición fundamental y de que el criterio de un individuo particular es menos fiable que el de un grupo de personas en igualdad de condiciones, en general utiliza e investiga la opinión de expertos. “La técnica Delphi es como un método de investigación sociológica, que independientemente de que pertenece al tipo de entrevista de profundidad en grupo, se aparta de ellas agregando características particulares”.

El principio de este método es la inteligencia colectiva tratando de lograr un consenso de opiniones expresadas individualmente por un grupo de personas seleccionadas cuidadosamente como expertos calificados en torno al tema, por medio de la iteración sucesiva de un cuestionario retroalimentado de los resultados promedio de la ronda anterior, aplicando cálculos estadísticos. En este caso en particular se utiliza el método para validar la propuesta de diseño de interfaces de interacción para la Plataforma de Televisión Informativa PRIMICIA, en el cual se indaga en dos aspectos como: aceptación, y posibles resultados de su aplicación todo resultado de la presente investigación.

Las principales características del método están dadas por el anonimato de los participantes (excepto el o los investigadores), iteración (manejar tantas rondas como sean necesarias), retroalimentación controlada, respuesta de grupo en forma estadística y justificación de respuestas. Existen tres etapas o fases esenciales en la aplicación del método:

- Fase preliminar: Se delimita el contexto, los objetivos, el diseño, los elementos básicos del trabajo y la selección de los expertos.
- Fase exploratoria: Elaboración y aplicación de los cuestionarios.
- Fase final: Análisis estadísticos y presentación de la información.

3.4.1 Fase Preliminar

Durante esta primera fase primeramente se determina el objetivo a alcanzar con la aplicación del método seleccionado, después se realiza la selección del tipo de Delphi a utilizar y por último se produce la selección de los expertos que protagonizarán la evaluación de los resultados.

Objetivo a alcanzar

El principal objetivo a alcanzar con la aplicación del método Delphi es que basándonos en la opinión de un grupo de expertos, los cuales utilizan diariamente la plataforma PRIMICIA se logre la validación de la propuesta de diseño planteada de una manera rápida e íntegra, demostrando su efectividad y aplicación desde el punto de vista de los mencionados expertos en el tema.

Selección del tipo de Delphi a utilizar

El método Delphi se puede clasificar de diferentes maneras teniendo en cuenta varios aspectos; por objetivos, por conducción, entre otros tipos. Para realizar la selección del tipo idóneo para esta investigación se tuvieron en cuenta las siguientes características: disminución de tiempo y flexibilidad de las respuestas.

Después de un estudio realizado se llega a la conclusión que el método Delphi conocido como Delphi cara-cara, es el seleccionado para realizar la validación de la propuesta solución. En esta variante del método Delphi, el cuestionario se lleva personalmente a cada integrante del panel, a quien se le hace la entrevista en forma individual, lo cual permite aumentar la flexibilidad de las respuestas, pues el entrevistador puede resolver cualquier duda o ambigüedad que se le presente al panelista en relación a las preguntas del cuestionario. Por otra parte se logra considerables ventajas de tiempo y se logra disminuir el porcentaje de deserción de los expertos.

Selección de expertos

Existen varios métodos para determinar la competencia de los expertos, entre los que se encuentran:

- Autovaloración
- Coeficiente de competencia K
- Efectividad de la actividad profesional

El criterio seleccionado para la selección de los expertos fue de Efectividad en su actividad profesional, este es el criterio más utilizado por ser el que logra mayor objetividad en la evaluación del resultado y a la vez es el más cómodo para proceder a realizar la selección de los expertos.

Para aumentar la calidad de la evaluación se trata de seleccionar expertos de reconocida experiencia profesional avalada por su alta calificación científico-técnica, reconocido prestigio profesional, conocimiento profundo del tema objeto de investigación y resultados satisfactorios en el trabajo pedagógico.

La elección de los expertos siguiendo las características antes mencionadas favorece agenciar resultados con calidad y aceptación, sumados a otros valores propios de estas personas como son: la honestidad, la responsabilidad y la integridad entre otros valores.

Esta selección se realizó además atendiendo a las posibilidades reales de participación de los candidatos, siendo todos profesionales de la UCI que tienen como promedio 5 años de experiencia en el proceso productivo de la Universidad. Poseen amplios conocimientos en temas relacionados con la propuesta a evaluar, lo que ha permitido que las opiniones brindadas sean confiables y válidas para avalar el trabajo realizado.

Confirmación de la participación de los expertos

Una vez conformado el panel de expertos se le comunicó a cada uno ellos, por diálogo directo, la intención de que participaran en el proceso de validación y aceptación de la propuesta realizada. En ese primer encuentro se les explicó cuáles eran los objetivos del proceso y qué consistía el trabajo. El cien por ciento de los invitados aceptó la tarea, por lo que se pasó a elaborar el listado final de los integrantes del panel, que quedaría integrado por un total de 7 especialistas, entre ellos 3 integrantes de la dirección de PRIMICIA y 4 integrantes del Departamento de Señales Digitales.

3.4.2 Fase Exploratoria

Elaboración de la Encuesta

El cuestionario se hizo con planteamientos de mediana longitud, debido a que si es un cuestionario muy largo puede inducir al desinterés de los expertos y uno muy corto puede provocar que no se cumplan los objetivos del mismo. Durante la realización, se hizo énfasis en las preguntas de votación las cuáles consisten en poner más de una alternativa con el objetivo de que el experto pueda decidirse por una. Preguntas de tipo abiertas, lo que permitió al encuestado brindar criterios sobre el tema en cuestión sin estar limitado, en sus respuestas, por un margen. Este tipo de preguntas puede ofrecer información importante de la que no se haya tenido referencias durante la investigación. Sin embargo también fueron incluidas preguntas de tipo cerradas, pues están enfocadas a intereses específicos de la investigación y desde este punto de vista aportan una mayor riqueza a las respuestas.

3.4.3 Fase Final

La actividad fundamental de esta fase es el procesamiento y estudio de la información. La evaluación del cuestionario se hizo de forma manual pues la cantidad de información y datos a procesar no son excesivamente voluminosos, ni complicados en su procesamiento. La principal meta de este método es llegar a un consenso entre los expertos, pero como no existe una única forma de medir consenso, se utilizaron las siguientes reglas las cuales son las adoptadas para los estudios Delphi a nivel mundial:

- Se entenderá por consenso en las preguntas con dos alternativas cuando una de ellas acumula al menos el 70% de los votos ponderados por nivel de confianza y grado de los expertos.
- Para las preguntas con más de dos alternativas se entenderá por consenso, cuando una de las alternativas acumula al menos el 50% del total de las alternativas ponderadas por nivel de confianza y grado de los expertos.

Pregunta 1.

- La pregunta fue respondida por los 7 entrevistados todos relacionados al departamento de sistemas digitales y 3 de ellos pertenecientes al proyecto PRIMICIA.

Pregunta 2.

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 100% de los panelistas concuerdan que la necesidad de la aplicación de una propuesta de diseño de interfaces de comunicación con el exterior para la Plataforma de Televisión Informativa PRIMICIA es alta, se logra una concordancia entre los expertos.

Pregunta 3.

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 100% de los panelistas concuerda en que el diseño de interfaces de interacción para la Plataforma de Televisión Informativa PRIMICIA con el exterior ayuda a recibir y exportar información desde y hacia otros sistemas, se logra una concordancia entre los expertos.

Pregunta 4.

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 71.4% de los panelistas concuerda en que la posibilidad de aplicación del diseño de interfaces de interacción para la Plataforma de Televisión Informativa PRIMICIA es regular, se logra un consenso entre los expertos.

Pregunta 5.

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 100% de los panelistas concuerda en que la interacción de la Plataforma de Televisión Informativa PRIMICIA con el exterior ayude a esta a proveer de información actualizada a sus usuarios, se logra un consenso entre los expertos.

Pregunta 6. Inciso a

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 57.2% de los panelistas concordaron en que le atribuyen una importancia media (valor 3) al Intercambio de información entre aplicaciones de escritorio mediante Sockets, se logra un consenso entre los expertos en esa pregunta.

Pregunta 6. Inciso b

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 57.2% de los panelistas concordaron en que le atribuyen una importancia media (valor 4) al Intercambio de información entre aplicaciones de escritorio mediante RPC (Llamada a procedimientos remotos), se logra un consenso entre los expertos en esa pregunta.

Pregunta 6. Inciso c

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 100% de los panelistas concordaron en que le atribuyen la máxima importancia (valor 5) al intercambio de información entre Aplicaciones Web mediante Web Services (Servicios Web), se logra un consenso entre los expertos en esa pregunta.

Pregunta 6. Inciso d

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 71.4% de los panelistas concuerdan en que atribuyen una importancia alta (valor 4) al intercambio de información entre Aplicaciones Web mediante RSS (Sindicación realmente simple), se logra un consenso entre los expertos en esa pregunta.

Pregunta 6. Inciso e

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 100% de los panelistas concordaron en que le atribuyen la máxima importancia (valor 5) al Intercambio de información entre una aplicación web y una plataforma web mediante ICE (Motor de comunicaciones por Internet), se logra un consenso entre los expertos en esa pregunta.

Pregunta 6. Inciso f

- La pregunta fue respondida por el 100% de los entrevistados lo que significa el 100% del panel de expertos. Como el 71.4% de los panelistas concuerdan en que atribuyen una importancia media (valor 3) Intercambio de información entre una aplicación web y una plataforma web mediante DCOM (Modelo de Objetos de Componentes Distribuidos), se logra un consenso entre los expertos en esa pregunta.

Pregunta 7.

Esta pregunta no fue respondida por ningún miembro del panel de expertos.

3.5 Conclusiones Parciales

En este capítulo se plantea la solución que hace referencia a las principales formas de comunicación de la Plataforma de Televisión Informativa PRIMICIA con agentes externos, lo que permitirá que la información sea importada de otros sistemas mediante varios procedimientos si es así necesario partiendo de los requerimientos que presente la misma, las cuales permitirán tener información fresca y actualizada proveniente de otros entornos web. En este capítulo se ha validado la propuesta de mejoras para el proceso de redacción de PRIMICIA, para ello se ha empleado el método Delphi mediante el panel de expertos. Los panelistas tuvieron un alto grado de concordancia en todas las preguntas por lo que solo fue necesario aplicar una sola ronda de entrevistas de donde se concluye según el criterio de los expertos que:

- La necesidad de la aplicación de una propuesta de diseño de interfaces de comunicación con el exterior para la Plataforma de Televisión Informativa PRIMICIA es alta.
- El diseño de interfaces de interacción para la Plataforma de Televisión Informativa PRIMICIA con el exterior permitirá a recibir y exportar información desde y hacia otros sistemas
- La interacción de la Plataforma de Televisión Informativa PRIMICIA con el exterior permitirá a esta a proveer de información actualizada a sus usuarios

Teniendo en cuenta los resultados obtenidos con la aplicación del método Delphi para certificar la propuesta realizada, se puede concluir que la misma ha sido validada satisfactoriamente.

Conclusiones Generales

Durante todo el desarrollo de la presente investigación se han adquirido una serie de conocimientos que han posibilitado la obtención de una serie de resultados. Con la realización de este trabajo se dio cumplimiento al objetivo general del diseño de interfaces de comunicación externa para la Plataforma de Televisión Informativa PRIMICIA con servicios externos. Primeramente se dio un acercamiento teórico sobre una serie de conceptos ligados a la web y a la interacción que ayudarían al entendimiento de los objetivos a lograr. Posteriormente se dio un acercamiento a los modelos de intercambio de información donde se analizaron una serie de tecnologías que permiten la comunicación entre ambientes heterogéneos.

Se puede decir entonces que se ha dado cumplimiento a todos los objetivos específicos planteados al inicio de la investigación logrando desarrollar un diseño de interfaces de comunicación para la Plataforma de Televisión Informativa Primicia acorde con las necesidades existentes. Lo cual será una gran mejora tanto para el producto como tal, al ganar en funcionalidad y competitividad en el mercado internacional, como para la Universidad y el país

Recomendaciones

Después de haber concluido la investigación y haber cumplido totalmente los objetivos de ésta, se recomienda agregar las funcionalidades propuestas a la Plataforma de Televisión Informativa Primicia que permitan aplicar el diseño de interfaces de comunicación previamente definido.

Bibliografía

1. **McGraw-Hill**. *Microsoft Diccionario de Informática e Internet*. 2004.
2. **Española, Real Academia de la Lengua**. RAE. [En línea] www.rae.es.
3. **Salaberria, Ramon**. *Redacción Periodística en Internet*. s.l. : Pablo de la Torriente Brau, 2005.
4. <http://www.w3c.es/>. [En línea] <http://www.w3c.es/>.
5. [En línea] 01 de 2010. http://www.norte.uni.edu.ni/estudiantes/cs_1.pdf.
6. **Fretzen, J y Madrid, McGraw-Hill**. *Superutilidades para JavaScript (Español)*. 1999. ISBN 84-481-2124-4.
7. **Tecnologico**. [En línea] 01 de 2010. <http://www.mitecnologico.com/>.
8. Chudiang. [En línea] 02 de 2010. http://www.chuidiang.com/clinux/sockets/sockets_simp.php.
9. [En línea] 03 de 2010. <http://ditec.um.es/laso/docs/tut-tcpip/3376c410.html>.
10. [En línea] 5 de 2010. http://www.dis.eafit.edu.co/cursos/st725/material/lect_rpc.htm.
11. [En línea] 2 de 2010. <http://www.desarrolloweb.com/articulos/1545.php>.
12. [En línea] 06 de 2010. <http://www.w3c.es/divulgacion/a-z/#x>.
13. **Mesa, Silvia**. [En línea] <http://silmc1515.wordpress.com/2008/10/15/uddi-description-discovery-and-integration/>.
14. [En línea] <http://www.rss.nom.es/>.
15. [En línea] [Citado el: 3 de 3 de 2010.] <http://www.w3c.es/Divulgacion/Guiasbreves/WebSemantica>.
16. **Henning, Michi y Spruiell, Mark**. *Distributed Programming with ICE*. 2003.
17. [En línea] [Citado el: 2 de 04 de 2010.] <http://www.zeroc.com/ice.html>.

18. **Murcia, Universidad de.** [En línea] [Citado el: 15 de 1 de 2010.] <http://ditec.um.es/ssdd/trabajos/0506/Memoria-icegrid.doc>.
19. [En línea] [Citado el: 11 de 11 de 2009.] <http://agamenon.uniandes.edu.co/~revista/articulos/corba/corba.htm/>.
20. [En línea] [Citado el: 12 de 11 de 2009.] <http://www.mico.org/>.
21. [En línea] [Citado el: 20 de 1 de 2010.] <http://ditec.um.es/ssdd/orbacus.pdf>.
22. www.ucentral.edu.co. [En línea] [Citado el: 25 de 1 de 2010.] <http://www.ucentral.edu.co/NOMADAS/nunme-ante/11-15/Pdfs%20Nomadas%2015/19c-Agentes.pdf>.
23. www.prismtech.com. [En línea] [Citado el: 25 de 11 de 2009.] <http://www.prismtech.com/section-item.asp?id=634>.
24. **Cabañas, Luis Miguel Peña.** [En línea] [Citado el: 4 de 4 de 2010.] <http://grasia.fdi.ucm.es/sensei/docs/sensei.pdf>.
25. **Hurd, Jim.** Clarin.com. [En línea] [Citado el: 1 de 12 de 2009.] <http://www.clarin.com/suplementos/informatica/1999/01/13/t-01001d.htm>.
26. **Blatecky, Alan R. y Sun, Xian-He.** Middleware: the key to next generation computing (Preface). *Journal of Parallel and Distributed Computing*. 2004, 64.
27. **Bray, Mike.** Software Technology Roadmap, Middleware. [En línea] 1997. http://www.sei.cmu.edu/str/descriptions/middleware_body.html.
28. [En línea] www.docentes.utonet.edu.bo/.

Bibliografía Consultada

1. **GARCÍA, R. H.** PRIMICIA, Plataforma de Televisión Informativa. 2008, n°
2. PRIMICIA, Plataforma de Televisión Informativa. XVI Fórum de Ciencia y Técnica, 2008, n°
3. **Tamayo, Jorge Daniel Olivares y Rey, Bernardo Almaguer.** Desarrollo del Canal Informativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela: Subsistema de Administración. Ciudad Habana: s.n., 2008.
4. IEEE Standard Glossary of Software Engineering Terminology, 1990.
5. Utilización del método Delphi en la pronosticación: una experiencia inicial. Instituto de Investigaciones Económicas de la JUCEPLAN, La Habana. **OÑATE, N. RAMOS**, 1978.
6. **Rolando Alfredo Hernández León, Zayda Coello González.** EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. Ciudad de la Habana: EDUNTIV, 2002. ISBN 978-959-16-0343-2.
7. **Henning, Michi; Spruiell, Mark.** Distributed Programming with Ice. 2003.
8. **W3C.** RSS 2.0 specification. <http://validator.w3.org/feed/docs/rss2.html>
9. **Lima Córdova, Orestes y Suárez Contreras, Eridniel.** *Diseño de una Plataforma de Transmisión Abierta para la Radio y la Televisión.* 2008. [Consultado el: Enero, 2009].
10. **Hernández Bustio, José Andrés.** *Desarrollo del Canal Informativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela: Subsistema de Transmisión.* 2008. [Consultado el: Enero, 2009].
11. **Departamento de Técnicas de Programación.** *Introducción a la Programación Cliente - Servidor.* Entorno Virtual de Aprendizaje. [En línea] 2009. [Consultado el: Enero, 2010] [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=10530>].

Glosario

APIs:(Application Programming Interface): Interfaz de Programación de Aplicaciones.

Aplicación: Programa informático creado para facilitar al usuario un determinado tipo de trabajo. Esto lo caracteriza frente a otros programas como los sistemas operativos, las utilidades y los lenguajes de programación.

Callback: En programación, un es un código ejecutable que recibe como parámetro la dirección o puntero de otra función, cuando el callback es llamado este recurre al puntero de la función y la ejecuta.

Cliente/Servidor: Método de distribución de información o de archivos en el cual la agrupación central, servidor, almacena los archivos y los hace disponibles para solicitudes de aplicaciones cliente.

Demultiplexor: En el campo de las telecomunicaciones el demultiplexor es un dispositivo que puede recibir a través de un medio de transmisión compartido una señal compleja multiplexada y separar las distintas señales integrantes de la misma encaminándolas a las salidas correspondientes. La señal compleja puede ser tanto analógica como digital y estar multiplexada en cualquiera de las distintas formas posibles para cada una de ellas.

DLL: Una biblioteca de enlace dinámico o más comúnmente **DLL** (sigla en inglés de *dynamic-link library*) es el término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.

Handless: es un puntero que es usado cuando una aplicación hace referencia a bloques de memoria u objetos gestionados por otros sistemas, como una base de datos o un sistema operativo.

Hipertexto:(Hypertext): Es un sistema para escribir y mostrar texto que enlaza a información adicional sobre ese texto.

HTML: Lenguaje de marcado de hipertexto, es el lenguaje autoritario para crear documentos en la World Wide Web. Define la estructura de un documento web usando etiquetas y atributos.

Infocintas: Cintillo informativo que se muestra en las pantallas de tipo texto o texto-imagen, promocionando eventos de importancia para la institución, noticias relevantes que se transmitirán o información en general para los televidentes.

ISMS: Sistema de Información de Gestión de la Seguridad.

Little Endian y Big Endian: La nomenclatura de los criterios *little-endian* y *big-endian* proviene de la novela *Los viajes de Gulliver* de Jonathan Swift, y se puede entender como "dé comienzo por el extremo pequeño" y "de comienzo por el extremo mayor", aunque es propenso a confundirse con "acaba en pequeño" y "acaba en grande" respectivamente. Su etimología proviene de un juego de palabras en inglés con los términos compuestos *little-end-in* y *big-end-in*. El término inglés *Endianness* designa el formato en el que se almacenan los datos de más de un byte en un ordenador.

Marshalling/UnMarshalling: Debido a problemas de alfabetos (ASCII/ABCDIC), representación de enteros (complemento a uno o a dos), punto flotante, ordenamiento de bytes (Little Endian y Big Endian), se hace necesario establecer una forma canónica de representar los distintos tipos de datos. Se ejecuta un proceso conocido como Marshalling/UnMarshalling, el cual consiste en transformar los tipos de datos locales de una máquina en un formato estándar para ser transmitidos por la red. Los tipos básicos como escalares entre otros se pasan por valor, cuando son arreglos tanto en los clientes como servidores se realiza una copia local, se manipulan y se envían por la red.

MHP: (Multimedia Home Platform) es un sistema intermedio (middleware en inglés), diseñado por el proyecto DVB y estandarizado por la ETSI. MHP define una plataforma común para las aplicaciones interactivas de la televisión digital, independiente tanto del proveedor de servicios interactivos como del receptor de televisión utilizado. De este modo, MHP favorece la creación de un mercado horizontal donde aplicaciones, red de transmisión y terminales MHP pueden ser suministrados por proveedores o fabricantes independientes.

Multiplexor: En el campo de las telecomunicaciones el **multiplexor** se utiliza como dispositivo que puede recibir varias entradas y transmitir las por un medio de transmisión compartido. Para ello lo que hace es dividir el medio de transmisión en múltiples canales,

para que varios nodos puedan comunicarse al mismo tiempo. Una señal que está multiplexada debe demultiplexarse en el otro extremo.

PRIMICIA: Plataforma de Televisión Informativa.

Skeleton: Esqueleto.

SSL (Secure Sockets Layer): Protocolo de Capa de Conexión Segura- (SSL) y Transport Layer Security -Seguridad de la Capa de Transporte- (**TLS**), su sucesor, son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

Stubs: Son librerías que emplean los programadores las cuales se generan de forma automática, se emplea un lenguaje de descripción de interfaces para su generación (IDL). Estas descripciones se procesan por un precompilador que genera el código fuente de los stubs cliente y servidor. Resuelven problemas de heterogeneidad gracias a XDR o eXternal Data Representation. Son un formato independiente de datos.

WWW: (World Wide Web): Sistema de información distribuido con mecanismos de hipertexto. Es el universo de servidores *HTTP*, que permiten mezclar texto, gráficos y archivos de sonido juntos.

Web: Servidor de información *www*. Se utiliza también para definir el universo *www* en su conjunto.

W3C:(*Consortio World Wide Web*): Consorcio internacional que produce estándares para la *World Wide Web*. Está dirigida por Tim Berners Lee, creador original de *URL* (Uniform Resource Locator, localizador uniforme de recursos), *HTTP* (Hyper Text Transfer Protocol, protocolo de transferencia de hipertexto) y *HTML* (lenguaje de marcado de hipertexto) que son las principales tecnologías sobre las que se basa la Web.

XML:(Extensible Markup Language): Conocido como lenguajes de marcas. Es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (*W3C*), no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

Índice de Anexos

Anexo 1: Encuesta a los miembros del panel de expertos.

Anexo 1: Encuesta a los miembros del panel de expertos.

Encuesta a los miembros del Panel de Expertos

Compañero(a):

Con motivo de la realización del trabajo de diploma sobre el diseño de interfaces de interacción para la Plataforma de Televisión Informativa Primicia, el autor del mismo se encuentra realizando la presente encuesta, en aras de conocer algunos criterios relevantes sobre el tema y de obtener información importante proveniente de expertos que por su calificación y experiencia en su profesión puedan ofrecer una evaluación de los resultados teóricos de esta investigación.

Muchas Gracias

- 1- ¿Cuál es su función dentro del proyecto Plataforma de Televisión Informativa PRIMICIA o dentro del Departamento de Señales Digitales?

- 2- ¿Cómo usted valora la necesidad de la aplicación de una propuesta de diseño de interfaces de comunicación con el exterior para la Plataforma de Televisión Informativa PRIMICIA?

Alta___ Media___ Baja___.

- 3- ¿Usted cree que el diseño de interfaces de interacción para la Plataforma de Televisión Informativa PRIMICIA con el exterior ayude a recibir y exportar información desde y hacia otros sistemas?

Si___ No___.

- 4- ¿Cómo usted valora la posibilidad de aplicación del diseño de interfaces de interacción para la Plataforma de Televisión Informativa PRIMICIA?

Bien___ Regular___ Mal___.

5- ¿Usted cree que la interacción de la Plataforma de Televisión Informativa PRIMICIA con el exterior ayude a esta a proveer de información actualizada a sus usuarios?

Si_____ No_____.

6- Teniendo en cuenta la importancia que para PRIMICIA significaría la comunicación con el exterior evalúe el uso de las siguientes tecnologías con un valor de 0 a 5.

a) Intercambio de información entre aplicaciones de escritorio mediante Sockets.

__Cero __Uno __Dos __Tres __Cuatro __Cinco.

b) Intercambio de información entre aplicaciones de escritorio mediante RPC (Llamada a procedimientos remotos).

__Cero __Uno __Dos __Tres __Cuatro __Cinco.

c) Intercambio de información entre Aplicaciones Web mediante Web Services (Servicios Web).

__Cero __Uno __Dos __Tres __Cuatro __Cinco.

d) Intercambio de información entre Aplicaciones Web mediante RSS (Sindicación realmente simple).

__Cero __Uno __Dos __Tres __Cuatro __Cinco.

d) Intercambio de información entre una aplicación de escritorio y una plataforma web mediante CORBA (Arquitectura común de intermediarios en peticiones a objetos).

__Cero __Uno __Dos __Tres __Cuatro __Cinco.

e) Intercambio de información entre una aplicación de escritorio y una plataforma web mediante ICE (Motor de comunicaciones por Internet).

__Cero __Uno __Dos __Tres __Cuatro __Cinco.

f) Intercambio de información entre una aplicación de escritorio y una plataforma web mediante DCOM (Modelo de Objetos de Componentes Distribuidos).

__Cero __Uno __Dos __Tres __Cuatro __Cinco.

- 7- Cualquier sugerencia que desee hacer acerca del diseño de interfaces de interacción para la Plataforma de Televisión Informativa PRIMICIA.

Ciudad de la Habana, Junio 2010

“Año 52 de la Revolución”