

INTRODUCCIÓN

Los adelantos tecnológicos presentados en las últimas décadas han causado una evolución, cambio y crecimiento continuo en el entorno empresarial, ocasionando que sea cada vez más competitivo y agresivo. Por esta razón las empresas de la actualidad deben mejorar sus procesos al mismo ritmo que evoluciona la tecnología, ya que corren el riesgo de quedar en obsolescencia, lo cual permitiría que la competencia tome ventaja en factores críticos como efectividad, eficiencia y tiempo de respuesta. Además, el hecho de que las empresas empleen sistemas computacionales para que asistan sus procesos internos, brinda la posibilidad de atender mucho más rápido las nuevas solicitudes de sus clientes, lo cual aumenta las posibilidades de extender su mercado actual y atraer nuevos clientes potenciales.

“Las compañías exitosas entienden que un alto nivel de rendimiento en su cadena de abastecimiento es un factor clave para su continuo crecimiento y es por eso que constantemente invierten mucho trabajo y esfuerzo en mejorar la calidad de sus servicios y productos, marketing, satisfacción del cliente, rendimiento, eficiencia y rentabilidad, ya que los ganadores del mañana serán quienes logren reunir y reaccionar rápidamente ante información crítica”¹. Tomar decisiones de esta forma permite a las organizaciones cumplir con sus expectativas de crecimiento y al mismo tiempo les permite utilizar la información como una ventaja competitiva.

Para lograr cumplir esta visión, las compañías requieren una infraestructura que proporcione información a través de todas las funciones y localidades de la misma, para esto se hace necesario el uso de un software de Planeación de Recursos Empresariales (ERP²). Actualmente las compañías poseen uno o más de estos sistemas, los cuales se encargan de controlar la información en cada uno de sus departamentos y áreas, hoy en día estos sistemas son vitales en las organizaciones ya que permiten que la información sea correctamente almacenada y procesada, además son indispensables para llevar al máximo la productividad y eficiencia de los procesos internos.

En algunos casos se hace necesario la interoperabilidad de los sistemas ERP permitiendo que se comuniquen entre sí, compartiendo información y recursos que no están disponibles dentro de sus arquitecturas individuales; así mismo, este tema se ha convertido en un componente clave para todas las empresas, debido a que cada departamento tiene diferentes maneras de organizar y estructurar la

¹ GARG, Vinodkumar. N. K. Venkitakrishnan. Enterprise Resource Planning: Concepts and Practice. 2 Ed. New Delhi. India: Prentice-Hall of India Pvt. Ltd., 2004. 200 p. ISBN 81-203-2254-1.

² El termino ERP se define como “Enterprise Resource Management o Sistema de planificación de recursos empresariales” y se define como “un sistema de planificación de los recursos y de gestión de la información que, de una forma estructurada, satisface la demanda de necesidades de la gestión empresarial.” (MUÑIS, 2004).

información que genera y por lo general, esta es la razón para que haya más de un sistema ERP implementado.

Otra situación que se presenta en muchas ocasiones es que estos sistemas están situados en lugares muy distantes geográficamente y los usuarios se conectan a través de redes virtuales privadas (VPN).

El modelo actual de trabajo funciona correctamente cuando la información que debe pasar por los equipos de los usuarios es poca, pero en el momento en que esta información alcanza altos volúmenes de datos, se generan complicaciones en las labores del usuario, debido que la información a procesar es cargada en su equipo y éste no posee los recursos necesarios para el correcto y ágil procesamiento de dicha información, haciendo que se ralentice su sistema o incluso pueda provocar un bloqueo total de su equipo, lo que llevaría al usuario a un lapso de improductividad mientras el equipo se restablece, además probablemente cause congestión en la red de datos, disminuyendo así la efectividad y eficiencia de la empresa.

En este trabajo se aborda el diseño e implementación de un conjunto de soluciones las cuales logren exponer un nuevo modelo de comunicación, el cual permita reducir el consumo de recursos hardware del equipo de los usuarios, disminuyendo la cantidad de información que estos deben procesar, realizando peticiones pequeñas al servidor de aplicaciones las cuales le indiquen el tipo de procesamiento al que debe someter la información y la clase de respuesta que debe entregar, de esta manera se planea disminuir las cargas en la red y acelerar la respuesta a los usuarios finales.

1. EL PROBLEMA

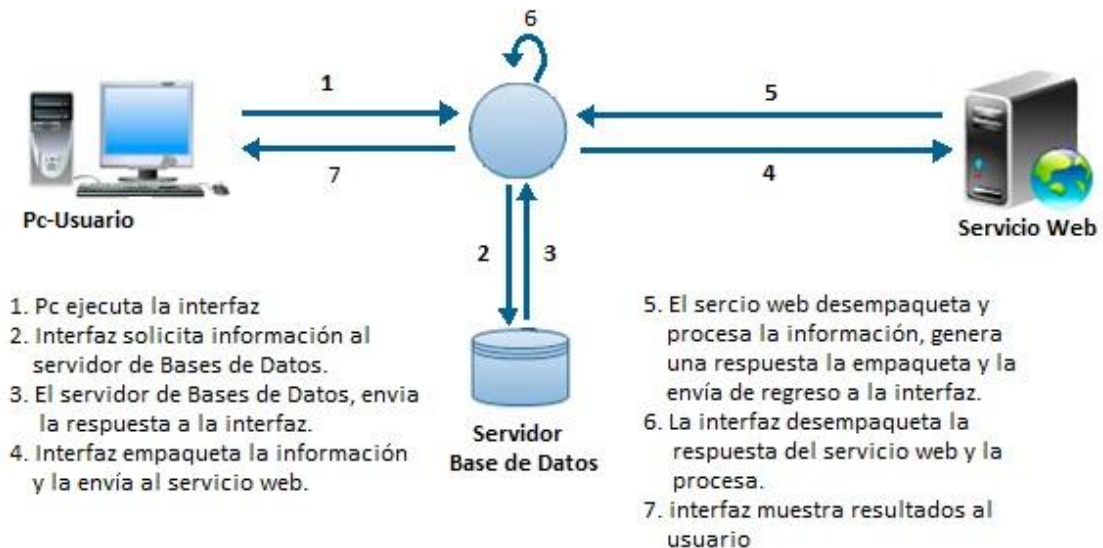
1.1. DESCRIPCIÓN DEL PROBLEMA

Debido a la necesidad de comunicación e integración entre ERP's diferentes, y para garantizar la interoperabilidad de los componentes de hardware y software, nace lo que hoy en día conocemos como "Interfaz de Comunicación", ésta se encarga del proceso de envío y recepción de datos entre ambos sistemas, basándose en modelos de comunicación distribuidos y no distribuidos, dicha interfaz puede ser desarrollada basándose en diferentes tipos de comunicación entre sistemas, entre ellos está el uso de archivos planos, procedimientos remotos (CORBA, RMI, RPC, etc.) y las más nuevas se están apoyando en los servicios web.

Actualmente las interfaces están siendo enfocadas hacia las tecnologías distribuidas, ya que éstas facilitan y solucionan en gran medida los problemas típicos que se tienen al tratar de comunicar dos sistemas diferentes; normalmente las interfaces de comunicación están diseñadas para ser ejecutadas periódicamente, es decir, mensual, quincenal, semanal, en algunos casos a diario y en muy pocos casos ante cada interacción del usuario con el software, esto sucede porque el software está diseñado para trabajar de manera independiente con respecto de los otros ERP.

En los ambientes corporativos se acumulan grandes cantidades de datos durante el periodo de tiempo en que las interfaces están inactivas, y en el momento que se ejecutan, éstas deben tomar la información requerida, procesarla (filtrarla, totalizar, etc.), empaquetarla y enviarla al Servicio Web. Además, es muy común que el Servicio Web responda con una cantidad de información similar, debido a que éste toma los registros y envía de regreso la información detallada de la operación para cada registro, luego el sistema receptor debe desempaquetar la respuesta, evaluarla y almacenarla. La siguiente imagen ilustra el proceso que se lleva a cabo en la comunicación entre dos ERP:

Imagen 1. Modelo actual de comunicación entre ERPs.



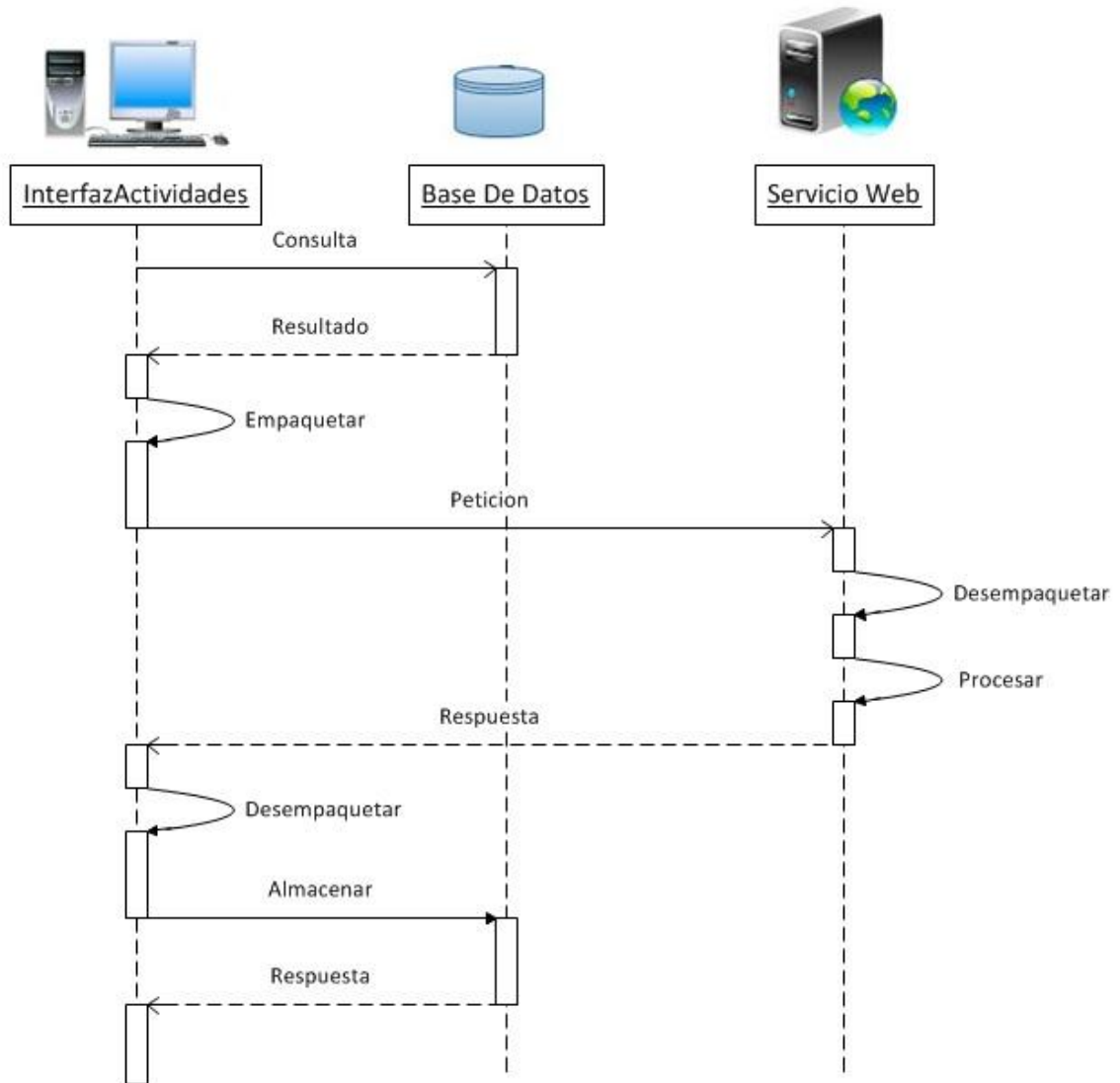
Se debe tener en cuenta que la cantidad de memoria requerida por la interfaz es equivalente a la cantidad de datos que envía y recibe, además que es el equipo del usuario quien provee dicha cantidad de memoria requerida.

Fuente: los autores.

En la imagen anterior se ilustra el modelo comunicación actual entre dos sistemas ERP a través de servicios web, donde se puede observar que la interfaz de comunicación alojada en el equipo cliente, es el actor principal en el proceso y por esta razón consume una cantidad de recursos de hardware proporcional a la cantidad de datos a procesar, como anteriormente se mencionó este modelo de comunicación funciona correctamente cuando la información que debe pasar por los equipos de los usuarios es poca, pero en el momento en que esta información alcanza un alto volumen de datos, el equipo del usuario presenta una disminución en su velocidad de respuesta, debido a que no posee los recursos necesarios para el correcto y ágil procesamiento de dicha cantidad de información, haciendo que se ralentice su sistema, o incluso provoque el bloqueo total del equipo, lo que llevaría al usuario a un lapso de inproductividad mientras el equipo se restablece y probablemente esto genere una mayor congestión en la red de datos, disminuyendo así la efectividad y eficiencia de la empresa.

Para mejorar la comprensión del modelo actual de comunicación se presenta el siguiente diagrama de secuencias.

Diagrama 1. Diagrama de secuencia del modelo actual de comunicación entre ERP's.



En la gráfica anterior se puede ver claramente que es el equipo del usuario quien realiza la mayor parte del trabajo, preparando la información y procesando la respuesta proveniente del servicio web. Dicha operación requiere cierta cantidad de recursos para ser llevada a cabo, los cuales aumentan o disminuyen de acuerdo a la cantidad de información que tenga que ser procesada.

El enfoque principal de este análisis se centra en sistemas que utilizan el motor de bases de datos Oracle como contenedor de la información, ya que cuenta con

las características necesarias de interoperabilidad, robustez, potencia y confiabilidad necesarias para crear una solución acorde y a la altura del problema expuesto en el capítulo anterior.

Se espera que la solución propuesta en este documento, ayude a las empresas a mejorar la eficiencia y rendimiento de los sistemas que requieren el uso de interfaces de comunicación con servicios web.

1.2. FORMULACIÓN DEL PROBLEMA

¿Es posible solucionar el problema del excesivo consumo de recursos en los equipos cliente, si se traslada al servidor la carga computacional que se requiere para ejecutar las interfaces de comunicación?

2. JUSTIFICACIÓN

Las empresas que utilizan el modelo actual de comunicación entre ERP's, presentan serios inconvenientes cuando la información a consultar a través de la interface de comunicación es de gran tamaño; debido a que en estos casos el equipo de un usuario común no posee las características necesarias para procesar rápidamente dicha cantidad de información, lo que provoca que se ralentice su sistema e incluso se pueda bloquear el equipo, lo cual disminuye la productividad laboral y por ende la eficiencia y el rendimiento de la actividades de la empresa; debido a que se hace necesario que el usuario espere a que el equipo responda a dicha solicitud, para así recobrar recursos de hardware y descongestionar el ancho de banda ocupado en la red de datos. Por lo anterior se hace necesario abordar esta problemática creando un nuevo modelo de comunicación entre ERP, el cual libere los equipos cliente del excesivo consumo de recursos hardware, trasladando dicha carga computacional a un equipo servidor, el cual logre mejorar el rendimiento del sistema y permita mayor agilidad en las consultas a distintas ERP, además de disminuir la el tráfico de datos en la red de datos de la empresa.

2.1 ALCANCE

Crear una solución basada en el lenguaje PL/SQL que actúe como un intermediario entre los equipos de los usuarios y los servicios web de integración, la cual permita que el procesamiento de la información sea realizado por la base de datos y no por las limitadas máquinas de los usuarios.

Construir un manual del programador, donde se especifique el modo como debe ser usada la librería y el modo como deben comunicarse las ERP para su correcto funcionamiento.

Realizar un laboratorio o caso de estudio donde se demuestren las ventajas de liberar a los equipos de los usuarios del procesamiento innecesario de grandes volúmenes de información.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Diseñar, proponer e implementar una solución integral de comunicación para los sistemas de planificación de recursos empresariales (ERP's) que utilicen servicios web como motor para sus interfaces de comunicación, la cual traslade la carga computacional desde los equipos de los funcionarios al servidor en el cual reside el motor de bases de datos.

3.2 OBJETIVOS ESPECÍFICOS

- Estudiar el estado del arte de los Servicios Web.
- Presentar los ERP's más utilizados actualmente.
- Comparar las diferentes interfaces de comunicación de los ERP's estudiados, pensando en su integración con los servicios web.
- Examinar la utilidad del protocolo HTTP y el uso del lenguaje XML en el motor de bases de datos ORACLE.
- Crear una interfaz de comunicación en PL/SQL para comunicar los ERP's a través de Servicios Web.

4. MARCO DE REFERENCIA

4.1 MARCO TEÓRICO

4.1.1 La computación distribuida: Fue a principio de los años 90 cuando la computación distribuida hizo su entrada en el mundo de la informática y en el mercado comercial, como la idea de la llamada “Utility computing que supone una revolución que dará mucho que hablar”³, pues es una tecnología que permite el uso sincronizado de recursos tanto hardware de diferentes arquitecturas, las cuales podrían ser un equipo de escritorio, supercomputadores, clúster, entre otros, sin estar supeditados a un control central, como software, bases de datos o información en general. El propósito de la computación distribuida es facilitar el uso y la integración de los distintos recursos computacionales para intercambio de datos o de tiempo de computación. “Universidades, laboratorios de investigación o empresas se asocian para formar un sistema de computación distribuido, para lo cual utilizan algún tipo de software que implemente este concepto”⁴.

4.1.1.1 Historia: Gracias a la evolución computacional y al abaratamiento de la misma, se dio lugar al uso de la computación clúster que en su momento eran insostenibles para las empresas por su relación costo rendimiento, éstos surgieron como opción muy trascendental tanto en el campo de la computación paralela como la distribuida. Con esta tecnología se lograba unir un conjunto de ordenadores o dispositivos mediante una red para llevar a cabo tareas complejas o que requieren de alta computación, “sin embargo la gran demanda tanto de computación como de espacio y gestión de almacenamiento requeridos por un gran número de aplicaciones que gestionan grandes cantidades de datos y han de hacerlo de forma eficiente y sencilla, exige el uso de nuevas tecnologías, como es el caso de la computación distribuida o distribuida computing.”⁵

Actualmente la computación distribuida ha tomado ventaja frente a otras tecnologías similares, debido a la potente colaboración que ofrece una cantidad casi ilimitada de dispositivos heterogéneos conectados mediante una red, la perfecta integración de hardware, además de ser una solución altamente escalable y flexible, lo que evita problemas de falta de recursos como se presentan en los cuellos de botella, debido a la posibilidad de modificar el número y características de sus componentes.

³ RENATA.EDU.CO. Curso de computación en grid [en línea]. [Fecha de consulta: 10 diciembre 2013]. Disponible en <http://www.renata.edu.co/index.php/experiencias-significativas/894-curso-de-computacion-en-grid-comenzara-el-primer-marzo.html>

⁴ FERREIRA Escutia, Rogelio. GRID COMPUTING. XUMARHU.NET. diciembre 10 de 2012. [En: http://www.xumarhu.net/grid_computing.ppt]

⁵ Computación GRID. Trabajo realizado por: Isaac David Barsallo. Fecha: Jueves 1 de julio de 2010. Disponible en: <http://www.slideshare.net/Savoy158/computacion-grid>

De igual manera la computación distribuida toma ventaja de la nueva generación del protocolo IP, pues el nuevo protocolo de Internet IPv6 permitirá trabajar con una Internet más rápida y accesible. Una de las ideas que se tienen en la superación de las limitaciones actuales del protocolo IPv4, es la aparición de nuevos niveles de servicio que harán uso de la nueva capacidad de la red para intercomunicar los dispositivos.

Los recursos están distribuidos en dicha red de manera transparente, pero teniendo estrictas pautas de seguridad y políticas de gestión.

Entre los requisitos que debe cumplir cualquier computación distribuida están:

- Los datos deben poder compartirse con miles de usuarios.
- Debe suministrarse total seguridad al sistema y a los datos.
- Los datos del sistema deben ser accesibles en cualquier lugar y en cualquier momento.
- El sistema debe aceptar las políticas de gestión de muchos centros diferentes.

4.1.1.2 Ventajas de los sistemas distribuidos con respecto de los centralizados:

- Alta disponibilidad. Gracias a la conexión de múltiples dispositivos, si un servidor presenta un problema, se pueden reasignar los servicios en los servidores restantes, por el contrario en un sistema centralizado si el servidor falla, todo el sistema se ve afectado.
- Reducción de costos: Debido a que los servicios son gestionados por grupos de equipos, no se hace necesario instalar servidores de gran tamaño
- Excelente capacidad de balanceo de sistemas: debido a que es posible reasignar la capacidad desde la granja de recursos a donde se necesite, así se evita el cálculo de la capacidad en función de los picos de trabajo
- Economía: Es una razón muy importante debido a que este tipo de sistemas posee en potencia una relación precio/desempeño mucho mejor que la de un sistema centralizado.
- Distribución inherente: Permite el alojamiento de partes del sistema en forma inherente; es decir, algunas aplicaciones utilizan máquinas que están separadas a cierta distancia.
- Velocidad: un sistema distribuido puede lograr mayor poder de cómputo que un computador central.
- Crecimiento por incrementos: si es necesario aumentar la capacidad de cómputo, mediante un sistema distribuido, se puede añadir más procesadores al sistema, lo que permite un desarrollo gradual conforme surjan las necesidades, en ocasiones un sistema central no permite añadir

capacidad de procesamiento sino que debe cambiarse el computador central por uno nuevo.

4.1.1.3 Ventajas de los sistemas distribuidos con respecto de las PC independientes:

- **Datos compartidos:** Un sistema distribuido hace que múltiples usuarios puedan acceder a una base de datos común con gran facilidad.
- **Dispositivos compartidos:** Mediante un sistema distribuido es posible compartir dispositivos entre múltiples usuarios como puede ser discos duros, unidades lectoras de DVD, y lo mejor compartir poder de procesamiento entre otros dispositivos.
- **Flexibilidad:** Un sistema distribuido soporta fácilmente los cambios y actualizaciones necesarias para lograr llevar a cabo las tareas de manera eficiente.
- **Comunicación:** un sistema distribuido permite la comunicación entre computadoras para el uso de aplicaciones compartidas.

4.1.1.4 Desventajas de los sistemas distribuidos:

- Se posee una alta dependencia al estado de la red, la pérdida o saturación de ésta puede alterar negativamente las ventajas que el sistema distribuido brinda.
- Si no se establecen normas estrictas y se mantiene una vigilancia del acceso de los datos, la información es vulnerable ante la conexión de múltiples usuarios.

4.1.1.5 Características clave de los sistemas distribuidos: De acuerdo a lo anteriormente nombrado se establece que son seis características principales que han sido responsables de la utilidad de los sistemas distribuidos. Se trata de compartición de recursos, apertura (openness), concurrencia, escalabilidad, tolerancia a fallos y transparencia.⁶

- **Compartir recursos:** Los sistemas multiusuario clásicos desde siempre han provisto la compartición⁷ –o multiusuarios- de recursos entre sus usuarios. Sin embargo, los recursos de una computadora multiusuario se comparten de manera natural entre todos sus usuarios. En los sistemas distribuidos es posible, además de compartir componentes hardware como discos e impresoras y procesador, entre otros, también se pueden compartir

⁶ Seguridad de los sistemas distribuidos. 28 de noviembre de 2012. Por: Facultad de Ciencias Exactas y Naturales y Agrimensura-UNNE. [En línea: Exa.unne.edu.ar]

⁷ De acuerdo con los autores del proyecto de grado, la palabra ‘compartición’ equivale a decir, que es un proceso que exige comunicación entre ordenadores por medio de un programa que gestiona dichos recursos.

elementos de software como ficheros, ventanas, bases de datos y otros objetos de datos.

- **Apertura:** Un sistema puede ser abierto o cerrado con respecto al hardware si se pueden agregar periféricos, memoria, procesadores, discos duros, entre otros; o con respecto a software si es posible hacer cambios al sistema operativo, protocolos de comunicación y servicios de compartición de recursos, etc. La apertura de los sistemas distribuidos se determina principalmente por el grado en que se puedan agregar nuevos servicios de compartición sin duplicar o dañar la configuración existente, las interfaces de comunicación son un software clave del sistema, están claramente especificadas y se ponen a disposición de los desarrolladores; en una palabra, las interfaces se hacen públicas.

Los sistemas distribuidos poseen un mecanismo uniforme de comunicación entre procesos e interfaces publicadas para acceder a recursos compartidos; los sistemas distribuidos abiertos pueden construirse a partir de hardware y software heterogéneo y es probable que procedan de vendedores diferentes, pero la conformidad de cada componente con el estándar publicado debe ser cuidadosamente comprobada y certificada para evitar problemas de integración.

- **Concurrencia:** cuando en un computador se corren varios procesos, decimos que se están ejecutando concurrentemente. En el caso de que el computador esté equipado con un único procesador central, la concurrencia tiene lugar entrelazando la ejecución de los distintos procesos, pero si el computador tiene N procesadores, entonces pueden ejecutarse estrictamente a la vez hasta N procesos.

Los sistemas distribuidos cuentan con muchos computadores o máquinas, cada uno posee uno o más procesadores centrales. Es decir, si hay N computadores en un sistema distribuido con un (1) procesador central, cada computador realiza entonces hasta N procesos en paralelo debido a que pertenece a un sistema que está basado en el modelo de compartición de recursos, la posibilidad de ejecución paralela ocurre por dos razones:

- Muchos usuarios pueden interactuar simultáneamente con programas de aplicación.
 - Muchos procesos servidores son ejecutados concurrentemente, cada uno respondiendo a diferentes peticiones de los procesos clientes.
- **Escalabilidad:** Los sistemas distribuidos permiten operar a escalas diferentes sin perder su efectividad y eficiencia. En algunos casos un

sistema distribuido puede estar formado por dos estaciones de trabajo y un servidor de ficheros y en otros casos puede estar construido alrededor de una red de datos de área local o extensa, lo que comprendería cientos o tal vez miles de computadores y/o dispositivos que conformarían el sistema. Lo anterior a menudo suele llamarse red de ordenadores o internetworks y forman un solo sistema distribuido, capaz de permitir que los recursos sean compartidos entre todos ellos; tanto el software de sistema como el de la aplicación no deberían alterarse o sufrir fallas cuando la escala del sistema se incrementa o deba cambiarse.

La necesidad de escalabilidad no es solo un problema de prestaciones de red o de hardware, sino que está íntimamente ligada con todos los aspectos del diseño de los sistemas distribuidos. El diseño del sistema debe reconocer explícitamente la necesidad de escalabilidad o de lo contrario aparecerán serias limitaciones.

- **Tolerancia a Fallos:** los sistemas informáticos pueden fallar en ocasiones debido a múltiples factores. Cuando se produce un fallo de hardware o en el software, los programas podrían arrojar respuestas incorrectas o podrían detenerse antes de terminar el proceso que estaban llevando a cabo. El diseño de sistemas tolerantes a fallos se basa en dos cuestiones complementarias entre sí, la primera es redundancia hardware (uso de componentes redundantes) y la segunda es la recuperación del software (diseño de programas que sean capaces de recuperarse de los fallos).

En los sistemas distribuidos la redundancia puede plantearse en un grano más fino que el hardware, pueden replicarse los servidores individuales que son esenciales para la operación continua de aplicaciones críticas.

La recuperación del software tiene una estricta relación con el diseño del mismo, siendo capaz de recuperarse (roll-back) al estado de los datos permanentes antes de que se ocurriera el fallo.

- **Transparencia:** se puede definir como ocultar las divisiones de los componentes de un sistema distribuido al usuario y al programador de aplicaciones, de manera que el sistema sea visto como un todo, en vez de una colección de componentes independientes. A continuación una breve descripción de los tipos de transparencia:
 - **“Transparencia de Concurrencia:** Los sistemas de gestión de base de datos distribuidas brindan transparencia de concurrencia si es que las transacciones independientes son lógicas y tienen similitud con que se puedan hacer al mismo tiempo, es decir los resultados serían

los mismos se hiciera de una sola vez. Esto sucede con la replicación, por ejemplo, dado que este proceso es asíncrono”⁸.

- **Transparencia de Acceso:** esta transparencia consiste en que el acceso a los segmentos de información remotos se puede realizar de la misma forma que a los segmentos de información locales.
- **Transparencia de Localización:** permite el acceso a los objetos de información sin necesidad de conocer su localización.
- **Transparencia de Replicación:** permite el uso de múltiples instancias de los objetos de información para aumentar las prestaciones y la fiabilidad y sin que los usuarios o los programas de aplicación perciban la existencia de las réplicas.
- **Transparencia de Fallos:** permite a los usuarios y programas de aplicación, terminar sus tareas o procesos a pesar que ocurran fallos en el software o en el hardware.
- **Transparencia de Migración:** permite el traslado de objetos de información dentro de un sistema sin perturbar a los programas de aplicación y/o a los usuarios.
- **Transparencia de Prestaciones:** permite reconfigurar el sistema para mejorar las prestaciones mientras la carga varía.
- **Transparencia de Escalado:** permite el aumento de potencia del sistema y de las aplicaciones sin perjudicar la estructura del sistema o los algoritmos de la aplicación.

En el mercado actual existen variedad de sistemas distribuidos, de acuerdo a las necesidades de cada empresa o usuario entre ellos están:

El Modelo cliente servidor: Es uno de los modelos más utilizados y reconocidos actualmente en este modelo existe un conjunto de procesos servidores, donde cada uno actúa como un gestor de recursos para una colección de recursos del mismo tipo, y también encontramos una colección de procesos clientes, donde cada uno lleva a cabo un proceso que requiere uno o más de los recursos hardware y/o software compartidos.

⁸ La transparencia en las bases de datos distribuidas. Transparencia de concurrencia. Por: Ing. En Sistemas, Abraham Sopla Maslucán. Edición: viernes, 21 de septiembre de 2012. Bajado: 25 octubre de 2013. En: [<http://amazonasopina.blogspot.com/2012/09/la-transparencia-en-las-bases-de-datos.html>]

Los procesos servidores pueden en ocasiones tener la necesidad de acceder a recursos compartidos manejados por otros procesos, así que algunos procesos son a la vez clientes y servidores. En este modelo, todos los recursos compartidos son mantenidos y manejados por los procesos servidores. Los procesos clientes realizan peticiones a los servidores cuando necesitan acceder a algún recurso, si dicha petición es válida, entonces el servidor lleva a cabo la acción requerida y envía una respuesta al proceso cliente.

El termino proceso se usa en el sentido clásico de los sistemas operativos, donde un proceso es un programa en ejecución.

El Middleware, es un software distribuido requerido para facilitar la interacción entre clientes y servidores. El fácil y transparente acceso a servicios y recursos no locales distribuidos a través de una red se provee a través del middleware, ya que sirve como marco para las comunicaciones entre las partes cliente y servidor de un sistema. Los middleware se encargan de definir las API que deben usar los clientes para hacer una petición de servicio a un servidor, la transmisión mediante la red de la petición y de la respuesta desde el servidor al cliente. Algunos “ejemplos de middleware estándar para dominios específicos incluyen: ODBC, para bases de datos, Lotus para groupware, HTTP y SSL para Internet y CORBA, DCOM y JAVA RMI para objetos distribuidos⁹.

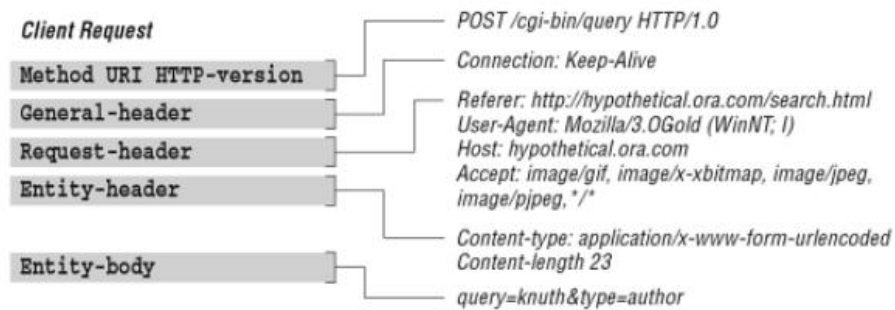
4.1.2 HTTP Hypertext Transfer Protocol: HTTP o Protocolo de transferencia de Hipertexto fue desarrollado por el World Wide Web Consortium o W3C, inicialmente conto con la colaboración de la Internet Engineering Task Force, y es el protocolo que se encuentra detrás de la World Wide Web ('www') en cada transacción web, bien sea de un documento, gráfica, en cada click de un hipervínculo, o en cada envío de un formulario, en cada intento de La web para tratar de distribuir información a través de internet, HTTP es el protocolo utilizado para lograrlo; debido a que entrega una manera estándar para que las computadoras se comuniquen entre sí, “HTTP especifica como los clientes solicitan la información, y como los servidores responden dichas peticiones”¹⁰.

4.1.2.1 Estructura de las transacciones HTTP: En HTTP todas las peticiones y las respuestas siguen la misma estructura general. La siguiente imagen muestra la estructura de una petición de un cliente.

⁹ COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. Sistemas distribuidos – concepto y diseño, 2005. Editorial ADDISON – WESLEY. ISBN 8478290494.

¹⁰ WONG, Cliton. HTTP Pocket Reference. O'Reilly & Associates, Inc. 2000.

Imagen 2. Estructura de una petición del cliente¹¹



Fuente: Estructura de una petición del cliente. WONG, Cliton. HTTP Pocket Reference. O'Reilly & Associates, Inc. 2000.

Las peticiones HTTP tienen los siguientes componentes generales:

1. La primera línea le indica al servidor que método usar, a que documento aplicarlo y que versión de HTTP está usando el cliente. Los métodos posibles en HTTP 1.1 son GET, POST, HEAD, PUT, LINK, UNLINK, DELETE, OPTIONS y TRACE. HTTP 1.0 no soporta los métodos OPTIONS y TRACE. No todos los métodos deben ser soportados por el servidor. La URL indica la ubicación del documento al cual aplicar el método seleccionado, cada servidor cuenta con su propia manera de traducir el contenido de la URL en un recurso utilizable.
2. Los encabezados generales (general headers) del mensaje son opcionales utilizados tanto en la petición del cliente como en la respuesta del servidor. Estos encabezados indican información general como la hora actual o la ruta a través de la red que usan el cliente y el servidor.
3. Las cabeceras de petición (request headers) le indican al servidor más información acerca del cliente. El cliente puede identificarse a sí mismo y al usuario ante el servidor y especificar el formato en el que desea visualizar los documentos desde el servidor.
4. Las cabeceras de entidad (entity headers) son utilizadas cuando una entidad (un documento) está a punto de ser enviado. Estas especifican información acerca de la entidad, como el esquema de codificación, tamaño, tipo y origen¹².

Nota: Los Esquemas de codificación estandarizan la codificación de los juegos de caracteres mediante la definición de un conjunto de reglas para representar los

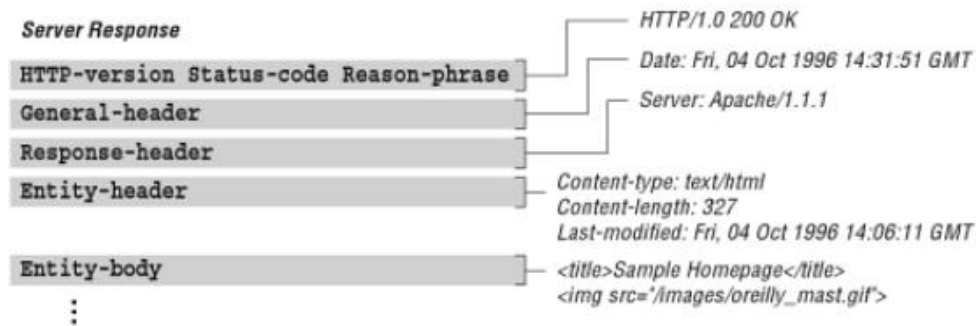
¹¹ Ibid. Pág. 8.

¹² WONG, Cliton. HTTP Pocket Reference. Editorial: O'Reilly & Associates, Inc. ISBN 1-56592-862-8. 2000.

datos de tipo carácter. Cada esquema de codificación consta de un número de páginas de códigos que cumplen estas reglas¹³.

En la siguiente imagen se ilustra la respuesta del servidor WEB, a la petición HTTP realizada por un cliente web.

Imagen 3. Estructura de una respuesta del servidor web¹⁴



Fuente: WONG, Cliton. HTTP Pocket Reference. Estructura de una respuesta del servidor web. Editorial: O'Really & Associates, Inc. ISBN 1-56592-862-8. 2000.

En la respuesta del servidor, las cabeceras generales (general headers) y las cabeceras de entidad (entity headers) son las mismas que se usan en la petición del cliente. La etiqueta contenido de la entidad (entity body) es la misma que se utiliza en la petición del cliente, excepto porque esta vez es utilizada como respuesta.

La primera parte de la primera línea indica la versión de HTTP que el servidor está usando, el servidor hará cualquier intento para ajustarse a la versión de HTTP más compatible con la que el cliente está usando, el código de estado (estatus-code) indica el resultado de la petición y la frase de explicación (Reason-phrase) es una descripción del código de estado entendible para los humanos.

La cabecera de respuesta (response-header) le indica al cliente la configuración del servidor, le informa al cliente los métodos que soporta el servidor, la autorización de peticiones, o indica que intente de nuevo¹⁵.

4.1.3 HTML HYPERTEXT MARKUP LANGUAGE: Es un lenguaje de programación diseñado para permitir la creación de sitios web. Estos sitios web pueden ser vistos por cualquier persona conectada a Internet. Es un lenguaje relativamente fácil de aprender, los temas básicos son accesibles a la mayoría de la gente en una sola sesión, y es muy potente en lo que le permite crear. Está

¹³ Esquemas de codificación. Por: IBM, 2010. [En: IBM Information Software for 2/OS Solutions Information Center. Enero 3 de 2003] [pic.dhe.ibm.com/infocenter/dzichelp/v'r2/indez.jsp?lopic=%2fcom.ibm.db2z10

¹⁴ WONG, Cliton. HTTP Pocket Reference. Editorial: O'Really & Associates, Inc. ISBN 1-56592-862-8. 2000. P.9

¹⁵ *Ibíd.* P. 10.

constantemente en proceso de revisión y evolución para satisfacer las demandas y necesidades de la audiencia de Internet que crece bajo la dirección de la W3C, el organismo encargado de diseñar y mantener el lenguaje.

La definición de **HTML** es **HyperText Markup Language**.

- HyperText o Hipertexto es el método por el cual nos movemos alrededor de la web, haciendo clic en texto especial que llamado hipervínculos el cual nos lleva a la siguiente página. El hecho de que sea “Hyper” sólo significa que no es lineal es decir, usted puede ir a cualquier lugar en Internet siempre que lo desee haciendo clic en los vínculos, no hay un orden establecido para hacer las cosas.
- Markup o Marcado es lo que las etiquetas HTML le hacen al texto en su interior. Ellas lo marcan como un cierto tipo de texto (texto en cursiva, por ejemplo).
- HTML es un lenguaje, ya que tiene palabras en clave y la sintaxis como cualquier otro idioma.

4.1.3.1 Como funciona HTML: HTML consta de una serie de códigos breves escritos en un archivo de texto por el autor del sitio y estas son las etiquetas. El texto se guarda como un archivo HTML, y es visualizado a través de un navegador web, como Internet Explorer o Netscape. Este navegador lee el archivo y traduce el texto en una forma visible, se espera que despliegue la página tal como el autor había previsto. Escribir código HTML implica el uso de las etiquetas correctamente para crear su visión. Puedes usar cualquier cosa, desde un rudimentario editor de texto a un editor gráfico de gran alcance para crear páginas HTML¹⁶.

4.1.4 XML EXTENSIBLE MARKUP LANGUAGE: XML o lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por la W3C, esta es una simplificación y adaptación del lenguaje SGML. XML permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML), por lo tanto XML no es considerado un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

Algunos de los lenguajes que lo usan para su definición son XHTML, SVG, MathML, Android.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Es posible usarlo en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable, debido a que es una tecnología muy sencilla, posee a su alrededor tecnologías que logran complementarlo y hacerlo

¹⁶ SHANNON, Ross. What is HTML? Hyper Text Markup Languaje explained en HTML Sources HTML Tutorials. Editado el 21 de agosto de 2012. Fecha consulta: 03 de enero de 2013. [en: <http://www.yourhtmlsource.com/starthere/whatishtml.html>]

mucho más grande, con posibilidades enormes y básicas para la sociedad de la información. XML en asocio con tecnologías relacionadas, representa una manera más avanzada de hacer las cosas, cuya novedad principal consiste en que permite compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, XML tiene parte importante en el desarrollo del mundo actual, ya que cada vez más el mundo tiende a la globalización y la compatibilidad entre sistemas. Esto debido a que XML será la tecnología que permita compartir la información de una manera fácil, confiable, y sobre todo segura. Así mismo, XML será quien permita al programador dedicar mayor tiempo y esfuerzo a tareas importantes mientras trabaja con los datos, ya que tareas como la validación de estos o el recorrido de las estructuras corren por cuenta del lenguaje y está especificado por el estándar, de modo que no tendrá que preocuparse por ello.

4.1.4.1 Ventajas del XML: una de las mayores ventajas de XML es capacidad de extender después de diseñado y puesto en producción. En XML es posible agregar nuevas etiquetas, de tal modo que se pueda continuar usando sin complicación alguna.

XML usa un analizador estándar, lo que evita crear un analizador específico para cada versión de lenguaje XML, esto facilita el empleo de cualquiera de los analizadores disponibles y de esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.

Debido a su estructura estándar se hace fácil la interpretación por parte de nuevos usuarios. Además logra comunicar aplicaciones de distintas plataformas, sin importar el origen de los datos. Es decir, comunicar diferentes bases de datos en distintos sistemas operativos.

4.1.5 Los servicios Web: Es una aplicación que puede ser descrita, publicada, localizada e invocada a través de una red, generalmente Internet. Combinan los mejores aspectos del desarrollo basado en componentes y la Web¹⁷.

Los servicios web poseen funcionalidades que están ocultos al público, estas pueden ser reutilizadas sin saber o preocuparse como fueron implementadas, los servicios web a diferencia de los componentes, son accedidos usando protocolos web como pueden ser HTTP y XML, y no mediante protocolos específicos del modelos de objetos como son RMI, DCOM p IIOP.

Los consumidores de los servicios web pueden ser implementados en cualquier plataforma y en cualquier lenguaje de programación, debido a que su interface está definida en términos de los mensajes que él mismo acepta y retorna, es decir,

¹⁷ WEB SERVICES. Diciembre 20 de 2012. Extechnical es el portal de acceso a recursos técnicos para los usuarios Genexus. [En: http://www.gxtechnical.com/gxdlsp/pub/genexus/internet/tecnic/papers/web_servic]

solo debe poder crear y consumir los mensajes definidos por la interface de los servicios web.

4.1.5.1 El modelo de los servicios web: “La arquitectura básica del modelo de web services describe a un consumidor, un proveedor y ocasionalmente un corredor (broker). Relacionados con estos agentes están las operaciones de publicar, encontrar y enlazar. La idea básica consiste en que un proveedor publica su servicios en un corredor, luego un consumidor se conecta el corredor para encontrar los servicios deseados y una vez que lo hace se realiza un lazo entre el consumidor y el proveedor. Cada entidad puede jugar alguno o todos los roles”¹⁸.

Para que el modelo funcione y se logren desarrollar o consumir correctamente los servicios web se deben cumplir los siguientes requerimientos.

- Se debe poseer una forma estándar de representar los datos.
- Se debe contar con un formato común y extensible de mensajes.
- Es necesario un lenguaje común y extensible para describir los servicios.
- Se debe tener una forma de descubrir los servicios en Internet.

4.1.5.2 Ventajas de los servicios web: los servicios web poseen excelentes características, las cuales pueden ser base para la nueva generación de sistemas distribuidos, entre ellas esta:

- Ubicuidad: Los servicios web se comunican utilizando HTTP y XML, gracias a esta razón cualquier dispositivo que logre manejar estas tecnologías puede implementar o acceder a dichos servicios. Se espera que muy pronto estén presentes en teléfonos, autos e incluso máquinas expendedoras, las que avisarán a la central cuando el stock sea menor al indicado.
- Fácil de Usar: El concepto detrás de los servicios web es fácil de entender, ya que incluso existen toolkits de vendedores como IBM o Microsoft que ayudan a desarrolladores a crear servicios web en forma fácil y rápida.

¹⁸ Web Services. El modelo de web service. Consulta agosto 27 de 2013. [En: http://www.gxtechnical.com/gxdlsp/pub/genexus/internet/technicalpapers/web_services.htm]

- **Interoperabilidad:** Un servicios web puede interactuar fácilmente con otro servicio web, debido a que pueden ser implementados en cualquier lenguaje, por lo cual no es necesario que el desarrollador se adapte a un ambiente de desarrollo en especial para producir o consumir un servicio web.
- **Reduce la complejidad al encapsular:** Esto sucede debido puesto que para consumir un servicio web lo importante es conocer la interface que el servicio provee y no como esta implementado.
- **Soporte de la Industria:** Muchas empresas apoyan actualmente el desarrollo de servicios web, debido a que las empresas más importantes de software soportan SOAP. Por ejemplo la nueva plataforma .Net del gigante Microsoft, está basada en servicios web, haciendo aún más simple el desarrollo de los mismos, que podrían ser luego consumidos por un servicio web desarrollado utilizando Visual Age de IBM.

4.1.6 SOAP (Simple Object Access Protocol): Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML¹⁹. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

El protocolo SOAP proporciona un mecanismo sencillo y ligero para el intercambio de información estructurada y no escrito entre los compañeros en un entorno descentralizado y distribuido usando XML. SOAP no define en sí misma ninguna semántica de la aplicación como un modelo de programación o la semántica de aplicación específica, sino que define un mecanismo simple para expresar la semántica de aplicaciones, proporcionando un modelo de paquetes modulares y la codificación de los mecanismos de codificación de datos dentro de los módulos. Esto permite SOAP para ser utilizado en una gran variedad de sistemas que van desde sistemas de mensajería RPC.

4.1.6.1 SOAP consta de tres partes:

Las cabeceras SOAP definen la construcción de un marco general para expresar lo que está en un mensaje, que debe realizar con él, y si es opcional u obligatorio.

Las reglas de codificación SOAP, definen un mecanismo de serialización que puede ser utilizado para el intercambio de los tipos de datos definidos por la aplicación.

¹⁹ Newcomer, Eric. Understanding Web Services XML, WSDL, SOAP, and UDDI. Editorial Pearson Education. Indianapolis, 2002. ISBN: 0-201-75081-3.

La representación SOAP RPC define una convención que se puede utilizar para representar llamadas a procedimientos remotos y respuestas.

4.1.6.2 Objetivos del diseño: un objetivo importante de SOAP es la simplicidad y extensibilidad. Esto significa que hay varias características de los sistemas de mensajería tradicionales y los sistemas de objetos distribuidos que no forman parte de la especificación SOAP núcleo. Tales características incluyen: de basura distribuidos colección, encajonamiento de hormigón o de los mensajes, objetos por referencia (que requiere la recolección de basura distribuida) y activación (que requiere que los objetos por referencia).

4.2 MARCO CONCEPTUAL

- **ERP o planificación de recursos empresariales:** son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía en la producción de bienes o servicios.
- **Utility computing:** El término viene de las llamadas utilities, que en inglés son las empresas públicas que tienen como modelo de negocios el cobro de lo que se consume. Utility Computing es un modelo de negocios donde se paga al proveedor del producto o al prestador de servicios, sólo lo que fue utilizado.
- **Clusters:** El término clúster (del inglés cluster) se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardware común y que se comportan como si fuesen una única computadora. Simplemente, un clúster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.
- **Middleware:** Middleware es un software de computadora que conecta componentes de software o aplicaciones para que puedan intercambiar datos entre éstas. Es utilizado a menudo para soportar aplicaciones distribuidas. Esto incluye servidores web, servidores de aplicaciones, sistemas de gestión de contenido y herramientas similares. Middleware es especialmente esencial para tecnologías como XML, SOAP, servicios web y arquitecturas orientada a servicios.
- **Roll-back:** En tecnologías de base de datos, un rollback es una operación que devuelve a la base de datos a algún estado previo. Los Rollbacks son importantes para la integridad de la base de datos, a causa de que

significan que la base de datos puede ser restaurada a una copia limpia incluso después de que se han realizado operaciones erróneas. Son cruciales para la recuperación de crashes de un servidor de base de datos; realizando rollback(devuelto) cualquier transacción que estuviera activa en el tiempo del crash, la base de datos es restaurada a un estado consistente.

- **SSL:** Siglas de Secure Sockets Layer, un protocolo de seguridad desarrollado por la compañía Netscape en el año 1994. A partir de ese momento y hasta la fecha, se ha ido actualizando y mejorando su funcionamiento hasta llegar a ser uno de los sistemas de cifrado y encriptación de datos más populares de Internet, llegando a su madurez absoluta con la versión 3.0. Pero las bondades de SSL no se limitan sólo a cifrar datos importantes que enviamos y recibimos a través de Internet, también permite entre otras cosas, autenticar tanto a los clientes como a los servidores que realizan conexiones TCP/IP, mejorando considerablemente el nivel de seguridad de sus versiones anteriores.
- **CORBA:** (Common Object Request Broker Architecture). CORBA es un estándar definido por el OMG (Object Management Group) que permite escribir componentes de software en múltiples lenguajes y ejecutarlos en múltiples computadoras para trabajar juntos. Es una arquitectura estándar para sistemas de objetos distribuidos. Permite una distribución, colección heterogénea de objetos para interoperar.
- **JAVA RMI:** (Java Remote Method Invocation) es un mecanismo ofrecido por Java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de Java y proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java. Si se requiere comunicación entre otras tecnologías debe utilizarse CORBA o SOAP en lugar de RMI.
- **ODBC:** Open DataBase Connectivity (conectividad abierta de bases de datos) es un estándar de acceso a las bases de datos desarrollado por SQL Access Group en 1992. El objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos (DBMS) almacene los datos. ODBC logra esto al insertar una capa intermedia (CLI) denominada nivel de Interfaz de Cliente SQL, entre la aplicación y el DBMS. El propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda. Para que esto funcione tanto la aplicación como el DBMS deben ser compatibles con ODBC, esto es que la aplicación debe ser capaz de producir comandos ODBC y el DBMS debe ser capaz de responder a ellos. Desde la versión 2.0 el estándar soporta SAG y SQL.

- **SGML:** Standard Generalized Markup Language Es un metalenguaje de donde deriva el HTML y el XML. Fue definido por la norma ISO 8879 en 1986 y desde entonces ha sido considerado el lenguaje estándar para mantener los depósitos centrales de la estructura documental. Se trata de un lenguaje para marcar y describir documentos con independencia total del hardware y software utilizados.
- **MathML:** (Mathematical Markup Language = Lenguaje de Marcas Matemático) es un lenguaje de marcado basado en XML, cuyo objetivo es expresar notación matemática de forma que distintas máquinas puedan entenderla, para su uso en combinación con XHTML en páginas web, y para intercambio de información entre programas de tipo matemático en general. El soporte de MathML es grande en programas matemáticos (como Maple o Mathcad), pero aún escaso en navegadores, solo Google Chrome cuenta con soporte estable de MathML. En Internet Explorer solo está disponible con plugin y cuenta con un soporte parcial en Mozilla.
- **WSDL:** (Web Services Description Language) es un protocolo basado en XML que describe los accesos al Web Service. Podríamos decir que es el manual de operación del mismo, porque nos indica cuáles son las interfaces que provee el Servicio web y los tipos de datos necesarios para su utilización.
- **UDDI:** Es un modelo de directorios para Web Services. Es una especificación para mantener directorios estandarizados de información acerca de los Web Services, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los Web Services. Es un lugar en el cual podemos buscar cuales son los Servicios web disponibles, una especie de directorio en el cual podemos encontrar los Web Services publicados y publicar los Web Services que desarrollemos.
- **W3C:** son las siglas de World Wide Web Consortium, un consorcio fundado en 1994 para dirigir a la Web hacia su pleno potencial mediante el desarrollo de protocolos comunes que promuevan su evolución y aseguren su interoperabilidad. El consorcio está compuesto por un grupo de programadores, desarrolladores web, ejecutivos de la industria y usuarios que ayudan a definir las especificaciones para el desarrollo de la tecnología web.
- **RPC:** (Remote Procedure Call - Llamada a Procedimiento Remoto). Protocolo que permite a una aplicación en una computadora ejecutar código en otra remota sin tener que preocuparse por la comunicación entre ambas. Las RPC suelen utilizarse en los programas tipo cliente/servidor. Las RCP

suponen un avance de los sockets. Existen diversos RPC no compatibles entre sí como ser, el RFC 1057 (RPC de Sun), DCOM (de Microsoft), etc.

- **GNU LGPL:** (GNU Lesser General Public License = Licencia Pública General Reducida de GNU) Es una licencia de software creada por la Free Software Foundation que pretende garantizar la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios.
- **XML:** son las siglas de Extensible Markup Language, una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de SGML, diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones.
- **URI:** Identificador de Recurso Uniforme (Uniform Resource Identifiers). Los URI son cadenas que funcionan como identificadores globales que hacen referencia a recursos en la Web tales como documentos, imágenes, archivos descargables, servicios, buzones de correo electrónico y otros.
- **On-demand:** es algo que se podría clasificar como un web tone, es decir, un servicio que siempre está disponible, pero que las empresas solo pagan cuando las utilizan.

4.3 MARCO LEGAL

Todo el trabajo o material producto de este proyecto de grado estará licenciado bajo la licencia GNU LGPL o GNU Lesser General Public License por las siguientes razones:

- El resultado de este trabajo se mantendrá bajo una licencia “Open Source” o de Código Abierto”.
- Se desea brindar la posibilidad de incluir este trabajo como parte o un elemento de un sistema privado, pero asegurándose que el usuario final recibe este material bajo la misma licencia.
- Se desea que cualquier persona pueda colaborar en el proyecto.
- Se busca que cualquier cambio o mejora realizado a este trabajo sea publicado bajo el mismo pensamiento “Open Source”, confiando en la buena voluntad de quien realiza la modificación.

- Esta licencia aplica del mismo modo en cualquier país donde sea utilizado el código fuente que dé como resultado este trabajo de grado.
- Se puede leer la licencia en <http://www.gnu.org/copyleft/lgpl.html> (inglés)

5. DISEÑO METODOLÓGICO

5.1 TIPO DE INVESTIGACIÓN

El éxito de este trabajo depende de la recolección correcta de la información producida durante las pruebas, para ello se deben utilizar una técnica de investigación descriptiva, mediante la cual se busca encontrar una justificación clara de los objetivos de este trabajo resaltando los valores más importantes recolectados durante las pruebas (entre ellos memoria, velocidad, etc.).

Además de la técnica de investigación, se planea utilizar un diseño experimental, debido a que en este diseño se pretende establecer una relación de causa y efecto de un fenómeno, por medio de la manipulación directa de las variables que ejercen incidencia sobre el fenómeno²⁰.

En nuestro caso este diseño es el más adecuado debido a que se requiere controlar la cantidad de datos a ser transferidos y procesados.

5.2 FUENTES DE INFORMACIÓN

5.2.1 Fuentes de información primarias: páginas web, EBooks, libros, artículos científicos, monografías, tesis relacionadas con sistemas distribuidos y servicios web, artículos de revistas especializadas en tecnología y parte de la experiencia ganada por los autores de la tesis durante la colaboración en algunos proyectos relacionados con el tema en empresas del valle del cauca.

5.2.2 Fuentes de información secundarias: Sitios web con pocas referencias bibliográficas, profesores y compañeros de trabajo en general

5.3 POBLACIÓN Y MUESTRA

Las pruebas serán efectuadas en ambientes controlados, en los cuales se realizarán comparaciones entre los dos modelos de comunicación presentados en este proyecto, que constan del mismo servidor y la misma cantidad de clientes para ambas pruebas.

²⁰ RAMÍREZ, Alberto. Metodología de la investigación científica. Edición, 26 de agosto de 2005. Consulta el 01 de abril de 2013. [En: <http://javeriana.edu.co/fear/ecología/documents/ALBERTORAMIREZMETODOLOGIADELAINVESTIGACION>]

5.3.1 Técnicas y procedimientos para la recolección de información: para hacer una correcta recolección se deben tener en cuentas los siguientes tres puntos.

- Registro y recolección de información a partir de pruebas individuales a cada software que actúa en el proceso de comunicación.
- Utilizar los mismos elementos computacionales para cada prueba para evitar variaciones en los resultados.
- Consultar libros virtuales y físicos para aumentar la confiabilidad y veracidad de la información.

6. ESTADO DEL ARTE DE LOS SERVICIOS WEB

Debido a la constante necesidad en la era moderna de automatización de procesos por medio de software y de la velocidad con que internet crece y extiende sus servicios por el mundo, se hace necesario que los desarrolladores busquen estrategias para sacar mayor provecho de cada salto que dan las tecnologías de información. Gracias a estos, ofrecer servicios a través de la red a los usuarios genera diversas ventajas para las empresas²¹, la más importante es la disponibilidad y la accesibilidad que esto agrega a sus productos, dándole un mayor control sobre el estado de la comunicación entre sus sistemas de información, además de mejorar considerablemente su escalabilidad, mantenimiento y redundancia.

Los servicios web pueden definirse como un “conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios”²². Dichos servicios son ofrecidos por proveedores los cuales son llamados procedimiento remotos y los usuarios hacen un llamado a este servicio a través de la web. Así mismo otros sistemas interactúan con el servicios web en una manera prescrita por su descripción usando mensajes SOAP, típicamente enviados usando HTTP con una serialización XML en relación con otros estándares relacionados con la web, finalmente se puede decir a manera de resumen que los servicios web son herramientas de software que permiten el intercambio de datos, recursos y funcionalidades entre dos sistemas conectados entre sí a través de una red de comunicaciones.

Básicamente los servicios web de la actualidad están basados en tres componentes²³: el solicitante, el proveedor y el servicio de registro, los cuales se ven reflejados en la infraestructura mínima para su implementación: un modo de comunicación (SOAP), un modo para describir el servicio (WSDL) y un servidor para el servicio (UDDI). En la actualidad WSDL, UDDI y SOAP son el núcleo de los servicios web ya que muchas otras especificaciones y arquitecturas son diseñadas basándose en ellas.

Nota: debido al alcance y a los objetivos de este trabajo, en los cuales solo se consumen servicios web privados, el catálogo de negocios de Internet UDDI no será tratado en los siguientes capítulos y solo será mencionado en caso de ser estrictamente necesario.

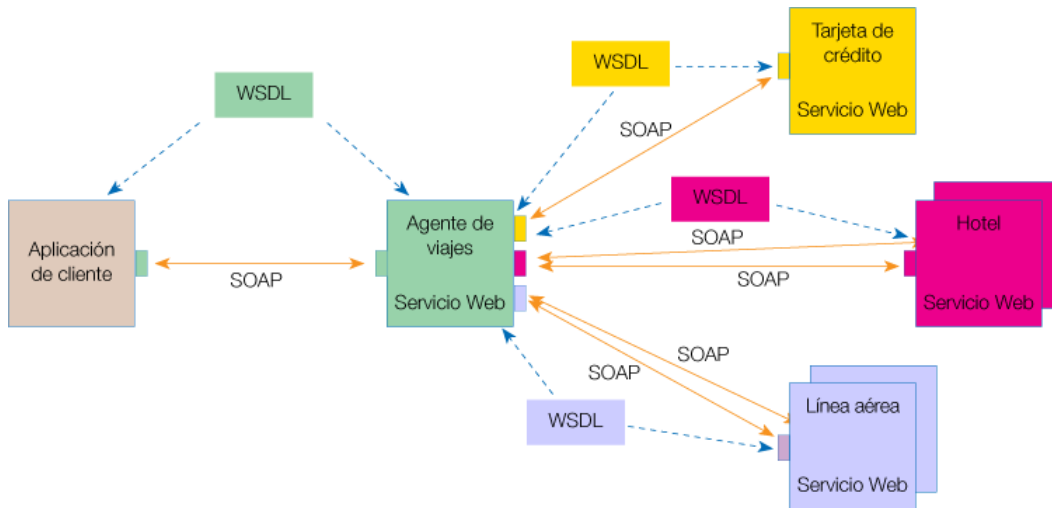
²¹ KUMAR, B. V. Web Services: An Introduction. Editorial Tata McGraw Hill. New Delhi. Número Estándar: 1259027805.

²² <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

²³ GUSTAVO, Alonso; CASATI, Fabio; KUNO, Harumi; MACHIRAJU, Vijay. Web Services: concepts, architectures and applications. Berlín, 2003. Editorial Springer. Número Estándar: B004CRTNAO.

Gracias, a esta arquitectura básica, los servicios web pueden a su vez ser clientes de otro servicio web como indica la imagen 4.

Imagen 4. Gráfica de interacción entre varios servicios web.²⁴



Fuente: Guía breve de servicios web. España. World Wide Web Consortium (W3C). Disponible en <http://www.w3c.es/divulgación/guíasbreves/serviciosweb>, 2012.

En la gráfica anterior, se muestra como se comunican los servicios web enviando y recibiendo paquetes SOAP entre ellos a través de la red, cada uno de estos paquetes es construido según las especificaciones contenidas en el archivo de descripción del servicio web WSDL definido para cada servicio.

6.1 PROTOCOLO HTTP COMPARTIENDO DATOS A TRAVÉS DE LA RED

Como se mencionó anteriormente, el envío de los paquetes SOAP desde y hacia los servicios web se hace a través del Protocolo de Transferencia de Hipertexto HTTP, este “es el protocolo que gobierna la transferencia de archivos a través de la red”²⁵ y fue diseñado para soportar Hipertexto, lo que quiere decir que posee la habilidad para interconectar documentos insertando enlaces entre ellos como parte del contenido. En particular HTTP soporta el Lenguaje de Marcado de Hipertexto (HTML) el cual define un conjunto estándar de indicadores textuales especiales para especificar como los navegadores web muestran las palabras e imágenes de un sitio web²⁶.

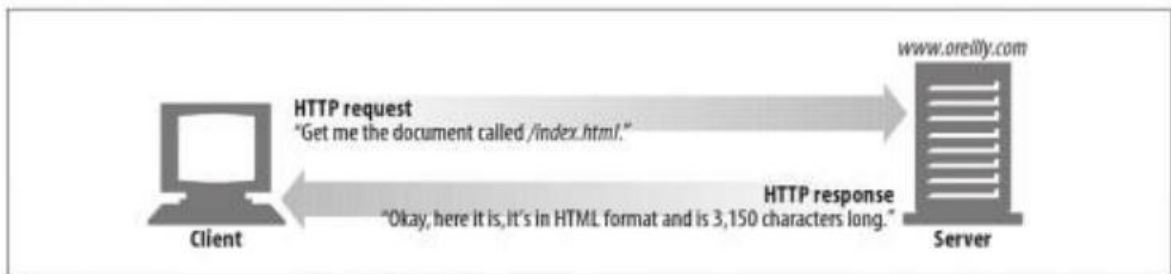
²⁴ Guía breve de servicios web. España. World Wide Web Consortium (W3C). Disponible en <http://www.w3c.es/divulgación/guíasbreves/serviciosweb>, 2012.

²⁵ *Ibíd*, p. 37.

²⁶ *Ibíd*, p. 37.

HTTP se basa en el modelo cliente / servidor, el cual generalmente utiliza sockets TCP. Un cliente HTTP al momento de solicitar un documento debe abrir una conexión hacia el servidor HTTP y enviar una petición la cual está formada por un método de petición, un identificador único de recursos (URI), la versión del protocolo seguido del contenido del mensaje en sí. El servidor emite una respuesta, la cual está formada por una línea de estado que contiene la versión del protocolo utilizado y el código de estado de la respuesta (éxito o error), seguida del contenido de la respuesta (por lo general el documento solicitado por el cliente) y finalmente cierra la conexión.

Imagen 5. Clientes Web y Servidores²⁷



Fuente: Cliente web y servidores: Gourley, David; Totty, Brian. HTTP: The Definition Guide, 2002. Pág. 4. Número estándar 1565925092.

En la imagen se explica de una manera más detallada la estructura de una petición y una respuesta HTTP.

²⁷ Gourley, David; Totty, Brian. HTTP: The Definition Guide, 2002. Pág. 4. Número estándar 1565925092.

Imagen 6. Ejemplo de GET para tools.html²⁸

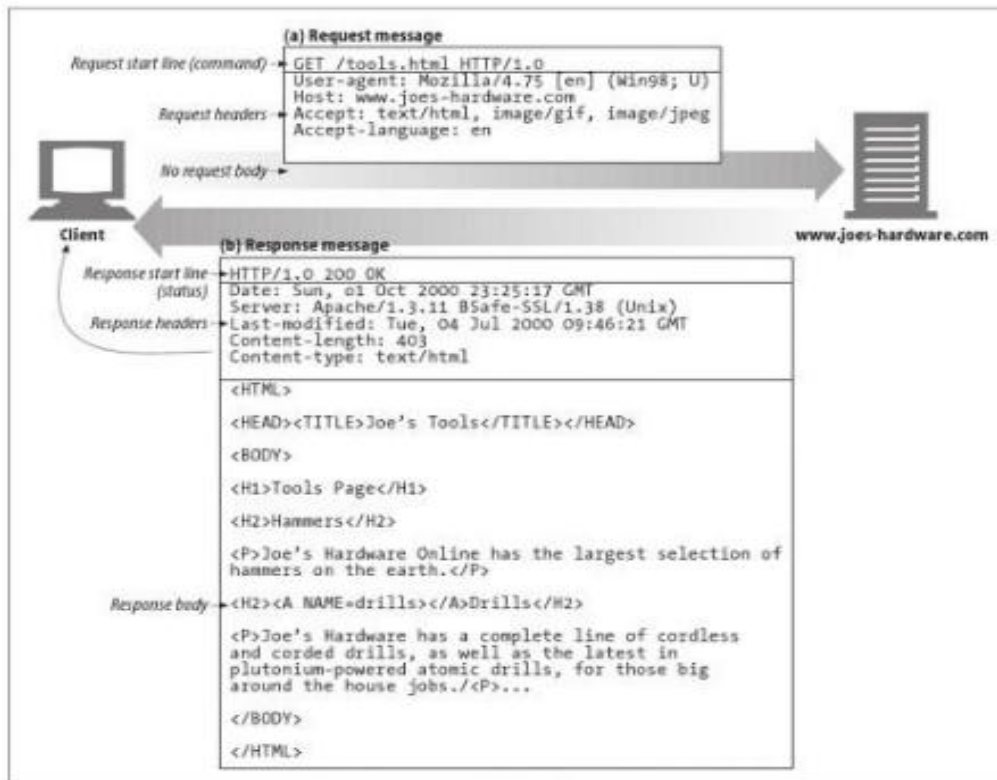


Figure 1-8. Example GET transaction for <http://www.joes-hardware.com/tools.html>

Fuente: Ejemplo de GET para tools.html: Gourley, David; Totty, Brian. HTTP: The Definition Guide, 2002. Pág. 4. Número estándar 1565925092.

En cuanto a la petición HTTP los detalles son:

- 1) GET /tools.html HTTP/1.0, se solicita el documento tolos.html a través del método GET utilizando la versión 1.0 del protocolo HTTP.
- 2) User-Agent es la línea que identifica al cliente ante el servidor.
- 3) Host, es la dirección del servidor a quien se le hace la petición.
- 4) Accept, indica que formato o formatos son esperados por el cliente.
- 5) Accept-lenguaje, es el lenguaje en el que el cliente espera la respuesta.

La respuesta desde el servidor está compuesta de los siguientes detalles:

- 1) HTTP/1.0 200 OK, se utiliza la versión 1.0 del protocolo HTTP para la respuesta, además se envía un código de estado (200) y una descripción legible del código, en este caso "OK".
- 2) Date, la fecha en que se envía la respuesta.
- 3) Server, nombre del servidor y la versión.
- 4) Last-modified, es la última fecha de modificación del documento solicitado.

²⁸ Ibíd. Pág. 12.

- 5) Content-length, es el tamaño del documento o del contenido de la respuesta HTTP.
- 6) Content-type, es el formato en el que se entrega el documento solicitado.
- 7) Más abajo encontramos el contenido de la respuesta, en este caso código HTML.

En el caso de los servicios web, el contenido de las peticiones y respuestas HTTP no se encuentra en lenguaje HTML (Hypertext Markup Language), sino en etiquetas del lenguaje XML (EXTENSIBLE MARKUP LANGUAGE).

6.2 EL LENGUAJE XML, PORQUE HTML NO ES DINÁMICO

XML fue desarrollado para superar las limitaciones de HTML, especialmente para brindar mejor soporte a la creación dinámica de contenido. HTML está bien para definir y mantener contenido estático, pero a medida que la web evoluciona hacia una plataforma de software, en la cual la información posee un significado asociado, el contenido debe ser generado y desplegado dinámicamente. Usando XML, es posible definir cualquier cantidad de elementos que le dan un significado a los datos; esto es, los datos son descritos y se especifica que hacer con ellos por medio de uno o más elementos creados con un propósito específico.

Imagen 7. Ejemplo XML²⁹

```
<Company>
  <CompanyName region="US">
    Skateboots Manufacturing
  </CompanyName>
  <address>
    <line>
      200 High Street
    </line>
    <line>
      Springfield, MA 55555
    </line>
    <Country>
      USA
    </Country>
  </address>
  <phone>
    +1 781 555 5000
  </phone>
</Company>
```

Fuente: Newcomer, Eric. Understanding Web Services XML, WSDL, SOAP, and UDDI. Ejemplo XML. Pearson Education, 2002. Ejemplo XML. Número estándar 0-201-7508-3.

En este ejemplo, XML permite definir no solo elementos que describen datos, sino también estructuras que permiten agrupar datos relacionados. Es fácil imaginar una búsqueda de elementos que cumplan con ciertos criterios como <country> y

²⁹ Newcomer, Eric. Understanding Web Services XML, WSDL, SOAP, and UDDI. Pearson Education, 2002. Ejemplo XML. Número estándar 0-201-7508-3.

<phone> para una <company> en particular o para todas y retornar una lista de elementos que se identifiquen como <company> en la web.

Además, XML permite que esquemas asociados validen la información de manera separada y describir otros atributos cualidades de la información, algo completamente imposible utilizando HTML.

6.3 COMPRENDIENDO SOAP Y EL PORQUÉ DE XML EN LOS SERVICIOS WEB

En el mundo de los servicios web, SOAP es un protocolo de empaquetado para los mensajes que son compartidos por las aplicaciones. La especificación no define más que un recubrimiento basado en XML para la información que es transmitida, y un conjunto de reglas para traducir datos específicos de cada aplicación y/o plataforma en representaciones XML. El diseño de SOAP lo hace viable para una gran variedad de aplicaciones de mensajería y patrones de integración.

SOAP es XML. Eso es todo, SOAP es una aplicación de la especificación XML. Se enfoca principalmente en los estándares XML como XML Schema y XML Namespaces para su definición y su función.³⁰ Las siguientes imágenes 8 y 9 muestran la estructura básica del contenido de los paquetes SOAP para las peticiones y las respuestas.

³⁰ Snell, James; Tidwell, Doug; Kulchenko, Pavel. Programing web services with SOAP. Edit O'Reilly & Associates, Inc. 2002.

Imagen 8. Estructura básica de una petición SOAP

```
SOAP Request
```

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:hello xmlns:ns2="http://ws/">
      <name>data</name>
    </ns2:hello>
  </S:Body>
</S:Envelope>
```

Fuente: Estructura básica de una petición SOAP. Por: Snell, James; Tidwell, Doug; Kulchenko, Pavel. Programing web services with SOAP. Edit O'Reilly & Associates, Inc. 2002.

Imagen 9. Estructura básica de una respuesta SOAP³¹

```
SOAP Response
```

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:helloResponse xmlns:ns2="http://ws/">
      <return>Hello data !</return>
    </ns2:helloResponse>
  </S:Body>
</S:Envelope>
```

Fuente: Estructura básica de una respuesta SOAP. Por: Snell, James; Tidwell, Doug; Kulchenko, Pavel. Programing web services with SOAP. Edit O'Reilly & Associates, Inc. 2002.

Tanto la petición como la respuesta contienen las siguientes etiquetas:

- 1) <?xml... Datos opcionales sobre la versión de xml y la codificación de caracteres utilizada.
- 2) <S:Envelope ... Es la etiqueta que se encarga de contener todo el paquete SOAP, su significado literal es "envoltura" ya que es la etiqueta que cubre todo el paquete.

³¹ Tomado del "tester" para servicios web de Netbeans 7.2

- 3) <S:Header/> Es la cabecera del paquete SOAP en esta se incluye información adicional como por ejemplo los datos de inicio de sesión y seguridad³².
- 4) <S:Body> es la etiqueta que contiene el cuerpo del paquete SOAP, dentro de esta se almacena la información que se desea transmitir puede contener la información de métodos a ejecutar en las peticiones y las estructuras de datos en las respuestas.
- 5) En la petición encontramos <ns2:hello ... esto indica que se va a ejecutar el método hello en el servidor y la etiqueta <name> indica que se envía un parámetro llamado “name” con el valor de “data”. En la respuesta desde el servidor encontramos <ns2:helloResponse... indica que las etiquetas contenidas dentro de ella son el resultado de la operación “hello”; además en dentro de la etiqueta <return> se especifica la información detallada de la respuesta, en este caso “Hello data !”

Básicamente lo que hace el cliente y el servidor es desempaquetar (tomar los datos de “S:Envelope”) la información y procesarla según se indique en el archivo de descripción del servicio web WSDL. Este archivo será tratado con más detalle en el siguiente apartado.

6.4 EL ARCHIVO WSDL Y LAS POSIBILIDADES DE LOS SERVICIOS WEB

El Lenguaje de Descripción de Servicios Web (WSDL por sus cifras en inglés) hace posible a las herramientas de generación automática de código construir clientes para servicios web existentes³³. Cada aplicación expone algún tipo de funcionalidad; esta funcionalidad es invocada a través de varios tipos de operaciones. Dichas operaciones requieren porciones de información específicas, las cuales son utilizadas para producir un resultado. Una vez las operaciones han finalizado la aplicación debería retornar dichos resultados.

Este intercambio completo de información debe ser realizado utilizando algún tipo de protocolo para empaquetar la información y enviarla en cualquier sentido. Sin embargo, la mayoría de las aplicaciones requieren que el desarrollador describa como se supone que debe suceder este intercambio. Los detalles específicos de como un servicio es implementado se convierten en un parte intrincada de la aplicación. En caso de que se deba realizar algún cambio, la aplicación debe ser cambiada y recompilada, este tipo de aplicaciones no son muy flexibles³⁴.

³² SKONNARD, Aaron. Understanding SOAP. Edición marzo de 2003. Consulta: enero 25 de 2013. [En: <http://msdn.microsoft.com/en-us/library/ms995800.aspx>.

³³ SNELL, James; TIDWELL, Dough; KULCHENKO, Pavel. Programming web services with SOAP. Editorial O’Reilly & Associates, Inc. Sebastopol, 2002. Número estándar 0596000952.

³⁴ *Ibíd.*

Con los Servicios Web las aplicaciones pueden descubrir toda esta información de manera dinámica mientras la aplicación se encuentra en ejecución. Esta opción hace los cambios mucho menos perjudiciales.

La especificación SOAP no se ocupa de la “descripción” de los servicios web. La especificación estándar utilizada para hacer servicios web autodescriptivos se llama “Web Services Description Language (WSDL)”³⁵. Utilizando WSDL un servicio web puede describir todo acerca de lo que hace, como lo hace y como sus clientes pueden utilizarlo (consumirlo).

Existen varias ventajas de usar WSDL:

- WSDL hace más fácil escribir y mantener servicios entregando un enfoque más estructurado para definir interfaces de servicios web.
- WSDL hace más fácil consumir servicios web reduciendo la cantidad de código (y errores potenciales) que una aplicación cliente debe implementar.
- WSDL hace más fácil implementar cambios que sean menos propensos a romper aplicaciones clientes SOAP. La lectura dinámica de descripciones WSDL permite a dichos cambios ser enviados a los clientes de manera automática usando WSDL así que esas posibles costosas modificaciones al código del cliente no tengan que ser realizadas cada vez que un cambio ocurre.

6.5 RESTFUL COMO ALTERNATIVA PARA LA CREACIÓN DE SERVICIOS WEB

REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedias distribuidos (entre ellos la Web). REST es un conjunto de principios para el diseño de arquitecturas en red, los cuales indican cómo se deben definir y utilizar los recursos (se le llama recurso a un documento, imagen, etc. disponible vía web); REST es generalmente utilizado cuando se definen interfaces que transmiten datos sobre HTTP sin utilizar una capa adicional, como lo hace SOAP (el cual requiere almacenar la información dentro un paquete adicional, el cual debe ser creado cada vez que se envía una petición o respuesta SOAP).

³⁵ Este tutorial está diseñado para que usted pueda comprender Web Services Description Language (Lenguaje de descripción de servicios web - WSDL). Está pensando para desarrolladores que desean exponer sus propios servicios al uso de otros utilizando WSDL y también para desarrolladores que tengan el archivo WSDL para un servicio al que desean acceder y necesitan crear un cliente. **Fecha de edición:** 08-08-2011. Consulta: 31-10-2013. En: <http://www.ibm.com/developerworks/ssa/webservices/tutorials/ws-understand-web-services2/>

³⁶ RODRÍGUEZ, Alex.

La aplicación de REST en este trabajo no se cubre ya que la solución propuesta se enfoca netamente en servicios web basados en SOAP.

6.6 USO ACTUAL DE LOS SERVICIOS WEB

Gracias a las grandes y poderosas ventajas que ofrecen los servicios web, en la actualidad están siendo implementados por diversidad de proveedores de software y aplicaciones no solo en temas netamente informáticos, sino también en distintos sectores tales como el financiero, administrativo, salud, química, agricultura, entre otros. Algunas de las ventajas que tienen los servicios web son: fácil integración, extensibilidad, poca complejidad, interoperabilidad con distintas aplicaciones, estandarización, independientes de la plataforma. Utilizando estas ventajas frente a otros tipos de sistemas distribuidos las empresas logran diseñar ecosistemas de software los cuales trabajan de manera independiente pero que a su vez se integran como parte de un todo para lograr aumentar el rendimiento de sus procesos internos.

Es difícil especificar de manera precisa como se están utilizando los servicios web en las empresas nacionales ya que debido a la importancia de la información que procesan, la variedad de trabajos que realizan, el avance individual en cuanto a optimización y rendimiento, tanto como los derechos de autor que residen sobre cada uno, lo más probable es que dichos servicios operen de manera privada y muy seguramente cuenten con sistemas anti espías que protejan la información vital de la empresa.

No obstante, algunas compañías publican sus servicios para que sean utilizados de manera pública, un ejemplo de esto es el caso de Amazon Web Service (AWS) quien ha hecho público su servicio con el siguiente anuncio: “Estamos muy contentos de anunciar el soporte para Auto Scaling en la aplicación móvil de AWS Console, disponible en Amazon Appstore, Google Play, y iTunes. Utilice la aplicación para ver el estado y la salud de sus recursos de AWS en su dispositivo móvil. La aplicación cuenta con soporte para EC2, Elastic Load Balancing, Amazon Relational Database Service, Auto Scaling, CloudWatch y el tablero de instrumentos del Servicio de Salud”³⁷. Soporte Amazon CloudFront para POST, PUT y otros métodos HTTP, Amazon EC2 ofrece ahora Modificación instancia reservada para el tipo de instancia, entre muchos otros.

En el siguiente apartado se mencionan algunos casos públicos donde el uso de servicios web ha mejorado significativamente los procesos realizados por cada uno de los actores involucrados en la noticia.

³⁷ AWS Console for iOS/Android Now Supports Auto Scaling [Octubre 21 de 2013]. Consulta: 21-11-2013. Disponible en: <http://aws.amazon.com/es/about-aws/whats-new>

6.6.1 Casos de estudios de la contribución de los servicios Web al mercado: como prueba de la diversidad y aplicación cada vez mayor de los servicios web, y su contribución al rendimiento de los procesos puntuales en las empresas, se presentan los siguientes casos de estudio de empresas que decidieron optar por su uso y mejorar trascendentalmente el rendimiento de sus procesos.

- **CASO DE ESTUDIO ROTECH HEALTHCARE**

“Rotech Healthcare, situada en Orlando, florida, provee servicios médicos a través de una red distribuida de pequeñas compañías en 650 lugares, además mantiene poco personal de TI y busca solucionar problemas del negocio por medio de iniciativas de integración ligeras.

Para dar un mejor uso de los recursos de ventas, Rotech deseaba comparar datos acerca de la localización de los doctores con la de los pacientes. Tras un estudio de la ubicación y tiempo de trabajo de sus doctores, se reveló altas incongruencias y anomalías, debido a que se envían más doctores de los necesarios a lugares con pocos pacientes, o lo contrario.

Para solucionar este problema, la empresa decide, establecer un enlace SOAP entre los orígenes de datos, para por medio de este alimentar una aplicación analítica propia. El uso de los servicios web ayudo a Rotech a establecer la conexión más rápidamente de lo que hubiera podido hacerlo por medios convencionales, gasto muy poco tiempo y dinero en el proyecto, utilizando uno o dos desarrolladores por día por menos de dos semanas.

Rotech afirma que los resultados proveen beneficios inmediatos, pues ahora puede ver con mayor claridad como sus recursos de ventas deben ser desplegados para mejorar la utilización de sus servicios principales de negocio, incluyendo en circunstancias en las que su número de pacientes implica que su número de doctores sea muy bajo”³⁸.

- **“CASO DE ESTUDIO DEPARTAMENTO DE AGRICULTURA DE COLORADO**

El departamento de agricultura de colorado necesitaba modernizar el acceso a múltiples recursos de información que usa para el seguimiento de la salud de la población de alces en cautiverio en el estado. Antes de comenzar el proyecto de servicios web, los empleados debían combinar

³⁸ Uso actual de los servicios web. Informes ejecutivos Gartner. [Fecha: 01-11-2013]. En: Web Services in Action: Case Studies From the Front Line. En: http://www.gartner.com/5_about/news/ws_sample.pdf

manualmente la información de múltiples fuentes en la aplicación HyperCard de Mac para sus análisis. Definiendo, ensamblando e inspeccionando los reportes se podía tardar hasta seis semanas.

Cuando un alce es diagnosticado con la enfermedad es crítico determinar con que otros alces ha tenido contacto, para rastrear el paciente cero de cualquier brote también para evaluar el riesgo de contagio en los otros alces. Por lo tanto la espera de seis semanas era un perjuicio significativo para la rápida respuesta ante la situación, adicionalmente, la aplicación con la que los datos eran analizados no residía en la red, por lo tanto era difícil acceder.

La fuente tecnológica dominante de información era una base de datos que generaba reportes impresos que el empleado de turno digitaba en la aplicación de Hyper tarjetas, donde la lógica de negocio y la información residía. La base de datos relacional consistía en acerca de veinte tablas, con catorce a treinta columnas por tabla. Eran comunes largas esperas para rastrear los contactos realizados por un alce enfermo.

En la solución con servicios web, los usuarios realizaban sus peticiones a través de una interfaz web ASP, la cual envía las peticiones a través de una librería dinámica de enlaces utilizando paquetes SOAP. Estos a su vez representan una capa de transacciones de objetos COM+, los cuales interactúan con la base de datos. Los resultados son entregados al usuario a través de la capa de presentación ASP. La capa SOAP para el transporte de datos fue adoptada para incrementar la flexibilidad futura de la aplicación.

Los oficiales reportaron que podían rastrear el contacto de un alce con otro alce de manera inmediata, en vez de esperar varias semanas un reporte. El modo universal de ver la información que provee el nuevo sistema es crítico para crear estrategias unificadas de publicación de información acerca de brotes o enfermedades en los alces.

• CASO DE ESTUDIO MAPPOINT

MapPoint muestra el potencial y los desafíos de la implementación completa de servicios Web. MapPoint es un candidato natural para el uso de los servicios Web porque el mapeo de datos tiene alto valor en muchos contextos, pero no es algo que la mayoría de las empresas quieren recopilar o administrar por sí mismas. Los datos mapeados aparentan ser estáticos pero en la práctica cambian significativamente todo el tiempo, especialmente cuando son complementados por datos como “puntos de interés”. MapPoint brinda la posibilidad de que otros propietarios de datos especializados distribuyan su información a través de servicios web.

MapPoint comprende una base de datos sustancial de datos cartográficos y un programa para buscar y visualizar estos datos. La aplicación MapPoint fue concebida y diseñada para ejecutarse como una aplicación para PC de un solo usuario. Se encuentra también disponible como un control ActiveX con una amplia interfaz de objetos, destinados para ser incrustados en otras aplicaciones. El reto para Microsoft MapPoint era convertirse en un sistema multiusuario, accesible a través de Internet, y capaz de integrarse con otros sistemas que requieren Mapeo de los datos.

Con una API SOAP, una especificación de interfaz WSDL y un motor de alto rendimiento de acceso a datos, el servicio web de MapPoint estaba "listo para volar." Los desafíos más importantes comprenden crear una infraestructura de despliegue, términos comerciales y condiciones de uso del servicio.

Algunas de las implementaciones iniciales de MapPoint .NET están dentro de los servicios de MSN (por ejemplo, Expedia y HomeAdvisor). Otros en implementar este servicio son Dollar Rent- ACAR y el hotel German con el servicio de reservas Tellme usan MapPoint. NET para proporcionar instrucciones de conducción de voz desde un teléfono móvil.

Una aplicación particularmente innovadora de este servicio Web fue desarrollada por Marks & Spencer, que utiliza MapPoint. NET para combatir el fraude con tarjetas de crédito. Cuando el mismo número de cuenta aparece como si hubiera sido utilizada en varios almacenes en un intervalo de tiempo breve, esto plantea una alerta. El sistema envía las ubicaciones de las tiendas de MapPoint. NET con la solicitud de devolver el tiempo de conducción entre dichas tiendas. Si el tiempo es significativamente mayor que el tiempo real entre las compras, la conclusión es que posiblemente un fraude de tarjetas de crédito se ha producido.

- ESTUDIO DE CASO GOOGLE

El popular motor de búsqueda Google ha publicado una Interfaz SOAP. Los usos que se hacen de esta interfaz por una comunidad de entusiastas anuncian las nuevas reglas de la Era de los servicios Web.

A principios de 2002, Google anunció que proporcionaría acceso a su índice de búsqueda y funciones particulares a través de una interfaz utilizando SOAP. Google ha agregado una variedad de integraciones de usuario a su índice y funciones de búsqueda, a partir de la simple inclusión de una caja de búsqueda de Google en las publicaciones de páginas web para utilizar de manera más sofisticada su motor de búsqueda y algoritmos de relevancia, los desarrolladores han creado:

- Un sistema que permite a los usuarios introducir los contenidos de sus páginas web en un comparador de relevancia de Google, junto con otras funciones analíticas de su propia creación, para revelar otras publicaciones web que comparten sus mismos intereses.
- Un generador de error tipográfico, que permite a los usuarios ingresar una palabra, ver cómo es más a menudo incorrectamente digitada en las búsquedas escritas (Con fines de accesibilidad web).
- Un motor de publicación de páginas web que analiza un el título de la entrada y le incorpora una cadena de búsquedas adecuadas, señaladas por Google.

El entusiasmo con el que los usuarios han incorporado la funcionalidad de búsqueda en estas otras aplicaciones, muestra como la forma de buscar en un sitio web es cada vez más capaz de ser reducida a componentes independientes. Al igual que otras aplicaciones, los motores de búsqueda eran originalmente aspectos de productos unitarios que incluían acceso, lógica del negocio y contenido, así como otros elementos de apoyo, como repositorios y autenticación de usuarios. A medida que esas funciones se han descontinuado y las aplicaciones de búsqueda se han convertido en su propia categoría, los proveedores han modularizado estas aplicaciones aún más³⁹.

³⁹ Ibid.

7. ERP'S MÁS UTILIZADOS

7.1 QUE ES UNA ERP

El termino ERP (Enterprise Resource Planning), se puede definir como “un sistema de planificación de los recursos y de gestión de la información que, de una forma estructurada, satisface la demanda de necesidades de la gestión empresarial”⁴⁰.

“Un sistema ERP es una solución informática integral que permite a las empresas aumentar su productividad, controlar y reducir sus costes y llegar a sus clientes de forma ágil y eficiente, haciendo cada vez más competitiva la empresa en el mercado. El ERP se utiliza como una herramienta de gestión estratégica, que ayuda a la empresa a conseguir la integración de todos sus procesos de trabajo y la mejor optimización de todos los recursos disponibles. La información y los medios para su captación, transformación y expedición, constituyen recursos escasos en las organizaciones por lo que se deben establecer políticas, estrategias, normas y procedimientos para optimizar su uso”⁴¹.

7.2 EL CONCEPTO DEL LOS ERP Y SU EVOLUCIÓN

“Estos sistemas comenzaron a desarrollarse en USA durante la segunda guerra mundial, con el objetivo de apoyar la gestión de los recursos materiales que demandaba el ejército.

Fueron llamados MRPS (Material Requirements Planning Systems), o sistemas de planeación de requerimientos de materiales. En la década de los 60, las compañías manufactureras retomaron la idea de MRPS con el fin de gestionar y racionalizar sus inventarios y planificar el uso de recursos acorde a la demanda real de sus productos, por lo que los MRPS evolucionan a MRP (Manufacturing Resource Planning). En los años 80 la utilización de estos sistemas incluían conceptos como “Just in Time”, manejo de la relación con clientes y proveedores, entre otros, es así como los MRP evolucionan completamente hasta lo que se conoce como MRP II.

⁴⁰ Fuente: los autores.

⁴¹ LUÍS MUÑIZ GONZÁLEZ, LICENCIADO EN CIENCIAS ECONÓMICAS Y EMPRESARIALES, AUDITOR Y CENSOR JURADO DE CUENTAS. Implantación de un ERP: El valor añadido a la empresa. Partida Doble, No. 139, Diciembre 2002, Editorial ESPECIAL DIRECTIVOS. Consulta: 01-11-2013. En: http://revistacontable.dev.nuatt.es/noticias_base/implantaci%C3%B3n-de-un-erp-el-valor-a%C3%B1adido-a-la-empresa

En la década de los 90, producto de la globalización, las empresas comenzaron a requerir de sistemas que apoyaran la gestión empresarial, integraran las partes del negocio, promovieran la eficiencia operativa y sirvieran de soporte aspectos críticos de la administración. Así la industria de software en un comienzo desarrolló aplicaciones para integrar los distintos sistemas MRP I y MRP II, que años más tarde se transformaron en los sistemas empresariales integrados, conocidos actualmente como ERP (Enterprise Resource Planning) o Sistemas de Planeación de Recursos Empresariales.

Los Sistemas del tipo ERP (Enterprise Resource Planning) se han definido como un sistema global de planificación de los recursos y de gestión de la información que de forma estructurada puede satisfacer la demanda de las necesidades de gestión de la empresa (Gallardo L.; González C.; Tapia F) Son paquetes de software de tipo World Class que permiten a las empresas evaluar, implementar, automatizar, integrar y gestionar de forma eficiente las diferentes operaciones que se presentan en éstas. Se puede utilizar por todo tipo de empresas, pero se requiere de una adaptación según sean las circunstancias y tipo de organización que tiene cada una de ellas.

Estos paquetes de software disponen de módulos específicos para cubrir las exigencias de cada una de las áreas funcionales de la empresa, de tal manera que crean un flujo de trabajo⁵ entre los distintos usuarios. Este flujo permite evitar tareas repetitivas, y mejora la comunicación en tiempo real entre todas las áreas que integran la empresa⁴².

7.3 VENTAJAS DE LA UTILIZACIÓN DE ERPS

- a) "Por qué la empresa debe adquirir un ERP: las tendencias comerciales actuales y futuras obligan a las empresas a ser cada vez más competitivas, para ello es necesario que una empresa tenga optimizado e integrado todos sus flujos internos de información y sus relaciones comerciales externas. Además debe conseguir los objetivos estratégicos como son las mejoras de la productividad, la calidad, el servicio al cliente y la reducción de costes. Las tecnologías de la información han permitido, en gran medida, la consecución de dichos objetivos. En este apartado, podemos reseñar la aportación de los ERP y las ventajas del comercio electrónico o intercambio electrónico de información con clientes y proveedores a través de proyectos EDI, proyectos basados en aplicaciones WEB, etc.

⁴² Benvenuto Vera, Ángelo. Universidad de Concepción. Implementación de Sistemas ERP, su impacto en la gestión de la empresa e integración con otras TIC/ El concepto de los ERP y su evolución. *CAPIV REVIEW Vol. 4 2006 ISSN 0718-4654 Versión impresa / ISSN 0718-4662 Versión en línea.* Consulta: 01-11-2013. En: <http://www.capic.cl/capic/media/ART3Benvenuto.pdf>

b) **Ventajas obtenidas al implantar un ERP:** es necesario conocer una serie de ventajas que la implantación del ERP aporta en mayor o menor amplitud a la empresa, a continuación se detallan algunas de ellas:

- La empresa dispone de una solución integrada para que las diversas funciones y departamentos trabajen de una forma más eficiente.
- La empresa consigue una garantía de actualización continua y más inmediata de la aplicación a las necesidades del negocio y a la reducción de los costes fijos.
- El máximo beneficio de la implantación de un ERP solamente se obtiene cuando todas las aplicaciones de la empresa están totalmente integradas, incluido sus proyectos de comercio electrónico, y cuando esta integración se realiza en el menor plazo de tiempo posible.
- Un ERP se debe componer de un determinado número de módulos, todos ellos adaptables a las necesidades de cada empresa y escalables (que se puedan ir adquiriendo poco a poco en función de las necesidades).
- Los sistemas ERP permiten utilizar y compartir las bases de datos, además ofrecen un tipo de interfaz para el usuario y tiene un cierto nivel de adaptación a las necesidades y particularidades de cada empresa.
- La mayoría de sistemas ERP existentes en el mercado pueden trabajar como aplicaciones para ser utilizadas en entornos *cliente/servidor*.
- La capacidad de obtener información, procesarla y transformarla de forma inmediata, agiliza todos los procesos de trabajo⁴³.

7.4 EJEMPLOS DE ERP

A continuación se describen algunos de los ERP más conocidos y usados en nuestro país, además se incluye una breve descripción de las opciones de comunicación de datos que poseen, incluidos claro está los servicios web.

7.4.1 Biosalc44: esta aplicación se originó en los inicios de la década de 1990 y hoy acumula más de 15 años de experiencia y especialización en sistemas para la agroindustria. Atiende más de 150 clientes, destacándose entre ellos, los mayores

⁴³ *Ibíd.*

⁴⁴ La Tecnología de la información aplicada al Agronegocio. Consulta: 01-11-2013. En: <http://www.biosalc.com.co/biosalc-colombia.html>

grupos sucroalcoholeros del Brasil y también grandes productores de semillas, frutas, hortalizas, palma africana, eucalipto, entre otros.

Las diferentes soluciones ERP que BIOSALC ofrece a sus clientes son:

- Agrícola
- Gerenciamiento Industrial
- Mantenimiento de Flota
- Corporativo ERP
- Medio Ambiente
- Planificación Presupuestal
- Mantenimiento Industrial
- Pecuaria

Todas las soluciones anteriores comparten las siguientes características principales:

- Elaboración ágil de presupuestos agrícolas.
- Control detallado de los costos por labor, por lote, por bloque, por tonelada Cosechada, etc.
- Acompañamiento en Tiempo Real de la planificado vs lo realizado por unidades y dinero.
- Acompañamiento de indicadores usando un cockpit (BI), en cada sistema.
- Conocimiento de las eficiencias de la maquinaria.
- Conocimiento de la productividad de los trabajadores según la labor que realizan.
- Evaluación de la calidad con la que se hacen las labores.
- Algoritmos de programación de labores configurable según el cultivo.
- Bloqueos automáticos de labores según presupuestos.
- Sistema de pesaje con contingencia.
- Basculas desatendidas (sin operarios).
- Controles Fitosanitarios avanzados con alertas automáticos.
- Digitación de apuntes utilizando tecnologías móviles.
- Acompañamiento de la actividad agrícola mediante de mapas.
- Generador de reportes, tablas dinámicas e indicadores que permite total independencia del usuario.
- Navegación intuitiva con pantallas definidas según el perfil del usuario.
- Auditoria y rastreabilidad total de cambios realizados en el sistema por los usuarios.
- Integración on-line con el sistema ERP usando tecnologías avanzadas (webservice, XML, entre otros).

7.4.1.1 Formas de comunicación con otras ERP: todas las soluciones que la empresa Biosalc brinda, poseen la capacidad de comunicarse con otros sistemas ERP principalmente por medio de tres sistemas básicos:

- Archivos planos
- Procedimientos remotos
- Servicios web

7.4.2 MAPINFO⁴⁵: MapInfo® Profesional, es una aplicación de Windows para análisis cartográfico y geográfico el cual permite visualizar fácilmente la relación entre los datos y la geografía. Actualmente es propiedad de Pitney Boews, quien asegura que es el mejor servicio de información geográfica de la actualidad, la última versión de MapInfo permite crear, compartir y usar mapas de una manera rápida y sencilla.

Algunas de sus características más importantes son:

- **Fácil de usar e integrar:** MapInfo es una herramienta cartográfica extremadamente flexible, ya que se integra directamente con diversos sistemas de IT, además no se necesita ser un experto de IT para poder usarla.
- **Datos fácilmente accesibles:** MapInfo profesional soporta múltiples formatos de archivo, así que permite visualizar datos traídos directamente desde Microsoft Excel, Microsoft Access, Oracle, Microsoft SQL Server y más.
- Opciones flexibles de despliegue de mapas
 - Territorios
 - Vías
 - Puntos
 - Entre otros.
- **Capacidad para compartir datos de manera efectiva:** MapInfo posee múltiples opciones para publicación de información, es posible hacer todo desde exportar sus datos a cualquier formato hasta publicar enormes mapas con descripciones y gráficas. Asistentes fáciles de usar los cuales permiten publicar mapas estáticos e interactivos sin ninguna complicación, así que se puede compartir sus resultados con quien lo necesite.

⁴⁵ MAPINFO. [Disponible en: <http://www.pb.com/software/Data-Mining-and-Modeling/Geographic-Data-Mining-Tools/MapInfo-Professional.shtml>]

7.4.3 SAP⁴⁶: SAP América Latina y Caribe inició sus operaciones en 1994 con el objetivo de ofrecer a sus clientes eficiencia y flexibilidad a través de aplicaciones enfocadas en el crecimiento rentable del negocio. Con más de 12,000 clientes y un ecosistema de 450 socios de negocios en la región, permite a las empresas de todos los tamaños y de cualquier sector de industria ser más competitivas y rentables, hacer más eficientes sus procesos, potenciar el resultado de sus negocios y reducir costos a través de la innovación tecnológica.

SAP ayuda a que las empresas de todos los tamaños se administren mejor. Desde las tareas administrativas hasta la mesa directiva, desde el almacén hasta la atención al cliente, desde los equipos de escritorio hasta los dispositivos portátiles, SAP potencia a las personas y a las organizaciones para trabajar juntos con mayor eficiencia y usar el conocimiento de negocios con mayor eficacia para estar siempre un paso adelante de la competencia. Lo logramos ampliando la disponibilidad de las instalaciones de software en las instalaciones, los despliegues on-demand y los dispositivos móviles.

7.4.3.1 Tipos de interfaces de comunicación:

- Posibilidad de importar y exportar archivos planos (txt).
- SAP .Net connector.
- Servicios web.

7.4.4 Cliquida: “la solución para planeación, control y liquidación de tiempos laborados consiste en un dispositivo biométrico que registra la entrada y salida de los trabajadores a la instalación donde laboran, este se integra con nuestro software “CLiquida” el cual basado en dichas marcaciones realiza la LIQUIDACIÓN DE TIEMPOS (horas y minutos laborados), discriminando los diferentes conceptos (horas extra y recargos) que define la legislación laboral. Además, el sistema es una útil herramienta que facilita la PLANEACION DE TURNOS y la administración del recurso humano (indicadores de AUSENTISMO Y CUMPLIMIENTO, horas extras no remuneradas, PROYECCIÓN DE NÓMINA, entre otros)”⁴⁷.

CLiquida es la solución para la planeación, control y liquidación de tiempos laborados que le permitirá ahorrar dinero mediante la optimización y el cálculo exacto y eficiente de las horas trabajadas por sus empleados, la proyección de sus costos de nómina y la asignación de metas (presupuestos) a las diferentes áreas de su empresa.

⁴⁶ NUESTRA EMPRESA, SAP. [En: <http://www.sap.com/latinamerica/about-sap/our-company.epx>]

⁴⁷ CLiquida es la solución para la planeación, control y liquidación de tiempos laborados. [En: <http://www.stt-solutions.com/producto-detalle.php?idProduct=3>]

Los conceptos liquidados por el sistema son:

- Horas Ordinarias
- Horas extra diurnas
- Horas extra nocturnas
- Horas extra diurnas festivas
- Horas extra nocturnas festivas
- Recargo nocturno
- Recargo dominical
- Recargo nocturno dominical
- Recargo festivo
- Recargo nocturno festivo
- Horas de descuento

Como un valor agregado, el sistema también presenta información que facilita la adecuada administración del recurso humano, como indicadores de ausentismo, cumplimiento y de distribución de las horas extras, permisos y los recargos entre un grupo de empleados. La manera como esta ERP se integra con otras aplicaciones es por medio de archivos planos (txt).

Finalmente este sistema puede ser utilizado de acuerdo a las necesidades de la empresa en forma monousuario y multiusuario.

- Monousuario: La aplicación CLiquidada, versión monousuario está orientada a que una sola persona se encargue de la programación y liquidación de la nómina.
- Multiusuario: La aplicación CLiquidada versión multiusuario está orientada a un esquema de trabajo colaborativo, en el que diferentes jefes de área, ejecutan la programación y el control de los empleados a su cargo y una sola persona valida la liquidación del tiempo laborado

7.5 CONCLUSIONES ERP's

En conclusión se puede decir que el objetivo principal de una ERP es el de organizar la información de tal manera que se convierta en un recurso útil para la entidad que lo usa, facilitando la toma de decisiones y permitiendo que el tiempo y los recursos sean utilizados de una manera más eficiente. Como se puede notar en la descripción hecha en cada uno de los sistemas ERP más comunes en nuestro país, dichos sistemas cuentan con interfaces que se relacionan con servicios web, lo cual demuestra la importancia que tiene la existencia de un conector o punto de encuentro entre ellos, que permita a los desarrolladores desconocer los lenguajes en que ha sido desarrollado, para simplemente enfocarse en enviar y recibir datos del servicios web, que es lo que realmente les interesa al momento de crear una interface de comunicación.

7.5.1 Comparación de las interfaces de comunicación de los ERP's estudiados: Para evaluar y determinar el rendimiento de un sistema ERP es importante conocer las interfaces de comunicación que posee, debido a que mediante estas es posible saber su grado de flexibilidad. Por lo general las empresas poseen varios sistemas ERP, que interoperan entre ellos compartiendo información por medio de alguno de los métodos mencionados a continuación.

Tabla 1. Comparación de las interfaces de comunicación de los ERP's estudiados

ERP	Servicios Web	Archivos Planos	Procedimientos Remotos	Archivos CSV, XLS, etc.
BIOSALC	Servicios web de integración	Archivos delimitados y de ancho fijo	Llamadas a procedimiento remoto RPC	Importa y exporta archivos de Office y separados por comas
MAPINFO				Importa y exporta archivos de Office y separados por comas
SAP	Servicios web de integración	Archivos de ancho fijo	Llamadas a procedimiento remoto RPC	
CLIQUIDA		Archivos delimitados y de ancho fijo		Importa y exporta archivos de Office y separados por comas

Fuente: Los autores.

En la tabla 1 se resume la manera como cada una de las ERP mencionadas, se comunican con otros sistemas computacionales.

8. LA BASE DE DATOS ORACLE, LA UTILIDAD DEL PROTOCOLO HTTP Y EL USO DE XML

Oracle se define como una herramienta potente basado en la tecnología cliente/servidor para la gestión de bases de datos, desarrollado por Oracle Corporation, se considera a Oracle como una de las bases de datos más completas del mundo, este producto por ser tan potente y de tan alto precio hace que sea utilizado mayormente en multinacionales y/o empresas muy grandes⁴⁸. Oracle es considerado como uno de los sistemas de bases de datos más completos.

Actualmente Oracle es una de las compañías de software empresarial más grande del mundo, contando con más de 300.000 clientes, entre los cuales 98 de las empresas que figuran en la lista Fortune 100, en más de 145 países, también se encuentra General Motors (EE.UU), General Electric (EE.UU), Intel Corporation (EE.UU), HP (Bélgica, Inglaterra y EE.UU.), Phillips (Bélgica, Holanda y EE.UU.), Yahoo (EE.UU.), Alcatel (Francia), British Gas (Inglaterra), cuenta también con 280.000 clientes de bases de datos Oracle, 80.000 clientes de Oracle Fusion Middleware, 43.000 clientes de Aplicaciones Oracle, 20.000 socios de negocios en todo el mundo, 84.000 empleados en todo el mundo, 20.000 desarrolladores, 7.500 profesionales de soporte y 8.000 expertos en servicios de consultoría. El negocio de Oracle actualmente es la información, ya que sabe cómo administrarla, compartirla y protegerla, este permite a las empresas tener una información más actualizada y precisa de sus sistemas comerciales⁴⁹.

El gran defecto de Oracle es el alto costo monetario según sus licencias y versiones, esta desventaja está provocando que sufra la competencia de Microsoft SQL Server de Microsoft y la oferta de otros sistemas de gestión de bases de datos relaciones como es PostgreSQL, MySql o Firebird ya que estas RDMS tienen licencia gratuita, y por obvias razones las personas y empresas, prefieren utilizar estas que son con de uso libre, aunque cabe decir que este tipo de bases de datos no son de gran utilidad cuando se trata de almacenar demasiada información.

Antes Oracle además de su costo, sufría con el defecto de no poder trabajar bajo GNU/Linux, hoy en día esta desventaja se ha convertido en una ventaja, porque las últimas versiones de Oracle se han certificado para este tipo de trabajo.

⁴⁸ MASIP, David. ¿Qué es Oracle. En desarrolloweb.com edición 19 de julio de 2002. Consulta 23 de febrero de 2013. [En: <http://www.desarrolloweb.com/articulos/840.php>]

⁴⁹ Oracle Global Fact Sheet. Por: Latina, Dirección Oracle América. Oracle Hardware and software, Engineered to work together. Edición 11 de agosto de 2008. Consulta: 23 de febrero de 2013. [En: <http://www.oracle.com/lad/corporate/press/oracle-fact-sheet-aug-08-322055-esa.pdf>]

Desde un principio Oracle empezó ofreciendo como primer servicio la gestión de base de datos, las bases de datos Oracle es una colección de datos tratada como una unidad. El propósito general es almacenar y recuperar información relacionada. ,ofreciendo a sus clientes confiabilidad, alto rendimiento en transacciones, seguridad entre otros, pero como todo, tiene algunos errores o desventajas que se han ido mejorando poco a poco a través del tiempo, esto se ha logrado con cada versión que ha ido sacando la compañía, hoy en día la versión más actual es Oracle 11g, esta es de gran confiabilidad para los profesionales de las tecnologías de la información ya que estos deben administrar más información con mayor calidad de servicio y de manera oportuna a sus usuarios⁵⁰.

Hoy en día las tecnologías de la información deben brindar exitosamente más información con mayor calidad de servicio y a la vez con menos riesgo de cambio dentro de las tecnologías de la información, estos requerimientos que se pide hoy en día, son los servicios que Oracle Database 11g ofrece a sus usuarios.

Los usuarios que desean utilizar Oracle Database 11g como la solución a muchos de los problemas que tras la gestión de la información, puede utilizar todo el poder de la base de datos líder del mundo para:

- Reducir los costos del servidor en un factor de 5
- Reducir los requisitos de almacenamiento en un factor de 12
- Mejorar el desempeño de los sistemas de misión crítica en un factor de 10
- Duplicar la productividad de los DBA
- Eliminar las redundancias del centro de datos
- Simplificar la cartera de software de TI

8.1 ORACLE Y LA WEB, UTL_HTTP

El paquete UTL_HTTP hace llamadas HTTP desde SQL y PL/SQL. Puede ser utilizado para acceder a datos en Internet a través de HTTP.

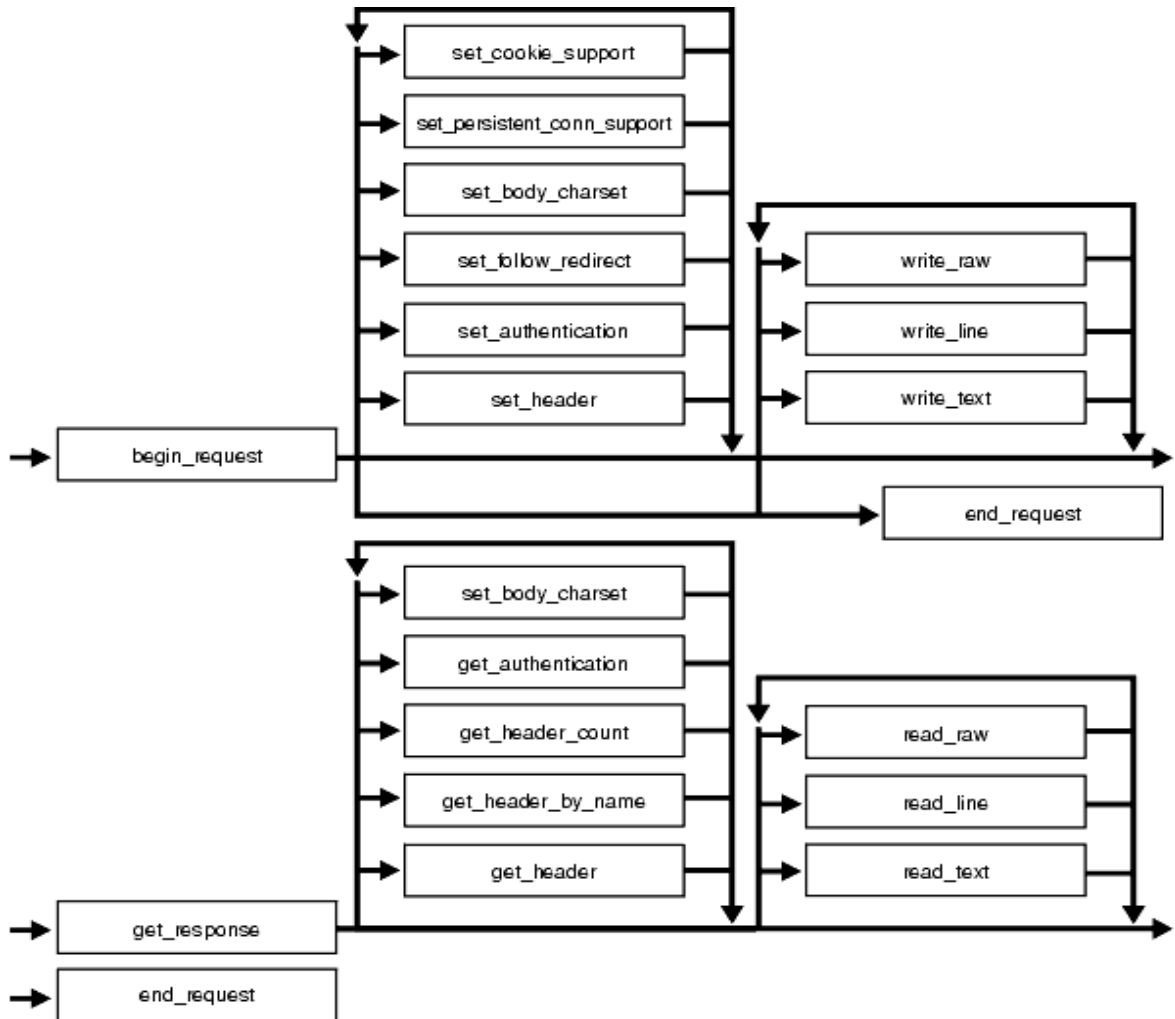
Con UTL_HTTP, es posible escribir programas PL/SQL que se comunican con servidores Web (HTTP). UTL_HTTP contiene una función que se puede utilizar en consultas SQL. Este paquete soporta HTTP sobre el protocolo Secured Socket Layer (SSL), también conocido como HTTPS, directamente o a través de un proxy HTTP. Otros protocolos relacionados con Internet de acceso a datos (como el File Transfer Protocol (FTP) o el protocolo Gopher) también son soportados utilizando un servidor proxy HTTP.⁵¹

⁵⁰ Ibid.

⁵¹ Docsoracle.com.Oracle documentación. [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258u_http.htm número estándarb14258-02].

La siguiente imagen muestra el flujo se sigue UTL_HTTP al momento de hacer una petición:

Imagen 10. Flujo del paquete UTL_HTTP⁵²



Fuente: Flujo del paquete UTL_HTTP. Docsoracle.com.Oracle documentación. [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258u_http.htm número estándar b14258-02].

8.2 LISTAS DE CONTROL DE ACCESO (ACL)

Debido a la inseguridad en las redes, los administradores deben contar con herramientas para denegar el acceso no autorizado de las aplicaciones a la red, la base de datos Oracle cuenta con una herramienta de protección llamada ACL (Acces Control List). A partir de la versión número 11 la base de datos Oracle

⁵² Ibíd.

posee una lista secuencial de sentencias para denegar o permitir el acceso a los archivos, directorios y recursos de capa superior, es decir, estas listas se usan para gestionar la seguridad del sistema, además se pueden otorgar permisos a un usuario, un grupo, un equipo o una unidad administrativa en concreto ya sea localmente o a través de la red⁵³.

La implementación de ACL por defecto está basada en la base de datos, estas son bastantes útiles para proporcionar un mayor control en el acceso a archivos. Estas se pueden usar cuando las protecciones de archivos UNIX tradicionales no son suficientes, puesto que estas protecciones proporcionan permisos de lectura, escritura y ejecución para el propietario, grupo y otros usuarios. Por tanto, en este caso una ACL da un nivel más específico de seguridad de archivos⁵⁴.

Las ACL permiten definir los siguientes permisos de archivo:

- Permisos de propietario de archivo.
- Permisos de archivo para el grupo del propietario.
- Permisos de archivo para otros usuarios que están fuera del grupo del propietario.
- Permisos de archivo para usuarios específicos.
- Permisos de archivo para grupos específicos.
- Permisos predeterminados para cada una de las categorías anteriores.

En un sistema de archivos UFS una ACL da una mayor seguridad para los archivos, ya que permite definir permisos de archivo para el propietario del archivo, el grupo, otros y usuarios y grupos específicos y definir permisos predeterminados para cada una de las categorías anteriores⁵⁵.

8.3 ORACLE Y EL USO DE XML

El XMLType es suministrado por Oracle y es utilizado para almacenar y consultar datos XML en la base de datos. XMLType tiene funciones propias que puede utilizar para acceder, extraer y consultar los datos XML mediante expresiones XPath. XPath es otro estándar desarrollado por el comité de W3C para atravesar los documentos XML. Las funciones XMLType de Oracle soportan muchas expresiones XPath del W3C. Oracle también proporciona un conjunto de funciones de SQL y paquetes PL / SQL para crear valores de tipo XML de datos relacionales u objeto-relacional.

XMLType es un sistema de tipo definido, por lo que se puede utilizar como un argumento de una función o el tipo de datos de una tabla o una columna de vista.

⁵³ PIQUERO, Julián Verón. Listas de control de acceso (ACL). Prácticas de redes, 2009.

⁵⁴ Docs.oracle.com Oracle Documentation. Consulta: 23 de febrero de 2013. [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm Numero Standar B14258-2.]

⁵⁵ Ibíd.

También puede crear tablas y vistas de XMLType. Cuando se crea una columna XMLType en una tabla, puede optar por almacenar los datos XML en una columna CLOB o un objeto relacional.

También puede registrar el esquema (usando el paquete DBMS_XMLSCHEMA) y crear una tabla o columna que se ajuste al esquema registrado.

- XMLType se puede utilizar en PL / SQL los procedimientos almacenados como parámetros, valores de retorno y variables.
- XMLType puede representar un documento XML como un ejemplo (de XMLType) en SQL.
- XMLType tiene funciones propias que operan sobre contenido XML. Por ejemplo, puede utilizar las funciones de tipo XML para crear, extraer e indexar datos XML almacenados en la base de datos Oracle
- La funcionalidad también está disponible a través de un conjunto de interfaces de programación de aplicaciones (API) que figuran con PL / SQL y Java.

Con XMLType y las capacidades de estos, los desarrolladores de SQL pueden aprovechar la potencia de la base de datos relacional mientras se trabaja en el contexto de XML. Del mismo modo, los desarrolladores de XML pueden aprovechar la potencia de los estándares XML mientras se trabaja en el contexto de una base de datos relacional.

Los tipos de datos XMLType se puede utilizar como el tipo de datos de columnas en las tablas y vistas. Las variables de XMLType se puede utilizar en PL / SQL los procedimientos almacenados como parámetros, valores de retorno, y así sucesivamente. También puede utilizar XMLType en SQL, PL / SQL y Java (mediante JDBC).⁵⁶

8.4 CLOB, GUARDANDO XML EN LA BASE DE DATOS

Un CLOB es un tipo de campo de Oracle más grande que el VARCHAR2, estos son equivalentes a los campos de tipo TEXT en Postgres o los de tipo MEMO en MYSQL, es decir, es el tipo de campo donde se puede almacenar mucha información de tipo texto, tanto así, que una columna CLOB puede tener hasta (4 gigabytes - 1) * (tamaño del bloque de la base de datos) de información. Si los tablespaces en la base de datos tienen un tamaño de bloque estándar, y si se ha

⁵⁶ Ibíd. [En: http://docs.oracle.com/cd/B10500_01/appdev.920/a96620/xd04cre.htm]

utilizado el valor predeterminado del parámetro CHUNK de almacenamiento cuando se crea una columna LOB, entonces esto es equivalente a (4 gigabytes - 1) * (tamaño del bloque de la base de datos) .

Debido a que CLOB es un objeto se hace un poco complicado obtener el valor de un campo en un SELECT, para eso se usa la librería DBMS_LOB.⁵⁷ Es recomendable usar CLOB o BLOB en lugar de LONG, ya que en general los campos LONG de Oracle tienen limitaciones en Oracle, por lo cual se decide reemplazarlos por los CLOB,⁵⁸ los objetos CLOB tienen soporte transaccional completo. Los cambios realizados a través de SQL, el paquete DBMS_LOB, o el Oracle Call Interface (OCI) participan plenamente en la transacción. La manipulación de los valores CLOB se pueden hacer y deshacer. Sin embargo, no se puede guardar un localizador de CLOB en una variable PL/SQL o OCI en una transacción y luego usarlo en otra transacción o sesión.⁵⁹

8.5 DBMS_XMLGEN

El paquete DBMS_XMLGEN convierte los resultados de una consulta SQL a un formato XML canónico. El paquete tiene una consulta SQL arbitrario como entrada, lo convierte a formato XML y devuelve el resultado como un CLOB. Este paquete es similar al paquete DBMS_XMLQUERY, excepto en que está escrito en C y compilado en el núcleo. Este paquete sólo se puede ejecutar en la base de datos. Un resumen de las funciones que tiene el DBMS_XMLGEN son las siguientes⁶⁰:

⁵⁷ Basado en: <http://www.acatos.es/2009/04/08/obtener-valores-de-un-campo-clob-en-oracle/>

⁵⁸ Release Notes de Generador Visual Basic 8.0. En: <http://ftpusa.artech.com.uy/files/1379282663/vb80u5sp.pdf>

⁵⁹ Docs.oracle.com Oracle Documentation. Consulta: 23 de febrero de 2013. [En: http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements001.htm]

⁶⁰ Docs.oracle.com Oracle Documentation. Consulta 23 de febrero de 2013. [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.html Número Estándar B14258-2]

Tabla 2. Los resultados de una consulta SQL a un formato XML canónico

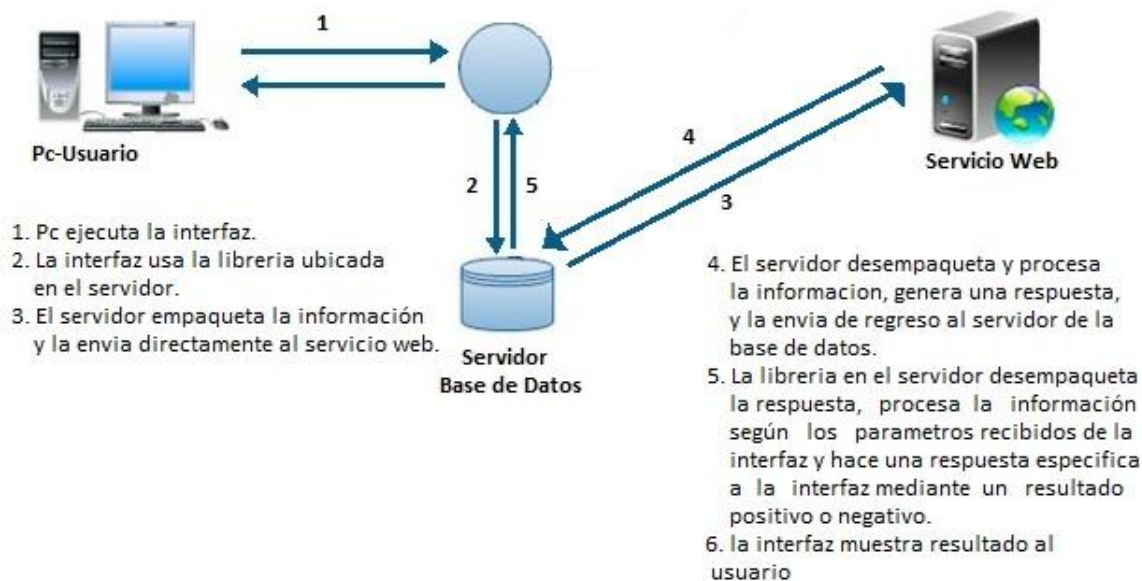
FUNCIÓN	DESCRIPCIÓN
CLOSECONTEXT Procedure	Cierra el contexto y libera todos los recursos
CONVERT Functions	Convierte el XML en su equivalente escapado. Ejemplo: “<nodo>” a “<node>”
GETNUMROWSPROCESSED Function	Obtiene el número de filas de SQL que se trataron en la última llamada a las funciones GetXml
GETXML Functions	Obtiene el documento XML
GETXMLTYPE Functions	Obtiene el documento XML y lo devuelve como XMLType
NEWCONTEXT Functions	Crea un nuevo identificador de contexto
RESTARTQUERY Procedure	Reinicia la consulta para iniciar la captación desde el comienzo
SETCONVERTSPECIALCHARS Procedure	Determina si los caracteres especiales, como \$, que son caracteres que no son XML, deben ser convertidos o no a su representación de escape
SETMAXROWS Procedure	Establece el número máximo de filas a consultar
SETNULLHANDLING Procedure	Define como NULL las opciones de manejo
SETROWSETTAG Procedure	Configura el nombre del elemento adjuntando el resultado completo
SETROWTAG Procedure	Configura el nombre del elemento que encierra cada fila del resultado
SETSKIPROWS Procedure	Establece el número de filas para ignorar antes de generar el XML.
USENULLATTRIBUTEINDICATOR Procedure	Especifica cuando utilizar un atributo XML para indicar un valor NULL, o ignorando la inclusión en la entidad del documento XML

Fuente: Los resultados de una consulta SQL a un formato XML canónico. Disponible en: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_xmlgen.htm

9. INTERFAZ DE COMUNICACIÓN EN PL/SQL PARA COMUNICAR ERP's A TRAVÉS DE SERVICIOS WEB

El nuevo modelo para el consumo de servicios web desde las interfaces desarrolladas por las distintas ERP quedaría de la siguiente manera:

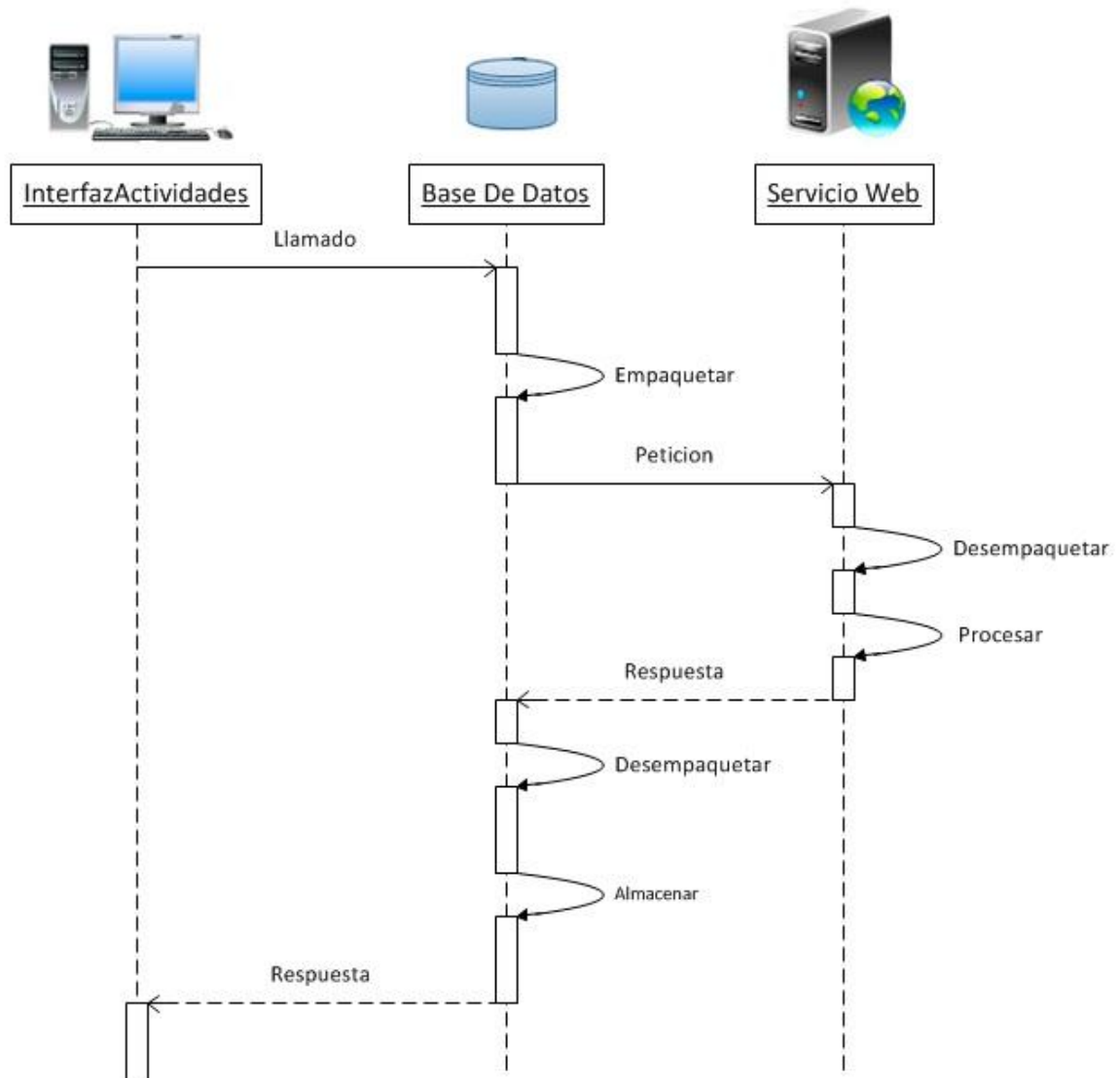
Imagen 11. Modelo propuesto de comunicación entre ERP's



Fuente: los autores.

En la imagen anterior se puede observar que mediante el nuevo modelo de comunicación no se hace necesario enviar toda la información a la interfaz alojada en el equipo del usuario, sino únicamente la respuesta positiva indicando que el procedimiento se ha llevado a cabo satisfactoriamente o negativa indicando que ha ocurrido un error en el proceso.

Diagrama 2. Diagrama de secuencia del nuevo modelo de comunicación entre ERP's.



Mediante este diagrama se logra observar con mayor facilidad, el traslado de la carga computacional que va dirigida al servidor de la base de datos, el cual se encarga de desempaquetar y procesar toda la información que entrega el servicio web, para luego entregar una respuesta concreta al usuario, logrando así un incremento en el rendimiento del sistema ya que:

- Se traslada la carga computacional al servidor.
- Aumenta la disponibilidad de recursos hardware del equipo cliente.
- Solo se envía información una vez hacia y desde el servicio web.

- Aumenta la velocidad de respuesta ya que el servidor de la base de datos empaqueta y desempaqueta la información más rápido que el equipo del cliente.
- Al estar instalado el paquete en la base de datos, éste tiene acceso directo a la información bien sea para consultarla o registrarla, lo que genera mayor velocidad de la respuesta.

En cuanto a las ventajas para los desarrolladores se encuentra:

- Independencia del lenguaje, debido a que todos los sistemas ERP poseen rutinas simples para conexión a base de datos, el desarrollador solo debe usar dichas rutinas para invocar la librería con los parámetros y configuraciones que desee.
- No se requiere ser un experto en el tema, solo se debe entender el modelo de comunicación y los datos que se envían y reciben. Para un ejemplo práctico de consumo de un servicio web que realice una suma, el desarrollador solo debe conocer:
 - La dirección URL del servicio web.
 - Los parámetros que envía (numero1, numero2).
 - Los valores que retorna (total).

Con estos datos es suficiente para que el desarrollador use la librería en su nivel más básico.

- La memoria requerida por la aplicación es mucho menor, ya que no debe almacenar ninguna de las estructuras de datos relacionadas con el servicio web.

9.1 QUE ES UNA LIBRERÍA EN PL/SQL

Básicamente cuando se habla de una librería PL/SQL, se hace referencia a un conjunto de funciones y procedimientos agrupados en un paquete⁶¹ PL/SQL. Este paquete será el encargado de contener todas las operaciones y funcionalidades que la librería pondrá a disponibilidad de los desarrolladores.

9.2 QUÉ DEBE PERMITIR LA LIBRERÍA

La librería debe permitir a los desarrolladores realizar las siguientes tareas relacionadas con el consumo de servicios web:

⁶¹ Un paquete es una estructura que agrupa objetos de PL/SQL compilados (procedures, funciones, variables, tipos...) en la base de datos. Esto nos permite agrupar la funcionalidad de los procesos en programas. [En: <http://www.devjoker.com/contenidos/articulos/55/Paquetes-en-PLSQL.aspx>]

- Establecer la conexión con el servidor remoto.
- Extraer la información desde la base de datos en una manera adecuada para su envío al servicio web.
- Crear el paquete SOAP a enviar al servicio.
- Extraer la información que llega en la respuesta desde el servicio web.

9.2.1 Conexión con el servidor remoto: esta operación es realizada de manera implícita por la librería, solo requiere la dirección URL del servicio web. Adicionalmente existen algunos datos adicionales que el desarrollador puede especificar para agregar cabeceras a la petición realizada por la librería.

9.2.2 Métodos de extracción de la información: la librería cuenta con procedimientos almacenados que se encargan de la extracción de la información en formato XML listo para ser enviado a los servicios web, estos métodos permiten realizar extracciones de dos tipos:

- **Sentencias SQL:** a partir de una sentencia SQL la librería genera las etiquetas XML a partir de los alias o nombres de las columnas en el resultado de dichas sentencias.
- **Especificando la tabla de origen:** a partir de la tabla especificada la librería consulta las columnas disponibles para dicha tabla y genera la información en formato XML según el orden y los nombres de las columnas detectadas.

9.2.3 Formas de creación de paquetes SOAP: los métodos de creación de paquetes SOAP para enviar al servicio web son muy flexibles ya que permiten al desarrollador enviar el XML generado en el proceso de extracción o enviar su propio XML preparado con anterioridad. Esta característica puede ser utilizada por desarrolladores con conocimientos de PL/SQL. Los procedimientos de creación de paquetes SOAP requieren especificar algunas cabeceras requeridas por los servicios web, por ejemplo el método a ejecutar en el servicio web.

9.2.4 Lectura de resultados del servicio web: se espera que el servicio web retorne información en formato XML la cual es extraída según los parámetros especificados por el desarrollador. La librería permite especificar de qué nodo o nodos XML extraer dichos datos, además permite realizar extracciones directo a una tabla donde cada nodo en la respuesta será almacenado en la tabla según el orden de los campos que esta posea.

9.3 CÓMO LOGRARLO

9.3.1 Realizar peticiones HTTP desde Oracle: como ya se mencionó anteriormente en este escrito la base de datos Oracle cuenta con la capacidad de realizar peticiones web a través del protocolo HTTP por medio del paquete UTL_HTTP utilizando el lenguaje PLSQL, en este capítulo se estudiará este paquete desde un punto de vista más técnico, siempre enfocándose principalmente en las características que sean relevantes para el propósito de este trabajo de grado.

9.3.1.1 UTL_HTTP, los tipos de datos a utilizar: básicamente para realizar una petición HTTP exitosa se requieren dos tipos de dato principales, uno para la petición y otro para la respuesta, UTL_HTTP.req y UTL_HTTP.resp respectivamente.

9.3.1.2 UTL_HTTP.req la petición HTTP en PLSQL62: este es un objeto del tipo RECORD de PL/SQL y es utilizado para representar una petición HTTP, está conformado por tres tipos de datos carácter "VARCHAR2" y su sintaxis es:

Imagen 12. Estructura del Objeto UTL_HTTP.req⁶³

```
TYPE req IS RECORD (  
    url          VARCHAR2(32767),  
    method       VARCHAR2(64),  
    http_version VARCHAR2(64));
```

Fuente: Estructura del Objeto UTL_HTTP.req Disponible en:
http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#i1012681

- url: la dirección URL de la petición HTTP.
- method: el método a ser utilizado para obtener el recurso identificado por la URL (ej. GET o POST).
- http_version: versión del protocolo utilizado para enviar la petición.

Existen algunos atributos adicionales cuyo nombre comienza con el prefijo "private_" estos campos son privados y han sido definidos únicamente para la implementación del paquete UTL_HTTP y no se deben modificar.

⁶² Oracle® Database PL/SQL Packages and Types Reference. 10g Release 2 (10.2). B14258-02. REQ Type. Consulta 12-11-2013. [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#i1012713]

⁶³ Ibíd.

Cada uno de los valores anteriores se establece después de que la petición es creada por el método BEGIN_REQUEST. (Que hace parte también del paquete UTL_HTTP)

El objeto UTL_HTTP.req que se crea después de hacer el llamado a BEGIN_REQUEST es de solo lectura así que modificar sus valores no tiene ningún efecto.

9.3.1.3 UTL_HTTP.resp la respuesta HTTP en PLSQL64: con los objetos del tipo UTL_HTTP.req no es suficiente para enviar realizar una petición HTTP exitosa, es necesario conocer la respuesta del servidor ya que puede ser de muchos tipos: no encontrado 404, prohibido 403, error interno 505, etc. Una vez creada la petición es necesario tener un lugar en donde almacenar los datos enviados desde el servidor, para esto se utiliza el objeto UTL_HTTP.resp.

UTL_HTTP.resp es utilizado para representar una respuesta HTTP y está conformado por dos campos del tipo carácter "VARCHAR2" y un entero con signo "PLS_INTEGER"⁶⁵.

Imagen 13. Estructura del Objeto UTL_HTTP.resp⁶⁶

```
TYPE resp IS RECORD (  
    status_code      PLS_INTEGER,  
    reason_phrase    VARCHAR2(256),  
    http_version     VARCHAR2(64));
```

Fuente: Estructura del Objeto UTL_HTTP.resp. Disponible en:
http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#i1012713

- status_code: es el código de estado retornado por el servidor. Es un entero de 3 dígitos que indica los resultados de la petición HTTP tal y como lo indica el servidor.
- reason_phrase: es el mensaje textual que envía el servidor para describir el código de estado. Da una pequeña descripción de los resultados de la petición HTTP.
- http_version: versión del protocolo utilizado en la respuesta HTTP.

⁶⁴ Docs.oracle.com Oracle Documentation. Consulta: 23 de febrero de 2013. [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#i1012713]

⁶⁵ PLS_INTEFER – Oracle FAQ. [Edición: 15-09-2011]. Consulta: noviembre 2013. [En: http://www.oraFAQ.com/wiki/PLS_INTEGER]

⁶⁶ Duplicate Topic IDs – Oracle. Ibíd [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#i1012713]

Existen algunos atributos adicionales cuyo nombre comienza con el prefijo “private_” estos campos son privados y han sido definidos únicamente para la implementación del paquete UTL_HTTP y no se deben modificar.

Cada uno de los valores anteriores se establece después de que la respuesta es creada por el método GET_RESPONSE. (Que hace parte también del paquete UTL_HTTP).

El objeto UTL_HTTP.resp que se crea después de hacer el llamado a GET_RESPONSE es de solo lectura así que modificar sus valores no tiene ningún efecto.

9.3.1.4 Leyendo los datos desde la respuesta HTTP: para leer el contenido de la respuesta HTTP, es decir el cuerpo de la respuesta (body) se hace necesario el uso del procedimiento READ_LINE⁶⁷. “Este procedimiento lee el cuerpo de la respuesta HTTP en modo texto hasta que llega al final de cada línea y almacena la información leída en la variable especificada al momento de realizar el llamado al procedimiento. Cuando se llega al final del contenido de la respuesta HTTP se dispara la excepción “end_of_body”, además este procedimiento se encarga de convertir automáticamente el texto leído a la codificación de caracteres configurada en la base de datos.

La sintaxis del procedimiento READ_LNE es la siguiente:

Imagen 14. Sintaxis del Procedimiento READ_LINE”.

```
UTL_HTTP.READ_LINE (  
    r          IN OUT NOCOPY resp,  
    data       OUT NOCOPY VARCHAR2 CHARACTER SET ANY_CS,  
    remove_crlf IN BOOLEAN DEFAULT FALSE);
```

Fuente: Sintaxis del Procedimiento READ_LINE. Disponible en:
http://docs.oracle.com/cd/B28359_01/appdev.111/b28419/u_http.htm#i1026901

- r: la respuesta HTTP a leer.
- data: el cuerpo de la respuesta HTTP en formato texto.
- remove_crlf: es una bandera que indica si remover o no los caracteres de salto de línea. (\n)

Si se especifica un tiempo límite en la petición a la que pertenece la respuesta que está leyendo READ_LINE, este procedimiento espera que cada paquete esté listo para ser leído hasta que se cumpla el tiempo límite. Si se alcanza el tiempo límite READ_LINE detiene la lectura y retorna los datos que fueron leídos exitosamente,

⁶⁷ Docs.oracle.com [En: http://docs.oracle.com/cd/B28359_01/appdev.111/b28419/u_http.htm#i1026901]

en caso que no se hayan leído datos se dispara la excepción **transfer_timeout**, así el error puede ser controlado y reintentar la operación de lectura puede ser realizada después.

9.3.1.5 La teoría aplicada, utilizando UTL_HTTP para realizar una petición web: En el siguiente ejemplo se hace una petición HTTP a la página <http://localhost/tesis/> utilizando el paquete UTL_HTTP:

Nota: Los siguientes códigos fueron elaborados por los autores, los colores en el código expuesto pueden variar debido al estándar utilizado por cada lenguaje de programación en cada uno de los IDE utilizados para escribirlos

- Respuesta vista desde el navegador

Imagen 15. Ejemplo de Petición HTTP



Respuesta HTTP

Fuente: Los autores.

Procedimiento que realiza la petición desde la base de datos

```
DECLARE
    HTTP_PET utl_http.req;  --PETICION HTTP
    HTTP_RESP utl_http.resp;--PETICION HTTP
    LINEA_LEIDA CLOB;--LINEA QUE SE LEERÁ
    TOTAL_LEIDO CLOB;--TOTAL LEIDO DE LA RESPUESTA HTTP
BEGIN
    HTTP_PET := utl_http.begin_request('http://127.0.0.1/tesis/');
    HTTP_RESP := utl_http.get_response(HTTP_PET);

    BEGIN
        LOOP
            utl_http.read_line(HTTP_RESP, LINEA_LEIDA, TRUE);
            TOTAL_LEIDO:=TOTAL_LEIDO||LINEA_LEIDA||chr(10);
        END LOOP;
    EXCEPTION
    WHEN utl_http.end_of_body THEN
        utl_http.end_response(HTTP_RESP);
    END;
    dbms_output.put_line(TOTAL_LEIDO);
END;
```

Fuente: los autores.

Resultado del procedimiento anterior:

```
<html>
<head>
<title>Hola mundo UTL_HTTP</title>
</head>
<body>
<p>Respuesta HTTP</p>
</body>
</html>
```

Fuente: los autores.

Nota: para que el paquete UTL_HTTP funcione adecuadamente, se debe otorgar permisos de ejecución al usuario de la base de datos que requiere utilizarlo. Esto se logra con el siguiente comando:

```
grant execute on utl_http to usuario;
--reemplazar "usuario" con el usuario que requiere utilizar el paquete
```

Fuente: los autores.

9.3.2 Conclusión Oracle y HTTP: el primer paso para lograr el objetivo de este trabajo de grado es poder utilizar el protocolo HTTP desde el motor de base de datos Oracle. Esto es de vital importancia para poder hacer un correcto uso de los servicios web, la librería PL/SQL debe por enviar y recibir datos a través de la red, debido a que los servicios web utilizan dicho protocolo como su puente de comunicación principal para envío y recepción de información.

Oracle nos permite escribir en el cuerpo de la petición HTTP cualquier tipo de dato, incluido XML, lo que quiere decir que es posible enviar y recibir información desde cualquier servicio web.

Podemos concluir entonces que el motor de base de datos Oracle brinda la posibilidad de comunicarse de manera nativa con cualquier servicio publicado sobre el protocolo HTTP, permitiendo personalizar las peticiones y manejar las respuestas como el desarrollador lo desee.

9.3.3 Hablando el lenguaje de los servicios web, Oracle y el uso de XML: como se explicó en capítulos anteriores, los servicios web utilizan el lenguaje XML para empaquetar la información que envían y reciben, por ende para cumplir con

el objetivo general de este proyecto se debe conocer como el motor de base de datos Oracle hace uso de este lenguaje.

9.3.3.1 CLOB y su importancia: el tipo de dato CLOB almacena cerca de 128 TeraBytes de datos tipo carácter en la base de datos.⁶⁸ Este tipo de dato será utilizado para crear los paquetes SOAP que serán enviados al servicio web, este elemento será quien contenga toda la información XML de la petición realizada al servicio web.

Es necesario utilizar este tipo de dato en este proyecto ya que se requiere enviar y recibir grandes volúmenes de información desde y hacia los servicios web desde el motor de bases de datos, con el fin de disminuir la cantidad de información que manejan los equipos cliente en los que se encuentra instalado el software de cada ERP.

9.3.3.2 Uso de XMLELEMENT y XMLTYPE: la función XMLELEMENT se usa normalmente para producir un documento XML, del tipo XMLTYPE - Es un tipo de dato especial para manejar información en formato XML. ->Mismo que 1-, con una estructura anidada, ejemplo:⁶⁹

Imagen 16. Ejemplo del uso de XMLELEMENT

```
SELECT XMLELEMENT("Emp", XMLELEMENT("Name",
  e.job_id||' '||e.last_name),
  XMLELEMENT("Hiredate", e.hire_date)) as "Result"
FROM employee e WHERE employee_id > 200;
```

RESULT

```
<Emp>
  <Name>MK_MAN Harstein</Name>
  <Hiredate>17-FEB-96</Hiredate>
</Emp>

<Emp>
  <Name>MK_REP Fay</Name>
  <Hiredate>17-AUG-97</Hiredate>
</Emp>

<Emp>
  <Name>MK_REP Marvis</Name>
  <Hiredate>07-JUN-94</Hiredate>
</Emp>
```

Fuente: los autores.

⁶⁸ Ibid. [En: http://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm]

⁶⁹ Ibid [En: http://docs.oracle.com/cd/B19306_01/server.102/b14200/functions220.htm]

Nota: Para el objeto de este trabajo de grado, la función XMLELEMENT será utilizada para crear el cuerpo (body) de los paquetes SOAP que serán enviados a los servicios web.

9.3.3.3 Oracle y XML en la práctica: la siguiente grafica ilustra un ejemplo de una petición SOAP enviado a un servicio de ejemplo, se realizará una consulta SQL la cual genera una estructura XML idéntica a la de la petición.

Imagen 17. Estructura básica de una petición SOAP

```
SOAP Request
```

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:hello xmlns:ns2="http://ws/">
      <name>data</name>
    </ns2:hello>
  </S:Body>
</S:Envelope>
```

Fuente: Los autores.

Imagen 18. Consulta SQL para crear una petición SOAP

```
1 select xmlelement('S:body',
2           xmlelement('ns2:hello', XMLATTRIBUTES('http://ws/' as 'xmlns:ns2'),
3           xmlelement('name', 'data')) as result
4 from dual
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 1 en 0,001 segundos

RESULT
1 <S:body><ns2:hello xmlns:ns2="http://ws/"><name>data</name></ns2:hello></S:body>

Fuente: Los autores.

Como se puede observar en la imagen anterior es posible crear el contenido de un paquete SOAP desde el motor de la base de datos con una estructura anidada valida.

9.3.4 Extraer valores de datos XML: continuando con el ejemplo anterior, se hace necesaria la lectura de la información almacenada en los datos XML, esto se logra por medio de una función SQL llamada EXTRACTVALUE.⁷⁰

La sintaxis de esta función es:

Imagen 19. Sintaxis de la función SQL EXTRACTVALUE⁷¹



Fuente: Sintaxis de la función SQL EXTRACTVAL.
http://docs.oracle.com/cd/B10500_01/appdev.920/a96620/xdbo4cre.htm#1030960

- Type_instance: es el valor XML de donde se extraerá la información.
- XPath_string: es la ruta que la función EXTRACTVALUE debe seguir para encontrar los datos solicitados.
- Namespace_string: es un parámetro opcional que es utilizado como prefijo para la lectura de la información.

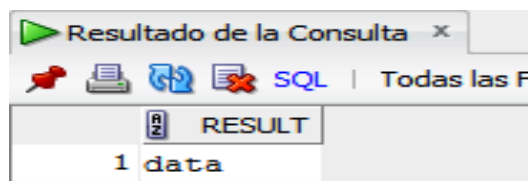
Tomando como base los datos XML del ejemplo utilizado en el capítulo anterior, el uso de la función EXTRACTVALUE para leer el atributo <name> debe ser el siguiente:

```
SELECT EXTRACTVALUE (XML, '/S:Body/ns2:hello/name') FROM DUAL
```

Fuente: los autores.

Y la ejecución de este comando da como resultado:

Imagen 20. Resultado de la función EXTRACTVALUE



Fuente: Los autores.

⁷⁰ Ibid, p. 68 [En: http://docs.oracle.com/cd/B10500_01/appdev.920/a96620/xdbo4cre.htm#1030960]

⁷¹ Ibid [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14259/img/extractvalue.gif]

9.3.4.1 Conclusión extracción de datos: esta función SQL es utilizada en la librería para entender de manera dinámica la respuesta entregada desde los servicios web y procesarla según el desarrollador indique.

Podemos concluir que utilizando de manera adecuada las herramientas del motor de bases de datos Oracle para la comunicación a través de HTTP y para el uso del lenguaje XML, es posible crear una librería la cual con parámetros adecuados, pueda comunicarse con cualquier clase de servicio web, tanto para crear una solicitud de manera dinámica, como para procesar de manera autónoma la respuesta proveniente del mismo. Una de las ventajas más grandes que presenta este tipo de modelo, es que no se requieren instalaciones adicionales ya que el soporte para estas tecnologías está incluido de manera nativa en el motor de base de datos, lo cual permite tener una independencia total de las máquinas de los usuarios, es decir que el sistema operativo, software instalado, recursos de hardware, entre otros, no puedan afectar la estabilidad y el rendimiento del sistema.

10. CONSUMO DE UN SERVICIO WEB DESDE LA BASE DE DATOS ORACLE

Siguiendo con el ejemplo planteado en el capítulo anterior, para que sea posible consumir servicios web desde el motor de base de datos Oracle, se debe enviar la estructura xml generada al servicio web a través de una petición HTTP. Para lograr este cometido se requiere escribir estos datos en el cuerpo de la petición HTTP que será enviada al servicio web del ejemplo, esto con la ayuda de UTL_HTTP y reutilizando parte del código expuesto anteriormente en este trabajo, se leerá el resultado de dicha petición, el cual será comparado con los datos de muestra producidos por el servicio web de ejemplo.

10.1 ESCRIBIENDO DATOS EN LA PETICIÓN DESDE UTL_HTTP

Para escribir datos en la petición HTTP desde el paquete UTL_HTTP se utiliza el procedimiento `WRITE_TEXT`⁷². Este procedimiento escribe en el cuerpo de la petición HTTP. Tan pronto como se envía texto al cuerpo de la petición, la sección de encabezado se completa y el texto es convertido automáticamente del juego de caracteres de la base de datos al juego de caracteres especificado en la petición.

La sintaxis del método `WRITE_TEXT` es:

Imagen 21. Sintaxis del procedimiento `WRITE_TEXT`

```
UTL_HTTP.WRITE_TEXT (  
  r      IN OUT NOCOPY REQ,  
  data  IN          VARCHAR2 CHARACTER SET ANY_CS);
```

Fuente: Sintaxis del procedimiento `WRITE_TEXT`.
http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#1027870

- r: la petición HTTP a escribir.
- data: el texto a escribir en el cuerpo de la petición HTTP.

Notas sobre el uso de `WRITE_TEXT`

Los clientes HTTP siempre deben permitir que el servidor remoto conozca el tamaño del cuerpo de la petición que se está enviando. Si el tamaño de la información es conocida de antemano se puede establecer la cabecera “**Content-Length**” en la petición HTTP, donde el tamaño es especificado en bytes y no en cantidad de caracteres. Si el tamaño de la petición no es conocido de antemano, se puede enviar la petición utilizando el formato “**chunked transfer-encoding**” de HTTP 1.1.

⁷² Ibíd, p. 70. [En: El procedimiento `WRITE_TEXT`. [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#1027870]

En el caso de este trabajo de grado, se conoce de antemano el tamaño del cuerpo de la petición HTTP a enviar y se utilizará el método “**set_header**”del paquete UTL_HTTP.

10.2 LAS CABECERAS DE LA PETICIÓN DESDE UTL_HTTP

El método SET_HEADER establece una cabecera HTTP. La cabecera es enviada al servidor web en cuanto es configurada.⁷³

La sintaxis del método SET_HEADER es:

Imagen 22. Sintaxis del procedimiento SET_HEADER

```
UTL_HTTP.SET_HEADER (  
  r      IN OUT NOCOPY req,  
  name   IN VARCHAR2,  
  value  IN VARCHAR2);
```

Fuente: Sintaxis del procedimiento SET_HEADER. Disponible en:
http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#i1027608

- r: la petición HTTP en la cual configurar la cabecera.
- name: nombre de la cabecera HTTP, ejemplo: content-Length.
- value: valor para la cabecera HTTP.

10.3 UNA PETICIÓN REAL A UN SERVICIO WEB REAL

Para el siguiente ejemplo se utilizará el servicio web de ejemplo que viene integrado con el entorno de desarrollo Netbeans 7.2. La siguiente imagen ilustra el contenido de una petición enviada a dicho servicio con su correspondiente respuesta, el objetivo de este ejemplo es demostrar cómo se puede realizar la misma operación utilizando solo el motor de base de datos Oracle.

⁷³ Docs.oracle.com [En: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_http.htm#i1027608]

Imagen 23. Código XML de una petición y respuesta SOAP



Fuente: Los autores.

Básicamente el consumo del servicio web desde la base de datos Oracle tiene tres etapas:

- Establecer el cuerpo de la petición (paquete SOAP⁷⁴)
- Configurar encabezados de la petición
- Leer el resultado de la petición

10.4 ESTABLECER EL CUERPO DE LA PETICIÓN

Inicialmente se deben conocer los datos a enviar ya que más adelante en la etapa de configuración de cabeceras es necesario indicar al servidor el tamaño de la petición HTTP que está siendo enviada.

⁷⁴ Código XML de una petición y respuesta SOAP. Más información en el capítulo 7 “Estado del Arte de los Servicios Web”

Imagen 24. Petición a realizar

```
declare --Bloque de definición de variables
    TAMANO NUMBER;--Tamaño de la petición a enviar
    CUERPO_PETICION XMLTYPE;--Petición a realizar
```

Fuente: Los autores.

El valor de la variable “CUERPO_PETICION” en formato XML es generado utilizando la función XMLELEMENT mencionada en capítulos anteriores:

Imagen 25. Cuerpo de petición en formato XML

```
select xmlelement("S:Body", xmlelement("ns2:hello",
    xmlattributes('http://ws/' as "xmlns:ns2"),
    xmlelement("name", 'data'))) into CUERPO_PETICION from dual;
```

Fuente: Los autores.

Con la sentencia anterior se crea el cuerpo de la petición a enviar⁷⁵.

Además para que la petición esté completa, se deben especificar las cabeceras del paquete SOAP a enviar y al final del bloque se almacena el tamaño del paquete SOAP en la variable “TAMANO”, para esto se concatenan al cuerpo de la petición las siguientes líneas.

⁷⁵ En el capítulo Oracle y XML se indica el resultado de esta sentencia SQL.

Imagen 26. Cabeceras del paquete SOAP

```
declare --Bloque de definición de variables
  TAMANO NUMBER;--Tamaño de la petición a enviar
  CUERPO_PETICION XMLTYPE;--Petición a realizar
  PETICIÓN CLOB;--Cabeceras del contenido de la petición

begin
  PETICION := '<?xml version="1.0" encoding="UTF-8"?>';
  PETICION := PETICION||
    '<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">';
  PETICION := PETICION|| '<S:Header/>';

  select xmlelement("S:Body", xmlelement("ns2:hello",
    xmlattributes('http://ws/' as "xmlns:ns2"),
    xmlelement("name",'data'))) into CUERPO_PETICION from dual;

  PETICION := PETICION||CUERPO_PETICION.GETCLOBVAL();
  PETICION := PETICION|| '</S:Envelope>';

  TAMANO := LENGTH(PETICION);
```

Fuente: Los autores.

De esta forma está completo el paquete SOAP que será enviado al servicio web.

10.5 CONFIGURANDO LOS ENCABEZADOS DE LA PETICIÓN

Para que la petición sea exitosa como mínimo se debe se deben especificar el tamaño, la versión del protocolo HTTP a utilizar, el método (GET o POST) y se debe escribir el paquete SOAP generado anteriormente en la petición.

Imagen 27. Respuesta desde el servicio web

```
--Se inicializa la petición HTTP a la URL del servicio web
--Utilizando POST y la versión 1.1 del protocolo
HTTP_PET :=
utl_http.begin_request('http://127.0.0.1:8084/Tesis/ServicioWeb', 'POST',
'HTTP/1.1');

--Se configura el tamaño de la petición
utl_http.set_header(HTTP_PET, 'Content-Length', TAMANO);

--Se escribe en la petición el paquete SOAP generado en las líneas
anteriores
utl_http.write_text(HTTP_PET,PETICION);

--Se obtiene la respuesta desde el servicio web
HTTP_RESP := utl_http.get_response(HTTP_PET);
```

Fuente: Los autores.

En el bloque de código anterior se almacena la respuesta desde el servicio web en la variable “HTTP_RESP” que es del tipo `utl_http.resp`, el cual fue mencionado en capítulos anteriores.

10.6 LEER EL RESULTADO DE LA PETICIÓN

Finalmente con el resultado de la petición realizada al servicio web almacenado en la variable “HTTP_RESP” se deben extraer la información línea por línea utilizando los métodos mencionados anteriormente en este mismo capítulo.

Imagen 28. Línea de respuesta HTTP

```
BEGIN
  LOOP
    --Se lee cada línea de la respuesta HTTP
    utl_http.read_line(HTTP_RESP, LINEA_LEIDA, TRUE);
    --Se concatena la línea leída con el total de líneas
    TOTAL_LEIDO:=TOTAL_LEIDO||LINEA_LEIDA||chr(10);--Salto de línea
  END LOOP;
EXCEPTION
WHEN utl_http.end_of_body THEN
  --Cuando se alcanza el final de la respuesta HTTP
  --Se cierra la conexión http
  utl_http.end_response(HTTP_RESP);
END;
dbms_output.put_line(TOTAL_LEIDO);--Se imprime el total leído
```

Fuente: Los autores.

La salida en el log del servidor de base de datos Oracle es la siguiente:

Imagen 29. Respuesta del servidor de base de datos

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:helloResponse xmlns:ns2="http://ws/">
      <return>Hello data !</return>
    </ns2:helloResponse>
  </S:Body>
</S:Envelope>
```

Fuente: Los autores.

El resultado en el log del servidor de base de datos Oracle es la respuesta obtenida desde el servicio web.

10.7 PRUEBAS DE RENDIMIENTO Y RESULTADO

Para la prueba se plantean el siguiente caso de estudio:

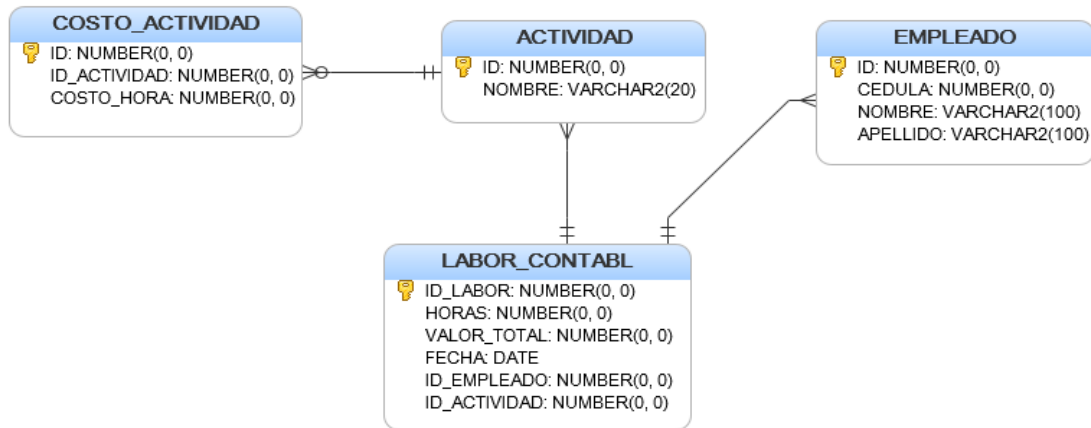
- Un sistema ERP contable, el cual publica un servicio web que se encarga de recibir un grupo de actividades realizadas por un conjunto de empleados, enviar dicha información al núcleo de la ERP contable y retornar el mismo número de actividades pero con un valor adicional el cual representa el valor pagado al empleado por dicha actividad.
- Un sistema ERP de manejo de actividades, el cual gestiona las actividades realizadas por un conjunto de empleados y que debe enviar dicha información al servicio web de la ERP contable, además almacena el valor de retorno entregado por el servicio web.

Se desarrollaron dos aplicaciones las cuales representan el sistema ERP de gestión de actividades, ambas fueron desarrolladas utilizando la tecnología Java FX – FXML. La única diferencia entre estas dos aplicaciones es la manera como ejecutan la interfaz de comunicación con el sistema ERP contable, una de ellas hace uso de la librería PL/SQL, la otra por el contrario no hace uso de la librería. Cada aplicación cuenta con una gráfica de barras la cual indica el consumo de memoria por segundo durante el tiempo de ejecución de la interfaz.

De igual manera, se diseñó un servicio web el cual representa el servicio publicado por el sistema ERP contable, este servicio almacena las actividades recibidas y retorna el valor a pagar por cada actividad.

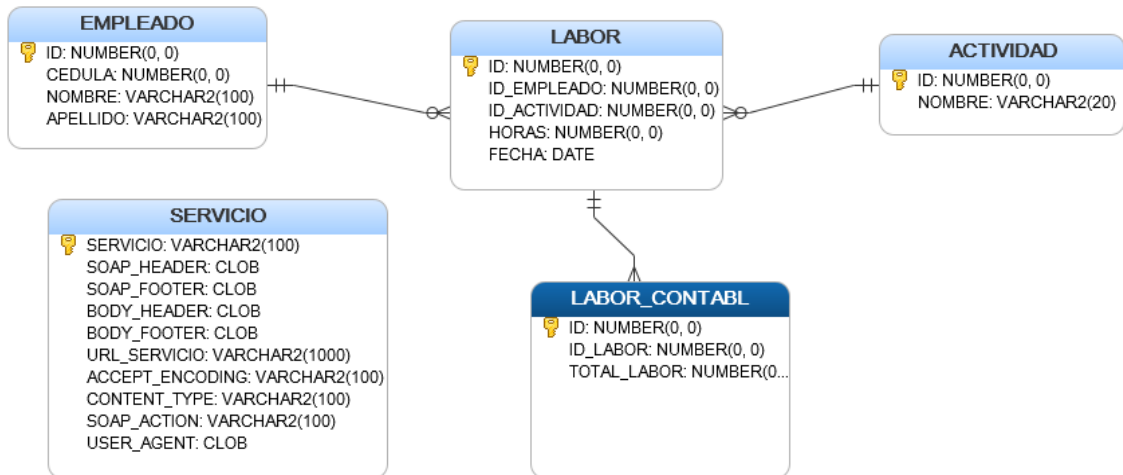
Los modelos entidad relación de cada sistema ERP son los siguientes:

Diagrama 3. Diagrama relacional de la base de datos del sistema contable



Fuente: Los autores.

Diagrama 4. Diagrama relacional de base de datos del sistema de gestión de actividades



Fuente: Los autores.

El equipo utilizado para la prueba cuenta con las siguientes características de Hardware.

- DELL XPS - CORE I7 - 2.20GH
- 8 GB RAM
- Oracle 11g
- 134000 registros

Resultados sin implementar la librería:

- Tiempo ejecución: 213 Segundos.
- Memoria utilizada en promedio: 181MB

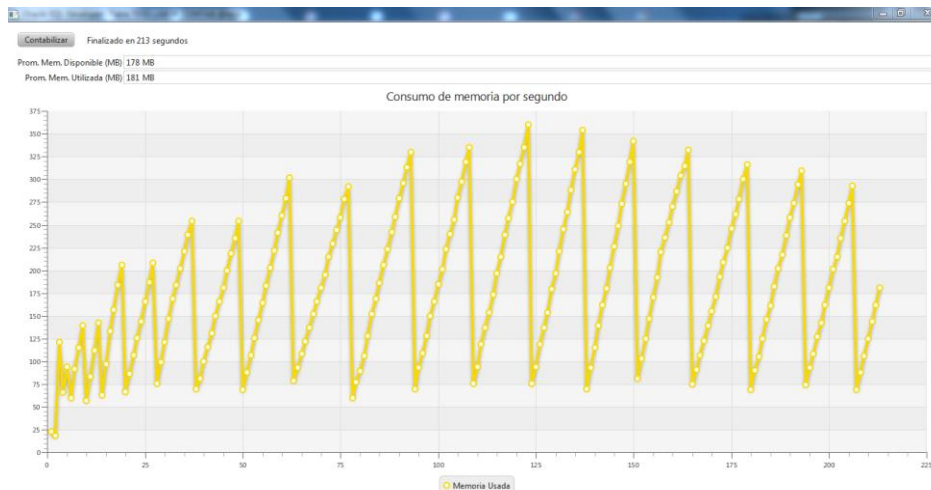
Resultados con la librería implementada:

- Tiempo ejecución: 81 Segundos.
- Memoria utilizada en promedio: 16MB

Calculo de resultados

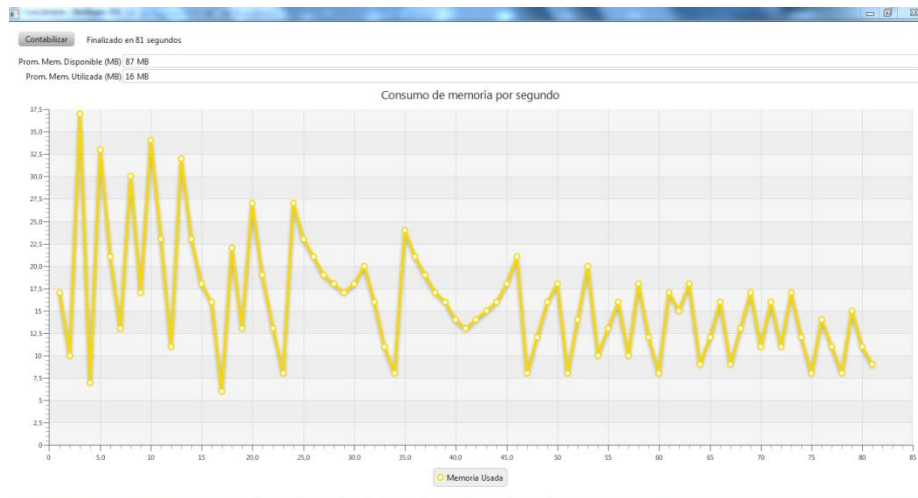
- Tiempo de ejecución
 - o Tiempo sin librería: 213 Segundos
 - o Tiempo con librería: 81 Segundos
 - o % de tiempo con librería: $81 * 100 / 213 = 38\%$
 - o % de tiempo ahorrado 62%
- Memoria utilizada
 - o Memoria promedio sin librería: 181 MB
 - o Memoria promedio con librería: 16 MB
 - o % Memoria utilizada: $16 * 100 / 181 = 8,8\%$
 - o % Memoria ahorrada: 91,2%

Imagen 30. Consumo de memoria sin implementar la librería en la máquina del cliente



Fuente: los autores.

Imagen 31. Consumo de memoria con la librería implementada en la máquina del cliente.



Fuente: los autores.

En las imágenes se logra observar el cambio que ocurre en el comportamiento de las aplicaciones, donde el consumo de memoria sin implementar la librería tiene un incremento sostenido el cual no se hace visible al momento de implementar la librería, claramente se nota un consumo de memoria mucho menor en la segunda aplicación. Lo anterior demuestra que mediante el uso de la librería se aumenta el nivel de disponibilidad del sistema, disminuyendo los tiempos de ejecución y la cantidad de memoria utilizada.

10.8 DOCUMENTACIÓN Y USO

10.8.1 Configuración de los servicios: los servicios que se deseen utilizar deben ser configurados en una tabla llamada “Servicio” la cual tiene la siguiente estructura:

Tabla 3. Tabla de Servicios

COLUMN_NAME	DATA_TYPE	NULL...	DATA_DEFAULT	COMMENTS
SERVICIO	VARCHAR2...	No	(null)	Identificador del servicio
SOAP_HEADER	CLOB	No	(null)	Encabezado del paquete SOAP
SOAP_FOOTER	CLOB	No	(null)	Cierre del paquete SOAP
BODY_HEADER	CLOB	No	(null)	Encabezado del cuerpo del paquete SOAP
BODY_FOOTER	CLOB	No	(null)	Cierre del encabezado del cuerpo del paquete SOAP
URL_SERVICIO	VARCHAR2...	No	(null)	Dirección URL del servicio web
ACCEPT_ENCODING	VARCHAR2...	Yes	'gzip,deflate'	Compresión a utilizar
CONTENT_TYPE	VARCHAR2...	Yes	'text/xml;charset=UTF-8'	Tipo de contenido que se envía al servicio web
SOAP_ACTION	VARCHAR2...	Yes	(null)	Acción SOAP a ejecutar
USER_AGENT	CLOB	Yes	'Mozilla/4.0'	Identificador del navegador que se va a utilizar

Fuente: los autores.

En esta tabla se almacenan los siguientes datos:

- **SERVICIO:** identificador del servicio, le indica a la librería de donde tomar la información necesaria para conectar con el servicio web a consumir.
- **SOAP_HEADER** y **SOAP_FOOTER:** Datos en formato XML los cuales almacenan las etiquetas de apertura y cierre del encabezado del paquete SOAP. (ver imagen 17, 23 y capítulo 7).
- **BODY_HEADER** y **BODY_FOOTER:** Datos en formato XML los cuales almacenan las etiquetas de apertura y cierre del cuerpo del paquete SOAP. (ver imagen 17,23 y capítulo 7).
- **URL_SERVICIO:** dirección URL donde se encuentra situado el servicio web.
- **“ACCEPT_ENCODING:** Modo de optimización encargado de indicar al servidor web que tipo de codificación es esperada la respuesta, por defecto se utiliza “gzip,deflate” para indicar que se espera información comprimida utilizando los métodos gzip y deflate”⁷⁶.
- **CONTENT_TYPE:** Tipo de contenido que se escribe en la petición realizada al servicio web, para el uso de servicios web se debe utilizar “text/xml” que indica que se espera texto en formato XML. Adicionalmente por defecto se indica que el juego de caracteres a utilizar es el “UTF8” pero en teoría

⁷⁶ Cómo funciona la compresión GZIP / Deflate. La compresión de una web es un proceso transparente para el usuario, pero es bueno que conozcamos un poco sobre cómo se realiza internamente en las comunicaciones HTTP. [Editado: 05 de agosto de 2010]. Por: Miguel Angel Alvarez. [Consulta: 12-11-2013. Disponible en: <http://www.desarrolloweb.com/articulos/como-funciona-compresion-gzip-deflate.html>]

puede ser reemplazado por cualquiera de los valores registrados por la “Autoridad de Asignación de Cifras en Internet (IANA)”⁷⁷

- SOAP_ACTION: Dependiendo de la documentación del servicio web, se debe especificar este parámetro o no. Indica el método SOAP a ejecutar en el servicio web.
- USER_AGENT: “es el identificador del navegador a utilizar, le dice al servidor que tipo de cliente web está realizando la petición HTTP”⁷⁸. Se recomienda utilizar el valor establecido por default.

Los datos especificados en esta tabla son utilizados por la librería para crear una petición HTTP adecuada para cada servicio.

10.8.2 Métodos disponibles: básicamente para utilizar un servicio web de integración los sistemas deben crear paquetes SOAP en formato XML a partir de la información en la base de datos, transportar estos paquetes por medio de HTTP y finalmente procesar la respuesta del servicio web (que llega nuevamente en formato XML) y almacenar los resultados en la base de datos.

Para cumplir de manera transparente con cada etapa del ciclo mencionado en el párrafo anterior, la librería brinda al desarrollador las siguientes alternativas:

CREATE_SOAP_PACKAGE

Crea un paquete SOAP utilizando las cabeceras de apertura y cierre del paquete y la información a empaquetar.

Parámetros:

- REQUEST_HEADER: encabezado del paquete
- REQUEST_CONTENT: contenido del paquete, información a enviar al servicio web
- REQUEST_FOOTER: etiquetas de cierre del encabezado del paquete

Salida:

Un paquete SOAP listo para ser enviado por medio de HTTP al servicio web

CONSUME_WS - GENERICO

⁷⁷ Character Sets. Last Updated. [Edition: 2013-10-01]. Registration Procedure(s), Expert Review Expert(s). Primary Expert Ned Freed and Secondary Expert Martin Dürst. [Consulta: 12-11-2013. Disponible en: <http://www.iana.org/assignments/character-sets/character-sets.xhtml>]

⁷⁸ Definition of User Agent. [This Wiki page is edited by participants of the User Agent Accessibility Guidelines working group (UAWG)]. [Consulta: 12-11-2013]. Disponible en: http://www.w3.org/WAI/UA/work/wiki/Definition_of_User_Agent

Realiza una petición a un servicio web utilizando como base la configuración especificada en sus parámetros. Es un método genérico, el cual no utiliza la tabla "Servicio" para darle mayores posibilidades al desarrollador de utilizarlo desde implementaciones propias.

Parámetros:

- CL_DATOS_ENVIO: datos que van a ser enviados al servicio web.
- CL_SOAP_ENVELOPE_HEADER: encabezado del paquete SOAP que será creado y enviado.
- CL_SOAP_ENVELOPE_FOTER: etiquetas de cierre del paquete SOAP.
- CL_SOAP_BODY_HEADER: etiqueta de cuerpo del paquete SOAP que será creado y enviado
- CL_SOAP_BODY_FOOTER: etiquetas de cierre del cuerpo del paquete SOAP
- CH_WS_URL: dirección URL del servicio web.
- CH_ACCEPT_ENCODING: tipo de codificación que se espera en la respuesta.
- CH_CONTENT_TYPE: tipo enel que se escribe el contenido de la petición. (ver apartado 10.8.2 para más detalles)
- CH_SOAP_ACTION: acción SOAP a ejecutar en el servicio web.
- CH_USER_AGENT: identificador de cliente a utilizar para realizar la petición al servicio web.

Salida

Respuesta recibida desde el servicio web en formato XML.

CONSUME_WS - EXTENSIÓN

Este método consume un servicio web utilizando como base el nombre del servicio configurado en la tabla "servicio" y los datos a enviar.

Parámetros:

- CH_SERVICIO: servicio configurado en la tabla servicio
- CL_DATOS_ENVIO: datos que van a ser enviados al servicio web.

Salida:

Respuesta recibida desde el servicio web en formato XML.

TALBA_TO_XML

Este método toma los datos de cualquier tabla y los convierte a formato XML, brinda la posibilidad de especificar que columnas de la tabla convertir a XML y permite filtrar los resultados utilizando una condición SQL genérica.

Parámetros:

- CH_TABLA: Nombre de la tabla de donde se tomarán los datos para convertir en XML.
- CH_CAMPOS: Campos a utilizar en la salida XML
- CH_CONDICION: Condición a utilizar para filtrar los resultados a partir de la tabla
- CH_ROWSET: es el nombre del nodo XML que agrupa todas las filas del resultado de la consulta.
- CH_ROW: es el nombre del nodo XML que agrupa cada fila del resultado de la consulta
 - o Ejemplo de ROWSET y ROW
En un resultado XML de esta función el ROWSET y el ROW se verían identificados de la siguiente manera:

Imagen 32. Ejemplo de ROWSET y ROW

```
<rowset>
  <row>
    <columna1>dato1</columna1>
    <columna2>dato2</columna2>
    <columna3>dato3</columna3>
    <columna4>dato4</columna4>
  </row>
</rowset>
```

Fuente: los autores.

Salida:

Contenido de la tabla en formato XML.

TABLA_TO_XML_UCAMPOS

Este método, al igual que TABLA_TO_XML toma los datos de cualquier tabla y los convierte a formato XML, pero sin utilizar el atributo CH_CONDICION, lo que quiere decir que convierte todos los datos de la tabla a formato XML.

TABLA_TO_XML_UCONDI

Este método, al igual que TABLA_TO_XML toma los datos de cualquier tabla y los convierte a formato XML, pero sin utilizar el atributo CH_CAMPOS, lo que quiere decir que convierte todas las columnas de la tabla a formato XML utilizando la condición especificada en CH_CONDICION.

TABLA_TO_XML_UTABLA

Este método, al igual que TABLA_TO_XML toma los datos de cualquier tabla y los convierte a formato XML, pero sin utilizar los atributos CH_CONDICION y CH_CAMPOS, lo que quiere decir que convierte todas las columnas y todos los datos de la tabla a formato XML.

XML_TO_TABLA

Este método inserta en cualquier tabla la información en formato XML especificada en sus parámetros.

Parámetros:

- CH_TABLA: Nombre de la tabla donde se almacenarán los datos extraídos desde el XML de entrada.
- CH_ROOT: Es el nombre del nodo raíz de donde se tomará la información, se utiliza este nombre ya que por lo general los datos provenientes de los servicios web en formato XML contienen algunos nodos con información adicional que completan el paquete SOAP, gracias a este parámetro se puede extraer la información sin confundir los nodos del paquete con los nodos donde se encuentra la información a almacenar.
- CL_XML: Información en formato XML de donde se extraerán los datos a almacenar en la tabla especificada en CH_TABLA

Salida:

Este método no produce ninguna salida.

UTILIZA_SERVICIO_FULL

Este método es una implementación básica donde se utilizan todos los demás métodos mencionados anteriormente, es incluida en la librería para ilustrar lo fácil que es utilizarla y extenderla. Se encarga de consumir el servicio web especificado utilizando los datos de la tabla origen y almacenar el resultado del servicio web en la tabla de destino.

Parámetros

- CH_SERVICIO: servicio configurado en la tabla servicio
- CH_TABLA_ORIGEN: Tabla de donde será extraída la información para enviarla al servicio web
- CH_CAMPOS_ORIGEN: Columnas que serán utilizadas desde la tabla origen.
- CH_CONDICION: Condición que será utilizada para filtrar los datos desde la tabla origen.
- CH_TABLA_DESTINO: Tabla donde será almacenada la información recibida desde el servicio web.
- CH_ROWSET: es el nombre del nodo XML que agrupa todas las filas del resultado de la consulta.
- CH_ROW: es el nombre del nodo XML que agrupa cada fila del resultado de la consulta
 - o Ejemplo de ROWSET y ROW

Imagen 33. En un resultado XML de esta función el ROWSET y el ROW se verían identificados de la siguiente manera:

```
<rowset>
  <row>
    <columna1>dato1</columna1>
    <columna2>dato2</columna2>
    <columna3>dato3</columna3>
    <columna4>dato4</columna4>
  </row>
</rowset>
```

Fuente: los autores.

Salida

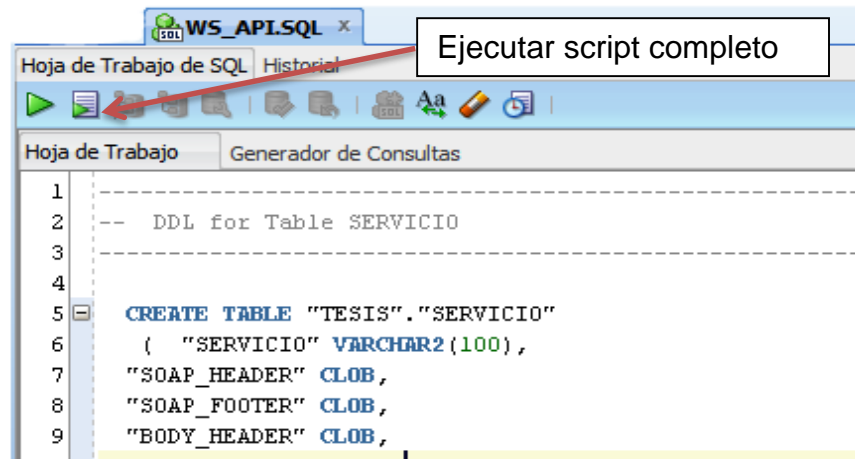
Este método no tiene salidas.

10.8.3 Instalación: el paquete de instalación es un archivo SQL el cual contiene las sentencias SQL necesarias para instalar la librería en el motor de base de datos Oracle. Existen dos maneras de instalar la librería:

- Desde un editor SQL

Se debe abrir el archivo WS_API.sql en una hoja de trabajo del editor favorito, y con la sesión donde se desea instalar la librería se debe ejecutar el script completo:

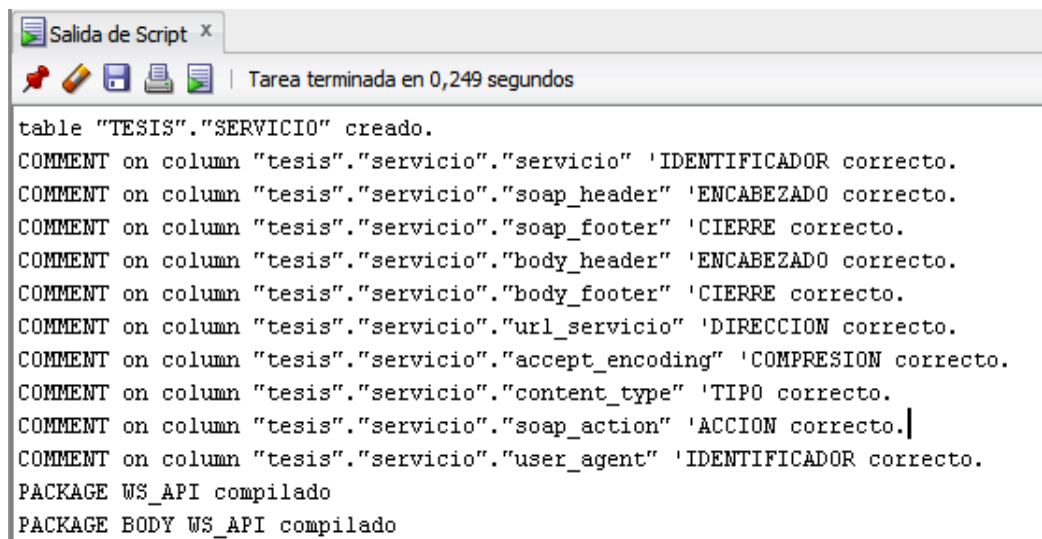
Imagen 34. Ejecución script completo



Fuente: los autores.

Nota: en este caso utilizamos SQL Developer de Oracle.

Imagen 35. La salida del script debe de ser la siguiente:



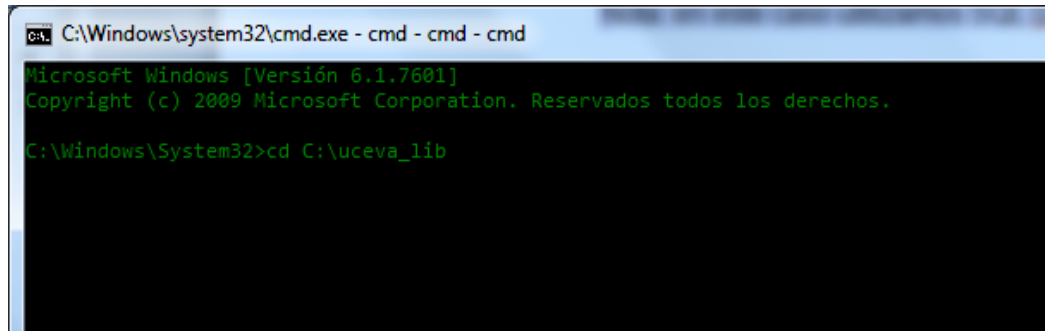
Nota: en este caso se utilizó el usuario "tesis" para instalar la librería, eso quiere decir que este usuario ahora tiene a su disposición las funcionalidades documentadas en los capítulos 10.8.1 y 10.8.2

Fuente: los autores.

- Desde la consola SQL

Desde la consola de comandos se debe navegar con el comando “cd” en Windows o Linux hasta el directorio donde está almacenado el script de instalación, en este caso “C:\uceva_lib”:

Imagen 36. Uso del comando “cd” desde la consola



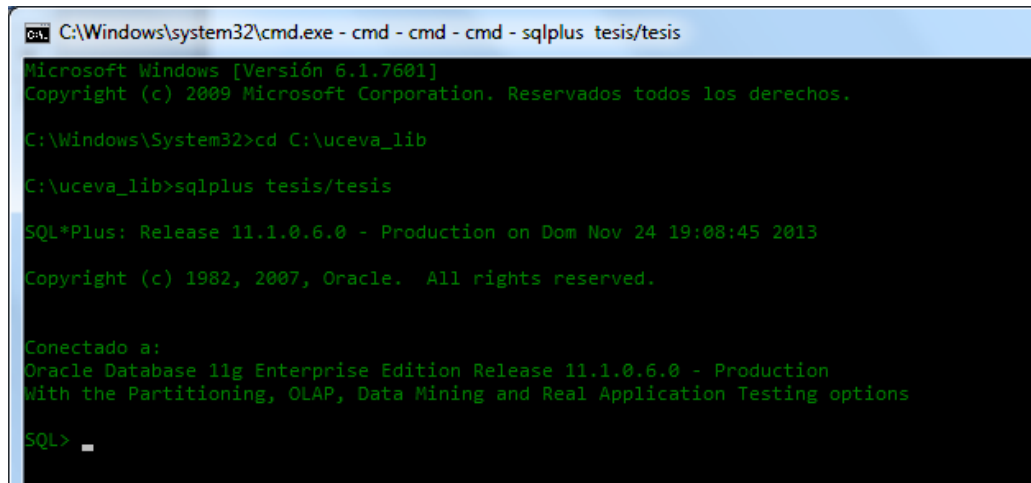
```
C:\Windows\system32\cmd.exe - cmd - cmd - cmd
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Windows\System32>cd C:\uceva_lib
```

Fuente: los autores.

Ahora se debe iniciar sesión en el motor de base de datos con el usuario en donde se desea instalar la librería:

Imagen 37. Ejecución del archivo WS_API.sql



```
C:\Windows\system32\cmd.exe - cmd - cmd - cmd - sqlplus tesis/tesis
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Windows\System32>cd C:\uceva_lib
C:\uceva_lib>sqlplus tesis/tesis

SQL*Plus: Release 11.1.0.6.0 - Production on Dom Nov 24 19:08:45 2013

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Conectado a:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> _
```

Fuente: los autores.

Ahora se debe ejecutar el archivo WS_API.sql con la instrucción “@WS_API.sql”, la salida del comando anterior debe ser la siguiente:

Imagen 38. El usuario “tesis” para instalar la librería

```
C:\Windows\system32\cmd.exe - cmd - cmd - cmd - sqlplus tesis/tesis

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Conectado a:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @WS_API.SQL

Tabla creada.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Comentario creado.

Paquete creado.

Cuerpo del paquete creado.

SQL>
```

Fuente: los autores.

En este caso se utilizó el usuario “tesis” para instalar la librería, eso quiere decir que este usuario ahora tiene a su disposición las funcionalidades documentadas en los items 10.8.1 y 10.8.2

11. CONCLUSIONES

Gracias a la investigación realizada al estado actual de servicios web, su funcionamiento y su utilización en el mercado actual, al igual que las ERP más usadas por las empresas, se encontró que algunos de los factores que más afecta el continuo desarrollo de las tareas cotidianas del usuario final son: el rendimiento del hardware, rendimiento del software, recurso humano y disponibilidad de la red y que una de las actividades que más impacto tiene sobre los equipos de los usuarios finales es la ejecución de interfaces de comunicación entre ERPs.

Con base en el estudio realizado, se detectó que en el modelo de comunicación entre ERP's actual existe una redundancia de pasos para dar respuesta a la petición hecha por el usuario, por esta razón se llevó a cabo la creación de un nuevo modelo de comunicación (ver imagen 11), que apoyado en la librería desarrollada por los autores, logra disminuir la utilización de memoria en el equipo cliente por parte de los sistemas ERP al momento de ejecutar interfaces de comunicación con servicios web, dejando la mayor parte de procesamiento a los servidores donde se encuentra alojado el motor de bases de datos Oracle, lo que brinda una reducción considerable de tiempos de respuesta y un aumento de la disponibilidad de los recursos hardware del equipo del usuario.

Dicho modelo se llevó a cabo aprovechando las tecnologías operadas en la empresa, como prueba de ello se demuestra mediante un caso real que el motor de base de datos Oracle cuenta con las herramientas necesarias para hacer uso de servicios web y además es posible realizarlo sin instalar componentes adicionales tanto en las máquinas de los usuarios como en el servidor.

Finalmente se concluye que mediante la implementación de dicha librería en el nuevo modelo de comunicación, se logró disminuir tanto la cantidad de memoria utilizada en los equipos clientes, como el tiempo requerido para la ejecución de la interfaz, logrando un ahorro del 62% del tiempo de ejecución y un 91% de la memoria utilizada por el sistema.

Los resultados mencionados anteriormente son de vital importancia para cualquier empresa, ya que el funcionamiento óptimo de sus sistemas ERP se ve reflejado en la eficiencia y eficacia de sus procesos internos y sus tiempos de respuesta, además, esto permite el desarrollo de las actividades de los usuarios con mayor agilidad y comodidad.

BIBLIOGRAFÍA

ALONSO, Gustavo. CASATI, Fabio. KUNO, Harumi. MACHIRAJU, Vijay. Web Services: Concepts, Architectures and Applications. Edición 2004. Palo Alto, CA, USA, Springer, 2004. 354 p. ISBN-10: 3540440089. ISBN-13: 978-3540440086

ALVAREZ Miguel Angel. Qué es XML [en línea]. Publicado: 13 junio 2001. [Fecha de consulta: 18 diciembre 2012]. Disponible en: <http://www.desarrolloweb.com/articulos/449.php>

Artech Consultores S.R.L. Web Services [en línea]. [Fecha de consulta: 15 diciembre 2012]. Disponible en http://www.gxtechnical.com/gxdlsp/pub/genexus/internet/technicalpapers/web_services.htm

AUGCyL. Introducción a los sistemas distribuidos [en línea]. [Fecha de consulta: 10 diciembre 2012]. Disponible en http://augcyl.org/?page_id=231

B. V, Kumar. Web Services an Introduction. West Patel Nagar, New Delhi. Tata McGraw-Hill, 2004. 447 p. ISBN-13: 978-0-07-059378-7

Benvenuto Vera, Ángelo. Universidad de Concepción. Implementación de Sistemas ERP, su impacto en la gestión de la empresa e integración con otras TIC/ El concepto de los ERP y su evolución. *CAPIV REVIEW Vol. 4 2006 ISSN 0718-4654 Versión impresa / ISSN 0718-4662 Versión en línea*. Consulta: 01-11-2013. En: <http://www.capic.cl/capic/media/ART3Benvenuto.pdf>

BOX Don, EHNEBUSKE David, KAKIVAYA Gopal, LAYMAN Andrew, MENDELSONH Noah, FRYSTYK Nielsen Henrik, THATTE Satish, WINER Dave. Simple Object Access Protocol (SOAP) 1.1. [En línea]. Publicado: 08 mayo de 2000. [Fecha de consulta: 15 diciembre 2012]. Disponible en: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

Character Sets. Last Updated. [Edition: 2013-10-01]. Registration Procedure(s), Expert Review Expert(s). Primary Expert Ned Freed and Secondary Expert Martin Dürst. [Consulta: 12-11-2013]. Disponible en: <http://www.iana.org/assignments/character-sets/character-sets.xhtml>

Computación GRID. Trabajo realizado por: Isaac David Barsallo. Fecha: Jueves 1 de julio de 2010. Disponible en: <http://www.slideshare.net/Savoy158/computacion-grid>.

Cómo funciona la compresión GZIP / Deflate. La compresión de una web es un proceso transparente para el usuario, pero es bueno que conozcamos un poco sobre cómo se realiza internamente en las comunicaciones HTTP. [Editado: 05 de agosto de 2010]. Por: Miguel Angel Alvarez. [Consulta: 12-11-2013. Disponible en: <http://www.desarrolloweb.com/articulos/como-funciona-compresion-gzip-deflate.html>]

COULOURIS, George. DOLLIMORE, Jean. KINDBERG, Tim. Sistemas Distribuidos – Conceptos y Diseño. 3b ed. ADDISON-WESLEY, 2005. 744 p. ISBN 8478290494.

Definition of User Agent. [This Wiki page is edited by participants of the User Agent Accessibility Guidelines working group (UAWG). [Consulta: 12-11-2013]. Disponible en: http://www.w3.org/WAI/UA/work/wiki/Definition_of_User_Agent

DEPARTAMENTO de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura – UNNE. Seguridad De Los Sistemas Distribuidos. Actualización: 20 octubre 2008. [Fecha de consulta: 28 noviembre 2012]. Disponible en: http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Seguridad_Sistemas_Distribuidos.pdf

GARG, Vinodkumar. N. K. Venkitakrishnan. Enterprise Resource Planning: Concepts and Practice. 2 Ed. New Delhi. India. Prentice-Hall of India Pvt. Ltd., 2004. 200 p. ISBN 81-203-2254-1

GNU.ORG. GNU Lesser General Public License [en línea]. Actualizada: 10 junio 2012. [Fecha de consulta: 03 septiembre 2012]. Disponible en: <http://www.gnu.org/copyleft/lesser.html>

GXTECHNICAL. Web Services [en línea]. [Fecha de consulta: 20 diciembre 2012]. Disponible en http://www.gxtechnical.com/gxdlsp/pub/genexus/internet/technicalpapers/web_services.htm

IBM. Esquemas de codificación [en línea]. [Fecha de consulta: 03 enero 2013]. Disponible en http://pic.dhe.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=%2Fcom.ibm.db2z10.doc.char%2Fsrc%2Ftpc%2Fdb2z_encodingscheme.htm

Información comprimida utilizando los métodos gzip y deflate. Disponible en: <http://www.desarrolloweb.com/articulos/como-funciona-compresion-gzip-deflate.html>

MASIP, David. ¿Qué es Oracle. En desarrolloweb.com. Edición 19 de julio de 2002. Consulta 23 de febrero de 2013. [En: <http://www.desarrolloweb.com/articulos/840.php>

MUÑIS, Luis. ERP Guía Práctica para la selección e implementación. Impreso en España, Ediciones Gestion 2000, 2004. 241 p. ISBN 84-8088-359-6.

NEWCOMER, Eric. Understanding Web Services: Xml, Wsdl, Soap, and Uddi. 75 Arlington Street, Suite 300 Boston, Pearson Education Inc., 2002. 368 p. ISBN-10: 0201750813.

Pineda Díaz, Amanda Yasmine. Sistema para el acceso, recuperación y procesamiento de archivos multimedia en ambiente Java/Jini. Tesis de grado Licenciatura en Ingeniería en Sistemas Computacionales. Cholula, Puebla, México: Universidad de las Américas Puebla. Departamento de Ingeniería en Sistemas Computacionales. 2004.

PIQUERO, Julián Verón. Listas de control de acceso (ACL). Prácticas de redes, 2009.

RAMÍREZ, Alberto. Metodología de la investigación científica. Edición, 26 de agosto de 2005. Consulta el 01 de abril de 2013. [En: <http://javeriana.educ.co/fear/ecología/documents/ALBERTORAMIREZMETODOLOGIADELAINVESTIGACION>

RENATA.EDU.CO. Curso de computación en grid [en línea]. [Fecha de consulta: 10 diciembre 2013]. Disponible en

<http://www.renata.edu.co/index.php/experiencias-significativas/894-curso-de-computacion-en-grid-comenzara-el-primerode-marzo.html>

ROJO, J. Oscar. Introducción a los Sistemas Distribuidos [en línea]. Actualizado: 2003. [Fecha de consulta: 12 enero 2013]. Disponible en: http://augcyl.org/?page_id=231

Roy T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures", PhD thesis, UC Irvine, 2000, en: <http://roy.gbiv.com/pubs/dissertation/top.htm>

SHANNON, Ross. What is HTML? Hyper Text Markup Languaje explained en HTML Sources HTML Tutorials. Editado el 21 de agosto de 2012. Fecha consulta: 03 de enero de 2013. [En: <http://www.yourhtml/source.com/starthere/whatishtml.html>

W3C. Extensible Markup Language (XML) [en línea]. Actualizada: 01 enero 2012. [Fecha de consulta: 18 diciembre 2012]. Disponible en: <http://www.w3.org/XML/>
B V Kumar. Web Services an Introduction. West Patel Nagar, New Delhi. Tata McGraw-Hill, 2004. 447 p. ISBN-13: 978-0-07-059378-7

WONG, Cliton. HTTP Pocket Reference. 101 Morris Street, Sebastopol, O'Reilly & Associates, Inc., 2000. 75 p. ISBN 1-56592-862-8.

WORLD WIDE WEB Consortion Oficina España. Guía Breve de Servicios Web [en línea]. [Fecha de consulta: 22 diciembre 2012]. Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

W3C. Extensible Markup Language (XML) [en línea]. Actualizada: 01 enero 2012. [Fecha de consulta: 18 diciembre 2012]. Disponible en: <http://www.w3.org/XML/>

XUMARHU.NET. GRID COMPUTING [en línea]. [Fecha de consulta: 10 diciembre 2012]. Disponible en http://www.xumarhu.net/grid_computing.ppt

YOURHTMLSOURCE. What is HTML? [En línea]. [Fecha de consulta: 03 enero 2013]. Disponible en <http://www.yourhtmlsource.com/starthere/whatishtml.html>

Páginas electrónicas:

- http://books.google.com.co/books?id=etQc3_PXnQoC&printsec=frontcover#v=onepage&q&f=false página 27
- http://books.google.com.co/books?id=etQc3_PXnQoC&printsec=frontcover#v=onepage&q&f=false página 22
- http://books.google.com.co/books?id=etQc3_PXnQoC&printsec=frontcover#v=onepage&q&f=false páginas 22 y 23
- <http://www.biosalc.com.co/biosalc-colombia.html>
- <http://www.pbinsight.com/welcome/mapinfo/>
- <http://www.pb.com/software/Data-Mining-and-Modeling/Geographic-Data-Mining-Tools/MapInfo-Professional.shtml>
- <http://www.sap.com/latinamerica/about-sap/index.epx>
<http://www.slideshare.net/rlavi/integracion-sap-con-aplicaciones-net>
- <http://www.stt-solutions.com/producto-detalle.php?idProduct=3>