



## 2. OBJETIVOS DEL PFC.

El objetivo del proyecto es el diseño e implementación de dos aplicaciones software, un cliente y un servidor, así como el protocolo de comunicación por el cual se rigen ambas. La finalidad que tienen dichas aplicaciones es la de ser integradas en un sistema de información, que permita proporcionar un servicio de recomendación de itinerarios intermodales para el transporte público de pasajeros. Dicho sistema se valdrá de las ventajas de las nuevas tecnologías para proporcionar a sus usuarios una interfaz sencilla, atractiva y eficaz que oculten toda la complejidad inherente al paradigma de las comunicaciones entre diferentes tipos de redes de transmisión de datos y en tiempo real.

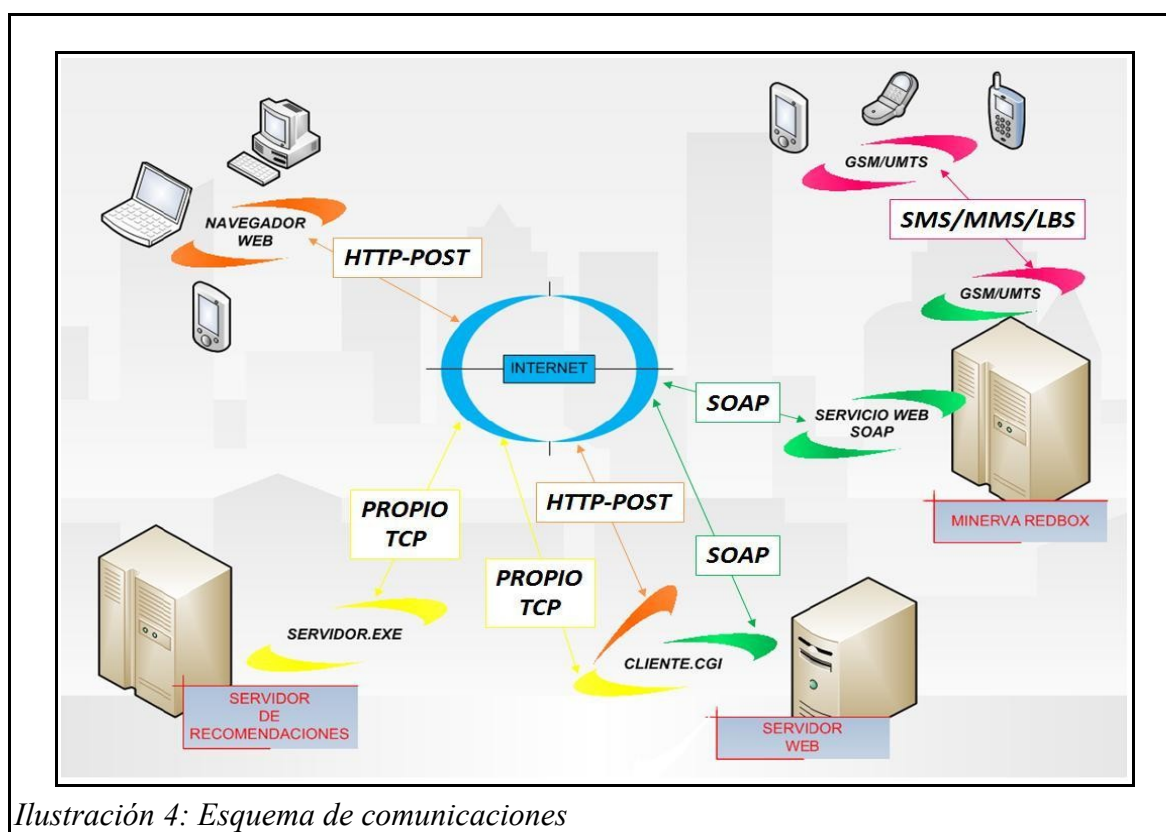


Ilustración 4: Esquema de comunicaciones

En el siguiente esquema se expone de forma simplificada las diferentes redes que intervienen en la transmisión de la información y los principales protocolos empleados en las comunicaciones que pueden llevarse a cabo durante su funcionamiento.

Son ámbito del proyecto los siguientes aspectos:

- Diseño e implementación del CLIENTE y el SERVIDOR.
- Diseño e implementación del protocolo que rige la comunicación entre ambos.
- Elección e implementación del método a seguir para dar un servicio a través de Internet.
- Elección e implementación de una interfaz SOAP, para que el CLIENTE pueda comunicarse a través de ella con el sistema MINERVA-REDBOX.
- Elección e implementación de un servidor web para que el CLIENTE pueda ofrecer sus servicios y realizar las diferentes peticiones.
- Elección y puesta a punto del entorno de desarrollo necesario para la implementación de las aplicaciones, así como para llevar a cabo las diferentes pruebas de funcionamiento de las mismas.

## **2.1. SOLUCIÓN ADOPTADA.**

### **2.1.1. El paradigma Cliente-Servidor.**

Esta arquitectura consiste básicamente en que una aplicación, denominada cliente, realiza peticiones a otro programa, el servidor, que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora, es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Las principales ventajas de esta arquitectura son de tipo organizativo, debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se

ejecuta necesariamente sobre una sola computadora ni es necesariamente un sólo programa.

Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica es generalmente la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. La arquitectura cliente-servidor sustituye así a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

### **2.1.1.1. Características del Cliente.**

En la arquitectura C/S el remitente de una solicitud es conocido como Cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo maestro o amo).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interacciona directamente con los usuarios finales mediante una interfaz gráfica de usuario.

### **2.1.1.2. Características del Servidor.**

En los sistemas C/S el receptor de la solicitud enviada por cliente se conoce como servidor. Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un

papel pasivo en la comunicación (dispositivo esclavo).

- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interaccionen directamente con los usuarios finales.

### **2.1.2. Protocolo base para las comunicaciones.**

El protocolo básico sobre el que desarrollan todas las comunicaciones de nuestro sistema será el tandem formado por la pareja TCP/IP, el cual se ha convertido prácticamente en un estándar de las comunicaciones entre computadoras gracias a su uso exhaustivo. Por tanto, TCP/IP será la base sobre el que se apoye el protocolo definido para que se comuniquen CLIENTE y SERVIDOR. Dicho protocolo se expondrá más adelante en el apartado, Desarrollo del Proyecto.

El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN). TCP/IP fue desarrollado y demostrado por primera vez en 1972 por el Departamento de Defensa de los Estados Unidos, ejecutándolo en ARPANET, una red de área extensa de dicho departamento.

#### **2.1.2.1 TCP.**

TCP (Transmission-Control-Protocol, en español Protocolo de Control de Transmisión)[17] es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 - 1974 por Vint Cerf y Robert Kahn.

Muchos programas dentro de una red de datos compuesta por computadoras pueden usar TCP para crear conexiones entre ellos a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

TCP es un protocolo de comunicación orientado a conexión y fiable del nivel de transporte, actualmente documentado por IETF RFC 793. Es un protocolo de capa 4 según el modelo OSI.

En la pila de protocolos TCP/IP, TCP es la capa intermedia entre el protocolo de Internet (IP) y la aplicación. Habitualmente, las aplicaciones necesitan que la comunicación sea fiable y, dado que la capa IP aporta un servicio de datagramas no fiable (sin confirmación), TCP añade las funciones necesarias para prestar un servicio que permita que la comunicación entre dos sistemas se efectúe: libre de errores, sin pérdidas y con seguridad.

Los servicios provistos por TCP corren en el anfitrión (host) de cualquiera de los extremos de una conexión, no en la red. Por lo tanto, TCP es un protocolo para manejar conexiones de extremo a extremo, y como una serie de conexiones de extremo a extremo pueden existir a través de una serie de conexiones punto a punto, estas conexiones extremo-extremo son llamadas circuitos virtuales. Éstas son las características fundamentales de TCP:

- Orientado a conexión: dos computadoras establecen una conexión para intercambiar datos. Los sistemas de los extremos se sincronizan con el otro para manejar el flujo de paquetes y adaptarse a la congestión de la red.
- Operación Full-Duplex: una conexión TCP es un par de circuitos virtuales, cada uno en una dirección. Sólo los dos sistemas finales sincronizados pueden usar la conexión.
- Error Checking: una técnica de checksum es usada para verificar que los paquetes no estén corrompidos.
- Acknowledgements: sobre recibo de uno o más paquetes, el receptor regresa un

acknowledgement (reconocimiento) al transmisor indicando que recibió los paquetes. Si los paquetes no son notificados, el transmisor puede reenviar los paquetes o terminar la conexión si el transmisor cree que el receptor no está más en la conexión.

- Flow Control: si el transmisor está desbordando el buffer del receptor por transmitir demasiado rápido, el receptor descarta paquetes. Los acknowledgement fallidos alertan al receptor para bajar la tasa de transferencia o dejar de transmitir.
- Servicio de recuperación de Paquetes: el receptor puede pedir la retransmisión de un paquete. Si el paquete no es notificado como recibido (ACK), el transmisor envía de nuevo el paquete.

Los servicios fiables de entrega de datos son críticos para aplicaciones tales como transferencias de archivos, servicios de bases de datos, proceso de transacciones y otras aplicaciones de misión crítica en las cuales la entrega de cada paquete debe ser garantizada.

TCP usa el concepto de número de puerto para identificar a las aplicaciones emisoras y receptoras. Cada lado de la conexión TCP tiene asociado un número de puerto (de 16 bits sin signo, con lo que existen 65536 puertos posibles) asignado por la aplicación emisora o receptora. Los puertos son clasificados en tres categorías: bien conocidos, registrados y dinámicos/privados. Los puertos bien conocidos son asignados por la Internet Assigned Numbers Authority (IANA), van del 0 al 1023 y son usados normalmente por el sistema o por procesos con privilegios. Las aplicaciones que usan este tipo de puertos son ejecutadas como servidores y se quedan a la escucha de conexiones. Algunos ejemplos son: FTP (21), SSH (22), Telnet (23), SMTP (25) y HTTP (80). Los puertos registrados son normalmente empleados por las aplicaciones de usuario de forma temporal cuando conectan con los servidores, pero también pueden representar servicios que hayan sido registrados por un tercero (rango de puertos registrados: 1024 al 49151). Los puertos dinámicos/privados también pueden ser usados por las aplicaciones de usuario, pero este caso es menos común. Los puertos dinámicos/privados no tienen significado fuera de la conexión TCP en la

que fueron usados (rango de puertos dinámicos/privados: 49152 al 65535, recordemos que el rango total de 2 elevado a la potencia 16, cubre 65536 números, del 0 al 65535).

## 2.2. ENTORNO DE DESARROLLO.

### 2.2.1. Lenguaje de Programación C.

Para implementar las aplicaciones CLIENTE y SERVIDOR se ha optado por utilizar el lenguaje C. C [16] es un lenguaje de programación creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix.

C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones. Se trata de un lenguaje débilmente tipificado de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

C tiene las siguientes características de importancia:

- Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de ficheros, proporcionadas por bibliotecas.
- Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado no llevado al extremo (permitiendo ciertas licencias rupturistas).

- Un sistema de tipos que impide operaciones sin sentido.
- Usa un lenguaje de preprocesado, el preprocesador de C, para tareas como definir macros e incluir múltiples ficheros de código fuente.
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Interrupciones al procesador con uniones.
- Un conjunto reducido de palabras clave.
- Por defecto, el paso de parámetros a una función se realiza por valor. El paso por referencia se consigue pasando de forma explícita a las funciones las direcciones de memoria de dichos parámetros. Punteros a funciones y variables estáticas, que permiten una forma rudimentaria de encapsulado y polimorfismo.
- Tipos de datos agregados (estructor) que permiten que datos relacionados (como un empleado, que tiene un id, un nombre y un salario) se combinen y se manipulen como un todo (en una única variable "empleado").

La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia por lo que, si los programas creados lo siguen, el código es portátil entre plataformas y/o arquitecturas. En la práctica, los programadores suelen usar elementos no-portátiles dependientes del compilador o del sistema operativo.

### **2.2.2. El Sistema Operativo y Aplicaciones de Desarrollo.**

Con el fin de sumarnos a la tendencia actual de utilizar código abierto y de libre distribución, y al mismo tiempo disponer de un entorno de desarrollo eficaz, en lo que respecta al lenguaje C, el sistema operativo empleado será una distribución de Linux [17].

Linux es un término genérico para referirse a sistemas operativos similares a Unix basados en el núcleo de Linux. Su desarrollo es uno de los ejemplos más prominentes de software libre; normalmente todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la Licencia Pública General de GNU[1] (GNU GPL) y otras licencias libres.

Las variantes de estos sistemas se denominan distribuciones de Linux y su objetivo es ofrecer una edición que cumpla con las necesidades de determinado grupo de usuarios. Algunas distribuciones de Linux son especialmente conocidas por su uso en servidores y computadores. No obstante, es posible instalar Linux en una amplia variedad de hardware como computadores de escritorio y portátiles, computadores de bolsillo, teléfonos celulares, dispositivos empotrados, videoconsolas y otros.

El nombre Linux viene del núcleo Linux, inicialmente escrito por Linus Torvalds en 1991. El software de sistema viene, normalmente, del sistema operativo GNU, iniciado por Richard Stallman en 1983 y mantenido por la FSF. La contribución de GNU es la razón por la que algunos prefieren llamar GNU/Linux al sistema operativo formado por las herramientas de GNU y el núcleo Linux en su conjunto. La pronunciación, para cualquier idioma, según su autor[5] es muy cercana a como se pronuncia en español: /lí.nux/ o /lnəks/.

En concreto la distribución de Linux empleada ha sido Linux Mandriva. Mandriva Linux (fusión de la distribución francesa Mandrake Linux y la brasileña Conectiva Linux) es una distribución Linux aparecida en julio de 1998 propiedad de Mandriva, enfocada a principiantes o usuarios que se están introduciendo al mundo de GNU/Linux y al Software Libre.

Se distribuye mediante la Licencia pública general de GNU, y es posible descargar su distribución en formato ISO, sus asistentes o sus repositorios. La primera edición se fundamentó en Red Hat Linux (versión 5.1) y escogió el entorno gráfico de KDE (versión 1.0). Desde entonces ha

seguido su propio camino, separado de Red Hat y ha incluido numerosas herramientas propias o modificadas, fundamentalmente dirigidas a facilitar la configuración del sistema. Mandrake (su anterior nombre) también es conocida por compilar sus paquetes con optimizaciones para procesadores Pentium y superiores, incompatibles con versiones más antiguas tales como 386 y 486.

Mandriva Linux, que forma parte del grupo LSB (Linux Standard Base), viene con aproximadamente 19.000 paquetes de software, incluyendo juegos, programas de oficina, multimedia, gráficos, servidores y utilidades de Internet. A diferencia de otras distribuciones, no se basa en un único entorno de escritorio. a pesar de que el entorno de escritorio oficial es KDE, Mandriva proporciona apoyo tanto a este último como a Gnome (GTK) y Xfce, apoyando tanto el desarrollo de programas QT (Kat, buscador integrado en KDE) como GTK (las herramientas de administración de Mandriva están escritas en GTK).

Para la administración de programas, Mandriva utiliza Urpmi y RPMDrake, una herramienta disponible tanto en formato gráfico y como en formato Texto. Urpmi es una herramienta totalmente comparable a APT. Urpmi se encarga de resolver las dependencias de los paquetes rpm, facilitando enormemente la instalación, desinstalación de programas y la actualización del sistema.

Mandriva Linux tiene una gran comunidad de usuarios, que proporcionan ayuda, soporte y software para el usuario de Mandriva Linux. Basado en foros y comunidades agrupadas según idioma. Las características que más nos interesan de la distribución Mandriva Linux empleada son las siguientes:

- Mandriva Linux Free 2008.1 (versión del núcleo 2.6.24.7)
- Compilador gcc Versión: 4.2.3-6mnb1
- Make Versión: 3.81-2mdv2008.1. El propósito de la utilidad make es determinar automáticamente qué piezas de un programa necesitan ser recompiladas, y lanzar las órdenes para recompilarlas.

- GDB Versión: 6.6-5.1mdv2008.1. El propósito de un depurador como GDB es permitir al usuario ver lo que está ocurriendo “dentro” de otro programa mientras que se está ejecutando lo que estaba haciendo en el momento que falló. GDB puede hacer cuatro tipo de cosas (más otras cosas para dar soporte a éstas para ayudarte a capturar errores en el acto:
  - Comenzar tu programa, especificando cualquier cosa que pueda afectar a su comportamiento.
  - Hacer que tu programa se pare al cumplirse las condiciones especificadas.
  - Examinar que ha pasado, cuando tu programa se ha parado.
  - Cambiar cosas de tu programa, de tal manera que puedas experimentar corrigiendo los efectos de un fallo y continuar para descubrir otros.
- gSOAP Versión: 2.7.11-2mdv2008.1.
- XAMPP Linux 1.6.8a.
- Joomla: es un gestor de contenidos web (CMS) de distribución libre y gratuita.
- Emacs Versión: 22.1-7.3mdv2008.1.

A todo ello hay que añadir que la pareja formada por Linux y el lenguaje de programación C, nos proporciona una herramienta muy potente en lo que se refiere a la programación de aplicaciones de red (programación de Sockets). El lenguaje C nos proporciona toda una API para el manejo de sockets en Linux. La unidad fundamental de toda la programación de red en Linux (y en la mayoría de los sistemas operativos) es el socket. De la misma forma que la E/S de archivos permite la conexión con el sistema de archivos, el socket permite establecer conexión con la red. El socket es un medio que el programa utiliza para direccionar, enviar y recibir mensajes. Así de esta forma es posible la comunicación entre aplicaciones remotas o incluso situadas en la misma computadora.

### **2.3. ENTORNO DE DESARROLLO PARA APLICACIONES WEB.**

Como el objetivo es implementar un servicio de recomendación de itinerarios intermodales a través de Internet y de telefonía móvil, el cual se implementa mediante un Servicio Web (más adelante explicaremos con cierta profundidad el concepto de Servicio Web). Para ello necesitamos implementar un entorno de desarrollo de aplicaciones web sobre nuestro sistema operativo Linux. Como uno de los requisitos que nos hemos impuesto desde el principio es el uso de código abierto (open source) utilizaremos un paquete que integra todos los componentes necesarios para el desarrollo y experimentación de aplicaciones web, se trata de XAMPP.

Para ofrecer el servicio a través de Internet realizaremos un formulario en HTML, este pasará sus parámetros a través del método POST a nuestra aplicación en lenguaje C, CLIENTE. Por lo tanto, emplearemos la tecnología CGI (Common Gateway Interface) para implementar el cliente de nuestro sistema.

Gracias al empleo del Servicio Web para implementar la interfaz con la red de telefonía móvil, la aplicación CLIENTE también podrá atender peticiones por SMS y/o MMS añadiendo el código C apropiado.

Por último, para proporcionar un entorno flexible y sencillo para la gestión de contenidos web instalaremos un Sistema Gestor de Contenidos Web (CMS). El Sistema de Gestión de Contenidos Web (el diseño del sitio web partiendo del CMS no es ámbito de este proyecto) para diseñar en el futuro un sitio web con plena funcionalidad del sistema, con un contenido más completo y complejo será Joomla, que es software libre y gratuito.

### 2.3.1. XAMPP.

XAMPP [10] es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas.

Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X. La filosofía detrás de XAMPP es la construcción de una versión fácil de instalar para los desarrolladores que entran al mundo de Apache. Para hacerlo más conveniente para los desarrolladores, XAMPP está configurado con todas las funciones activadas. La configuración por defecto no es buena desde el punto de vista de la seguridad y no es suficientemente segura para un ambiente de producción. Por lo que para sistemas reales es recomendable la instalación de cada uno de los componentes por separado y configurados con los requisitos de seguridad necesarios para su exposición a Internet.

La versión más reciente en el momento del desarrollo de este proyecto y por tanto la empleada es XAMPP Linux 1.6.8a y los componentes que incluye son: Apache 2.2.9, MySQL 5.0.67, PHP 5.2.6 & 4.4.9 & PEAR + SQLite 2.8.17/3.3.17 + multibyte (mbstring) support, Perl 5.10.0, ProFTPD 1.3.1, phpMyAdmin 2.11.9.2, OpenSSL 0.9.8h, GD 2.0.1, Freetype2 2.1.7, libjpeg 6b, libpng 1.2.12, gdbm 1.8.0, zlib 1.2.3, expat 1.2, Sablotron 1.0, libxml 2.6.31, Ming 0.3, Webalizer 2.01, pdf class 009e, ncurses 5.3, mod\_perl 2.0.4, FreeTDS 0.63, gettext 0.11.5, IMAP C-Client 2004e, OpenLDAP (client) 2.3.11, mcrypt 2.5.7, mhash 0.8.18, eAccelerator 0.9.5.3, cURL 7.18.2, libxslt 1.1.8, phpSQLiteAdmin 0.2, libapreq 2.08, FPDF 1.53, XAMPP Control Panel 0.6MD5 checksum: 88a51571a0e37dcd0c5e183113ec485c.

### 2.3.2. Apache.

Se ha decidido usar Apache por varios aspectos que, de forma general, consideramos que todo servidor web debe satisfacer. Algunos de estos aspectos son:

- Es uno de los servidores web mas utilizados a nivel mundial.
- Es un sistema multiplataforma.
- Posee infinidad de paquetes y módulos que nos permiten trabajar con gran cantidad de lenguajes de programación web, así como intérpretes de SQL y otras funciones.
- Permite transacciones seguras mediante SSL (Secure Socket Layer).
- Contiene soporte para Hosts virtuales.

Apache [18] es, hoy por hoy, uno de los servidores web mas utilizados a escala mundial, en gran parte se debe a que es Freeware bajo licencia GNU, y en parte también se debe a su robustez y sus múltiples posibilidades. Son ya muchas las empresas que gestionan su propio host (ya sea para Intranets o Internet) mediante un servidor Apache, lo cual les permite ahorrar las tasas de hosting privado o público que ofrecen numerosas empresas dedicadas a este tipo de servicios. Con Apache cualquiera puede montar su propio servidor Web y hacer uso de él.

El hecho de ser multiplataforma, es decir, el hecho de que soporte infinidad de sistemas operativos distintos entre si, es también un gran aliciente para su enorme proliferación. Hay versiones de Apache para los sistemas operativos mas usados (Windows, Linux, Unix, Solaris, Mac...).

Otra de las características más atractivas de este servidor, es que esta continuamente añadiendo nuevas características y mejoras, lo cual nos garantiza un crecimiento futuro. Además es relativamente fácil de configurar, puesto que solo existe 1 fichero de configuración.

### 2.3.3. HTML.

Tal como se ha comentado en apartados anteriores, se desea que el usuario final tenga la posibilidad de realizar peticiones por medio de un formulario web. Los formularios son posiblemente la herramienta más utilizada en Internet para obtener datos e información acerca de la gente que navega por un sitio web. La idea de los formularios es recolectar información online en la interacción con el usuario y luego ejecutar una determinada acción con la misma, que podría ser por ejemplo, quizás el caso más utilizado, un formulario de venta que el usuario completa y luego es enviado vía email al vendedor en forma encriptada. En este caso, la información recolectada por el formulario se le pasará a la aplicación CLIENTE. Y como es lógico la información que el usuario introducirá en el formulario serán los parámetros a partir de los cuales se calcularán las recomendaciones de transporte intermodal. Para realizar dicho formulario emplearemos el código HTML, a partir del cual se generan todas las páginas web.

HTML, siglas de HyperText Markup Language [7, 8, 9] (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML se compone de varios componentes vitales, incluyendo elementos y sus atributos, tipos de data, y la declaración de tipo de documento. Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo: <nombre-elemento>) y una etiqueta de

cierre (por ejemplo: `</nombre-elemento>`). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo: `<nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>`). Algunos elementos, tales como `<br>`, no tienen contenido ni llevan una etiqueta de cierre.

El marcado estructural describe el propósito del texto. Por ejemplo, `<h2>Golf</h2>` establece a "Golf" como un encabezamiento de segundo nivel, el cual se mostraría en un navegador de una manera similar al título "Marcado HTML" al principio de esta sección. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores web han estandarizado el formato de los elementos. Un formato específico puede ser aplicado al texto por medio de hojas de estilo en cascada (CSS).

El marcado de formato describe la apariencia del texto, sin importar su función. Por ejemplo, `<b>negrita</b>` indica que los navegadores web visuales deben mostrar el texto en negrita, pero no indica qué deben hacer los navegadores web que muestran el contenido de otra manera (por ejemplo, los que leen el texto en voz alta). En el caso de `<b>negrita</b>` e `<i>itálica</i>`, existen elementos que se ven de la misma manera pero tienen una naturaleza más semántica: `<strong>énfasis fuerte</strong>` y `<em>énfasis</em>`. Es fácil ver cómo un lector de pantalla debería interpretar estos dos elementos. Sin embargo, son equivalentes a sus correspondientes elementos de formato: un lector de pantalla no debería decir más fuerte el nombre de un libro, aunque éste esté en itálicas en una pantalla. La mayoría del marcado dedicado al formato de presentación ha sido desechado en HTML 4.0, en favor de Hojas de estilo en cascada (CSS).

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico o cualquier otro editor que admita texto sin formato como GNU Emacs. Existen además, otros programas para la realización de sitios Web o edición de código HTML. Estos programas se les conoce como editores WYSIWYG o What You See Is What You Get (en español: "lo que ves es lo que obtienes"). Esto significa que son editores en los cuales se ve el resultado de lo que se está editando

en tiempo real a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino que una forma un tanto más simple ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML la cual va generando todo el código a medida que se va trabajando.

Combinar estos dos métodos resulta muy interesante, ya que de alguna manera se ayudan entre sí. Por ejemplo; si se edita todo en HTML y de pronto se olvida algún código o etiqueta, simplemente me dirijo al editor visual o WYSIWYG y se continúa ahí la edición, o viceversa, ya que hay casos en que sale más rápido y fácil escribir directamente el código de alguna característica que queramos adherirle al sitio, que buscar la opción en el programa mismo.

### **2.3.4. CGI.**

Para ofrecer el servicio a través de Internet realizaremos un formulario en HTML, este pasará sus parámetros a través del método POST a nuestra aplicación en lenguaje C, cliente.cgi. Por lo tanto, emplearemos la tecnología CGI (Common Gateway Interface) para implementar el cliente nuestro sistema.

Un script CGI es un pequeño programa escrito en general en lenguaje PERL o C (muy populares en ambientes Unix) alojado en los servidores web que facilitan el intercambio y la transferencia de información entre el servidor y el cliente.

CGI son las siglas de Common Gateway Interface [5, 6]. Dicha interfaz estándar permite la transferencia de información entre un servidor web y un programa externo al servidor. Mediante CGI se establece un conjunto de normas que deben seguir los servidores web y las aplicaciones para poder interactuar entre sí. Aunque un CGI se puede programar en cualquier lenguaje soportado por el sistema operativo del servidor, los lenguajes más comunes son C y Perl, ambos son altamente portables entre sistemas operativos. El funcionamiento básico de una aplicación web basada en la

tecnología CGI es la siguiente:

- El navegador web realiza una petición HTTP al servidor web. Esta petición es posible que lleve consigo datos codificados por el navegador, como es el caso de la información introducida a través de un formulario.
- El servidor web recibe la petición, tras analizar la URL detecta que se trata de un programa CGI. Entonces el servidor ejecuta el programa CGI lanzando un proceso hijo.
- El CGI recibirá los datos introducidos por el usuario y otros parámetros en forma de variables de entorno. Llegados a este punto el programa CGI puede actuar de varias formas: realizando un procesamiento directo sobre los datos; interactuar con otras aplicaciones que son las verdaderas destinatarias de los datos recibidos; o ambas cosas.
- El CGI genera su resultado (una página HTML, una imagen, un archivo de sonido,...) y los envía al servidor web.
- El servidor web procesa el resultado del CGI añadiéndole el código necesario para formar un encabezado HTTP correcto y reenvía el resultado al navegador web.
- Finalmente el navegador muestra al usuario el resultado del programa CGI.

### **2.3.5. Joomla.**

Joomla! es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL [19]. Este administrador de contenidos se usa para publicar en Internet e intranets utilizando una base de datos MySQL. En Joomla! se incluyen características como: hacer caché de páginas para mejorar el rendimiento, indexado web, feed RSS, versiones imprimibles de páginas, flash con noticias, blogs, foros, polls (encuestas), calendarios, búsqueda en el sitio web, e internacionalización del lenguaje. Su nombre es una pronunciación fonética para anglófonos de la palabra swahili jumla que significa "todos juntos" o "como un todo". Se escogió como una reflexión

del compromiso del grupo de desarrolladores y la comunidad del proyecto.

La primera versión de Joomla! (Joomla! 1.0.0) fue publicada el 16 de septiembre de 2005. Se trataba de una versión mejorada de Mambo 4.5.2.3 combinada con otras modificaciones de seguridad y anti-bugs. Actualmente los programadores han publicado Joomla! 1.5 estable bajo un código completamente reescrito y construido bajo PHP 5.

Joomla surge como el resultado de una bifurcación o mejora de Mambo, de la corporación Miro de Australia, quien mantenía la marca del nombre Mambo en esa época y el grupo principal de desarrolladores. Joomla nace con esta división el 17 de Agosto de 2005. La corporación Miro formó una organización sin ánimo de lucro con el propósito inicial de fundar el proyecto y protegerlo de pleitos. El grupo de desarrollo reclamó que muchas de las cláusulas de la estructura de la fundación fueron acuerdos previos hechos por el comité directivo de Mambo, el que no tiene la consultoría necesaria de quienes mantienen el proyecto e incluye cláusulas que violan los valores principales del código abierto. El grupo de desarrollo creó un sitio Web que se llamó OpenSourceMatters para distribuir información a los usuarios, desarrolladores, diseñadores Web y a la comunidad en general. En ese momento el líder Andrew Eddie, conocido como "MasterChief", escribió una carta abierta para la comunidad, que apareció en la sección de anuncios del foro público en [www.mamboserver.com](http://www.mamboserver.com).

Al siguiente día, 1000 personas ingresaron al sitio web [opensourcematters.org](http://opensourcematters.org) para expresar su apoyo y estímulo por las acciones ejecutadas por el grupo de desarrollo. El sitio web recibió un aviso de temporalmente fuera de servicio debido al excesivo tráfico. Este evento apareció en [newsforge.com](http://newsforge.com), [eweek.com](http://eweek.com), y [ZDnet.com](http://zdnet.com). Peter Lamont CEO de Miro dio una respuesta en el artículo titulado "The Mambo Open Source Controversy - 20 Questions with Miro". Durante ese periodo de tiempo, Joomla se hizo más grande.

El 1 de septiembre, 2005, el nuevo nombre, "Joomla", que es el Inglés de la ortografía (en árabe:

جملة) jumla palabra que significa "todos juntos" o "en su conjunto".

A continuación se muestra una ilustración en la cual se muestra de forma simplificada y esquemática como deben quedar instalados los elementos vistos para ofrecer un servicio de recomendaciones a través de Internet.

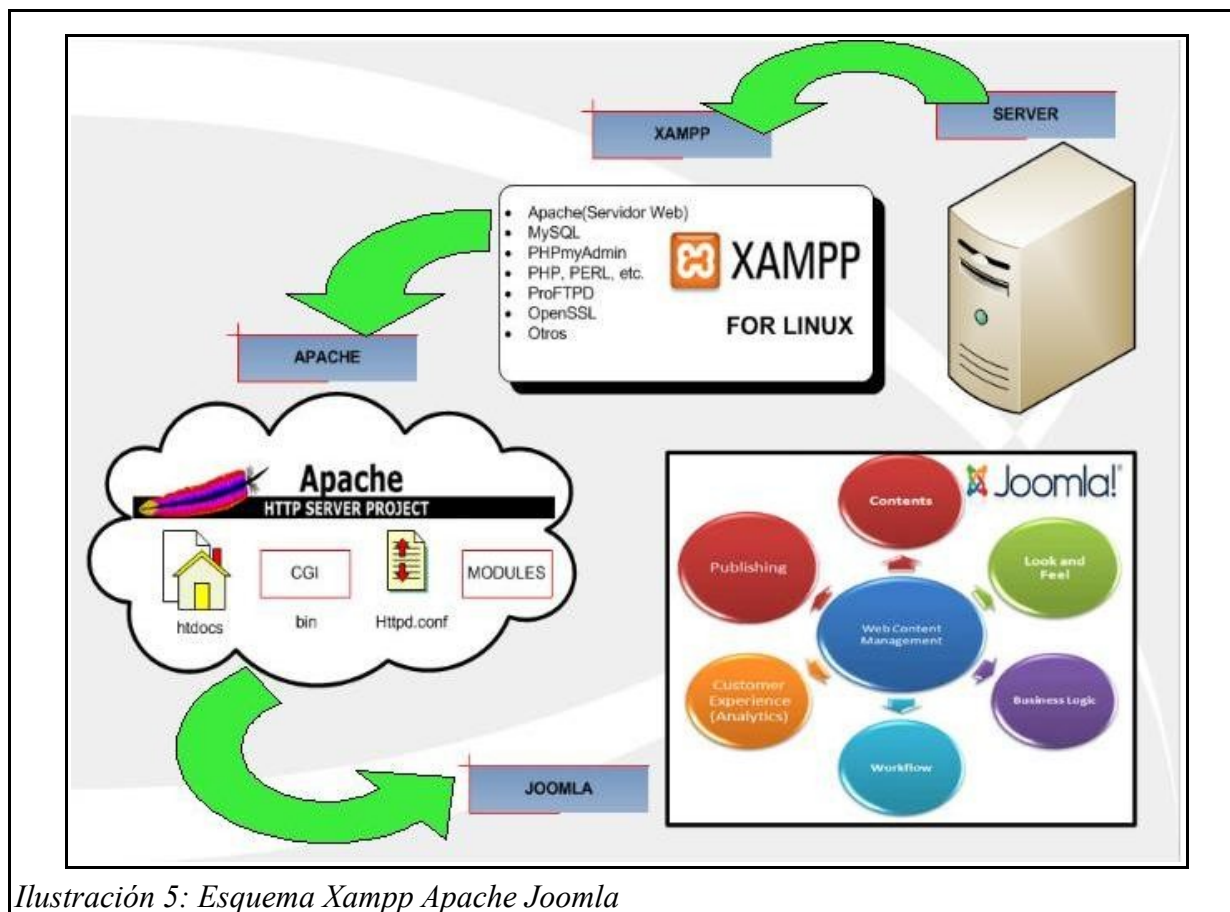


Ilustración 5: Esquema Xampp Apache Joomla

## 2.4. MINERVA-REDBOX, SOAP Y LOS SERVICIOS WEB.

Con la finalidad de implementar la interfaz con la plataforma Minerva-RedBox para que nuestro sistema pueda recibir peticiones de recomendación y enviar las recomendaciones resultantes mediante SMS y/o MMS, utilizaremos SOAP. La tecnología SOAP que nos permite desarrollar interfaces para servicios web. En concreto, para implementar la interfaz SOAP nosotros

utilizaremos el conjunto de herramientas de código abierto que nos proporciona GSOAP.

### **2.4.1. Minerva-RedBox.**

La plataforma Minerva-RedBox [20] permite el envío y recepción de mensajes SMS y MMS, así como el envío de peticiones de localización LBS y la recepción de las respuestas asociadas. Para ello, dispone de un descriptor estándar de servicios web (WSDL), basados en el estándar SOAP (HTTP/XML) con el que puede acceder a una serie de interfaces a través de los cuales podrá realizar envíos y obtener información sobre la plataforma. Dicho descriptor simplifica la integración con sus aplicaciones, ya que las versiones más recientes de los entornos de desarrollo permiten la creación automatizada de bibliotecas a partir de un WSDL (Este es el caso de gSOAP). Por otra parte, los mensajes procedentes de los dispositivos móviles se reenvían a las empresa mediante conexiones HTTP; se puede elegir si los datos se recibirán usando el método POST o el método GET, para cada uno de los canales (En nuestro caso como veremos se utilizará el método POST). A continuación se muestra un diagrama de la plataforma Minerva-RedBox.

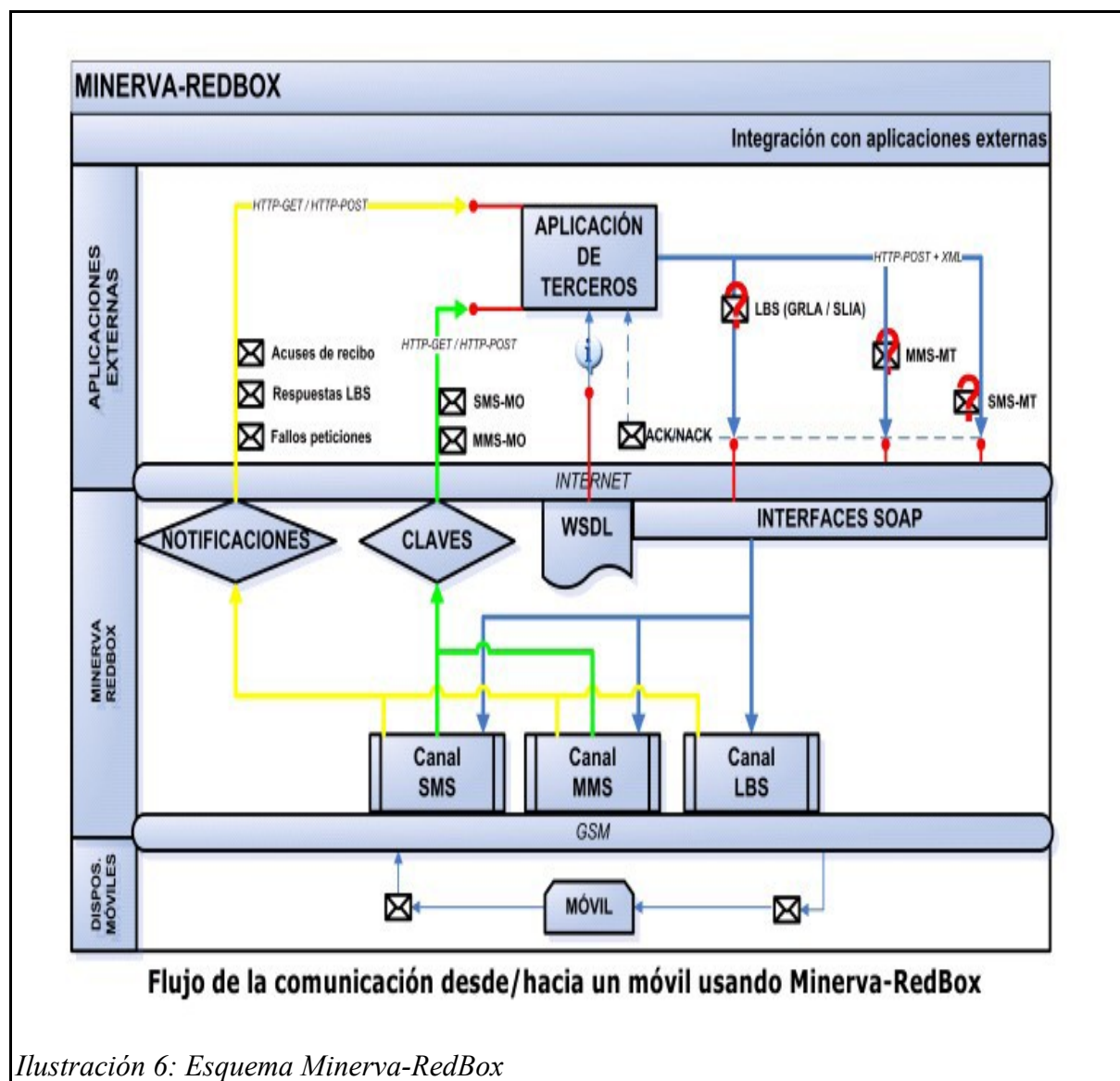


Ilustración 6: Esquema Minerva-RedBox

En nuestro caso particular CLIENTE sería la aplicación de terceros. El objetivo es que CLIENTE reciba una petición de recomendación vía sms (SMS-MO) y solicite al SERVIDOR el cálculo de las recomendaciones. Una vez obtenga las recomendaciones procedentes del SERVIDOR las enviará vía sms (SMS-MT) al terminal móvil que realizó la petición. Además, CLIENTE informará de los posibles errores de ejecución a través de SMS.

### 2.4.2. Servicio Web.

Un servicio web (en inglés Web Service) [12, 13] es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WSI, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Por lo tanto, un servicio web consiste en una función disponible en un servidor conectado al web. Esta función puede consistir en cualquier cosa: realizar un simple cálculo con unos datos que se le envían como parámetro; acceder a una base de datos para recuperar un conjunto de registros, validar la corrección de una información o contrastarla frente a otros datos, etc. El servicio web podrá ser solicitado desde otro programa informático que se ejecute en un ordenador conectado al web. Junto a la solicitud de la ejecución, se pueden enviar al ordenador que ofrece el servicio unos parámetros que el servicio web remoto tomará como base para el cálculo o la función.

La aplicación que actúa como cliente debe conocer la URL del servidor remoto que ofrece el servicio, el nombre del servicio que se solicita, y los parámetros que se deben enviar junto con la llamada al servicio. Estos datos se enviarán mediante HTTP. El servidor que ofrece el servicio web leerá los parámetros que se le han enviado, llamará a un componente o programa encargado de implementar el servicio, y los resultados que se obtengan de su ejecución serán devueltos al servidor que solicitó la ejecución del servicio.

#### **2.4.2.1. Estándares para servicios web.**

Los servicios web se definen a partir de las siguientes especificaciones:

- SOAP (Simple Object Access Protocol).
- WSDL (Web Services Description Language).
- UDDI (Universal Description, Discovery and Integration): a medida que el número de proveedores de servicios web aumente, será necesario disponer de un sistema de referencia que permita localizar estos servicios. Este es el propósito del proyecto UDDI. No obstante, no es uno de los componentes básicos de la tecnología sobre la que se construye el paradigma de los servicios web.

Del mantenimiento de las dos primeras, SOAP y WSDL se encarga el W3C. En el caso de UDDI, se trata de un proyecto en el que participan distintas empresas. Por último, decir que el lenguaje XML constituye la base de todos ellos.

#### **2.4.3. SOAP.**

La especificación SOAP [11] indica cómo se deben codificar los mensajes que circularán entre las dos aplicaciones. Fue definido inicialmente por Microsoft, Userland Software y DevelopMentor, a día de hoy se trata de una especificación mantenida por el W3C que cuenta con el apoyo de otros fabricantes como IBM, HP, Oracle, etc.

La especificación SOAP define dos modelos de mensajes:

- A) Un mensaje que se enviará desde la aplicación cliente a la aplicación servidor, solicitando la ejecución de un método al que se pasan una serie de parámetros.
- B) Un mensaje que se enviará desde la aplicación servidor a la cliente, y que contendrá datos XML con los resultados de la ejecución del método solicitado.

Muchos especialistas han discutido sobre las ventajas e inconvenientes técnicos de SOAP relativos a su utilización junto con otras tecnologías, pudiéndose extraer las conclusiones que se citan a continuación.

### **2.4.3.1. Ventajas del uso de SOAP.**

- ✓ El uso de SOAP sobre HTTP facilita la comunicación a través de próxies y firewalls.
- ✓ SOAP es lo suficientemente versátil como para permitir el uso de diferentes protocolos de transporte. La pila de protocolos estándar muestra a HTTP como protocolo de transporte, pero es posible emplear otros como SMTP.
- ✓ SOAP es independiente de la plataforma.
- ✓ SOAP es independiente del lenguaje.
- ✓ SOAP es sencillo y extensible.

### **2.4.3.2. Desventajas del uso de SOAP.**

- x Debido a lo relativamente detallado y exhaustivo que es el formato de los documentos XML, SOAP puede ser considerablemente más lento que sus tecnologías competidoras como CORBA. Esto no es un problema cuando se intercambian mensajes cortos. Para mejorar este aspecto están surgiendo nuevos modelos de procesamiento XML como es el caso de VTD-XML.
- x En la mayoría de las ocasiones se utiliza HTTP como protocolo de transporte porque se desconoce la posibilidad de usar otros. Ello trae consigo que se implementen servicios web desconociendo como funciona HTTP y podría darse el caso de que éste protocolo no fuera el adecuado para la aplicación que se quiere implementar. Por ello, un buen análisis de los

protocolos y las necesidades de la aplicación web es necesario para elegir un protocolo de transporte adecuado, el cual no tiene porque ser HTTP.

- x Debido al uso de HTTP, un firewall diseñado solo para permitir la navegación web puede que no desempeñe su función de análisis de paquetes de forma eficiente.

### **2.4.4. WSDL.**

Son las siglas de Web Services Description Language. Permite describir los distintos métodos o funciones que están disponibles en un servicio web, así como su signatura, es decir, el número de argumentos o parámetros que se les debe pasar, y el tipo de dato que devolverá la función como resultado. Se establece una equivalencia entre el documento WSDL y un “contrato” que especifica los servicios que el servidor se compromete a ofrecer al cliente, siempre que éste los solicite de la forma adecuada. Los documentos WSDL deben estar disponibles en el servidor web que ofrece los servicios. Como su creación resulta compleja, las distintas implementaciones de SOAP permiten generar estos archivos de forma sencilla, sin necesidad de conocer los elementos y la estructura del esquema XML en el que se basan.

### **2.4.5. Servicio Web/SOAP de Minerva-RedBox.**

Las aplicaciones de terceros que utilicen los servicios de Minerva-RedBox [20] realizarán sus peticiones usando el protocolo estándar SOAP (Simple Object Access Protocol). Las aplicaciones que realicen peticiones a los servicios web de Minerva-RedBox: invocarán una URL (ligeramente diferente para cada tipo de servicio) y pasarán los parámetros embebidos en un XML, usando el método HTTP- POST.

Cada servicio tendrá una serie de parámetros específicos, pero al invocarlos se tendrán que pasar

obligatoriamente:

- El identificador de usuario de la empresa
- La clave de ese identificador de empresa
- El canal por el que se quiere realizar el envío, que será:
  - “SMS” para el envío de los SMS
  - “MMS” para el envío de los MMS
  - “LBS” para realizar un petición de localización

| SERVICIO      | DESCRIPCIÓN                                                                                                                                                                                          | URL MINERVA                                                       |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| enviosms      | Permite el <b>envío de un SMS</b> a un número de teléfono determinado                                                                                                                                | http://193.147.165.85:9002/minerva/envios.asmx/<br>/enviosms      |
| enviomms      | Permite el <b>envío de un MMS</b> a un número de teléfono determinado, <b>con un máximo de 10 adjuntos</b> . No es necesario rellenar todos los adjuntos, sólo los que se necesite enviar            | http://193.147.165.85:9002/minerva/envios.asmx/<br>/enviomms      |
| urlmime       | <b>Detecta el tipo MIME de un fichero presente en la URL que se suministra.</b> Usa la extensión del fichero                                                                                         | http://193.147.165.85:9002/minerva/envios.asmx/<br>/urlmime       |
| urlbase64     | <b>Convierte un fichero presente en la URL que se suministra a una cadena codificada en base64</b>                                                                                                   | http://193.147.165.85:9002/minerva/envios.asmx/<br>/urlbase64     |
| enviolbs      | Permite enviar una petición para <b>localizar la posición</b> (geolocalización) <b>de un terminal móvil</b> ; las coordenadas devueltas vendrán en formato geográfico (latitud/longitud) o UTM       | http://193.147.165.85:9002/minerva/envios.asmx/<br>/enviolbs      |
| enviolbs_GRLR | Permite enviar una petición de geolocalización inversa, para <b>obtener una dirección</b> (calle, ciudad, etc...) <b>a partir de unas coordenadas</b> en formato geográfico (latitud/longitud) o UTM | http://193.147.165.85:9002/minerva/envios.asmx/<br>/enviolbs_GRLR |

*Ilustración 4: Canales de Minerva-RedBox*

La finalidad de implementar una interfaz SOAP en nuestra aplicación CLIENTE es disponer de una vía de comunicación con la plataforma Minerva-RedBox. Ello permitirá a nuestra aplicación poder recibir peticiones de recomendación de itinerarios a través de mensajes de texto procedentes de un terminal telefónico móvil; y también responder a las mismas con las correspondientes recomendaciones calculadas en el servidor de recomendaciones o los posibles mensajes de error.

Por lo tanto, para desarrollar esta interfaz en lenguaje C utilizaremos una plataforma de desarrollo software que nos permita implementar las funciones necesarias en lenguaje C, sin tener

que adentrarnos en los detalles de SOAP y XML. Dicha plataforma generará el código fuente necesario a partir del descriptor de servicios WSDL, y posteriormente insertaremos este código C en nuestra aplicación.

### **2.4.6. GSOAP.**

GSOAP [4, 21] es una plataforma de desarrollo software de código abierto (open source) que permite la implementación de Servicios Web/SOAP basados tanto en XML-RPC como en SOAP. El conjunto de herramientas de gSOAP ofrece una solución eficiente y transparente en C/C++ para las comunicaciones basadas en XML.

Para ello emplea técnicas de generado automático de código que permiten al programador ahorrar tiempo en la implementación de Servicios Web SOAP/XML en C/C++. Además, simplifica significativamente el uso en aplicaciones basadas en XML mediante el mapeo automático de XML a tipos de datos C/C++. De esta forma, los desarrolladores de aplicaciones no tendrán ya más la necesidad de implementar librerías específicas para el uso de XML en sus aplicaciones.

El conjunto de herramientas gSOAP implementa su código mediante tecnologías de compilado que permiten mapear los esquemas XML en código C/C++ y viceversa. Además la portabilidad del código a un número amplio de plataformas está garantizado.

#### **2.4.6.1. Las ventajas de gSoap.**

- Portabilidad: gSOAP está soportado por la mayoría de las plataformas, incluyendo sistemas reducidos y pequeños SO (por ejemplo: WinCE, Symbian, and PalmOS). La portabilidad ha sido comprobada en Windows (98, XP, Vista), Linux, Unix, Mac OS X, Solaris, HP-UX, AIX, FreeBSD, TRU64, Irix, QNX, y VxWorks.
- Estabilidad: es un software maduro. Su desarrollo y depuración ha tenido lugar durante años

desde 2001. El software es usado por muchos proyectos y productos industriales.

- Paquete todo en uno: su independencia de herramientas y librerías pertenecientes a terceras partes asegura el desarrollo exitoso y fiable de las aplicaciones.
- Código abierto: es open source. Por tanto, existen opciones de licencias gratis y comerciales.
- Soporte para C y C++: soporta tanto ANSI C como C/C++.
- Mapeo exhaustivo de datos XML: gSOAP es solo una herramienta SOAP/XML que proporciona el mapeo puro y nativo de C/C++ a XML y viceversa, sin florituras.
- Conforme a los protocolos estándares de la industria: SOAP 1.1/1.2, WSDL 1.1, UDDI v2.
- Soporta diferentes protocolos de transporte: HTTP/S, TCP, UDP, MIME (SwA), DIME (streaming), MTOM (streaming), HTTP1.0/1.1, IPv4, IPv6, RSS, XML-RPC, WSAddressing, WS-Enumeration, y muchos otros protocolos WS-\*. HTTP soporta HTTP/1.1 POST/GET SOAP/XML con mensajes comprimidos, chunking, keep-alive, logging, y encriptación SSL.
- Seguridad: HTTPS y WS-Security proporcionan autenticación, tokens y firmas digitales.
- Velocidad: la generación del código es rápida.
- Ejemplos: la plataforma incluye ejemplos de demostración.
- Integración con el servidor web: incluye Apache\_mod, IIS, Winlnet, interfaces CGI y FastCGI para integrar tus servicios.
- Mantenimiento y desarrollo activos: el software es rico en características, pero siempre se desean tener más. Por ello, gSOAP está en continuo mantenimiento y desarrollo.

### **2.4.6.2. Usuarios de gSoap**

Adobe Systems, AOL, BEA, Boeing, Cisco Systems, CNR, eBay, Ericsson, Exxon/Mobile, HP, IBM, Intel, Microsoft, Nokia, Pfizer, Siemens, WindRiver, Xerox, y muchos otros.