

2011
2012

Modulo Intranet de Gestión de Horarios de Tutorías y Listas de Asistencia.

Memoria Proyecto Final de Carrera

Titulación: Ingeniería Técnica en Informática de Sistema.

Director: Sergio Sáez Barona.

Código: DISCA-146

Escuela Técnica Superior de Ingeniería Informática





PROYECTO FINAL DE CARRERA.

Director: Sergio Sáez Barona.

Código: DISCA-146.

Título: Módulo intranet de gestión de horarios de tutorías y listas de asistencia.

Material entregado:

- Memoria Proyecto Final de Carrera
- Código de la Aplicación



Contenido

1 - INTRODUCCIÓN	6
1.1 - Motivaciones.....	6
1.2 - Planteamiento Técnico.....	7
2 - ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	9
2.1 - Descripción General De La Aplicación.....	11
2.2.1 - Perspectiva del producto	11
2.2.2 - Funciones del Producto.....	12
2.2.3 - Características del Usuario.....	13
2.2.4 - Restricciones Generales	14
2.2.5 - Supuestos y Dependencias.....	14
2.2 - Requisitos Específicos	15
2.3.1 - Requisitos Funcionales.....	15
2.3.2 - Requisitos de Interfaces Externas	18
2.3.3 - Requisitos de Eficiencia	19
2.3.4 - Obligaciones de Diseño	19
2.3.5 - Atributos.....	21
2.3.6 - Otros Requerimiento	22
3 - ANÁLISIS	23
3.1 - Diagrama de Clases	23
3.2 - Diagramas De Usuarios	26
3.3 - Diagramas de Caso De Usos.....	26
3.4 - Diagrama de Secuencia	28
4 - DISEÑO	30
4.1 - Nivel de Presentación o Interfaz de Usuario.....	31
4.1.1 - Página de Inicio.	31
4.1.2 - Página del Alumno.	32
4.1.3 - Página del Profesor	34
4.1.4 - Diagrama de Navegabilidad	36
4.2 - Nivel de Lógica de Funcionalidad o Aplicación	37
4.3 - Nivel de Acceso a Datos o de Persistencia.....	37
4.3.1 - Diseño Lógico	38
5 - IMPLEMENTACIÓN E INTEGRACIÓN.....	41
5.1 - Tecnologías.....	41
5.1.1 - XHTML	41



5.1.2 - CSS.....	42
5.1.3 - HTTP y Apache, Arquitectura Cliente-Servidor	42
5.1.4 - JavaScript.....	43
5.1.5 - PHP	43
5.2 - Herramientas	44
5.3 - Detalles de la Implementación.	44
5.3.1 - Perfiles de usuario.....	44
5.3.2 - Autenticación de usuarios.....	46
5.3.3 - Especificación y Aclaración en Operaciones de Actividad sobre los Horarios de Consulta.....	47
5.4 - Detalles de la Integración.....	48
6 - EVALUACIÓN Y PRUEBAS	49
6.1 - Evaluación	49
6.2 - Pruebas.....	50
6.2.1 - Validación XHTML y CSS.	50
6.2.2 - Comprobación de Enlaces Rotos.....	51
6.2.3 - Comprobación de la Resolución.....	52
6.2.3 – Pruebas de Uso.....	52
7 - CONCLUSIONES	56
8 - BIBLIOGRAFÍA.....	58
8.1 - Referencias Web	58
8.1.1 - ERS.....	58
8.1.2 - HTML y CSS.....	58
8.1.3 - JavaScript.....	58
8.1.4 - PHP	58
8.1.5 - MOSKitt	58
8.2 - Libros de Consulta.....	58
8.2.1 - HTML y CSS.....	58
8.2.2 - JavaScript.....	59
8.2.3 - AJAX.....	59
8.2.4 - PHP	59
8.2.5 - APACHE	59
8.2.6 - MySQL	59
ANEXO A - REQUISITOS	60
A.1 - Objetivos del Sistema	60
A.2 - Requisitos de Información	61



A.3 - Requisitos Funcionales.....	63
A.3.1 - Definición de Actores.....	63
A.3.2 - Casos de Uso del Sistema. Interfaces de transmisión de información	64
A.4 - Requisitos no Funcionales	69
ANEXO B - TECNOLOGÍAS UTILIZADAS	70
B.1 - ARQUITECTURA CLIENTE/SERVIDOR.....	70
B.2 - AJAX.....	71
B.3 - HOJAS DE ESTILO (CSS).....	73
B.4 - JAVASCRIPT	74
B.5 - PHP	75
B.6 - JSON	77
B.7 - APACHE	78
B.8 - MySQL	79
ANEXO C - SEGURIDAD DEL SISTEMA.....	81
C.1 - HTTPS	81
C.2 - SSL/TLS	82



1 - INTRODUCCIÓN

El Módulo intranet de gestión de horarios de tutorías y listas de asistencia, supone la creación de una herramienta, orientada a la Web, de gestión global de citas para alumnos y profesores, dentro del marco de las Asignaturas y sus Horarios de Consulta, especificados por los profesores que imparten las Asignaturas.

Esto significa fundamentalmente un acceso y consulta a los Horarios de Consulta de las asignaturas impartidas en una Carrera, y su manipulación (agregar, editar y eliminar) por parte del personal autorizado, y la posibilidad de realizar operaciones básicas como inscribirse o borrarse.

Partiendo de estos requisitos mínimos, básicos para una herramienta web de este tipo, se desarrollara nuestro Web hasta abarcar un completo abanico de posibilidades, y que giran en torno a un eje central basado en el fin máximo de la "Sociedad de la Información", que es el de satisfacer a un usuario que busca cierta información o que desea realizar una tarea determinada.

1.1 - Motivaciones

Las motivaciones que me ha llevado a realizar este Proyecto Final de Carrera, se pueden basar en los siguientes puntos: mi experiencia en la realización de este tipo de aplicación y la intención de realizar una aplicación con unas características de interface, de código y de utilización más avanzadas que lo que he hecho.

Estas han sido la razón de emprender la realización de una aplicación, básica y como base, que nos sirva para desarrollar futuras aplicaciones más complejas y de uso más generalizado, además de que sea una aplicación con una calidad, tanto en su tecnología como en su manejo, que nos proporcione experiencia antes de salir de la Universidad y tener que desenvolverse en un entorno laboral.

Las aspiraciones técnicas tienen como fin realizar una aplicación de calidad con un manejo sencillo. Además, de tener que usar como base un lenguaje y una forma de programar hasta ahora no realizada, en parte, y ofreciendo unas opciones graficas hasta ahora no utilizadas en mí caso.

Esto ha supuesto poner a prueba mi capacidad de aprendizaje, así como la capacidad de asimilar una forma de programar y adaptar mi estilo de programación a una nueva forma de programación, y de aplicación que se impone en el panorama web.

Estos factores han supuesto, no un "hándicap" como se podría suponer, sino al contrario, una motivación. Además, el ver como se van cumpliendo los objetivos, paso a paso... es lo que ha ido generando y motivando el interés, no solo por concluir el proyecto, sino también por realizarlo de una manera adecuada a las especificaciones, de un modo riguroso e incluyendo acciones e ideas que me han parecido atractivas para el proyecto.

Como aspiraciones humanas, puedo decir, que el principal reto personal es el desarrollo del proyecto en sí. Poner a prueba mi capacidad para desarrollar un proyecto, cumpliendo con unas especificaciones previstas y de un modo profesional, detallando el proceso de evolución.

Tengo que decir, que si ha habido un par de cosas que han supuesto una losa. La primera ha sido la de realizar el trabajo de manera independiente, ya que esto supone un trabajo extra en momentos del proyecto y a nivel personal en particular, el no tener un equipo con el que se trabaja, conlleva una relajación en los plazos de entrega, una relajación, que supone una demora en el tiempo. La segunda supone una carga añadida a la del proyecto, un doble esfuerzo, esta es la de estar trabajando a la vez que estoy finalizando el proyecto. Esto, unido al primer punto conlleva un esfuerzo complementario, a decir verdad, se convierte en una carga a nivel mental.

Por todo lo explicado anteriormente, entiendo que quizá hay estructuras más adecuadas a la hora de plantear el desarrollo de un proyecto y que ayudan más a los integrantes a llevar a cabo los trabajos que se encomiendan.

Aun así debo añadir que, a pesar de todo, todo esto supone para mí un reto de superación, una barrera que superar para llegar a la meta y conseguir el propósito final. El proceso no es fácil, pero una vez finalizado, el esfuerzo habrá valido la pena.

1.2 - Planteamiento Técnico

Hablando exclusivamente de lo que es la aplicación, lo destacable del proyecto serían los siguientes aspectos:

- Interfaz de usuario atractiva, sencilla e intuitiva.
- Desarrollado pensando en que la interfaz responda con gran velocidad.
- Portabilidad y “futurabilidad” de tecnologías.
- Su desarrollo está basado en tecnologías libres (WAMP).
- Interfaz web, con acceso total desde un navegador web cualquiera.

Se ha puesto especial empeño en una interfaz de usuario que simplifique la interacción entre el usuario y el programa, al mismo tiempo buscando que su diseño sea atractivo y elegante. Entendiendo que hacer una aplicación de gestión de citas, no sirve de nada si no se puede gestionar de una manera clara, eficiente y sencilla. Ya que de lo contrario, puede ser poco utilizada y llegar a no ser válida.

He trabajado bastante en conseguir un entorno gráfico muy cuidado, potente, intuitivo y de muy fácil aprendizaje, pero sin sacrificar las prestaciones ni la calidad del servicio. Cumpliendo con los estándares establecidos por W3C y CSS para separar el contenido de la presentación.

He buscado el modo en que la aplicación se ejecute lo más rápido posible, teniendo en cuenta el código que se puede ejecutar en el PC de usuario y el que se ejecuta en el Servidor. Por lo tanto, el PC que se conecte a la programa, limitara en parte la velocidad de ejecución, otra parte estará limitada por el servidor y una última por el ancho de banda del que se disponga.

Se ha buscado la compatibilidad y portabilidad entre navegadores web, siendo más exacta la representación cuanto más exacto sea el navegador interpretando dichos estándares.



De las tecnologías utilizadas se destaca la portabilidad y lo que se puede llamar “futurabilidad”. Por portabilidad se entiende la capacidad para ser multiplataforma y que, en caso de necesidad, poder cambiar estas manteniendo la funcionalidad. Por “futurabilidad” quiero que se entienda la capacidad de la web para su mantenimiento en el futuro al estar basado en tecnologías que son estándares y en lenguajes establecidos.

El desarrollo de la aplicación, hay que matizar, está basado en tecnologías libres exceptuando el SO (Windows), que no forma parte del desarrollo de la aplicación, pero si es la plataforma sobre la que se ha desarrollado el proyecto. Esto se denomina WAMP (Windows - Apache - MySQL - PHP) teniendo en cuenta que la aplicación en un entorno libre 100% LAMP (Linux - Apache - MySQL - PHP) funcionaria exactamente igual, solo cambiaría la plataforma sobre la que se ejecuta la aplicación. Teniendo en cuenta esto, un desarrollo con tecnologías libres permite implementar la aplicación bajo unos requisitos de presupuesto mínimos, ya que un PC de media capacidad sería capaz de dar el servicio. Esto es así en la mayoría de casos, aunque habría que tener en cuenta el número de alumnos, profesores y registros de citas que se manejan en la BD, ya que esto podría suponer una ralentización del sistema debido a los tiempos de búsqueda y afectar a la aplicación. Por lo que en un determinado momento apostar por hardware profesional como servidores dedicados, no hay que descartarlo. El soporte que la práctica totalidad de los servidores proporcionan en PHP y MySQL está asegurado.

Gracias a esta completa funcionalidad desde un navegador web, es posible el acceso desde cualquier ordenador conectado a internet, y el usuario una libertad completa de acceso, uso o consulta. Esto se traduce en comodidad de cara al usuario.

2 - ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Esta especificación de requisitos pertenece al desarrollo del Proyecto Final de Carrera “Módulo intranet de gestión de horarios de tutorías y listas de asistencia”. Se desarrollará siguiendo las directrices **IEEE Std 830-1998**, para la Especificación de Requisitos Software.

Propósito

El propósito del siguiente apartado consistirá en la especificación de requisitos de software para la aplicación web de gestión de citas en horarios de consulta. Esta especificación será útil para establecer uno prerrequisitos y una base sobre la que empezar el trabajo de realización, así como un documento de información y ayuda para cualquiera que quiera conocer más sobre el proyecto.

Introducción

El siguiente documento describe el proyecto de final de carrera “Módulo intranet de gestión de horarios de tutorías y listas de asistencia” para la Universidad Politécnica de Valencia, en la Escuela Técnica Superior de Ingeniería Informática – Valencia.

El objetivo es el desarrollo de una aplicación con una funcionalidad basada en la gestión de citas, y orientada en gestionar las tutorías de las asignaturas matriculadas.

Las técnicas aplicadas para el desarrollo de la aplicación, serán modelos de dirección del sistema de información enfocados a la Web.

Ámbito

El Módulo intranet de gestión de horarios de tutorías y listas de asistencia consistirá en una interfaz web dinámica que facilitara la consulta y la gestión de la BD de las Citas o Reservas, así como, de los Horarios de Consulta. Esta gestión se corresponde con las posibles Reservas que realice un alumno en las asignaturas en las que ha sido matriculado, y con los Horarios de Consulta que pongan a disposición del alumno los profesores que imparten las asignaturas.

La aplicación permitirá realizar operaciones básicas, como son la realización de la reserva por parte del usuario, así como su eliminación, la gestión de los horarios de consulta (operaciones de crear, eliminar, detalles). La aplicación estará restringida para todo aquel usuario que no sea alumno matriculado y profesor que no imparta docencia.

La principal función es la de que cualquier alumno que este matriculado pueda consultar o informarse del estado de los Horarios de Consulta disponibles por su profesor, y su disponibilidad desde cualquier lugar con una conexión disponible a Internet.

De igual manera, permitirá a los profesores que imparten docencia, una administración remota de los Horarios de Consulta que ponen a disposición del alumno. La Aplicación ofrece un



sistema de avisos, por el cual, ante la eliminación de un Horario de Consulta por parte del profesor, se avisará por medio de Email de dicha acción. Esto permite una coordinación ante posibles suspensiones de Eventos.

Definiciones, Acrónimos y Abreviaturas

- **Web:** La World Wide Web, la Web o WWW, es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador web para extraer elementos de información (llamados "documentos" o "páginas web") de los servidores web y mostrarlos en la pantalla del usuario.
- **Web dinámica:** Existen dos tipos de páginas web, de contenido estático (HTML) y de contenido dinámico que se generan a partir de lo que un usuario introduce en un web o formulario y que utiliza el servidor para construir una web personalizada que envía al cliente.
- **Apache:** Es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc), Windows y otras, implementa el protocolo HTTP/1.1 y la noción de sitio virtual.
- **MYSQL:** MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. Una de las razones para el rápido crecimiento de popularidad de MySQL, es que se trata de un producto Open Source, y por lo tanto, va de la mano con este movimiento.
- **XHTML:** Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje informático diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.
- **SHTML:** Los SSI (Server Side Includes) son directivas insertadas en páginas HTML que nos permiten inserción de contenido generado dinámicamente en nuestras páginas web. Como PHP pero mucho más fácil. Para que los SSI funcionen, las páginas HTML que las incluyen deben ser evaluadas por Apache antes de mostrar su contenido al navegador cliente. Las extensiones empleadas suelen ser .SHTML o .shtml.
- **CSS:** Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal de ordenador usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La W3C (World Wide Web Consortium) es la encargada de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.
- **W3C:** El World Wide Web Consortium (Consortio de la Web), abreviadamente W3C, es una organización que produce estándares para la Web.
- **PHP:** Acrónimo de "PHP: Hypertext Preprocessor". Es un lenguaje de programación de scripts, concebido en el tercer trimestre de 1994 por Rasmus Lerdorf. Se utiliza principalmente para la programación de CGIs para páginas web, destaca por su capacidad de ser embebido en el código HTML.
- **API:** (Application Programming Interface, en español Interfaz de Programación de la Aplicación) es un conjunto de especificaciones de comunicación entre componentes software.



- **HTTP:** Es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan Hyper Text Transfer Protocol, es decir, protocolo de transferencia de hipertexto.
- **Usuario:** Se llamará usuario a cualquier persona que acceda a la Aplicación, ya sea un alumno o un profesor.
- **FNMT (CERES):** (CERTificación ESpañola) que lidera la Fábrica Nacional de Moneda y Timbre, y que en líneas generales, consiste en establecer una Entidad Pública de Certificación, que permita autentificar y garantizar la confidencialidad de las comunicaciones entre ciudadanos, empresas u otras instituciones y administraciones públicas a través de las redes abiertas de comunicación.

Referencias

- [1] IEEE Recommended Practice for Software Requirements Specifications - IEEE Std 830-1998
- [2] The World Wide Web Consortium (W3C) for Standards: <http://www.w3.org/>
- [3] Wikipedia en español: <http://es.wikipedia.org>

Visión General

Lo que sigue en este documento, se encargara de profundizar y describir los conceptos sobre los cuales se desarrolla la aplicación así como los objetivos que se pretenden alcanzar.

El desarrollo del Proyecto Final de Carrera “Módulo intranet de gestión de horarios de tutorías y listas de asistencia” y su Especificación de Requisitos de Software se desarrollará siguiendo las directrices **IEEE Std 830-1998**, para la Especificación de Requisitos Software.

2.1 - Descripción General De La Aplicación

2.2.1 - Perspectiva del producto

La Aplicación se desarrolla en XHTML (con la opción SSI) junto con los lenguajes de programación PHP y JavaScript. Accederá a una BD MySQL para obtener los datos necesarios y generar las páginas personalizadas y con la información que corresponde con cada usuario. Es necesario por lo tanto trabajar con un sistema que nos ofrezca un servidor web con soporte PHP junto con una BD MySQL.

He optado por utilizar una plataforma junto a unas herramientas comúnmente llamadas WAMP (Windows Apache MySQL y PHP) que supone el uso de un SO Windows sobre el que se ejecuta un servidor Apache junto a una BD MySQL, y permite el uso de PHP. La razón de utilizar Windows, una plataforma de pago, es que en esto momentos nos permite la instalación de todo el software necesario para poner en marcha el proyecto. También hay que destacar la comodidad de tenerlo instalado en Windows y no tener que instalar todo desde el principio SO Linux incluido, pero todo lo que se ha hecho funcionaria igual en una plataforma LAMP (Linux Apache MySQL y PHP) siendo en este caso totalmente gratuitos los programas que se han utilizado en el proyecto.

La Aplicación tratará de ser lo más intuitiva y amigable posible de cara a los usuarios, ofreciendo una interfaz atractiva a la vista. Se tratará de implementar la web para que presente las mínimas diferencias en la presentación que hagan un navegador web y otro. En concreto se buscará la compatibilidad con los navegadores más utilizados, estos son: Microsoft IE, Firefox, Opera y Chrome.

La presentación en pantalla deberá mostrar la página, incluidos los elementos que la componen (botones, menús, etc... excepto las listas generadas al consultar la base de datos que si son muy extensas se dividirán en diferentes páginas), sin problemas en una **resolución de pantalla mínima de 1024x768**. La resolución de pantalla cambia según la resolución seleccionada por el usuario para su monitor, por lo que resoluciones de pantalla menores a la resolución mínima, no mostrarán la página completa en la pantalla. Así mismo, en resoluciones de pantalla mayores, la página se expandirá y se adaptará. **Las resoluciones de pantalla manejadas por el sistema oscilan entre los 1024x768 de los monitores de 19", y los 1920x1200 de los monitores de 25"**. Teniendo resoluciones para la Aplicación entre estos parámetros: 19", 22" y 25".

El servidor que mantenga la Aplicación, es necesario que ejecute Apache y MySQL sobre un sistema Windows o Linux (cualquiera). Para un servidor de contenido dinámico que ejecute el intérprete de PHP junto con las consultas a la BD MySQL y que proporcione agilidad al SO, es recomendable trabajar con una máquina Dual Core (Doble Núcleo) o superior, y con al menos 2 GB de memoria RAM. Hay muchas opciones, pero las más comunes serían la arquitectura Intel o AMD, con las opciones que nos ofrecen, es posible alojar los servidores en máquinas con arquitectura diferente, como Sparc, Alpha, etc.

El software necesario ya ha sido nombrado anteriormente: un SO Windows/Linux cualquiera, servidor web Apache con módulo PHP, servidor de bases de datos MySQL, y navegadores para realizar las pruebas (Chrome, Opera, IE 9 y Firefox).

Las comunicaciones entre cliente y servidor de la Aplicación se realizarán a través de Internet sobre TCP/IP, a través del protocolo HTTP. Las comunicaciones se realizan entre el navegador del cliente y el servidor.

Se tendrá en cuenta el consumo de memoria que producen las consultas a la BD junto con la ejecución del código PHP. Por lo tanto, se establecerá un límite usuarios que se podrán conectar de forma simultánea. Este aspecto está relacionado con el hardware del que dispongamos en la máquina, tanto velocidad del procesador como memoria disponible, así como el ancho de banda de la conexión.

2.2.2 - Funciones del Producto

Las funciones que realizará la Aplicación serán las siguientes:

1.- Operaciones sobre la Agenda.

- Insertar Horarios de Consulta.

2.- Operaciones sobre las Asignaturas.

- Seleccionar Asignatura de la Matricula.



3.- Operaciones sobre la Lista de Asistencia.

- Mostrar Alumnos Matriculados en una Asignatura.

4.- Operaciones sobre los Horarios de Consulta.

- Eliminar Horarios de Consulta.
- Añadir Alumno al Horario de Consulta.
- Eliminar Alumno del Horario de Consulta.

5.- Operaciones de Actividad sobre los Horarios de Consulta.

- Recordar al Profesor el Horario de Consulta (Email)*.
- Recordar al Alumno la Inscripción en Horario de Consulta (Email)*.
- Informar al Alumno la Eliminación en Horario de Consulta (Email)*.
- Recordar al Profesor el Horario de Consulta.
- Informar al Alumno la Eliminación de un Horario de Consulta.
- Informar al Alumno de la Eliminación en un Horario de Consulta.

6.- Operaciones de Información.

- Visualizar (Alumno) lista de Reservas en Horarios de Consulta en Asignaturas.(Detalles)

7.- Cambio sobre la Aplicación.

- Inicio de Sesión como usuario Alumno o Profesor.

(*) Véase apartado [5.3.3 - Especificación y Aclaración en Operaciones de Actividad sobre los Horarios de Consulta](#) en **Detalles de la Implementación**.

2.2.3 - Características del Usuario

Habrán dos tipos básicos de usuarios que puedan acceder a la Aplicación: Alumno y Profesor.

En este caso no existe ninguna posibilidad de existir un usuario anónimo como tal, ya que el entorno de uso que se le ha dado a la Aplicación no permite esta posibilidad.

Los Alumnos y Profesores que formen parte de la BD, y que se autentifiquen, podrán realizar las operaciones propias de dicho usuario, y en ningún caso, el Profesor podrá realizar las operaciones de Alumno y viceversa. Estas operaciones ofrecen el manejo y el uso de la Aplicación para cada usuario, de manera que se le permita desarrollar el fin para la que ha sido programada.



Los Administradores, serán los que se encarguen de proporcionar acceso a los usuarios, tanto Alumnos como Profesores, por medio del mantenimiento actualizado de la BD (el mantenimiento actualizado de la BD comprende el alta y baja de Alumnos y Profesores que vayan a hacer uso de esta Aplicación).

El Administrador no es usuario de la Aplicación y su acceso a los datos que contiene la BD no será por medio de esta interfaz. Las operaciones que desempeñe, las realizará directamente en la BD o por medio de una aplicación hecha para tal propósito.

2.2.4 - Restricciones Generales

El Administrador gestiona la BD a nivel básico (añadir, modificar y gestionar registros), pero no podrá en ningún caso modificar la estructura de la BD que será fija. En esta Aplicación, hay que entender que puede formar parte de un Intranet mayor, por lo que he supuesto que la BD que se utiliza en este caso, se crea de una BD general más completa.

Una modificación en la estructura de la BD (debido a la necesidad de añadir/eliminar algún campo), requerirá una actualización en el software. Esto solo lo podrá hacer el Administrador.

Una actividad intensa sobre la Aplicación, podría provocar un acceso inadecuado (lentitud, error en el acceso) debido a la saturación del servidor. Esto puede deberse a los accesos a la BD y la cantidad de registros que contiene. Esto se puede solucionar de varias maneras, desde aumentar la potencia del servidor o aplicar técnicas de trabajo distribuido (poner el servidor MySQL en otra máquina accesible por red, introducir un proxy inverso en la entrada que actúe como acelerador o construir un clúster de servidores web).

El Ancho de Banda del que dispondrá el servidor puede convertirse en un embudo a la hora de responder a las peticiones realizadas por los usuarios. Hay que hacer una elección adecuada y basar esta elección en datos que nos proporcionen una media de accesos producidos en horas punta y al día.

Existirá una política de seguridad en el servidor para evitar posibles intrusiones en el sistema, que puedan crear graves daños en el sistema o en la Aplicación. Una de ellas es la configuración de Apache (servidor web) y el uso de la encriptación, así como el mantenimiento y actualización en el caso en que se descubran fallos de seguridad. Con el servidor de BD MySQL se procederá de la misma forma.

El propio SO dispondrá de un sistema de seguridad para evitar posibles intrusiones (cortafuegos, detectores de intrusos, configuración del sistema, configuración de los usuarios y sus permisos,...).

La realización de BackUp`s (copias de seguridad) serán periódicas y contendrán los elementos que corresponden a la BD, páginas web, configuración y datos del servidor MySQL y la configuración del servidor Apache. También se dispondrá de BackUp del SO.

2.2.5 - Supuestos y Dependencias

En una posible puesta en marcha de la Aplicación a nivel funcional, se podría poner en incluir el sistema de avisos a los usuarios en relación con los Horarios de Consulta, siendo necesario que se disponga de un Servidor de Correo.



Los usuarios (Profesores, Alumnos,...) serán introducidos en la BD por el Administrador Web, Responsable de la Aplicación,..., y en ningún caso, los usuarios podrán inscribirse o introducir datos que no sean los que proporcionan las herramientas de la Aplicación.

2.2 - Requisitos Específicos

2.3.1 - Requisitos Funcionales

Usuario Alumno

Inicio de sesión como usuario autenticado

La Aplicación dispondrá de una opción de inicio de sesión que permitirá a los Alumnos poder autenticarse en la Aplicación. Mediante un código de usuario compuesto por la combinación de entre 1 a 3 vocales y consonantes que componen el nombre y los dos apellidos, y una clave que el Alumno habrá elegido en el proceso de matrícula, el usuario podrá acceder a la Aplicación de Gestión de Citas para Horarios de Tutorías.

Una vez autenticado (comprobada la identidad del Alumno), la Aplicación mostrará la página con la que realizar las reservas y que contiene las opciones de selección de asignatura, así como las demás operaciones que son operativas para este tipo de usuario.

Seleccionar asignaturas de la matrícula

Este tipo de selección, ofrece al alumno la posibilidad de elegir de entre las asignaturas matriculadas y realizar la reserva de horarios de consulta dispuestos por el profesor en la asignatura seleccionada. Esta opción se mostrara en la parte superior derecha en un formato persiana.

Esta información se obtiene una vez el usuario es autenticado. En este caso el sistema accede a la BD con los datos introducidos por el alumno y retorna las asignaturas de las que está matriculado en el formato: COD(X) – Nombre de la asignatura.

NOTA: (X) Se refiere al número de la asignatura cuando se imparte en varias partes.

Sistemas Operativos 1 – SO1

Sistemas Operativos 2 – SO2

Añadir Alumno al Horario de Consulta

Una vez el Alumno ha iniciado la sesión, la Aplicación mostrará la página con la que realizar las reservas y que contiene la opción de selección de asignaturas.

Una vez el Alumno ha seleccionado la asignatura en la que quiere realizar la reserva del Horario de Consultas, aparecerá en el calendario de reservas, los Horarios de Consulta que el profesor ha puesto a disposición del alumno.

Habiendo cumplido estos pasos, el Alumno solo tendrá que realizar un “*doble click*” del botón izquierdo del ratón para realizar la reserva en el Horario de Consulta.



Visualizar (Alumnos) lista de Reservas en Horarios de Consulta en Asignaturas (Detalles)

Una vez el Alumno ha iniciado la sesión, la Aplicación mostrará la página, en la que previa selección de la asignatura que se quiere consultar, se mostraran los Horarios de Consulta disponibles.

Una vez hecho esto y en el calendario (modo de visualización de semana) mostrándose los Horarios de Consulta de la asignatura seleccionada, el Alumno puede seleccionar cualquier Horario de Consulta con un “*click*” en el cuerpo del mismo, entonces le aparecerá la opción de detalles del Horario de Consulta en la parte superior izquierda.

Haciendo un “*click*” sobre este icono, aparecerá en detalle la descripción del Horario de Consulta así como de las reservas que tiene hechas. Mostrará al Alumno con su fotografía y mostrando el nombre y los apellidos junto con la franja horaria que tiene reservada. El resto de Alumnos aparecerán ocultos y solo se verá la franja horaria reservada.

Eliminar Alumno del Horario de Consulta

Una vez el Alumno ha iniciado la sesión, la Aplicación mostrará la página, en la que previa selección de la asignatura que se quiere consultar, se mostraran los Horarios de Consulta disponibles.

En los Horarios de Consulta disponibles aparecerán todos los alumnos que han reservado cita, pero solo en el que se corresponde con el Alumno que se ha identificado, aparecerá un aspa para, haciendo un “*click*” con el botón izquierdo en el aspa, eliminar al usuario del Horario de Consulta.

Requisito fundamental, aunque obvio, para poder eliminar/borrar a un Alumno de un Horario de Consulta, es que este usuario haya realizado una reserva en el Horario de Consulta correspondiente.

Recordar al Alumno la Inscripción en Horario de Consulta

La Aplicación se encargará de recordar a los Alumnos incluidos en un Horario de Consulta, por medio de un email y con 1 día de antelación, de que tiene una reserva en el Horario de Consulta de la asignatura correspondiente y la hora a la que tiene que asistir.

Informar al Alumno la Eliminación en Horario de Consulta

La Aplicación se encargará de informar a los Alumnos incluidos en un Horario de Consulta que sea eliminado, por medio de un email y en el momento mismo en que se produzca este hecho, de que el Horario de Consulta de la asignatura correspondiente y la hora a la que estaba prevista, ha sido eliminado.

Mostrar Alumnos Matriculados en una Asignatura

La Aplicación dispondrá de un botón a través del cual los alumnos podrán visualizar los alumnos matriculados en las asignaturas propias en las que el alumno se ha matriculado.



Usuario Profesor

Inicio de sesión como usuario autenticado

La Aplicación dispondrá de una opción de inicio de sesión que permitirá a los Profesores poder autenticarse en la Aplicación. Mediante un código de usuario y una clave que el Profesor habrá elegido, modificado o asignado, el Profesor podrá acceder a la Aplicación de Gestión de Citas para Horarios de Tutorías.

Una vez autenticado (comprobada la identidad del Profesor), la Aplicación mostrará la página en la que se insertaran los Horarios de Consulta que posteriormente estarán disponibles para los Alumnos, así como las demás operaciones que son operativas para este tipo de usuario.

Insertar Horarios de Consulta

Una vez el Profesor ha iniciado la sesión, la Aplicación mostrará la página en la que el Profesor podrá colocar los Horarios de Consulta. Los Horarios de Consulta que ponga el profesor a disposición de los Alumnos, se asociaran a la Asignatura y Grupo.

La página en la que el profesor colocará los Horarios de Consulta, mostrara como parte principal un calendario de semana de 5 días, y otro calendario más pequeño de mes. Ambos tienen botones para interactuar sobre ellos, estando sincronizados entre sí.

Eliminar Horarios de Consulta

Una vez el Profesor ha iniciado la sesión, la Aplicación mostrará la página inicial en la que se muestra el calendario semanal y le situará en la semana actual. El Profesor podrá visualizar los Horarios de Consulta correspondientes y tiene la opción de desplazarse en el tiempo por el calendario a través de los botones habilitados para ello.

El Profesor puede eliminar los Horarios de Consulta que aún no hayan “caducado”, por medio de un botón con icono de un cubo de basura. Este botón aparecerá en el momento se seleccione el Horario de Consulta.

En los Horarios de Consulta disponibles aparecerán todos los alumnos que han reservado cita, por lo que, en el momento en el que un Profesor elimine un Horario de Consulta, todos los alumnos inscritos serán avisados por medio de un email.

Requisito fundamental, aunque obvio, para poder eliminar/borrar un Horario de Consulta, es que el Profesor haya colocado un Horario de Consulta en el sistema y cumpla las condiciones.

Visualizar lista de Reservas en Horarios de Consulta en Asignaturas (Detalles)

Una vez el Profesor ha iniciado la sesión, la Aplicación mostrará la página inicial en la que se muestra el calendario semanal y le situará en la semana actual. El Profesor podrá visualizar los Horarios de Consulta correspondientes a la primera asignatura activa en el cuatrimestre

Una vez hecho esto y en el calendario (modo de visualización de semana) mostrándose los Horarios de Consulta de la asignatura, el Profesor puede seleccionar cualquier Horario de

Consulta con un “*click*” en el cuerpo del mismo, entonces le aparecerá la opción de detalles del Horario de Consulta en la parte superior izquierda.

Haciendo un “*click*” sobre este icono, aparecerá de un modo más gráfico y en detalle la descripción del Horario de Consulta así como las reservas que los alumnos tienen hechas. Mostrará a los Alumnos con su fotografía y mostrando el nombre y los apellidos junto con la franja horaria que tienen reservada.

Mostrar Alumnos Matriculados en una Asignatura

La Aplicación dispondrá de un botón a través del cual el profesor podrá visualizar los alumnos matriculados en las asignaturas en las que imparte docencia.

Recordar al Profesor el Horario de Consulta y los Alumnos inscritos

La Aplicación se encargará de recordar a los Profesores sus Horario de Consulta, por medio de un email y con 1 día de antelación, de que tiene un Horario de Consulta de la asignatura correspondiente y la hora. También se indicara la cantidad de Alumnos que tiene y los que están incluidos en el Horario de Consulta.

2.3.2 - Requisitos de Interfaces Externas

Interfaces de Usuario

El acceso a la interfaz de usuario de la Aplicación para la Gestión de Citas en Horarios de Consulta se realizara a través de la Web, por lo que se hace imprescindible el uso de un navegador con el que conectarnos al servidor y mostrar la interfaz web de la Aplicación. El diseño de la web será la que defina la interfaz para los usuarios, y que debería ser igual para todos los navegadores.

Todas las consultas y la gestión de las reservas, se realizara a través del navegador. Dependiendo del tipo de usuario que inicie la sesión, la interfaz se verá modificada sensiblemente de acuerdo a cada usuario, aunque el diseño general es común.

Interfaces de Hardware

Los Alumnos y Profesores que quieran acceder a la Aplicación deberán disponer de ordenador persona (PC) o MAC y tener un navegador web instalado. Actualmente casi cualquier ordenador es capaz de satisfacer los requisitos necesarios para poder ejecutar la Aplicación.

La interfaz gráfica requiere una resolución mínima de 1280x1024 para su correcta visualización, por lo que una tarjeta gráfica que permita esta resolución, junto a un monitor de un tamaño mínimo de 17” sería lo básico para el uso de la Aplicación.

Es necesario disponer de una interfaz de red (tarjeta de red) ya sea WIFI, Ethernet, ADSL, Fibra Óptica... que permita el acceso a Internet o LAN y la conexión a la Aplicación.

Interfaces de Software

Los Alumnos y Profesores que harán uso de la Aplicación no necesitarán el uso de un SO y navegador determinado, ya que se podrá visualizar desde cualquier navegador que cumpla con los estándares.

El servidor que hará de hosting para la Aplicación, estará basado en el software de Apache con la ejecución de PHP y la BD MySQL (siempre haciendo uso de software libre), disponibles en diferentes SO como Windows, Linux, BSD... Podremos elegir la plataforma que queramos para la ejecución de Software. En nuestro caso. Será Windows 7 Enterprise Edition 32b.

Interfaces de Comunicaciones

La comunicación entre los Alumnos y Profesores con el servidor de la Aplicación se realizará a través de Internet o de una red local (LAN). Por regla general, los Alumnos se conectarán a través de internet, desde sus casas o algún otro lugar con acceso a internet, por el contrario, los Profesores y el personal de Administración se conectarán desde la red LAN o WIFI de la propia Universidad Politécnica de Valencia.

En ambos casos los protocolos de comunicación son idénticos: se trata de una conexión TCP/IP entre los usuarios y el servidor, y el protocolo de comunicación HTTP entre los navegadores y el servidor web.

2.3.3 - Requisitos de Eficiencia

La Aplicación está pensada para el acceso simultáneo de los Alumnos y Profesores, es decir, que coincidan varios usuarios en el sistema en un mismo intervalo de tiempo. En principio, el servidor Apache en el que se aloja la Aplicación es capaz de proporcionar acceso simultáneo a un gran número de peticiones de acceso. Esta opción se puede configurar desde las opciones que proporciona el propio Apache, permitiendo elegir el límite máximo de conexiones que se permiten de modo simultáneo.

Apache, además de la configuración del número de conexiones, permite configurar otras opciones que afectan al rendimiento del sistema de forma directa como: nº de procesos en caliente, nº mínimo y máximo de procesos,...

En cualquier caso, la Aplicación deberá asegurar un acceso en condiciones óptimas en relación con la carga de la Aplicación y su correcta interacción con los Alumnos y Profesores. En casos en los que no se controlen este tipo de parámetros, la Aplicación podría no funcionar de manera correcta y ocasionar problemas como la ralentización en la navegación de los usuarios.

Nota: Un gran número de peticiones de acceso en un intervalo corto de tiempo podría provocar que el servidor se viera sobrecargado y no pudiera atender a las peticiones entrantes.

2.3.4 - Obligaciones de Diseño

Estándares cumplidos

La Aplicación se ha desarrollado intentando cumplir los estándares o recomendaciones para los lenguajes que se han utilizado.

Estos son:

- Recomendaciones para lenguajes de marcado como el lenguaje de marcas de hipertexto (XHTML), de W3C.
- Recomendaciones para hojas de estilo, especialmente hojas de estilo en cascada (CSS), de W3C.
- Estándares para JavaScript, de ECMA International.

La elección de XHTML sobre otros formatos:

- Compatibilidad con los navegadores.
- Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido...).
- Facilidad de edición directa del código y de mantenimiento.
- Formato abierto, compatible con los nuevos estándares que actualmente están desarrollando el W3C como recomendaciones para futuros agentes de usuario i navegadores.

La razón por la que se ha utilizado CSS, es la de separar la estructura de un documento de su presentación. Por esta razón, en casi todos los casos, la especificación del estilo se ha definido en varios documentos por separado y separadas del documento XHTML al que dan el estilo. Las principales razones de la elección de CSS son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Determinados navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web remoto o incluso deshabilitarla por completo, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser imprimida, mostrada en un dispositivo móvil, o ser leída por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

La utilización de JavaScript como código base utilizado en la Aplicación tal y como se entiende, es decir, la parte que proporciona el servicio, ha sido fundamentalmente, la posibilidad que da este lenguaje de ejecutar el código en el propio PC del usuario, liberando y distribuyendo la ejecución de código por parte del servidor.

- Es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y las ejecuta adecuadamente.
- JavaScript es un lenguaje orientado a eventos. Cuando un usuario pulsa sobre un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante

JavaScript se pueden desarrollar guiones que ejecuten acciones en respuesta a estos eventos.

- Es un lenguaje con una alta compatibilidad con los navegadores, si bien entre IE y los demás navegadores suele haber cierto tipo de incompatibilidad, Microsoft está girando hacia la estandarización.
- JavaScript es el complemento ideal de HTML, al permitir a la página realizar algunas tareas por sí misma, sin necesidad de estar sobrecargando el servidor del cual depende.

Limitaciones del Hardware.

Como se ha comentado, la instalación que mantenga la Aplicación y que ejecute Apache y MySQL sobre un sistema Windows o Linux (cualquiera) junto al intérprete de PHP y las consultas a la BD MySQL proporcionando agilidad, debería ajustarse a una máquina Dual Core (Doble Núcleo) o superior, y con al menos 2 GB de memoria RAM.

Ajustándonos a unos requisitos mínimos, podríamos elegir hasta un Pentium IV con 512MB de memoria RAM, aunque las limitaciones de un sistema como este, son evidentes.

El sistema no requerirá de ningún hardware especial adicional, exceptuando algún elemento para la conexión de red (una controladora de red por ejemplo), junto con una conexión adecuada a Internet.

Los sistemas de los clientes serán de cualquier tipo, con un acceso a internet para poder acceder al portal y la susodicha capacidad de conexión a la red. Evidentemente, deberán disponer de un navegador con el que accederán a la interfaz de la Aplicación.

2.3.5 - Atributos

Seguridad.

La seguridad en la Aplicación a la hora de proteger la privacidad y protección de los datos de los Alumnos y Profesores, es un apartado fundamental de la Aplicación. Por lo que los accesos a la Aplicación estarán protegidos con cifrado de datos y los datos personales almacenados, encriptados.

Las diferentes operaciones que cada usuario puede realizar están sujetas al tipo de usuario que se identifique. Así por ejemplo, ningún Alumno deberá poder añadir o eliminar Horarios de Consulta de la BD. Para asegurar la identidad del usuario se requerirá una cuenta con nombre de usuario y contraseña que le autentificara en la Aplicación y le dará acceso a interfaz.

La información de los usuarios acerca de las cuentas se guardará en la BD, con las contraseñas cifradas. El acceso o autenticación de los Alumnos y Profesores llevará al cliente a una página cifrada mediante el protocolo HTTPS, de modo que sus operaciones estarán protegidas mientras dure su sesión.



Mantenimiento

El servicio de mantenimiento, entendiendo este como la gestión de los Horarios de Consulta y las Reservas que se hagan de estos, lo realizarán los propios usuarios a través de las operaciones que cada uno tiene permitido realizar.

El mantenimiento del servidor Apache, la BD MySQL y los Alumnos y Profesores que están en la BD (que son ellos los usuarios finales de la Aplicación), será llevado a cabo por el departamento de informática, que será el encargado de tener actualizadas las BD y la consistencia de la misma.

El HW dependerá de quien de soporte a la Aplicación, por lo que en este caso dependerá del servicio de informática.

El mantenimiento del SW de la Aplicación se facilitará en la medida de lo posible. El código contendrá comentarios en lo relativo a la funcionalidad de las funciones, además se dispondrá de la ERS.

2.3.6 - Otros Requerimiento

Base de Datos

La Aplicación utilizará una BD en MySQL, donde almacenará toda la información referida a las Reservas realizadas por los Alumnos y a los Horarios de Consulta que pongan a disposición los Profesores, junto con otros datos pertenecientes a los Alumnos y Profesores comunicarse entre sí y recibir los avisos del sistema.

Las consultas a la base de datos se realizarán por parte del servidor web mediante PHP y su API de acceso a bases de datos MySQL.

Servidor de Correo (*)

La Aplicación utilizará un Servidor de Correo a través del cual se comunicarán a los usuarios las Asistencias a una cita y el Recordatorio de la existencia de cita, así como, cualquier modificación que haga el profesor en relación con los Horarios de Consulta. Se mantendrá a los usuarios relacionados con Horarios de Consulta informados ante la eliminación tanto del alumno como del mismo Horario de Consulta, por medio del Sistema de Avisos.

(*) En el proceso de puesta en marcha del servicio, se constata un problema que impide la implementación: Información personal para la configuración del Servidor de Correo de los usuarios. Se decide cambiar el método de interacción por mensajes internos de acceso a través de la Interfaz de la Aplicación.

3 - ANÁLISIS

El análisis que se ha hecho de la Aplicación es un **Modelado UML**, y se compone de los siguientes diagramas:

- Diagrama de Clases
- Diagrama de Usuarios
- Diagramas de Casos de Uso
- Diagramas de Secuencia

Los diagramas ayudan a comprender la Aplicación y proporcionan información específica acerca de la misma. La información que puede obtenerse, va desde una visión global de la Aplicación, las clases que la componen y las funciones que se pueden llevar a cabo (Diagrama de Clases); los usuarios que tomarán parte en la Aplicación (Diagrama de Usuarios); las funciones que podrán realizar cada tipo de usuario y su enlace a otras funciones (Diagrama de Casos de Uso); a la secuencia que sigue la ejecución de una función (Diagrama de Secuencia).

3.1 - Diagrama de Clases

A continuación se muestra el diagrama UML de la Aplicación, como ayuda para entender las funciones que se pueden desarrollar, quién las puede realizar, cómo se pueden llevar a cabo y qué tipos de objetos y elementos interactúan en ella y cómo se relacionan entre sí. Ahora pasamos a explicar las principales características de nuestra Aplicación en base a este diagrama.

Hay un grupo de clases que son fundamentales y que alrededor de ellas se realizan todas las acciones. Estas clases son: **Matricula**, **Asignatura** y **Horarios de Consulta**, y son las que ofrecen opciones e interactúan de un modo principal con todos los demás objetos de la Aplicación.

Ahora se explicará de manera detallada la funcionalidad de cada una de las clases que son representadas en el diagrama UML, al igual que sus métodos, variables y relaciones con otras clases.

-La clase **Matricula** es una de las principales clases y es una de las bases en la que se sostiene nuestra Aplicación. Indica la información inicial a partir de la cual todo comienza y cobra sentido. La matrícula se identifica con un *“id”* único y con un *“cod_asig”* y *“asignaturas”* que la definen. Esta clase dispone de la información que asocia a las clases **Alumno** con **Asignatura**, representando a las asignaturas en las que el alumno se ha matriculado. Existirán tantas matrículas como alumnos matriculados, y el acceso a esta clase está restringido a los gestores del servicio (se imita una situación real).

-La **Asignatura** es otra de las clases principales de la Aplicación y es sobre la que gira todo esto, sin esta clase, no hay aplicación. Asocia a la clase **Profesor** con la asignatura que imparte. Además, desde esta clase se descuelgan otras tres clases más, que son: **Horarios de Consulta**, **Lista de Asistencia** y **Cuatrimestre**. La primera de las clases proporciona el elemento básico de

interacción entre alumnos y profesores. Las otras dos proporcionan información a la Aplicación en relación con las asignaturas, que son los alumnos matriculados en la asignatura y el cuatrimestre al que pertenece *"cuatrimestre"*. En esta última se detallan la *"fecha de inicio"* y *"fecha fin"*, que habilitaran o no las asignaturas mostradas en la interfaz. Se ha establecido que haya tres grupos de docencia por asignatura excepto el PFC, *"cod_asig"* especificara a la asignatura y al grupo al que pertenece.

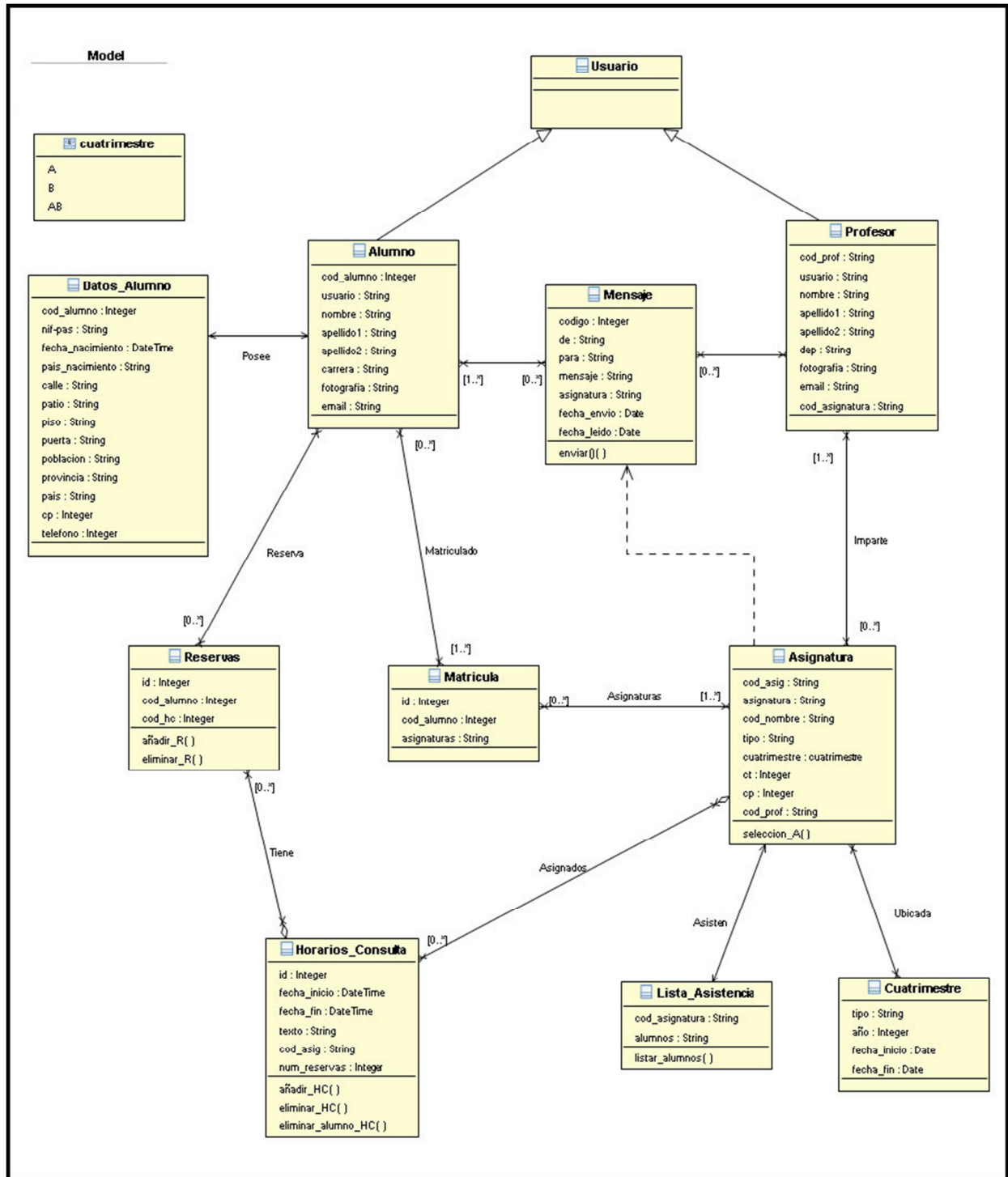
-Los **Horarios de Consulta** y las **Reservas**, son dos clases que van de la mano en la Aplicación y que a través de ellas, todos los usuarios interactúan entre si. El Horario de Consulta hace referencia a la Asignatura *"cod_asig"*, mientras que la Reserva hace referencia al Usuario *"cod_alumno"*. Un horario de consulta se compone de reservas relacionadas a él, es decir, alumnos inscritos en un horario de consulta de una asignatura en la que se han matriculado. Asociando estas clases se cierra el círculo en la comunicación entre **Alumno** y **Profesor**. Los profesores podrán añadir y eliminar horarios de consulta de las asignaturas en las que imparten docencia, y podrán eliminar las reservas de estos horarios. Estas acciones de los profesores se acompañarán de un **Mensajes** de aviso a los implicados. Los alumnos, podrán inscribirse o borrarse de los horarios que habiliten los profesores.

-El mensaje anteriormente citado, esta determinado por la clase **Mensaje**. Es el objeto encargado de avisar y recordar a los implicados en un Horario de Consulta de cualquier acción sobre estos, y de la inminencia del evento en el tiempo. Se hará referencia a la asignatura a través de *"asignatura"* y del remitente por medio del campo *"de"*.

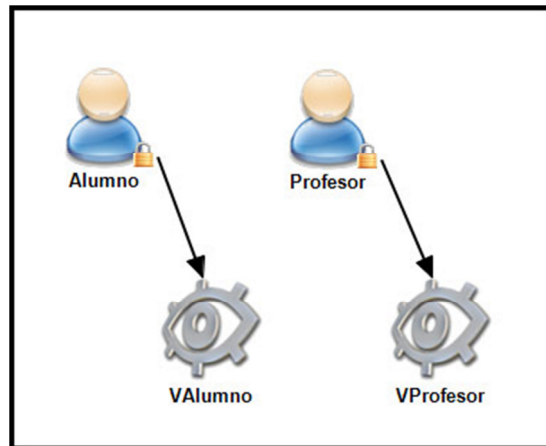
-La clase **Usuario** se encarga de identificar a los usuarios que vayan a utilizar la Aplicación, pudiéndose distinguir dos tipos de usuarios: **Alumno** y **Profesor**. A través de estas clases se definirán los roles que cada usuario tendrá en el sistema. Cada clase de usuario define las acciones de las que dispondrán en la Aplicación pero con unas características, permisos y funciones diferentes para cada tipo

-El **Alumno**, representa a un usuario identificado en el sistema como alumno de la UPV, que se ha matriculado y tiene una o varias asignaturas. Este usuario tendrá un completo control de la clase **Reservas**, haciendo uso de todos sus métodos. Ante un objeto de la clase **Horario de Consulta**, actuara de un modo pasivo, pudiendo solo inscribirse y o quitarse, pero sin tener ninguna acción asociada sobre este tipo de objetos salvo una, *"detalles"* del objeto. Dispondrá de objetos de la clase **Asignatura**, tantos como de las que se haya matriculado, pudiendo acceder a ellas en cualquier momento.

-El **Profesor**, representa a un usuario identificado en el sistema como profesor de la UPV, teniendo una o varias asignaturas en las que imparte docencia. Este usuario tendrá un completo control de la clase **Horario de Consulta**, haciendo uso de todos sus métodos. Ante un horario de consulta en el que se tenga asociados objetos de la clase **Reserva**, solo podrá eliminar este tipo de objetos reportándose al **Alumno** asociado a este objeto un mensaje de anulación/eliminación de la reserva o del horario de consulta. Se le asociaran los objetos de la clase **Asignatura** de las que él es docente.

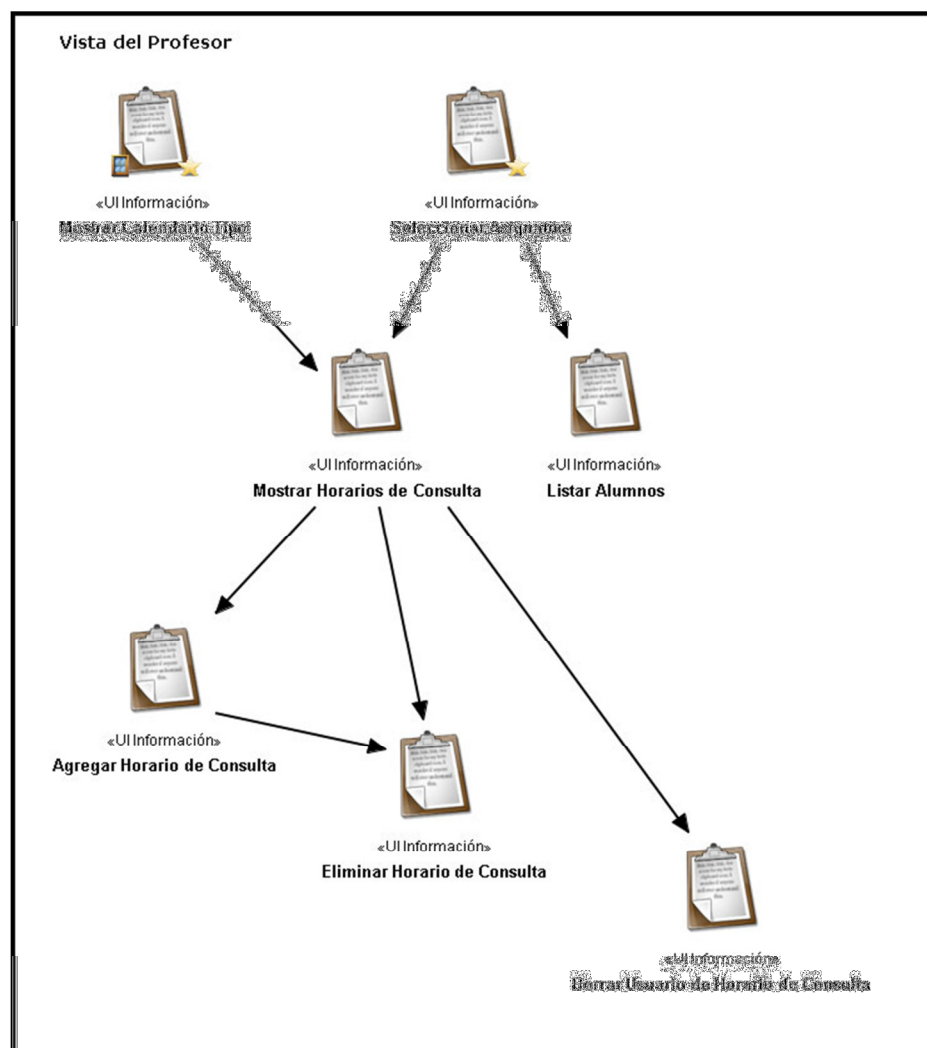


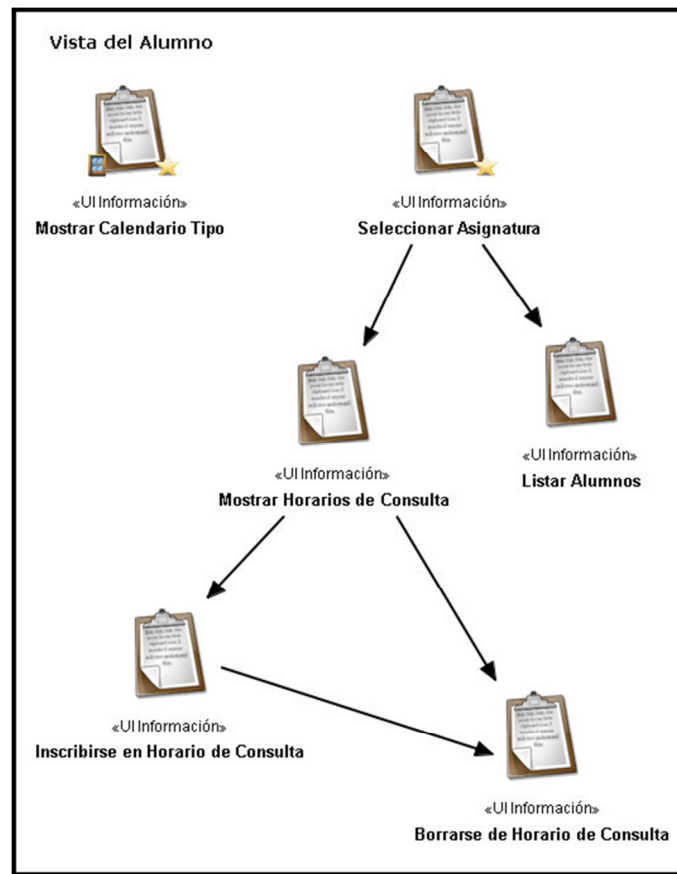
3.2 - Diagramas De Usuarios



3.3 - Diagramas de Caso De Usos

Muestran las acciones que pueden realizar cada tipo de usuario.

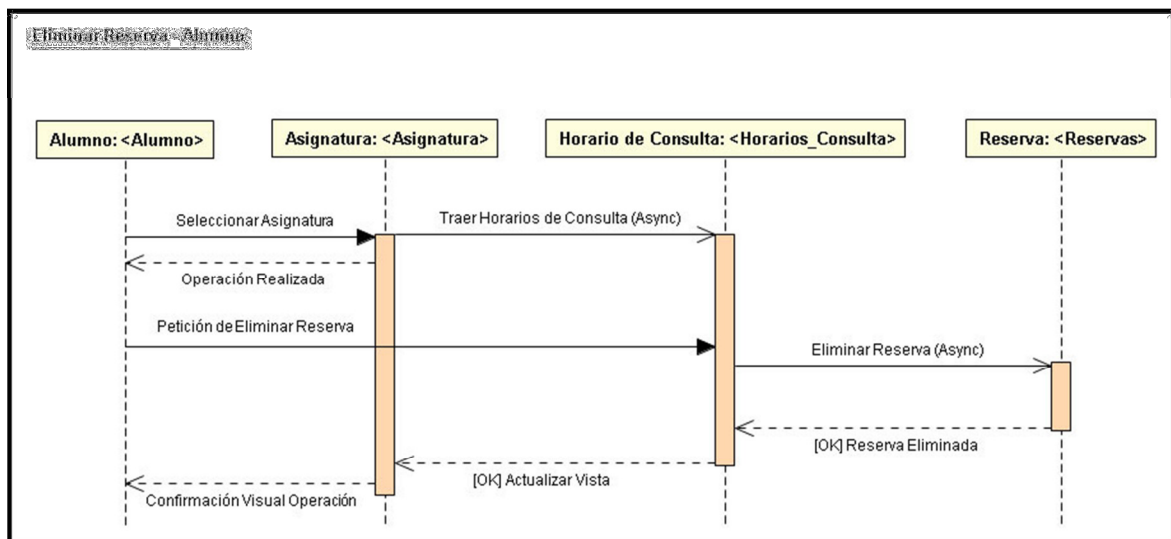
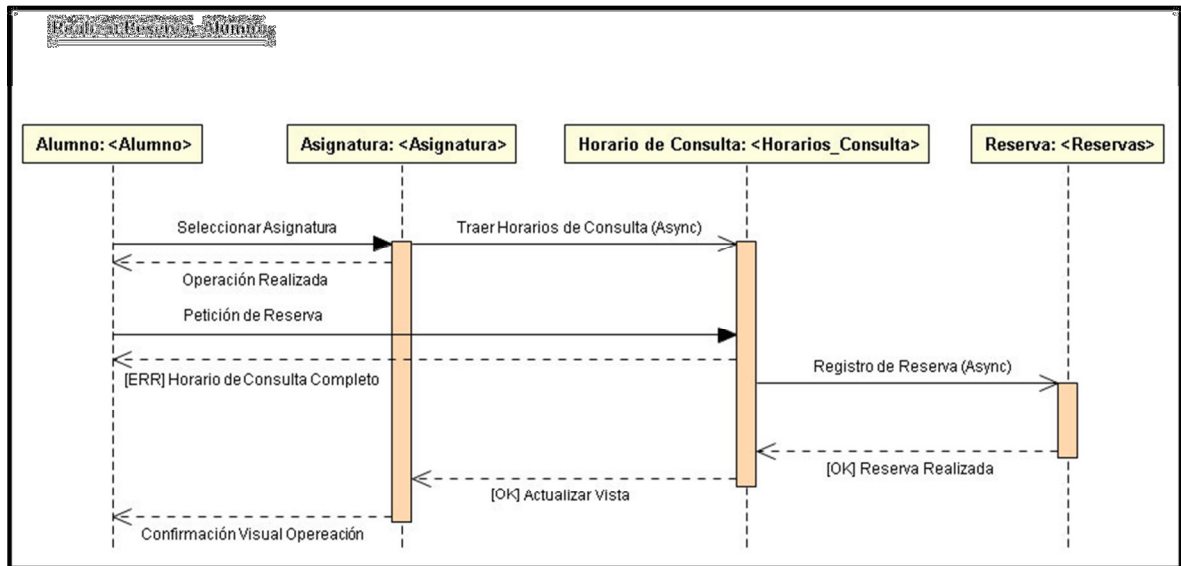


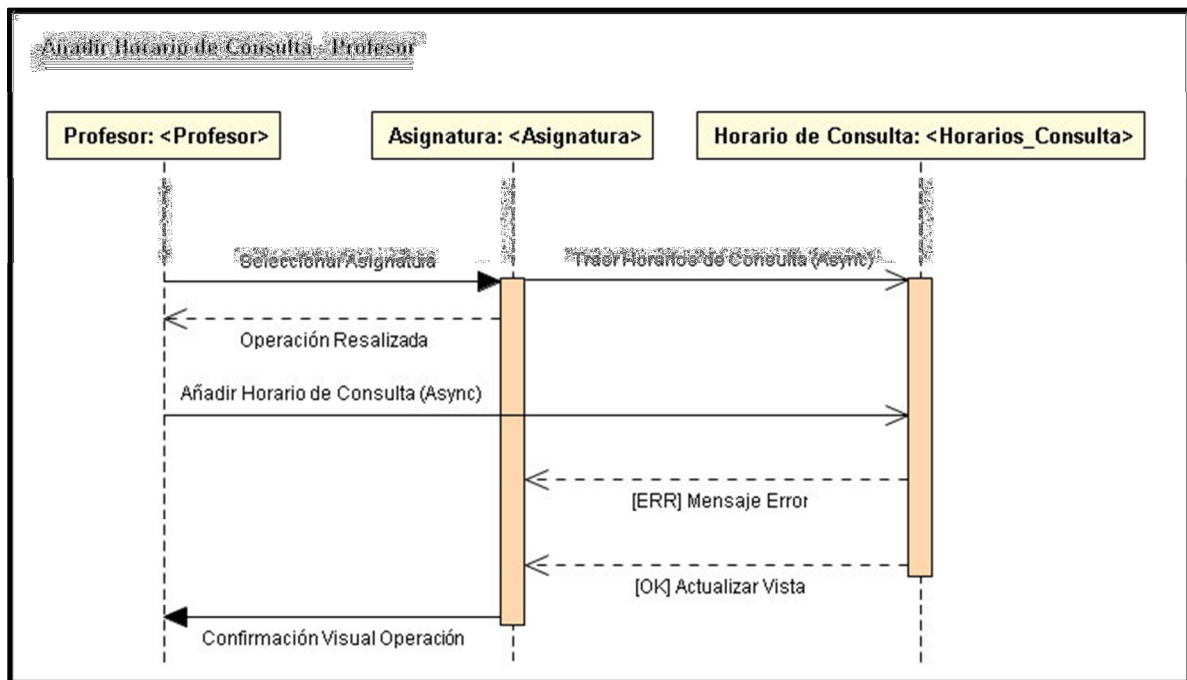
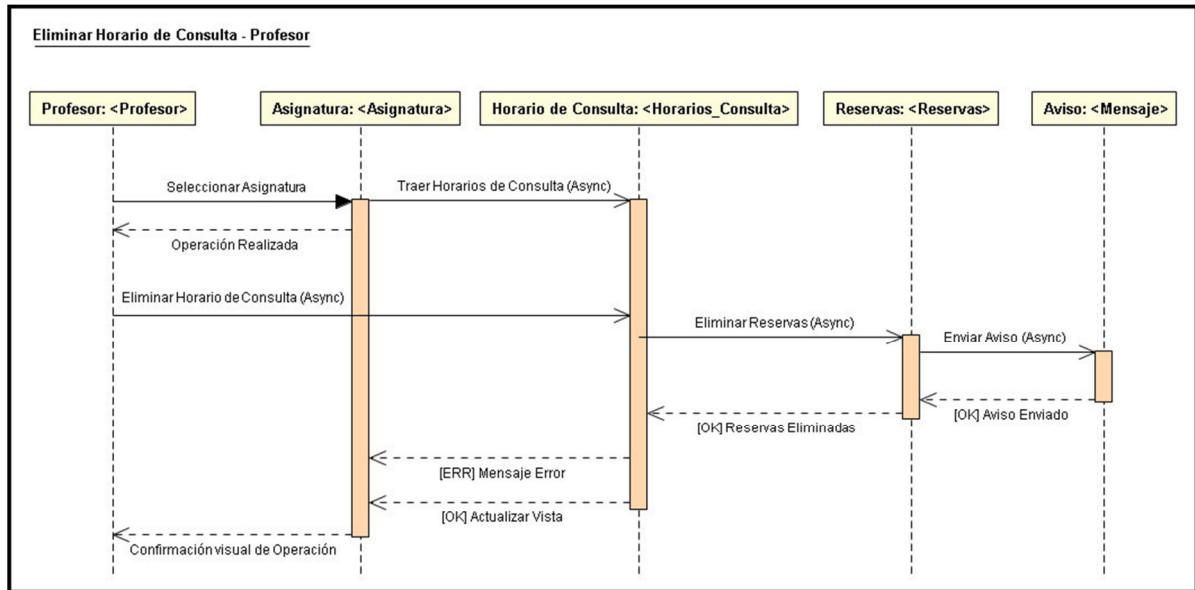


Para más detalles sobre los casos de uso principales consulte el **Anexo A** en el apartado de [Casos de Uso](#), donde están explicados individualmente y con alto nivel de detalle.

3.4 - Diagrama de Secuencia

A continuación se muestran los cuatro diagramas de secuencia mas utilizados, y en los que se detallan los pasos que siguen las operaciones descritas.





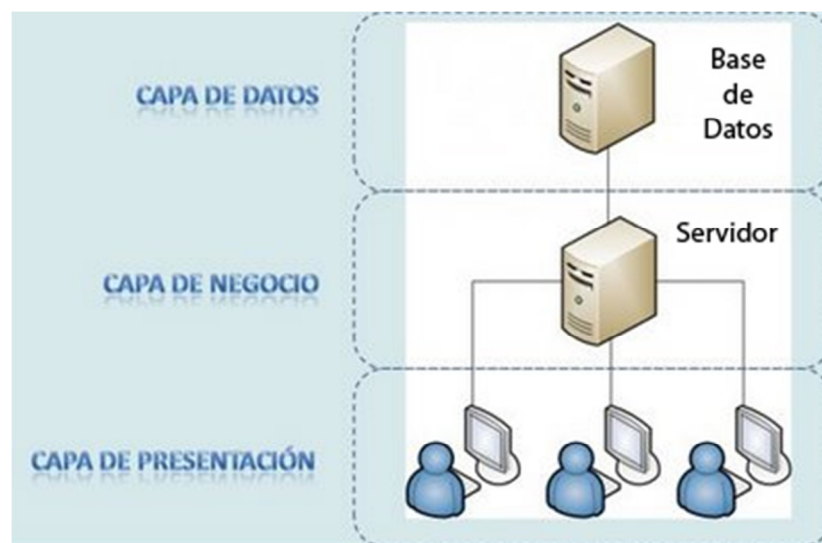
4 - DISEÑO

El desarrollo que se ha seguido para la Aplicación, ha estado basado en el modelo de desarrollo de tres capas. La arquitectura de tres capas se basa en la división del proyecto en las siguientes capas:

- Nivel de presentación o interfaz de usuario.
- Nivel de lógica de funcionalidad o aplicación.
- Nivel de acceso a datos o persistencia.

Con este modelo nos aseguramos de avanzar en la programación del proyecto de una forma ordenada, lo cual beneficia en cuanto a reducción de costos por tiempo, debido a que se podrá avanzar de manera más segura en el desarrollo, al ser dividida la aplicación general en varios módulos y capas que pueden ser tratados de manera independiente y hasta en forma paralela.

Otra característica importante a recalcar es la facilidad para las actualizaciones de la aplicación.



El **nivel de presentación o interfaz de usuario**, está formado por todos los documentos que envía el servidor al navegador, y que éste presenta al usuario de forma que le proporcione la información que ha solicitado y le permita interactuar por medio de enlaces y formularios.

La interfaz, debe facilitar al usuario la interacción con la Aplicación, ser amigable y fácil de utilizar, ya que es este el que se va a encargar de utilizar el sistema y de informar al equipo de desarrollo en caso de que haya algo que mejorar.

El **nivel de lógica de funcionalidad o aplicación**, está formado por un conjunto de librerías que implementan las clases del dominio. Este nivel es el encargado de realizar todas las operaciones a nivel de aplicación.

El **nivel de acceso a datos o persistencia**, lo forman la base de datos y el SGBD, los encargados de almacenar la información de la Aplicación y de permitir el acceso a la misma de forma controlada y segura.

4.1 - Nivel de Presentación o Interfaz de Usuario

La interfaz de la Aplicación ha sido diseñada de modo que su uso resulte sencillo e intuitivo para los Alumnos y Profesores.

El inicio, común a todos los usuarios, es una página sencilla que ofrece el acceso a la Aplicación previa autenticación. La autenticación tiene los campos de usuario y contraseña tanto para Alumnos como para Profesores, por lo que cada tipo de usuario deberá utilizar el que tiene asignado. Esta página se caracteriza por:

- Cabecera en la parte superior de la página y que se extiende a lo ancho la misma, donde se presenta el nombre de la aplicación.
- Cuerpo central que contiene los formularios de acceso de los Alumnos y Profesores. Ambos se encuentran separados por un línea central para diferenciar una parte de la otra, además, en la parte superior de ambas partes se identifica a el tipo de usuario que tiene acceso.

La página principal de la Aplicación se ha diseñado con le intención de que resulte atractiva, limpia e intuitiva, que la presentación de la información no resulte agobiante y que su utilización no sea confusa. La estructura es similar en el caso de Alumnos y Profesores, que se compone de las siguientes partes:

- Cabecera en la parte superior de la página y que se extiende a lo ancho la misma, donde se presenta el nombre de la aplicación.
- Cuerpo central que contiene la Aplicación que interactuara con los usuarios. Esta está compuesta por varios campos:
 - Panel de Navegación: En la parte superior derecha se encuentra este panel, te indicará en que página te encuentras y la ruta de acceso que has seguido hasta llegar a ella.
 - Panel de Selección: En la parte superior izquierda se encuentra este panel. La ventana desplegable te da la opción de seleccionar todas las asignaturas de las que te has matriculado y ver sus Horarios de Consulta, puestos por el Profesor a disposición de los Alumnos.
 - Panel de Información:

A continuación se detallan las interfaces existentes en la Aplicación.

4.1.1 - Página de Inicio.

La página inicial de la Aplicación sirve como presentación, y permite al Alumno o Profesor registrarse en ella. La página está compuesta por el nombre de está (he llamado a la Aplicación "Ágora") junto a la imagen que lo acompaña y el logo, todo en la parte superior de la página.

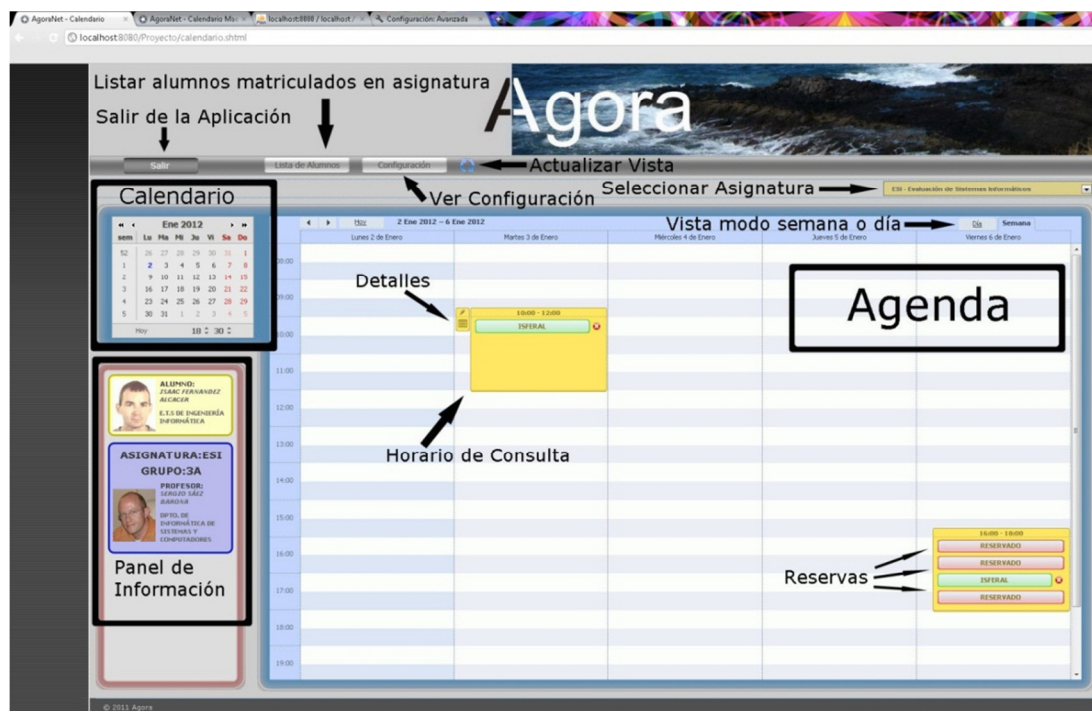
En la parte central de la página se encuentra el formulario en el que los Alumnos y Profesores se autenticaran (cada tipo de usuario tiene su formulario), además de unas notas de ayuda o aclaración para los usuarios.

A continuación se muestra una imagen de la página.

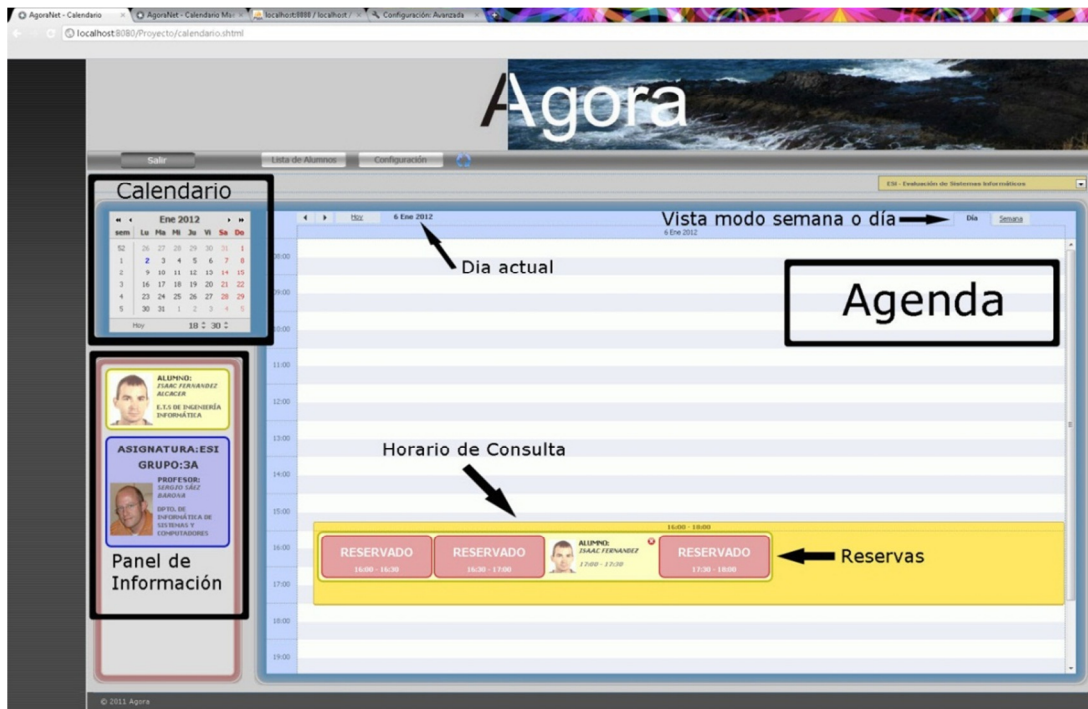


4.1.2 - Página del Alumno.

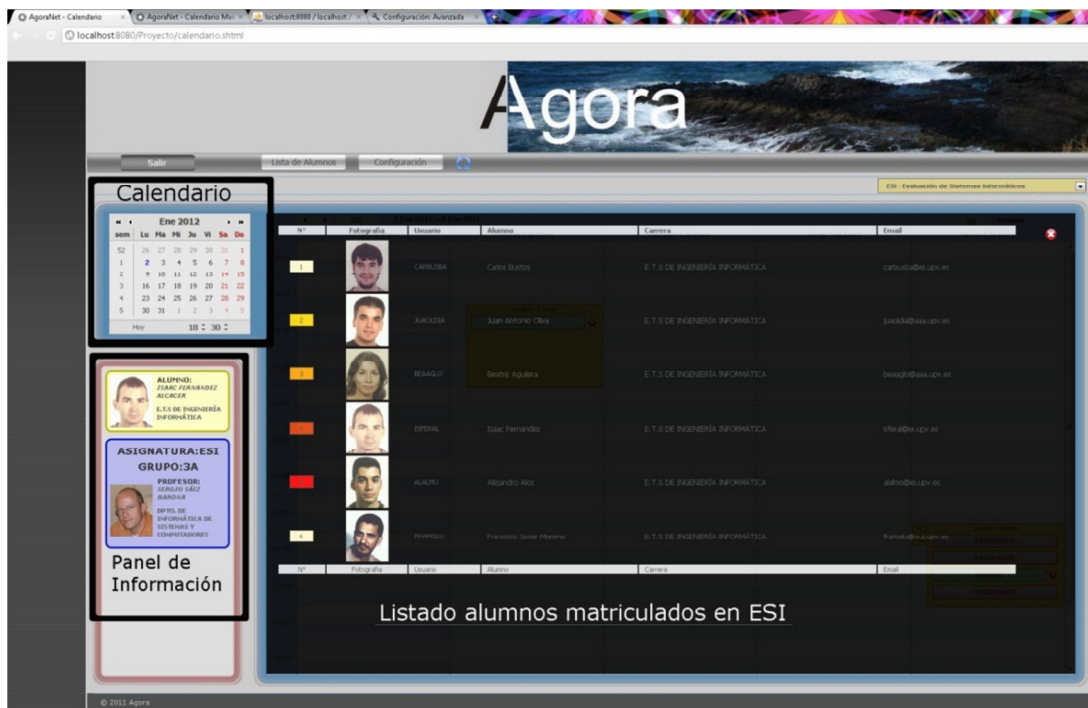
La página de la Aplicación para el Alumno cambia completamente de la inicial, excepto en la cabecera, que es igual en todas las páginas de la Aplicación. Al final del apartado se comentan las características de cada captura.



(Captura 1.)



(Captura 2.)



(Captura 3.)

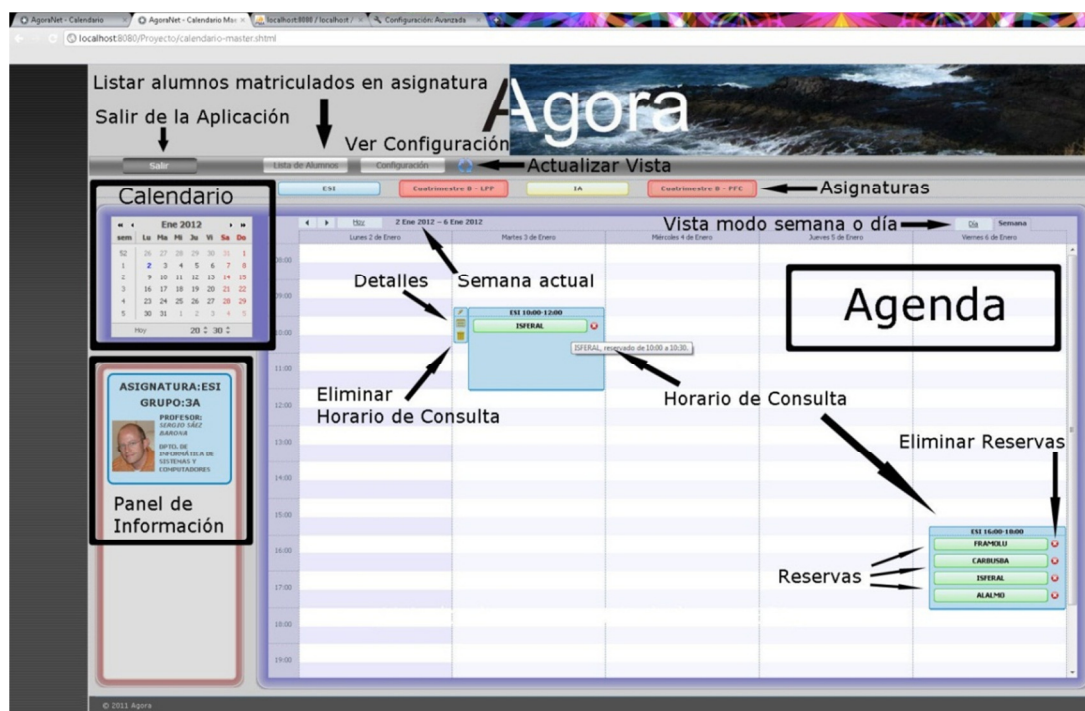


(Captura 4.)

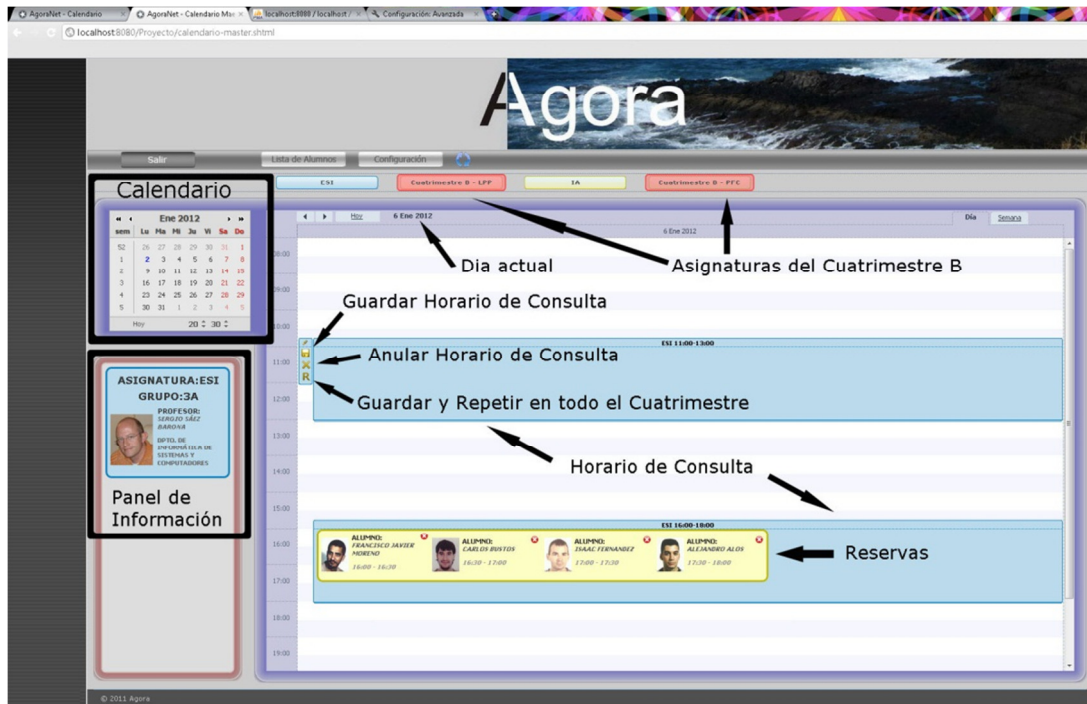
La página inicial del Alumno (captura 1), se caracteriza por tener en la parte superior derecha una ventana desplegable en la que aparecen las asignaturas en las que el Alumno se ha matriculado. Inicialmente, la interfaz se configura con las siguientes características: agenda de 5 días (lunes a viernes) para mostrar los Horarios de Consulta, un calendario para navegar por la agenda de una manera más rápida y un panel de información. En la agenda se cargaran los Horarios de Consulta previa selección de la asignatura. La agenda tiene un segundo modo de presentar los Horarios de Consulta, que es el modo “día” (captura 2), mostrando de este modo los datos de una manera más gráfica. También se presenta por medio de un “slide” los alumnos matriculados en las asignaturas, y también la configuración del sistema para los Alumnos (captura 3). Cuando se hace un “clic” en el icono de detalles de un Horario de Consulta, se muestra en detalle la reserva del Alumnos y los horarios reservados (captura 4).

4.1.3 - Página del Profesor

La página inicial para el Profesor cambia poco a la de los Alumnos. Esta se caracteriza por tener en la parte superior iconos de las asignaturas en las que imparte docencia.



(Captura 1.)



(Captura 2.)

La página inicial del Profesor (captura 1) se configura con las siguientes características: agenda de 5 días (lunes a viernes) para mostrar los Horarios de Consulta, un calendario para navegar por la agenda de una manera más rápida y un panel de información. En la agenda se cargaran los Horarios de Consulta en un primer momento de la primera asignatura activa en la que imparte docencia. La selección de las demás asignaturas se realizara con los iconos relacionados a las mismas y ubicados en la parte superior. La agenda tiene un segundo modo de presentar los Horarios de Consulta, que es el modo “día” (captura 2), mostrando de este modo los datos de una manera más gráfica.

Como en la página del Alumno, el Profesor también tiene la opción de ver los alumnos matriculados en las asignaturas por medio de un “slide”, así como la configuración del sistema para los Profesores. El profesor también tiene acceso a la opción de detalles de un Horario de consulta, el cual muestra en detalle las reservas de los Alumnos y los horarios reservados.

4.2 - Nivel de Lógica de Funcionalidad o Aplicación

El nivel lógico o de aplicación de la Aplicación está formado por un conjunto de librerías creadas que implementan las clases necesarias para el funcionamiento de la misma. Estas librerías realizan diversas funciones relacionadas con operaciones de cálculo, comprobación de condiciones, generación de peticiones a la base de datos, transformación y validación de datos, etc.

Este nivel permite una independencia entre el nivel de interfaz y el nivel de persistencia, de forma que sería posible realizar cambios importantes en dicho nivel (por ejemplo, cambiar el SGBD o incluso substituir la propia base de datos por otro sistema de almacenamiento) sin que esto afectase al nivel de interfaz, simplemente realizando algunos cambios en el nivel lógico para adaptarse al nuevo nivel que tiene por debajo.

En este nivel se ha intentado seguir la programación orientada a objetos, en el que funcionalidades relacionadas son englobadas dentro de clases, y con relaciones de herencia entre ellas. Así pues, en el nivel superior (interfaz) se instanciarán estas clases creando los objetos necesarios para realizar las operaciones.

Existe un amplio conjunto de componentes que conforman las librerías relacionadas con las operaciones sobre la BD. Estos componentes tienen una relación de herencia con una clase padre que les proporciona las herramientas de conexión a la base de datos, de forma que el resto sólo tiene que preocuparse de realizar las operaciones sobre la misma.

Aparte de este gran conjunto, existen varios componentes más que realizan otro tipo de tareas:

- Generación de partes de página comunes a todo el portal para el nivel de interfaz: menús, cabeceras, control de sesiones...
- Funciones generales usadas en casi todo el nivel de interfaz y sin relación entre sí como codificación/decodificación de variables o funciones de cálculo..., así como definición de variables globales a todo el portal.

4.3 - Nivel de Acceso a Datos o de Persistencia

En el nivel de persistencia o de acceso a datos, la Aplicación utilizará una BD relacional, donde se almacenará toda la información referida a las reservas de Horarios de Consulta y los Horarios de Consulta disponibles, Alumnos y Profesores, datos de la Matricula, ...

Para llevarlo a cabo, en la Aplicación es necesario tener una BD bien estructurada y diseñada con una serie de entidades con sus correspondientes atributos como CP (claves primarias), CA (claves ajenas), VNN (valores no nulos)..., y relacionadas entre sí de una manera lógica y con un cierto nivel de eficiencia. Todas las entidades disponen de una CP, pero en ciertas ocasiones esta clave primaria se compone de más de un atributo.

El planteamiento de las entidades (tablas) que son necesarias, son el primer paso en el desarrollo de la BD y del Nivel de Acceso a Datos o Persistencia. Estas son: ALUMNO, ASIGNATURA, CUATRIMESTRE, DATOS ALUMNO, EVENTOS, LISTA_ASISTENCIA, MATRICULA, PROFESORES.

Luego, más adelante, diseñando el modelo de UML, nos dimos cuenta de que eran necesarias otras entidades para poder completar la base de datos y algunas de las anteriores entidades deberían ser débiles, esto es, que no es posible identificarlas sólo por sí mismas, sino que es necesario además de su clave primaria la clave primaria de la entidad de la cual dependen (la entidad EVENTO depende de la entidad ASIGNATURA para identificar de forma única a sus elementos).

4.3.1 - Diseño Lógico

-Alumno.

Campo	Tipo
<u>cod_alum</u>	double
usuario	varchar(8)
nombre	tinytext
apellido1	tinytext
apellido2	tinytext
carrera	tinytext
fotografia	tinytext
email	tinytext

Nombre de la clave	Campo	Único
PRIMARY	cod_alum	Sí
usuario	usuario	Sí

Clave Primaria: cod_alum.

Clave Ajena:

Unico: usuario.

Valor No Nulo: Todos.

-Datos del Alumno.

Campo	Tipo
<u>cod_alumno</u>	int(11)
nif-pas	varchar(11)
fecha_nacimiento	timestamp
pais_nacimiento	varchar(25)
calle	varchar(50)
patio	varchar(4)
piso	varchar(2)
puerta	varchar(4)
poblacion	varchar(20)
provincia	varchar(20)
pais	varchar(25)
cp	int(5)
telefono	int(13)

Nombre de la clave	Campo	Único
PRIMARY	cod_alumno	Sí
nif-pas	nif-pas	Sí

Clave Primaria: cod_alum.

Clave Ajena:

Unico: nif-pas.

Valor No Nulo:

-Matricula.

Campo	Tipo
id	double
cod_alumno	double
asignaturas	text

Nombre de la clave	Campo	Único
PRIMARY	id	Sí
cod_alumno	cod_alumno	Sí

Clave Primaria: id.

Clave Ajena:

Unico: cod_alumno.

Valor No Nulo: Todos.

-Profesor.

Campo	Tipo
cod_prof	varchar(6)
usuario	varchar(8)
nombre	text
apellido1	tinytext
apellido2	tinytext
email	tinytext
fotografia	tinytext
dep	tinytext
cod_asignatura	text

Nombre de la clave	Campo	Único
PRIMARY	cod_prof	Sí
usuario	usuario	Sí

Clave Primaria: cod_prof.

Clave Ajena:

Unico: usuario.

Valor No Nulo: Todos.

-Asignaturas.

Campo	Tipo
cod_asig	varchar(5)
asignatura	text
cod_nombre	varchar(4)
tipo	text
id	double
cod_prof	varchar(6)

Nombre de la clave	Campo	Único
PRIMARY	cod_asig	Sí

Clave Primaria: cod_asig.

Clave Ajena:

Unico:

Valor No Nulo: Todos.

-Eventos.

Campo	Tipo
<u>id</u>	double
fecha_inicio	datetime
fecha_fin	datetime
texto	text
cod_asig	varchar(5)
num_reservas	int(2)

Nombre de la clave	Campo	Único
PRIMARY	id	Sí

Clave Primaria: id.

Clave Ajena:

Unico:

Valor No Nulo: id, fecha_inicio, fecha_fin, cod_asig.

-Cuatrimestre.

Campo	Tipo
<u>tipo</u>	set('A','B')
año	int(4)
fecha_inicio	datetime
fecha_fin	datetime

Nombre de la clave	Campo	Único
PRIMARY	tipo	Sí

Clave Primaria: tipo.

Clave Ajena:

Unico: usuario.

Valor No Nulo: Todos.

-Lista Asistencia.

Campo	Tipo
<u>cod_asignatura</u>	varchar(5)
alumnos	text

Nombre de la clave	Campo	Único
PRIMARY	cod_asignatura	Sí

Clave Primaria: cod_asignatura.

Clave Ajena:

Unico:

Valor No Nulo: Todos.

5 - IMPLEMENTACIÓN E INTEGRACIÓN

Para dar soporte a la Aplicación, realizar el desarrollo de la misma y finalmente, permitir su correcto funcionamiento, se han utilizado elementos SW que son detallados más adelante (programas, lenguajes de programación,...). Todos estos elementos permiten el correcto desarrollo y funcionamiento de la Aplicación, se encargan de la interpretación del código y de enlazar la aplicación con los componentes externos. Todo ello, sobre el HW que se encarga de procesar y ejecutar todas las peticiones.

A continuación se detallan estos elementos.

5.1 - Tecnologías

El Módulo intranet de gestión de horarios de tutorías y listas de asistencia se ha desarrollado como una solución WAMP junto a JavaScript, es decir, una aplicación Web creada utilizando la siguiente combinación de tecnologías:

- Windows (Windows 7), el sistema operativo.
- Apache, el servidor Web.
- MySQL, el servidor de bases de datos.
- PHP, el lenguaje de programación.
- JavaScript, el lenguaje de programación.

Se ha optado por estas tecnologías porque reúnen una serie de características que considero fundamentales:

- La gran cantidad y calidad de documentación existente sobre dichas tecnologías, que permiten un aprendizaje rápido pero profundo sobre la materia.
- La buena interacción de estas tecnologías entre sí.
- El hecho de que sean tecnologías libres y gratuitas, que son constantemente mantenidas por la comunidad de software libre.
- Son ampliamente utilizadas a nivel profesional, y se ha demostrado su alta calidad y productividad.

Las tecnologías Windows, Apache y MySQL, no han sido desarrolladas directamente en nuestra aplicación, son componentes externos a esta.

5.1.1 - XHTML

XHTML es la abreviatura de eXtensible HyperText Markup Language, y es el lenguaje que todos los programas navegadores usan para presentar información en la World Wide Web (WWW).

Este es un lenguaje muy sencillo que se basa en el uso de **etiquetas**, consistentes en un texto ASCII encerrado dentro de un par de paréntesis angulares (<...>). El texto incluido dentro de los

paréntesis nos dará una explicación de la utilidad de la etiqueta. Así por ejemplo la etiqueta `<TABLE>` nos permitirá definir una tabla.

Las etiquetas podrán incluir una serie de atributos o parámetros, en su mayoría opcionales, que nos permitirán definir diferentes posibilidades o características de la misma. Estos atributos quedarán definidos por su nombre (que será explicativo de su utilidad) y el valor que toman separados por un signo de igual. En el caso de que el valor que tome el atributo tenga más de una palabra deberá expresarse entre comillas, en caso contrario no será necesario. Así por ejemplo la etiqueta `<TABLE border=2>` nos permitirá definir una tabla con borde de tamaño 2.

Entre otras cosas, el manejo de estas *etiquetas* nos permitirá:

- Definir la estructura lógica del documento HTML.
- Aplicar distintos estilos al texto (**negrita**, *cursiva*,...).
- La inclusión de hiperenlaces, que nos permitirán acceder a otros documentos relacionados con el actual.
- La inclusión de imágenes y ficheros multimedia (gráficos, vídeo, audio).

5.1.2 - CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en XHTML o XML.

Las hojas de estilo se pueden insertar directamente en el código XHTML o bien en un archivo separado de hoja de estilos, al que el documento principal enlazará. La elección ha sido precisamente esta última opción, con el cual el estilo se centraliza en un único punto, con lo que ganamos en eficiencia, los cambios se limitan a un único archivo y además el tamaño de los documentos HTML se reduce al no contener ya información referente al estilo. Otra ventaja añadida radica en el hecho de que un cambio en la hoja de estilos desemboca en un cambio de estilo general para todo el portal, mientras que de otra forma habría que revisar uno a uno todos los documentos para aplicarles los estilos correspondientes.

Las hojas de estilo en el portal se han empleado precisamente para presentar las páginas web generadas por el servidor de forma estructurada, así como de aplicarle cierto diseño al portal en general para que los navegadores puedan mostrarlo al usuario de forma más agradable, intuitiva y amigable.

5.1.3 - HTTP y Apache, Arquitectura Cliente-Servidor

La comunicación entre la Aplicación Web y los usuarios que harán uso de ella, corresponde a un modelo cliente-servidor. En el Web los clientes demandan los documentos de hipertextos a los servidores mediante el protocolo HTTP. El hipertexto es el contenido de las páginas Web, en este caso HTML, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de páginas Web, y la respuesta del servidor, que incluye dichos documentos.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. En nuestro caso, para posibilitar un acceso con autenticación hemos utilizado el seguimiento de sesiones que ofrece JavaScript: **sessvars.js**.

Las páginas Web son textos ASCII escritos en el lenguaje HTML, que se transfieren entre los servidores Web y los navegadores de los clientes mediante el protocolo HTTP. El desarrollo de aplicaciones Web profesionales conlleva la utilización de tecnologías que permiten recoger información proveniente de clientes y de bases de datos, así como el envío de información personalizada al usuario. Esta comunicación entre cliente y servidor ha avanzado hasta la creación de páginas Web dinámicas, en mi caso utilizando una implementación con PHP y JavaScript.

Aunque los usuarios cuenten con la plataforma y el explorador adecuados, no siempre están dispuestos a introducir componentes ejecutables (JavaScript, Applets, Flash, ActiveX, etc...).

- Las páginas con *scripts* PHP proporcionan un adecuado soporte de procesamiento en el servidor, a la vez que permiten su coexistencia con páginas activas en el cliente. El acceso a las bases de datos es sencillo y potente.
- Las páginas Web generadas mantienen un formato común que se puede mantener de forma sencilla. Además, el código que se manda al usuario es XHTML estándar, con lo que se puede asegurar que la información se visualizará correctamente con independencia de la plataforma y el visualizador del cliente.

5.1.4 - JavaScript

JavaScript es un lenguaje de programación basado en objetos y en eventos de usuario. Los 'scripts' de JavaScript pueden ser introducidos dentro de las páginas de HTML. Con JavaScript se puede dar respuesta a eventos iniciados por el usuario (el observador de nuestras páginas, por ejemplo), eventos tales como la entrada de datos en un formulario o algún enlace. Esto sucede sin ningún tipo de transmisión. De tal forma que cuando un usuario escribe algo en un formulario, no es necesario que sea transmitido hacia el servidor, verificado y devuelto. Las entradas son verificadas por la aplicación cliente y pueden ser transmitidas después.

Aunque JavaScript se parece a Java, no es lo mismo. Java es un lenguaje de programación mucho más complejo que JavaScript. JavaScript está hecho para ser un lenguaje bastante fácil de entender. A los autores del JavaScript no les debió haber importado mucho el tema de la programación. Por esta razón, algunos elementos de Java no son aceptados en JavaScript.

5.1.5 - PHP

PHP es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML o en ficheros. Este lenguaje es gratuito y multiplataforma. El uso de PHP es un pilar fundamental en la Aplicación.

Las páginas dinámicas creadas de esta forma permiten, entre otras cosas:

- Listados del Alumnos matriculados en una Asignatura.
- Modificaciones en las Reservas y Horarios de Consulta, que se reflejan instantáneamente en la página.
- Envío de distintos elementos según los parámetros enviados desde el cliente.
- Acceso a la BD donde se encuentran todos los datos de la Aplicación, así como las modificaciones que se realicen sobre los mismos.
- Realización de tareas automatizadas, como pueden ser los avisos a Alumnos y Profesores de Eliminación del Horario de Consulta, así como, el Recordatorio al Alumno de una asistencia a un Horario de Consulta.

En la realización del proyecto se ha utilizado la versión de PHP 5.2.9-2, que ofrece una programación orientada a objetos.

5.2 - Herramientas

En el desarrollo de la aplicación web “Módulo intranet de gestión de horarios de tutorías y listas de asistencia” sean utilizado como elementos de soporte, tres elementos:

1. Sistema Operativo Windows 7 Enterprise
2. Servidor Web con soporte PHP Apache v.2.2.11 y soporte PHP v5.2.9-2
3. Base de Datos MySQL v5.1.33

Estos dos últimos elementos vienen con el programa gratuito WAMPSEVER, versión para Windows del software, en la versión 2.0.

Para la creación de documentos, tanto en formato XHTML y CSS, como JavaScript o PHP se ha utilizado la plataforma de desarrollo Adobe Dreamweaver CS5. Esta IDE de programación permite trabajar con multitud de lenguajes, entre ellos los utilizados, y ayuda al programador ofreciendo información relacionada con los objetos que se tratan. Aparte están las opciones visuales que ofrece en relación al código y al modo en el que se ve la página (diseño, código y mixta).

Ofrece también una administración global de los archivos de la Aplicación tanto de forma local como remota, permitiendo una conexión al host con ftp o sftp. Cada proyecto se guarda de forma independiente, pudiendo acceder a ellos de forma fácil y rápida.

La administración de la BD se ha realizado a través de phpMyAdmin, que es una herramienta PHP para el manejo de MySQL a través de páginas web, utilizando Internet. Con ella podemos crear y eliminar BD, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas. Se encuentra disponible bajo la licencia GPL.

Se ha utilizado la aplicación de retoque fotográfico Adobe Photoshop CS5 para la creación de la interfaz gráfica que los usuarios utilizan, como por ejemplo el logotipo, los botones, marcos y el resto de imágenes utilizadas en la Aplicación. He intentado que la interfaz sea clara, sencilla y atractiva, Aprovechando al máximo el espacio del que se disponía.

5.3 - Detalles de la Implementación.

5.3.1 - Perfiles de usuario

La Aplicación mostrara automáticamente la interfaz correspondiente al tipo de usuario que se esté utilizando, permitiendo realizar solo las opciones definidas el usuario actual y acceder a los recursos definidos para cada usuario en particular. Es decir, vamos a tener diferentes perfiles de usuarios, y cada usuario tendrá una interfaz diferente. Estos usuarios son: Alumno y Profesor.

Perfil del Alumno.

Es el perfil que utilizarán todos los Alumnos que sean dados de alta en la Aplicación y previa identificación, serán redirigidos a la página (calendario.shtml) en la que podrán interactuar con el sistema.

La identificación se realizará en la página inicial (index.html), que dispone de una zona para la identificación de los alumnos y los profesores, al estilo UPV.

El usuario alumno dispondrá de una agenda en la parte central de la página, eje de toda la aplicación. En el cuerpo de la aplicación, en la parte superior izquierda tendrá un calendario que interactúa con la agenda, y permite un desplazamiento de fechas más rápido que con las flechas de movimiento de que dispone la agenda en su parte superior. En la parte inferior izquierda se dispone un panel de información en el que aparece el alumno y posteriormente el profesor de la asignatura seleccionada.

En la parte superior tiene una ventana deslizante en la que se muestran todas las asignaturas en las que se ha matriculado el alumno y que podrá seleccionar. Puntualizar que se muestran todas las asignaturas sin tener en cuenta el cuatrimestre. Seleccionar una asignatura conlleva por un lado mostrar el profesor que imparte la asignatura, y por otro, mostrar los horarios de consulta que ha puesto disponibles el profesor junto con las reservas ya hechas.

En la parte superior también se muestran cuatro botones, de izquierda a derecha son los de Salir, Listar Alumnos, Mostrar Configuración y Actualizar Vista. El primero nos saca de la aplicación, como se puede deducir, el segundo y el tercero, mostrarán un *“slide”* sobre la agenda en el que se mostrarán los alumnos matriculados en la asignatura que está seleccionada, y la configuración de la Aplicación por si existen dudas de cómo están definidos los parámetros. El último actualiza los datos de la agenda mostrando las variaciones que puedan haber sufrido los Horarios de consulta.

Inicialmente el alumno verá la agenda en formato de semana, puede también cambiar a formato de día, mostrando un único día en la agenda. En modo día, la visualización de las reservas es más visual.

En todos los horarios de consulta que verá el alumno, al seleccionar uno, verá un pequeño icono en la parte superior izquierda, será el icono de detalles. Este icono muestra de una forma más visual el estado del horario de consulta.

Perfil del Alumno.

Es el perfil que utilizarán todos los Profesores que sean dados de alta en la Aplicación y previa identificación, serán redirigidos a la página (calendario-master.shtml) en la que podrán interactuar con el sistema.

La identificación se realizará en la página inicial (index.html), que dispone de una zona para la identificación de los alumnos y los profesores, al estilo UPV.

El usuario profesor dispondrá de una agenda en la parte central de la página, eje de toda la aplicación, y al inicio de la sesión se cargarán los Horarios de Consulta asociados a la primera asignatura activa en el cuatrimestre correspondiente. En el cuerpo de la aplicación, en la parte

superior izquierda tendrá un calendario que interactúa con la agenda, y permite un desplazamiento de fechas más rápido que con las flechas de movimiento de que dispone la agenda en su parte superior. En la parte inferior izquierda se dispone un panel de información en el que aparece el profesor y la asignatura que está viendo en ese momento.

En la parte superior, al profesor le aparecerán las asignaturas en las que imparte docencia, estando habilitadas las correspondientes al cuatrimestre en el que se encuentra, y deshabilitadas las que se corresponden al otro cuatrimestre. Estas asignaturas estarán diferenciadas por el color del botón y asociado a este color estará el fondo de los horarios de consulta y el fondo del profesor en el panel de información. Seleccionar una asignatura conlleva mostrar los horarios de consulta (con su color identificativo) junto con las reservas hechas. El profesor tendrá la opción de eliminar cada una de las reservas, así como eliminar el horario de consulta.

En la parte superior también se muestran cuatro botones, de izquierda a derecha son los de Salir, Listar Alumnos, Mostrar Configuración y Actualizar Vista. El primero nos saca de la aplicación, como se puede deducir, el segundo y el tercero, mostrarán un *“slide”* sobre la agenda en el que se mostrarán los alumnos matriculados en la asignatura que está seleccionada, y la configuración de la Aplicación por si existen dudas de cómo están definidos los parámetros. El último actualiza los datos de la agenda mostrando las variaciones que puedan haber sufrido los horarios de consulta.

Inicialmente el alumno verá la agenda en formato de semana, puede también cambiar a formato de día, mostrando un único día en la agenda. En modo día, la visualización de las reservas es más visual.

En todos los horarios de consulta que verá el profesor, al seleccionar uno, verá un pequeño icono en la parte superior izquierda, será el icono de detalles. Este icono muestra de una forma más visual el estado del horario de consulta.

5.3.2 - Autenticación de usuarios

En la Aplicación he aplicado dos métodos para gestionar la seguridad, el primero consiste en el uso de variables de sesión de JavaScript para controlar el acceso a las páginas que componen la Aplicación, el segundo es el uso del cifrado de datos, incluido la contraseña almacenada en la BD, en la comunicación entre el sistema y el pc local de cada usuario.

En el primer paso, es decir, acceder a la página de identificación, el usuario ya inicia con el servidor una comunicación que será cifrada. Esto es, configurar Apache para utilizar el puerto 443 y utilizar el protocolo HTTPS (HTTP seguro) que se ejecuta sobre este puerto. De esta manera, los datos transmitidos en la identificación del usuario ya serán tratados con un cifrado de datos, proporcionando una comunicación más segura.

Una vez que el sistema ha comprobado que los datos introducidos son correctos, de que tipo de usuario se trata (Profesor o Alumno) y se confirma la identidad del usuario, todo ello mediante un acceso a la BD, al usuario se le asigna una *“ID”* en una variable de sesión que le acompañará durante todo el tiempo en que esté en la Aplicación. Si este *“ID”* se pierde o se corrompe, el usuario será expulsado de la Aplicación.

Una vez que el usuario accede a la Aplicación, sin olvidar que se produce a través del protocolo HTTPS, el sistema cargará las variables de sesión necesarias para el correcto funcionamiento

de la Aplicación. Además al usuario se le redirigirá a la interfaz correspondiente al tipo de usuario al que pertenece: Profesor o Alumno.

Mientras el usuario interactúe con la Aplicación, la sesión permanecerá abierta. Cuando llegue el momento de cerrar la Aplicación, abandonar el sistema o cerrar la sesión, el usuario tendrá que hacer un "click" en el botón de "Salir" que esta en la parte superior izquierda de la interfaz, o bien, cerrar la pestaña del navegador o el propio navegador. Con estas acciones las variables de sesión se destruyen, por lo que para acceder a la Aplicación, el usuario deberá identificarse de nuevo.

De esta manera, se tratara de controlar los accesos no deseados y de mantener la información utilizada por los usuarios protegida frente a ataques.

5.3.3 - Especificación y Aclaración en Operaciones de Actividad sobre los Horarios de Consulta

En un primer momento, se valoró la idea de que el sistema mandase un email a los usuarios implicados ante una modificación de los Horarios de Consulta, y también, que se encargase de recordar a los implicados en los Horarios de Consulta de la asistencia a la cita, como un buen método de aviso. Como referencia se tomó a la Biblioteca de la UPV y al Sistema de Avisos que implementa en el préstamo de libros.

Tomando como Servidor de Correo el de la UPV, al llevar a cabo la implementación y la configuración que se requiere para su funcionamiento, es cuando han surgido los problemas. A la hora de configurar la aplicación, se necesitan unos parámetros que en la práctica es complicado disponer de ellos, sencillamente por la falta de datos con respecto a los usuarios.

```
$mail = new PHPMailer();  
$mail -> From = "isferal@eui.upv.es";  
$mail -> FromName = "Isaac Fernandez";  
$mail->AddAddress("isferal@eui.upv.es", "Isaac UPV");  
$mail -> Subject = "Test";  
$mail -> Body = "<h3>From UPV!</h3>";  
$mail -> IsHTML (true);  
$mail->IsSMTP();  
$mail->Host = 'ssl://';  
$mail->Port = 465;  
$mail->SMTPAuth = true;  
$mail->Username = "";  
$mail->Password = "";
```

Aquí se muestran los parámetros que se tienen que configurar para poder enviar emails desde la Aplicación. Como se puede ver, la mayoría de elementos se rellenan sin mayor dificultad, pero hay dos campos que implican el uso de información personal: Usuario y Password. Esta información no se dispone y de ahí el hecho de buscar otras opciones al respecto.

Por lo tanto, una vez desechada esta opción se opta por utilizar mensajes internos y la BD como respaldo, simplificando la configuración. De esta manera, ya se hace uso de un servidor de correo y el almacenamiento y el acceso a la BD se realiza de forma interna. En definitiva, la



comunicación entre Alumnos y Profesores se realizaría por medio de mensajes almacenados en la BD y que los usuarios accedan a ellos a través de la interfaz de la Aplicación.

El funcionamiento sería similar al utilizado con los emails, pero con un acceso desde la Aplicación:

Profesor

El profesor una vez accede a la Aplicación, tendrá un botón que le permitirá acceder a los mensajes, además de avisarle de mensajes nuevos o no leídos con un número que aparecerá al lado del texto del botón con el que se accede. Una vez acceda, los mensajes aparecerán en bloques, según asignatura, y ordenado de más reciente a menos.

El profesor podrá enviar mensajes únicamente a los alumnos matriculados en su asignatura, pudiendo mandar mensajes a un alumno, varios alumnos o todos.

Alumno

El alumno una vez accede a la Aplicación, tendrá un botón que le permitirá acceder a los mensajes, además de avisarle de mensajes nuevos o no leídos con un número que aparecerá al lado del texto del botón con el que se accede. Una vez acceda, los mensajes aparecerán en bloques, según asignatura, y ordenados de más reciente a menos.

Los alumnos, podrán enviar mensajes únicamente a los profesores que imparten las asignaturas en las que se han matriculado.

5.4 - Detalles de la Integración

Los componentes de origen externo utilizados en la Aplicación son el servidor Web y el SGBD.

El servidor Web que se ha utilizado ha sido Apache, instalado sobre el SO Windows 7 Enterprise Edition. Se ha utilizado con sus correspondientes módulos, incluido el de PHP, para que sea capaz de interpretar los scripts en el lado del servidor, así como los módulos de SSL para realizar las conexiones seguras.

El SGBD utilizado ha sido MySQL, con él se ha implementado el nivel de persistencia de la Aplicación, por la facilidad del manejo de MySQL y por las ventajas de almacenar la información utilizando una BD. Cabe destacar que PHP se combina a la perfección con MySQL, por lo que incluirlo en el sistema no ha supuesto ninguna complicación.

6 - EVALUACIÓN Y PRUEBAS

Durante el proceso de desarrollo de la Aplicación se hacen pruebas del SW que se añade, se comprueba su correcto funcionamiento y se continúa con el desarrollo.

Ahora, cuando la Aplicación esta desarrollada en su totalidad, se tiene que comprobar que todo ese código que se ha ido añadiendo, funciona correctamente y no existen problemas en la ejecución de la Aplicación, ni en las operaciones que realiza.

Este es el momento en el que se tienen que aplicar los elementos de evaluación y realizar las pruebas. Con esto comprobaremos la robustez de la Aplicación y si existen partes que fallan para su revisión y modificación.

6.1 - Evaluación

La evaluación de la Aplicación supone un apartado del proceso de desarrollo muy importante. Por eso, se ha decidido realizar un análisis de la usabilidad y unas pruebas sobre el software para garantizar una herramienta que cumple con los estándares.

La usabilidad se centra en las necesidades del usuario, por lo que durante el desarrollo de la interfaz y el modo de mostrar la información, se ha tenido en cuenta lo siguiente:

- Pensar en la mejor manera para mostrar la información a los usuarios.
- Tener en cuenta las limitaciones de los recursos del usuario a nivel de ancho de banda, capacidad de procesamiento y resolución.
- Definir una interfaz clara y sencilla en la que se tiene todo en una misma ventana y el acceso a la información no confunda a los usuarios. Los usuarios identificarán claramente donde se presentan los diferentes tipos de información.
- Simplificar la navegación al usuario dentro de la Aplicación, eliminando los enlaces a otras páginas y los saltos de imagen utilizando tecnologías que permiten cargar información en las capas deseadas actualizando información, y aprovechar al máximo el espacio.

Estas son algunas ideas que se han tratado en la Aplicación, pero existen numerosas reglas y recomendaciones para conseguir medios dotados de una alta usabilidad.

Por lo tanto la usabilidad se ha tenido en cuenta desde un principio y en todas las fases del proyecto, a la vez que se diseñaba e implementaba la aplicación, intentando en todo momento que esta fuese lo más fácil, cómoda e intuitiva posible para el usuario final.

Cabe destacar que una aplicación con una usabilidad compleja no se utiliza, por lo que al usuario, hay que simplificarle el uso de las herramientas con las que debe trabajar.

6.2 - Pruebas

A continuación se mostraran los resultados de las pruebas que se han aplicado al software de la Aplicación. Estas son referentes a la accesibilidad como son las pruebas de verificación de los estándares y resolución para su correcta visualización por cualquier cliente, esta ultima prueba realizada de un modo manual.

Las pruebas referentes a la navegabilidad como es la prueba de enlaces rotos o el funcionamiento de la Aplicación en los diferentes navegadores se han hecho de forma manual, aunque por curiosidad se ha pasado también un programa que realiza el testeo de los enlaces de la web para comprobar sus resultados.

6.2.1 - Validación XHTML y CSS.

Las paginas HTML que forman parte de la Aplicación han sido comprobados con el validado de W3C (<http://validator.w3.org/>), y se han corregido todos los posibles errores que pudieran haber, hasta que se ha validado correctamente como “XHTML 1.0 Transitional” mostrando la siguiente página.

The screenshot shows the W3C Markup Validation Service interface. At the top, it says "W3C Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below that, it says "Jump To: Congratulations · Icons". A green banner states "This document was successfully checked as XHTML 1.0 Transitional!". The main section shows the validation details: "Result: Passed", "File: calendario-master.shtml", "Encoding: iso-8859-1", "Doctype: XHTML 1.0 Transitional", "Root Element: html", and "Root Namespace: http://www.w3.org/1999/xhtml". There is a "Donate" button and a "Flat" button. Below that, there are "Options" for "Show Source", "Show Outline", "List Messages Sequentially", "Group Error Messages by Type", "Validate error pages", "Verbose Output", and "Clean up Markup with HTML-Tidy". A "Revalidate" button is also present. The bottom section contains "Congratulations" and "valid icon(s) on your Web page" with an HTML code snippet for the icon. At the very bottom, there is a footer with "Home About... News Docs Help & FAQ Feedback Contribute" and "W3C open source" logo.

Se han comprobado también los ficheros CSS de la Aplicación con el validador del W3C para CSS (<http://jigsaw.w3.org/css-validator/>) con la opción del perfil **CSS3**. La validación mostrará esta imagen en el caso en el que sea correcto el código. Hay que decir que existe una propiedad utilizada en los CSS que no está homologada por el CSS3 y que se utiliza para el navegador IE, que utiliza en parte sus propias reglas. La propiedad es la siguiente: **filter: alpha(opacity=40);**

The screenshot shows the W3C CSS Validator interface. At the top, it says 'El Servicio de Validación de CSS del W3C' and 'Resultados del Validador CSS del W3C para resolucion-25mst.css (CSS versión 3)'. Below that, it indicates 'Ir a: Su Hoja de Estilo validada'. The main heading is 'Resultados del Validador CSS del W3C para resolucion-25mst.css (CSS versión 3)'. A green banner reads '¡Enhorabuena! No error encontrado.' Below this, it states '¡Este documento es CSS versión 3 válido!' and provides instructions on how to display the validation icon on a website. Two code snippets are shown: one for the standard 'CSS Válido!' icon and another for the 'CSS Válido!' icon with a blue checkmark. A note at the bottom says '(cierre la etiqueta img con > en lugar de /> si utiliza HTML <= 4.01)'. At the very bottom, there is an HP logo and text stating 'The W3C validators are hosted on server technology donated by HP, and supported by community donations.' with a 'Donate' link and a 'Flattr' button.

6.2.2 - Comprobación de Enlaces Rotos

La comprobación de enlaces se ha hecho de un modo manual ya que, realmente, saltos entre paginas existen cuatro, dos son los que se realizan desde la pagina de inicio a las paginas que corresponden a cada usuario una vez se han identificado, y dos mas desde estas mismas paginas a la pagina inicial, o bien para salir de la aplicación o ser redirigido a la misma.

Es lógico entender que no hay enlaces rotos, ya que, habría un problema en la aplicación y el resultado sería quedarte bloqueado en el inicio, no pudiendo acceder a la Aplicación, o no podrías salir de la Aplicación.

6.2.3 - Comprobación de la Resolución

Se han establecido tres configuraciones de resolución, que varían dependiendo de la resolución que cada usuario tenga seleccionada en su monitor. Esta resolución vendrá influida por el tamaño del monitor que cada uno use.

Se han establecido los siguientes rangos de aplicación de resoluciones:

- Resolución de **1280 x 1024** **Monitores < 22"**
- Resolución de **1680 x 1050** **22" ≤ Monitores < 25"**
- Resolución de **1920 x 1200** **Monitores ≥ 25"**

Se ha establecido como monitor estándar, un monitor de 19".

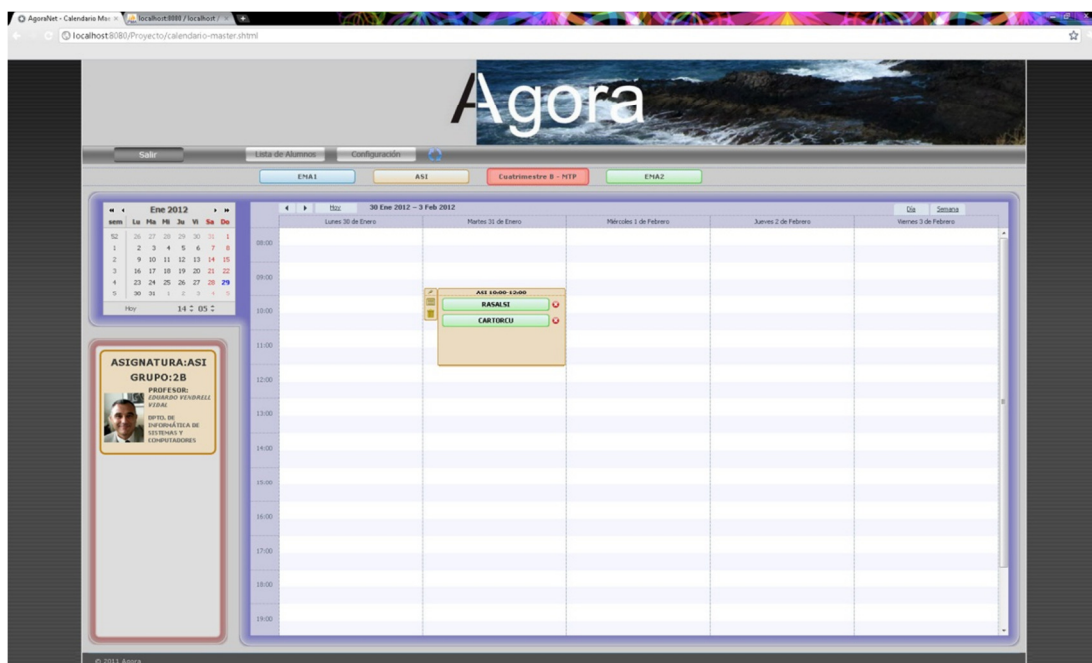
Las comprobaciones realizadas han sido satisfactorias.

6.2.3 - Pruebas de Uso

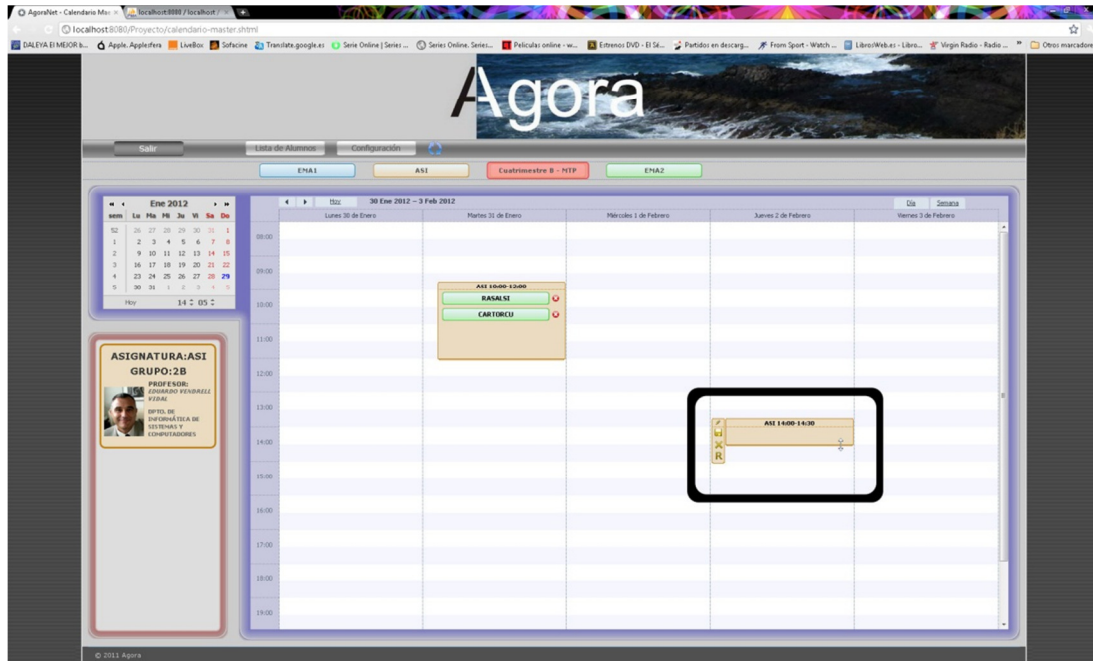
A continuación se realizarán dos pruebas de uso, que serán las operaciones que más se realizan en la Aplicación, una de un Alumno y otra de un Profesor.

Profesor: Agregar Horario de Consulta

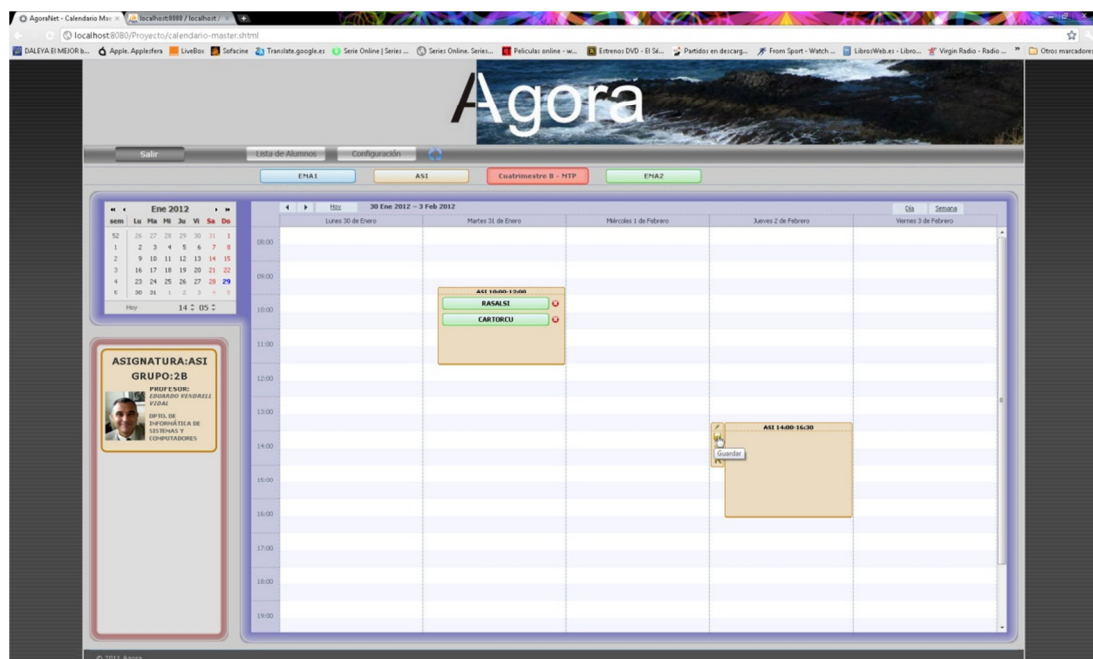
Un vez el Profesor ha accedido a la Aplicación le aparece la siguiente pantalla, teniendo que seleccionar la asignatura en la cual se quiere realizar las consultas. Para seleccionar la asignatura hay que hacer un "clic" en el icono de la asignatura. La asignatura seleccionada en la prueba ha sido ASI (Grupo B) y la imparte Eduardo Vendell.



Una vez se ha accedido a la Aplicación y se ha seleccionado la asignatura, para introducir un nuevo Horario de Consulta, hay que hacer “*doble clic*” en la franja de hora en la que se quiere iniciar el horario de consulta, es decir, si se quiere empezar el HC a partir de las 14.00, el “*doble clic*” se realizara en la franja de hora entre 14.00-14.30. Para redimensionar el HC hay que mantener apretado el botón en la parte inferior y arrastrar el ratón.

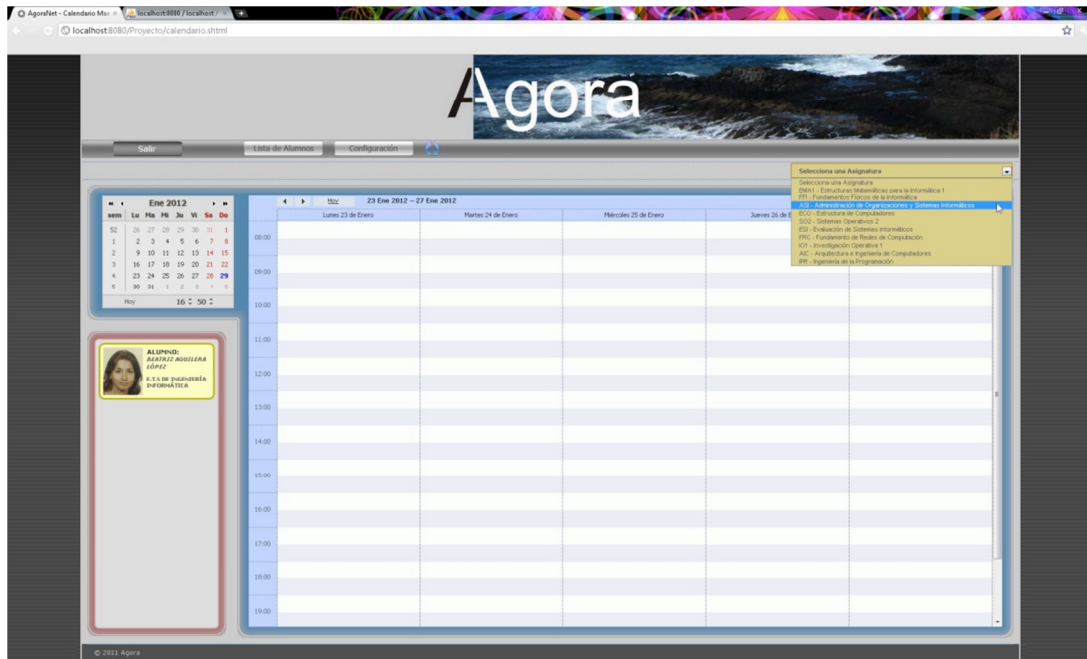


Una vez seleccionado la franja de hora en la que si situara el Horario de consulta, se libera el botón del ratón. Ahora lo único que queda es guardar el HC, esto se realiza haciendo un “*clik*” en el botón con una imagen de un disquete. **¡Importante!** Si esto no se hace el HC no será guardado por el sistema.

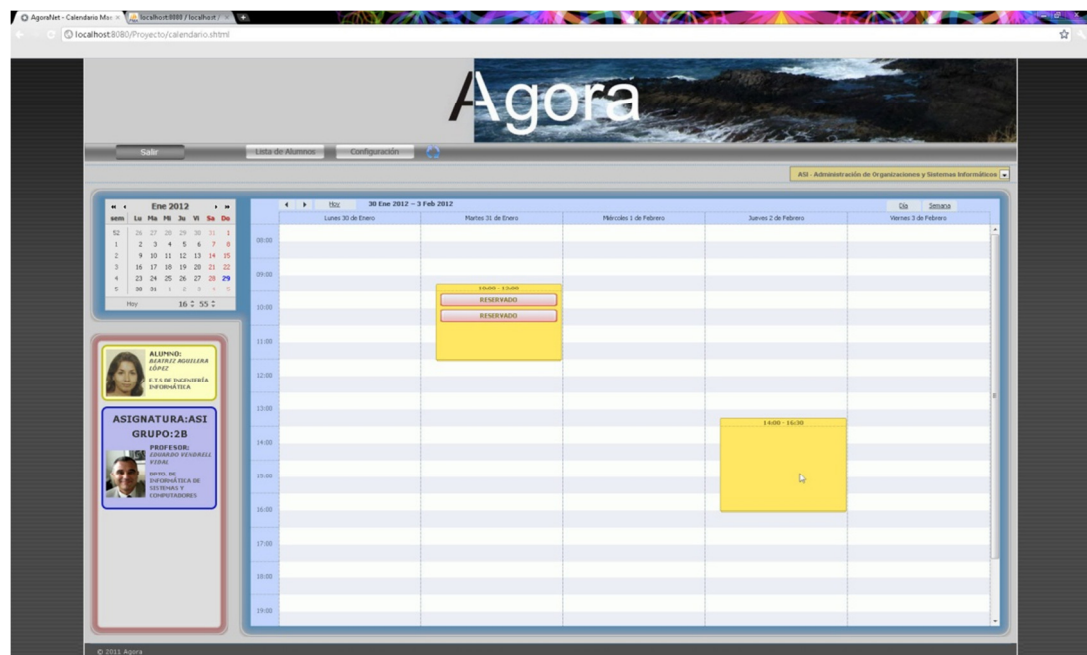


Alumno: Inscribirse en Horario de Consulta

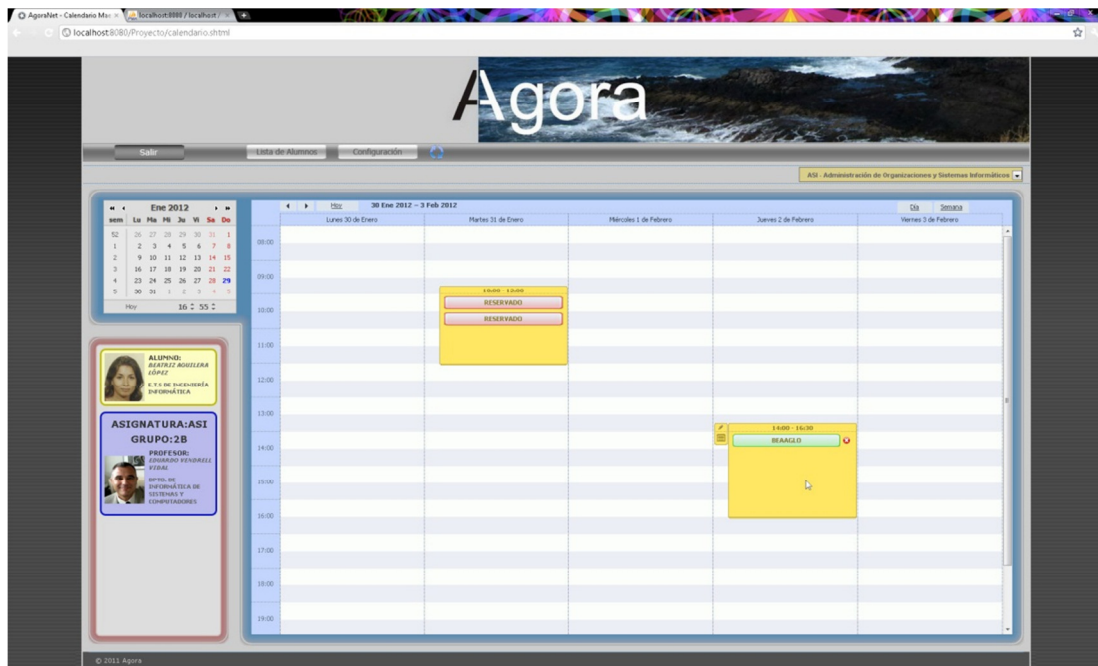
Aprovechando lo realizado anteriormente, he elegido un alumno que cursa la misma asignatura que la utilizada en el caso anterior. Una vez ha accedido el alumno a la Aplicación, la agenda le aparecerá sin ningún Horario de Consulta, debido a que no se ha seleccionado ninguna asignatura. En primer lugar, el alumno deberá seleccionar una asignatura. En este caso será la asignatura ASI (Grupo B) y de Alumno a Beatriz Aguilera.



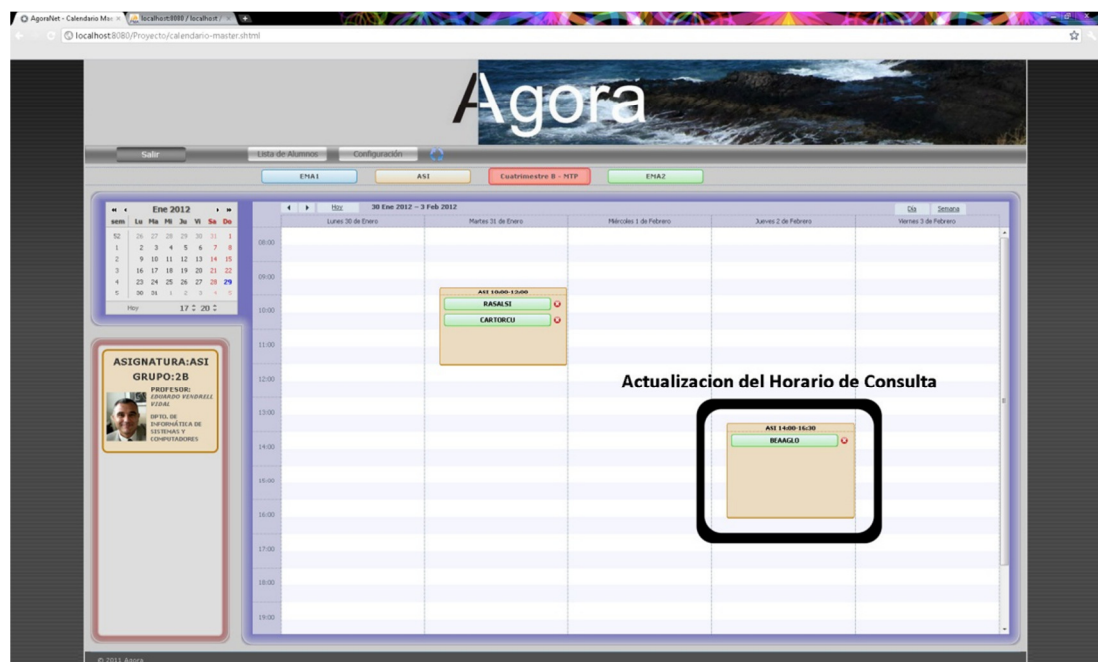
Una vez seleccionada la asignatura, aparecerán los Horarios de Consulta y los datos del Profeso.



En este momento, el Alumno tiene en la agenda cargados los Horarios de Consulta que el profesor pone a disposición de los Alumnos. El Alumno puede ver que horarios hay, en cuales puede inscribirse y que a que hora ha sido asignado, esto solo con poner el puntero del ratón por encima de la reserva realizada. Ante de eso, evidentemente hay que realizar una reserva, esto es, hacer "doble clic" en el cuerpo del HC. Con esto el Alumno habrá reservado cita para asistir a la tutoría.



En caso de tener una reserva realizada, no se podrá reservar una segunda reserva en el mismo Horario de Consulta. Ahora mostraremos la pantalla del Profesor de la asignatura actualizada para mostrar las reservas realizadas y la comunicación Alumno-Profesor.



7 - CONCLUSIONES

Como conclusión podría decir que el camino hasta la culminación del proyecto ha sido largo y con muchos momentos en los que el desanimo parece que lo va a echar todo por tierra. Hay que reconocer que no todo es así, también hay momentos en los que ves que las cosas van cuadrando y que avanzas en las tareas, y ves que se puede conseguir.

Ha sido mucho el trabajo y las tareas que se han tenido que desarrollar a lo largo de este Proyecto: planificar, diseñar, programar... volver a planificar, a diseñar y a cambiar la programación... Hay trabajos que han resultado más interesantes, que son los que van dando forma a la Aplicación, dar funcionalidad... y otros tediosos y desesperantes, como puede llegar a ser el análisis de código... Algunas cosas que aparentaban sencillas se han complicado, mientras que otras que parecían difíciles a simple vista al final han acabado siendo sencillas de implementar. El resultado final de todo este trabajo es la memoria que ahora concluye y el resultado visible que se puede consultar desde el navegador web.

La mayor alegría ha sido la de conseguir el propósito inicial y lograr desarrollar los objetivos planteados. Es decir, conseguir dar forma a una idea y ver como crece, siendo que hace un tiempo era algo intangible. Sin olvidar el hecho de sacarlo adelante, cosa que a nivel personal me satisface en gran manera, ya que ha sido un ejercicio de perseverancia, de tenacidad, de no abandonar una idea, de terminar lo empezado.

El resultado final ha sido plenamente satisfactorio y no se ha alejado mucho de la idea inicial que se tenía, si bien, todo ha tenido una evolución y algo siempre cambia.

En la introducción ya he comentado algunas cosas al respecto del proyecto, pero si hay algo que merece ser repetido, es la visión real que puede ofrecer el hacer un proyecto: desarrollar una idea, los pasos que hay que dar, la evolución de un proyecto, el esfuerzo que hay que invertir,... lo que se aprende.

Estoy convencido que afrontar este reto me ha servido para darme cuenta de muchas cosas, sobre todo, te das cuenta que las cosas no se hacen solas. Ahora solo cabe esperar, que todo lo aprendido me sirva a nivel laboral, como experiencia, como base para lo que se puede acercar.

De todas formas, aunque el proyecto ya este terminado, considero que la Aplicación no esta concluida, pues no existe un programa o un diseño que pueda darse por terminado al 100%. Siempre hay algo que se puede modificar, o nuevas ideas que se puedan aplicar, o nuevas opciones que añadir.

En ese sentido las posibilidades que ofrece la Aplicación para ser redimensionada, aplicarle nuevas opciones, y ofrecer a los usuarios mayores prestaciones son muy amplias. La opción de añadir nuevas utilidades o de aplicar nuevas ideas, esta abierta a todo aquel que quiera.



Quiero mencionar que en este proyecto se ha reutilizado código adaptándolo a las necesidades requeridas por la Aplicación, por lo que su uso a nivel comercial queda restringido al copyright que puedan tener estas aplicaciones.

Algunas ideas que se podrían desarrollar para seguir trabajando en una aplicación mas eficiente podrían ser: la búsqueda de optimización de la aplicación en las consulta a la BD, obtener recordatorios por email de las reservas o horarios de consulta, aplicar a la aplicación una versión en valenciano o en ingles,... Una vez desarrollado el núcleo no deberían se ser tan costosas.

No quiero terminar las conclusiones sin mencionar que algunas de las opciones arriba planteadas, en concreto la de *“obtener recordatorios de las reservas o horarios de consulta”*, se han intentado aplicar, pero los recursos que se necesitan (servidor de correo y datos de las cuentas de los usuarios) no son accesibles e imposibilitan el desarrollo. Con esto quiero decir que con los recursos adecuados se puede llevar a cabo cualquier opción, pero que en alguna circunstancia, uno se puede encontrar con que no es posible seguir adelante.

Yo termino aquí este proyecto y con el, también, el largo camino que me ha llevado hasta el final de esta Ingeniería Técnica en Informática de Sistemas.

8 - BIBLIOGRAFÍA

8.1 - Referencias Web

8.1.1 - ERS

- [1] IEEE Recommended Practice for Software Requirements Specifications - IEEE Std 830-1998
- [2] The World Wide Web Consortium (W3C) for Standards: <http://www.w3.org/>
- [3] Wikipedia en español: <http://es.wikipedia.org>

8.1.2 - HTML y CSS

- [1] Página de Desarrollo Web: <http://www.desarrolloweb.com>
- [2] Libros Gratuitos CSS: <http://www.librosweb.es/>

8.1.3 - JavaScript

- [1] Libros Gratuitos AJAX y JavaScript: <http://www.librosweb.es/>
- [2] DhtmlxScheduler: <http://dhtmlx.com/index.shtml>
- [3] Kwikcs for jQuery: <http://www.jeremymartin.name/projects.php?project=kwicks>

8.1.4 - PHP

- [1] Documentación oficial de PHP: <http://www.php.net/manual/es/>

8.1.5 - MOSKitt

- [1] Modeling Software KIT (MOSKitt): <http://www.moskitt.org/>

8.2 - Libros de Consulta

8.2.1 - HTML y CSS

- [1] HTML Manual - CHM
- [2] Introducción a CSS
Javier Eguíluz Pérez – LibrosWeb - <http://www.librosweb.es/>
- [3] CSS Avanzado
Javier Eguíluz Pérez – LibrosWeb - <http://www.librosweb.es/>



8.2.2 - JavaScript

- [1] Introducción a JavaScript
Javier Eguíluz Pérez – LibrosWeb - <http://www.librosweb.es/>
- [2] JavaScript Manual - CHM

8.2.3 - AJAX

- [1] Introducción a AJAX
Javier Eguíluz Pérez – LibrosWeb - <http://www.librosweb.es/>
- [2] DhtmlxScheduler Manual - CHM

8.2.4 - PHP

- [1] Manual Imprescindible de PHP5
Luis Miguel Cabezas Granado, (Anaya Multimedia, 2004. ISBN: 84-415-1785-1)
- [2] PHP Manual - CHM

8.2.5 - APACHE

- [1] La Biblia del Servidor Apache
Mohammed J. Kabir, (Anaya, 2003. ISBN: 84-415-1468-2)
- [2] Programación de Servidores Web con CGI, SSI e IDC
Sergio Lujan Mora (Editorial Club Universitario, 2001, ISBN: 978-84-8454-136-3)
- [3] HTTPS en Apache 2
Alberto Molina Coballes, José Muñoz Rodríguez, Jose Luis Rodríguez Rodríguez (Junta de Andalucía, Consejería de Educacion, 2011)
- [4] SSL, Secure Socket Layer y otros Protocolos
Jose Maria Morales Vázquez (UPM – Master Seguridad Informática, 2002)

8.2.6 - MySQL

- [1] La Biblia del Servidor Apache
Mohammed J. Kabir, (Anaya, 2003. ISBN: 84-415-1468-2)
- [2] MySQL Manual - CHM

ANEXO A - REQUISITOS

A.1 - Objetivos del Sistema

OBJ-01	Gestionar los horarios de consulta
Descripción	El sistema deberá gestionar la información correspondiente a los horarios de consulta de cada profesor en cada una de las asignaturas: INSERTAR, AGREGAR y ELIMINAR.
Estabilidad	Alta
Comentarios	El sistema informará a los elementos implicados en la acción.

OBJ-02	Gestionar las inscripciones de los alumnos
Descripción	El sistema deberá gestionar la información correspondiente a las inscripciones de los alumnos en los horarios de consulta: AGREGAR y ELIMINAR
Estabilidad	Alta
Comentarios	<ul style="list-style-type: none">• La información requerida de los alumnos, se obtendrá de la BD general de la UPV o bien importando los datos referentes a los alumnos que cursen asignaturas que utilicen la aplicación aquí descrita para gestionar los horarios de consulta.• El sistema informará a los elementos implicados en la acción.

OBJ-03	Gestionar los <u>mensajes</u> de información
Descripción	El sistema deberá gestionar los mensajes de información en relación a las acciones ejecutadas por los usuarios; además actuará como recordatorio para los usuarios del sistema con un tiempo de antelación de 24h
Estabilidad	Media
Comentarios	El sistema procederá de manera automática.



A.2 - Requisitos de Información

Información que el sistema recoge por medio de la interfaz gráfica.

IRQ-01	Información sobre los Horarios de Consulta	
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-01 <i>Gestionar los horarios de consulta</i> 	
Requisitos Asociados	<ul style="list-style-type: none"> • UC-01 <i>Insertar horarios de consulta</i> • UC-02 <i>Agregar horario de consulta</i> • UC-03 <i>Eliminar horario de consulta</i> 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>los horarios de consulta</i> , en concreto:	
Datos Específicos	<ol style="list-style-type: none"> 1. Fecha del Horario de Consulta. 2. Duración de Horario de Consulta. 3. Tiempo Asignado por Consulta. 	
Tiempo de Vida	Medio	Máximo
	4 meses	8 meses
Ocurrencias Simult.	Medio	Máximo
Estabilidad	Alta	
Comentarios	Ninguno	

CRQ-01	Relación entre Horarios de Consulta y Profesor	
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-01 <i>Gestionar los horarios de consulta</i> 	
Requisitos Asociados	<ul style="list-style-type: none"> • IRQ-01 <i>Información sobre los horarios de consulta</i> 	
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: un horario de consulta debe tener asignado un código de asignatura que relacione al profesor que imparte la asignatura con el horario de consulta asignado.	
Estabilidad	Alta	
Comentarios	Ninguno	



CRQ-02	Asignación de tiempo de Horarios de Consulta entre los Alumnos
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar las inscripciones de los alumnos</i>
Requisitos Asociados	<ul style="list-style-type: none"> • IRQ-01 <i>Información sobre los horarios de consulta</i>
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: un horario de consulta puede tener entre 0 y N alumnos, pero N no será mayor al resultado entero de dividir el tiempo de consulta entre el tiempo asignado por alumno (definido por el profesor).
Estabilidad	Alta
Comentarios	Ninguno

IRQ-02	Información sobre la inscripción de los alumnos	
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar las inscripciones de los alumnos</i> 	
Requisitos Asociados	<ul style="list-style-type: none"> • UC-04 <i>Añadir alumno al horario de consulta</i> • UC-05 <i>Eliminar alumno del horario de consulta</i> 	
Descripción	El sistema deberá almacenar la información correspondiente a las inscripciones de los usuarios en los diferentes horarios de consulta.	
Datos Específicos	<ol style="list-style-type: none"> 1. Nick del Alumno. 2. Orden de Consulta. 3. Fecha de Inicio y Fecha de Finalización. 4. Fecha de Realización de la Consulta. 	
Tiempo de Vida	Medio	Máximo
	3 días	7 días
Ocurrencias Simult.	Medio	Máximo
	6	16
Estabilidad	Alta	
Comentarios	Ninguno	



A.3 - Requisitos Funcionales

A.3.1 - Definición de Actores

ACT-01	Profesor
Descripción	Este actor representa a los profesores
Comentarios	Ninguno

ACT-02	Alumno
Descripción	Este actor representa a los alumnos
Comentarios	Ninguno



A.3.2 - Casos de Uso del Sistema. Interfaces de transmisión de información

UC-01	Insertar horarios de consulta	
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-01 <i>Gestionar los horarios de consulta</i> 	
Requisitos Asociados	<ul style="list-style-type: none"> • IRQ-01 <u><i>Información sobre los Horarios de Consulta</i></u> 	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando un profesor quiera insertar horarios de consulta.	
Precondición	El solicitante es profesor y no tiene asociados horarios de consulta.	
Secuencia Normal	Paso	Acción
	1	El usuario accede a la semana tipo en la que comienza el proceso de inserción de horarios de consulta.
	2	El usuario incluye en un calendario semana los horarios de consulta.
	3	El usuario solicita al sistema que inserte los horarios de consulta haciendo “ <i>click</i> ” en el icono destinado a tal efecto en el horario de consulta a insertar.
	4	El sistema inserta los horarios de consulta.
	5	El sistema informa al profesor de que el proceso ha terminado con éxito.
Postcondición	Los horarios de consulta del profesor se actualizan para el periodo lectivo actual.	
Excepciones (Paso en la Secuencia Normal)	Paso	Acción
Rendimiento	Paso	Cota de tiempo
	4	Inserción BD y Actualizar información < 1 segundo
Frecuencia		
Estabilidad	Alta	
Comentarios	Los horarios de consulta se agregan en bloque.	



UC-02	Agregar horarios de consulta	
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-01 <i>Gestionar los horarios de consulta</i> 	
Requisitos Asociados	<ul style="list-style-type: none"> • IRQ-01 <u><i>Información sobre los Horarios de Consulta</i></u> 	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando un profesor quiera añadir horarios de consulta.	
Precondición	El solicitante es profesor.	
Secuencia Normal	Paso	Acción
	1	El usuario accede a la semana del calendario para comenzar el proceso de inserción de un horario de consulta.
	2	El usuario incluye el horario de consulta.
	3	El usuario solicita al sistema que inserte los horarios de consulta haciendo “ <i>click</i> ” en el icono destinado a tal efecto en el horario de consulta a agregar.
	4	El sistema inserta el horario de consulta.
	5	El sistema informa al profesor de que el proceso ha terminado con éxito.
Postcondición	El horario de consulta se agrega al sistema y se actualiza para el periodo lectivo actual.	
Excepciones (Paso en la Secuencia Normal)	Paso	Acción
Rendimiento	Paso	Cota de tiempo
	4	Inserción BD y Actualizar información < 1 segundo
Frecuencia		
Estabilidad	Alta	
Comentarios	Los horarios de consulta se agregan de uno en uno.	



UC-03	Eliminar horario de consulta.	
Objetivos Asociados	<ul style="list-style-type: none">• OBJ-01 <i>Gestionar los horarios de consulta</i>	
Requisitos Asociados	<ul style="list-style-type: none">• IRQ-01 <u>Información sobre los Horarios de Consulta</u>	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando un profesor quiera eliminar un horario de consulta.	
Precondición	El solicitante es profesor y tiene asociados horarios de consulta.	
Secuencia Normal	Paso	Acción
	1	El usuario accede a la semana tipo o a la semana deseada del calendario.
	2	El usuario solicita al sistema comenzar con el proceso de eliminar un horario de consulta haciendo " <i>click</i> " en el icono destinado a tal efecto en el horario de consulta a eliminar.
	3	El sistema elimina los horarios de consulta.
	4	El sistema avisa al profesor de que la eliminación de los horarios de consulta ha tenido éxito, así como, a los alumnos implicados en la eliminación de los horarios de consulta.
Postcondición	El horario de consulta se elimina del sistema (Eliminar de la BD).	
Excepciones (Paso en la Secuencia Normal)	Paso	Acción
Rendimiento	Paso	Cota de tiempo
	4	Eliminación de BD y Actualizar información < 1 segundos
Frecuencia		
Estabilidad	Alta.	
Comentarios	Los horarios de consulta se eliminan de uno en uno. Los alumnos se eliminan en bloque.	



UC-04	Añadir alumno al horario de consulta.	
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar las inscripciones de los alumnos</i> 	
Requisitos Asociados	<ul style="list-style-type: none"> • IRQ-02 <i>Información sobre la inscripción de los alumnos</i> 	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando un alumno se añada a un horario de consulta.	
Precondición	El solicitante es un alumno.	
Secuencia Normal	Paso	Acción
	1	El usuario accede a la semana en la que quiere hacer la reserva en un horario de consulta.
	2	El usuario solicita al sistema comenzar con el proceso de añadir un alumno al horario de consulta correspondiente haciendo " <i>double click</i> " en el horario seleccionado.
	3	El sistema añade al alumno en el horario de consulta correspondiente.
	4	El sistema informa al alumno de la reserva para la consulta y del éxito de la acción. El sistema informa al profesor de la asignatura de la reserva realizada.
Postcondición	El alumno ha sido incluido en un horario de consulta.	
Excepciones (Paso en la Secuencia Normal)	Paso	Acción
	2	Si el horario de consulta esta completo, la acción " <i>double click</i> " no tendrá efecto alguno, y el sistema avisara al alumno del estado del horario de consulta. A continuación este caso de uso quedara sin efecto.
Rendimiento	Paso	Cota de tiempo
	3	Inserción en BD y Actualizar información < 1 segundo
Frecuencia		
Estabilidad	Alta.	
Comentarios	Los alumnos se añaden de uno en uno.	



UC-05	Eliminar alumno de horario de consulta.	
Objetivos Asociados	<ul style="list-style-type: none"> • OBJ-02 <i>Gestionar las inscripciones de los alumnos</i> 	
Requisitos Asociados	<ul style="list-style-type: none"> • IRQ-02 <u>Información sobre la inscripción de los alumnos</u> 	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando un alumno se elimine de un horario de consulta o bien un profesor elimine a un usuario de un horario de consulta.	
Precondición	El alumno tiene que estar inscrito en un horario de consulta. Los solicitantes serán alumnos y profesores.	
Secuencia Normal	Paso	Acción
	1	El usuario (alumno - profesor) accede a la semana en la que quiere eliminar a un alumno de un horario de consulta.
	2	El usuario solicita al sistema comenzar con el proceso de eliminar a un alumno de un horario de consulta, haciendo " <u>click</u> " en un icono aspa destinado a tal efecto.
	3	El sistema elimina al alumno del horario de consulta correspondiente
	4	El sistema informa al alumno de que ha sido eliminado de un horario de consulta. También informará al profesor que elimine a un alumno de un horario de consulta.
Postcondición	El alumno ha sido eliminado de un horario de consulta	
Excepciones (Paso en la Secuencia Normal)	Paso	Acción
Rendimiento	Paso	Cota de tiempo
	3	Eliminar de BD y Actualizar información < 0.5 segundos
Frecuencia		
Estabilidad	Alta	
Comentarios	Los alumnos se eliminan de uno en uno.	



A.4 - Requisitos no Funcionales

NFR-01	Copias de seguridad.
Objetivo Asociado	
Requisitos Asociados	
Descripción	El sistema deberá incorporar algún mecanismo que permita realizar copias de seguridad de los datos almacenados.
Comentarios	De alta prioridad.

NFR-02	Entorno de funcionamiento.
Objetivo Asociado	
Requisitos Asociados	
Descripción	El sistema deberá ser compatible con navegadores como:IE, Firefox, Chrome u Opera.
Comentarios	Ninguno.

ANEXO B - TECNOLOGÍAS UTILIZADAS

B.1 - ARQUITECTURA CLIENTE/SERVIDOR

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes.

Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y que funciona como un sistema gestor de base de datos (SGBD). Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.



Este tipo de arquitectura es la más utilizada en la actualidad, debido a que es la más avanzada y la que mejor ha evolucionado en estos últimos años.

Podemos decir que esta arquitectura necesita tres tipos de software para su correcto funcionamiento:

1. Software de Gestión de Datos: Este software se encarga de la manipulación y gestión de los datos almacenados y requeridos por las diferentes aplicaciones. Normalmente este software se aloja en el servidor.
2. Software de Desarrollo: Este software se aloja en los clientes y solo en aquellos que se dedican al desarrollo de aplicaciones.
3. Software de Interacción con los Usuarios: También reside en los clientes y es la aplicación gráfica de usuario para la manipulación de datos, siempre claro a nivel de usuario (consultas principalmente)

A parte de estos existen más aplicaciones software para el correcto funcionamiento de esta arquitectura pero ya están condicionadas por el tipo de sistema operativo instalado, el tipo de red en la que se encuentra...

1.1.- SSI: Server Side Include y SHTML

SSI son directivas de inclusión del servidor. Comandos que se incluyen en una página HTML y que son ejecutados por el servidor web antes de transmitir la página al cliente. Permite generar páginas web dinámicas.

Es una tecnología que permite crear páginas web dinámicas. Se basa en el uso de una serie de comandos que permiten, por ejemplo, incluir en una página web el contenido de un fichero o el resultado producido por un programa al ejecutarse. Desgraciadamente, no existe un estándar sobre SSI, por lo que cada servidor web puede definir los comandos SSI de diferente forma. Sin embargo, algunos comandos se han convertido en estándar de facto y su comportamiento es idéntico (o muy similar) en la mayoría de los servidores web.

Es conveniente emplear una extensión especial para aquellas páginas HTML que contienen directivas SSI. No es conveniente activar el intérprete SSI para todas las páginas, ya que entonces todas las páginas se van a interpretar (incluidas las que no contienen SSI) y el tiempo de respuesta del servidor web aumentará. Las extensiones predeterminadas normalmente son **.stm**, **.shtm** y **.sHTML**, aunque en la mayoría de los servidores web se pueden añadir otras.

B.2 - AJAX

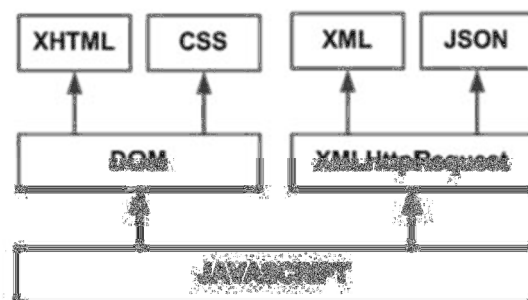
AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML".

Se puede definir AJAX de la siguiente forma:

“Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

Las tecnologías que forman AJAX son: ▪ XHTML y CSS, para crear una presentación basada en estándares.

- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.



Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve un nuevo código XHTML al navegador del usuario.

En el siguiente esquema, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX:

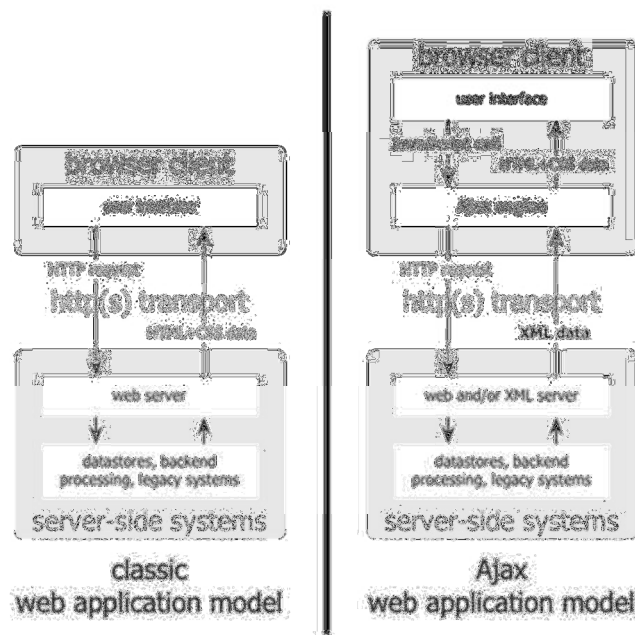


Figura 2. Comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX.

Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

El siguiente esquema muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. La imagen superior muestra la interacción

síncrona propia de las aplicaciones web tradicionales. La imagen inferior muestra la comunicación asíncrona de las aplicaciones creadas con AJAX.

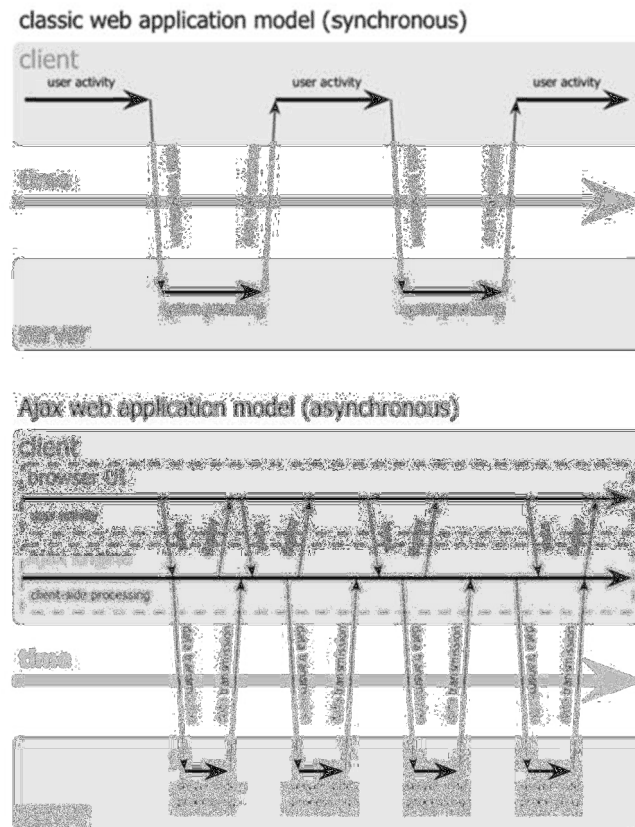


Figura 3. Comparación entre las comunicaciones síncronas de las aplicaciones web tradicionales y las comunicaciones asíncronas de las aplicaciones AJAX.

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

Desde su aparición, se han creado cientos de aplicaciones web basadas en AJAX. En la mayoría de casos, AJAX puede sustituir completamente a otras técnicas como Flash. Además, en el caso de las aplicaciones web más avanzadas, pueden llegar a sustituir a las aplicaciones de escritorio.

B.3 - HOJAS DE ESTILO (CSS)

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML / XHTML bien definidos y con significado completo (también llamados "*documentos semánticos*"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

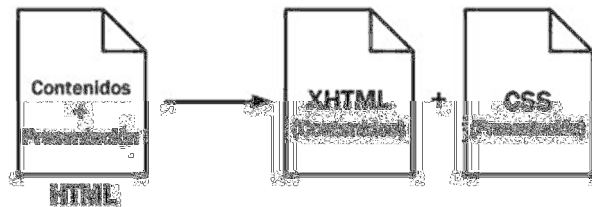


Figura 1. Esquema de la separación de los contenidos y su presentación

Al crear una página web, se utiliza en primer lugar el lenguaje HTML / XHTML para *marcar* los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

La siguiente tabla muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios:

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Internet Explorer	Trident	Completo desde la versión 6.0	Completo desde la versión 8.0	Prácticamente nulo
Firefox	Gecko	Completo	Casi completo	Selectores, pseudo-clases y algunas propiedades
Safari	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Google Chrome	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades

B.4 - JAVASCRIPT

JavaScript es un lenguaje de script multiplataforma orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una

fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

JavaScript desciende en espíritu de una línea de pequeños lenguajes de definición dinámica de tipos tales como HyperTalk y dBASE. Estos lenguajes de script ofrecen herramientas de programación a una audiencia mucho más amplia por su fácil sintaxis, su funcionalidad construida internamente y sus requisitos mínimos para la creación de objetos.

El núcleo de JavaScript contiene un conjunto central de objetos, tales como Array (arreglos), Date (fechas) y Math (objetos matemáticos), además de un conjunto central de elementos del lenguaje tales como los operadores, estructuras de control y sentencias. El núcleo de JavaScript puede ser extendido para una variedad de propósitos complementándolo con objetos adicionales; por ejemplo:

- JavaScript del lado Cliente extiende el núcleo del lenguaje proporcionando objetos para el control del navegador y su Modelo Objeto Documento [Document Object Model] (DOM). Por ejemplo, las extensiones del lado del cliente permiten a una aplicación ubicar elementos en un formulario
- HTML y responder a los eventos de usuario tales como el clic del mouse, entradas del formulario y navegación de páginas.
- JavaScript del lado Servidor extiende el núcleo del lenguaje proporcionando objetos relevantes para la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comunique con una base de datos relacional, proporcionar continuidad de la información desde una invocación de la aplicación a otra o efectuar la manipulación de archivos en un servidor.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java.

B.5 - PHP

PHP “Personal Home Page”, es un lenguaje de programación generalmente usado en la programación de sitios web dinámicos, es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. Actualmente es casi el lenguaje de desarrollo de sitios más usado en todo el mundo. El significado de las siglas PHP ha pasado a ser “PHP Hypertext preprocesor”.



Fue originalmente creado por Rasmus Lerdorf para presentar su portafolio de trabajo y otras cosas en el año de 1994. Su única intención fue la de crear un pequeño sistema de control para verificar el número de personas que leían su currículum vitae en la Web

Originalmente fue desarrollado en perl y para dar más potencia al sistema, Rasmus creó funciones en lenguaje C para permitir conexión a bases de datos. Este fue el comienzo de la potencia real del lenguaje.

En lugar de usar muchos comandos para mostrar XHTML (como en C o Perl), páginas PHP contienen HTML con código incluido en el mismo que hace "algo" (en este caso, mostrar "Hola ¡soy un script PHP!"). El código PHP está entre medio de etiquetas de comienzo y final especiales `<?php y ?>` que nos permitirán entrar y salir del "modo PHP".

```
<body>

  <?php
    echo "Hola, ¡soy un script PHP!";
  ?>

</body>
```

Lo que distingue a PHP de algo lado-cliente como JavaScript, es que el código es ejecutado en el servidor, generando XHTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales.

Características

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Y esto no es todo, se puede hacer mucho más.

Existen principalmente tres campos en los que se usan scripts en PHP.

- Scripts del lado-servidor: Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione. El intérprete PHP (CGI módulo), un servidor web y un navegador. Es necesario hacer funcionar el servidor, con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor web
- Scripts en la línea de comandos: Puede crear un script PHP y correrlo sin necesidad de un servidor web o navegador. Solamente necesita el intérprete PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en Unix o Linux) o el Planificador de tareas (en Windows).
- Escribir aplicaciones de interfaz gráfica: Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si conoce bien PHP, y quisiera utilizar algunas características avanzadas en programas clientes, puede utilizar PHP-GTK para escribir dichos programas

PHP puede ser utilizado en cualquiera de los principales SO del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario PHP de FastCGI, como lighttpd y nginx. PHP funciona ya sea como un módulo, o como un procesador de CGI.

Además, tiene la posibilidad de utilizar programación procedimental, programación orientada a objetos (POO), o una mezcla de ambas.

Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF e incluso películas Flash (usando libswf y Ming) sobre la marcha. También puede presentar otros resultados, como XHTML y cualquier otro tipo de ficheros XML. PHP puede autogenerar éstos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla, creando un caché en el lado-servidor para contenido dinámico.

Una de las características más potentes y destacables de PHP es su soporte para una gran cantidad de bases de datos. Escribir una página web con acceso habilitado a una base de datos es increíblemente simple utilizando una de las extensiones específicas (por ejemplo, para mysql).

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros.

PHP tiene útiles características de procesamiento de texto, las cuales incluyen las Expresiones Regulares Compatibles de Perl (PCRE), muchas extensiones, y herramientas para el acceso y análisis de documentos XML. PHP estandariza todas las extensiones XML sobre el fundamento sólido de libxml2, y extiende este conjunto de características añadiendo soporte para SimpleXML, XMLReader y XMLWriter.

Existen otras extensiones interesantes, pero aquí se comentan las de uso más habitual.

B.6 - JSON

JSON (*JavaScript Object Notation*) es un formato sencillo para el intercambio de información. El formato JSON permite representar estructuras de datos (arrays) y objetos (arrays asociativos) en forma de texto. La notación de objetos mediante JSON es una de las características principales de JavaScript y es un mecanismo definido en los fundamentos básicos del lenguaje.

En los últimos años, JSON se ha convertido en una alternativa al formato XML, ya que es más fácil de leer y escribir, además de ser mucho más conciso. No obstante, XML es superior técnicamente porque es un lenguaje de marcado, mientras que JSON es simplemente un formato para intercambiar datos.

La notación JSON para los arrays asociativos se compone de tres partes:

1. Los contenidos del array asociativo se encierran entre llaves (`{y}`)
2. Los elementos del array se separan mediante una coma (`,`)

3. La clave y el valor de cada elemento se separan mediante dos puntos (:)

Los objetos se pueden definir en forma de pares clave/ valor separados por comas y encerrados entre llaves. Para crear objetos vacíos, se utilizan un par de llaves sin contenido en su interior {}.

A continuación se muestra la notación JSON genérica para crear arrays y objetos:

Arrays: `var array = [valor1, valor2, valor3, ..., valorN];`

Objetos: `var objeto = {clave1: valor1, clave2: valor2, clave3: valor3,..., claveN: valorN};`

La notación abreviada se puede combinar para crear arrays de objetos, objetos con arrays, objetos con objetos y arrays, etc. A continuación se muestran algunos ejemplos de aplicaciones web reales que crean objetos mediante esta notación.

B.7 - APACHE

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. Actualmente se encuentra con una cuota de mercado del 65% (Estadísticas históricas y de uso diario proporcionadas por Netcraft3).

Una de las características más importantes de Apache es que se utiliza en casi todas las plataformas. Al principio únicamente funcionaba con los servidores Web Unix, pero la situación cambió. En la actualidad, no solo funciona con todas las versiones de Unix, sino que también lo hace con Windows 7, Mac OSX Lion, Linux, OS/2.

WAMPSEVER: Apache es el componente de servidor web en la popular plataforma de aplicaciones, junto a MySQL los lenguajes de programación PHP/Perl/Python (y ahora también Ruby). Siendo utilizada la versión 2.0 en el desarrollo del Proyecto.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos. Los programadores de aplicaciones web a veces utilizan una versión local de Apache con el fin de pre visualizar y probar código mientras éste es desarrollado.

Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (front end) del motor de búsqueda Google está basada en una versión modificada de Apache, denominada Google Web Server (GWS).

B.8 - MySQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés), multihilo y multiusuario con más de seis millones de instalaciones. Es un SGBD muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre bajo la GNU GPL para cualquier uso compatible con esta licencia.

Su principal objetivo de diseño fue la velocidad, por ello se suprimieron algunas características de los demás SGBDs, como las transacciones y las subselects. MySQL es un SGBD que ha ganado popularidad por una serie de atractivas características:

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- La API se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Cuenta con un rico conjunto de tipos de datos.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC.



- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.

ANEXO C - SEGURIDAD DEL SISTEMA

C.1 - HTTPS

Una vez que tenemos instalado un gestor de contenidos en nuestro servidor web que incluya autenticación de usuarios, uno de los pasos habituales es cifrar el proceso de autenticación de usuarios para evitar que alguien capture (se suele decir *esnife*) una contraseña de usuario y acceda de forma fraudulenta.

El cifrado de la comunicación entre el navegador y el servidor web se hace mediante el protocolo HTTPS, que tiene las siguientes características principales:

- Utiliza el protocolo SSL (actualmente TLS) para el cifrado de datos.
- El servidor utiliza por defecto el puerto **443/tcp**.
- Utiliza mecanismos de cifrado de clave pública y las claves públicas se denominan certificados.
- El formato de los certificados está especificado por el estándar X.509 y normalmente son emitidos por una entidad denominada **Autoridad Certificadora (CA por sus siglas en inglés)**. En el caso de HTTPS, la función principal de la CA es demostrar la autenticidad del servidor y que pertenece legítimamente a la persona u organización que lo utiliza. Dependiendo de los criterios utilizados para comprobar la autenticidad del servidor se emiten diferentes tipos de certificados X.509.
- El navegador contiene una lista de certificados de CA en las que confía y acepta inicialmente sólo los certificados de los servidores emitidos por alguna de estas CA.
- Una vez aceptado el certificado de un servidor web, el navegador utiliza éste para cifrar los datos que quiere enviar al servidor mediante el protocolo HTTPS y cuando llegan al servidor sólo éste podrá descifrarlos ya que es el único que posee la clave privada que los descifra.
- Las implementaciones modernas usan claves de 128-bit (o 256-bit) para claves de cifrado simétricas

Las dos formas de utilizar el protocolo HTTPS en Apache son: la más simple utilizando un certificado auto firmado (certificado que no está emitido por ninguna CA) y la utilización de un certificado emitido por la organización sin ánimo de lucro **CAcert**. Los pasos que habría que seguir para utilizar un certificado X.509 emitido por una CA comercial son exactamente los mismos que con CAcert, lo único que variaría sería el método que tendría que utilizar la CA para validar el servidor, la persona u organismo que solicita el certificado.

```
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBgQDwjPGyYKND0zQc9d8ysAm0w+IGt9kypWgGaPbzfuyX41ozk36e
U4xVUfUQ8gz3QPj47pNxCihxkgGD1ioF7Fq0/J7eet1++7MIoub7cVsBG6yPrRbY
65zAq7Tg/BJB/Aa3k7t191spbtMs8umKB0+/rRD4bPsgv9iwTD3L8d3QuwIDAQAB
AoGAbPdWhfcI1Ji+/fpukQs3n3ejIzBasptJcv/6dH7BIP9KP6mohDcArhCa+81D
hoxMJvncjmvPUKcL9+5c13eMNTMzomdLASBPmbThHMZMcr6Agpw1BruCjMs1DoUS
1L8AHgNng5gRH9th5fThcdauucvhp0iqbdCH76GqwUyhxaUkCQQD3b5syTatb9NM3
zGar3PbwQa22DBd6GfuLuZrtdvaX+f/XgmaQjiPgj/0oDmDJhyFtl+gp9/tcbzcu
LE+iBkF9AkEA3fn1X2Jqm4fFqyZSDCzpwzk4VMTpqfwaedrhcL60uXAaSSDREsMG
FxcFgB462/WHynQhCGqUIzaxa3djCGjwlwJAbTjNuAELTpcemVXXyGtscZd8V4y0
3EgV2pKeic3mgiVvwrCkF5E20wycQg8+uBa3McpJSCVXxnCtWetjZ3D9fQJAXRwW
IraDhU3EHTCaJeVJAMy1plkasms/KvbnDYXRBN/TBs2vJTGNA5zrjP/+Hfbr77C
hGdAUCOG6s8fSU3LbQJAX5hMWgvuUAktdpwwfMU0G1Gg7+pZMbl6CqpKtNIHseNp
aL2IOX0aoec02hVIUT5xwqRkQ+VUQBs9Nc8a/PI3TQ==
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIBsTCCARoCCQDtJd0YjQsT7DANBgkqhkiG9w0BAQUFADAdMRswGQYDVQQDEExJh
dmFOYXlIuZXhhbXBsZS5jb20wHhcNMTAwMzIxMTEONjA1WhcNMjAwMzE4MTEONjA1
WjAdMRswGQYDVQQDEExJhdmFOYXlIuZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEB
BQADgY0AMIGJAoGBANaM8bJgoOPTNBz13zKwCY7D4ga32TK1aAZo9vN+7JfjWjOT
fp5TjFVR9RdyDPdA+Pjuk3EKKHGSAYPWKqXsWrt8nt562X77swii5vtxWwEbrI+t
FtjrnMCrtOD8EkH8BreTu2X3Wylu0yzy6YoE77+tEPhs+yC/2LBMPcvx3dC7AgMB
AAEwDQYJKoZIhvcNAQEFBQADgYEAWNCqq9X+ykQ7NEQRUvVN06yHhFs+oCz5nPTJ
Qfkr30xtM6qI1wzMiRbakkIUTp5Mni/Nw5t1PTNMKFaGhNZKDnV+LRB1i2cgYk0B
uCJEPGFf1j8+BxeMUhYAX8hxHam86JpkTawUi2+v8QuHGbuLHszoRe3nYZ/nRul/
fgk3Wjk=
-----END CERTIFICATE-----
```

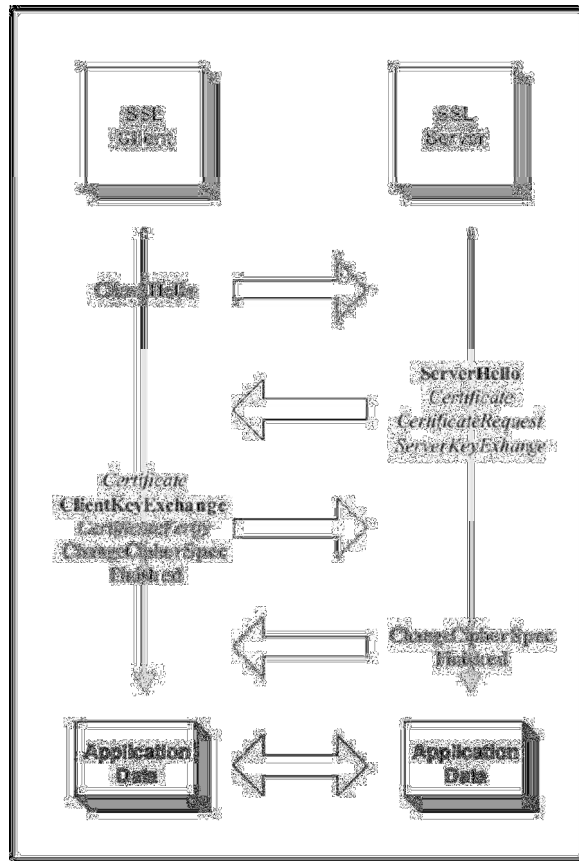
En esta imagen se puede apreciar dos certificados (privado y público) y su contenido. El certificado X.509 esta en formato PEM (base64)

C.2 - SSL/TLS

SSL 3.0, el protocolo dominante en la actualidad en el panorama del comercio electrónico, proporciona confidencialidad, integridad y verificación de la identidad de ambas partes (esta última característica sólo si se utiliza en conjunción de certificados digitales en ambos extremos, cosa que no suele ser frecuente). La disponibilidad es algo que se escapa del ámbito del protocolo y que se ve gravemente perjudicada por lo pesado del mismo (aunque existen soluciones hardware integradas con el y diseñadas a tal efecto). El estándar SSL no contempla la implementación del No Repudio de mensajes, aunque sería muy simple y fácil dar soporte a esta característica.

A grandes rasgos, podríamos decir que SSL trabaja de la siguiente forma: en primer lugar intercambiamos una clave de longitud suficiente mediante un algoritmo de cifrado asimétrico. Mediante esa clave establecemos un canal seguro utilizando para ello un algoritmo simétrico previamente negociado. A continuación, toma los mensajes a ser transmitidos, los fragmenta en bloques, los comprime, aplica un algoritmo hash para obtener un resumen (MAC) que es

concatenado a cada uno de los bloques comprimidos para asegurar la integridad de los mismos, realiza el cifrado y envía los resultados. El estado de todas estas operaciones son controladas mediante una máquina de control de estados. Una sesión SSL puede comprender múltiples conexiones. Adicionalmente, se pueden establecer múltiples sesiones SSL simultáneas.



TLS 1.0, la llamada versión 3.1 de SSL, es el sucesor natural de este. Lo tiene todo para que esta sucesión se realice sin traumas: como el alías que lo acompaña indica (nos referimos a la 'etiqueta' de SSL 3.1) es meramente un SSL mejorado y no propietario que posee mecanismos de compatibilidad con SSL y viene ya de serie con la gran mayoría de los navegadores. Y sin embargo, la inercia en el mundo de los servidores es tal que la transición hacía el no acaba de realizarse. Tanto IIS como Apache, los dos servidores más extendidos (juntos suman más del 90% del total del parque de Internet) llevan soporte de TLS y, sin embargo, los administradores de sistemas siguen decantándose por 'las viejas soluciones'.

Las principales diferencias entre SSL 3.0 y TLS 1.0 son las siguientes:

- En los mensajes *Certificate Request* y *Certificate Verify* del protocolo de *Handshake*. En SSL 3.0 si el servidor solicita un certificado al cliente para que se autentique, este debe de responder con el o con un mensaje de alerta advirtiendo de que no lo tiene. En TLS 1.0 si el cliente no posee certificado no responde al servidor de ninguna forma a este requerimiento.



- Cálculo de las claves de sesión. El mecanismo utilizado para construir las claves de sesión es ligeramente diferente en TLS 1.0.
- TLS 1.0 no soporta el algoritmo de cifrado simétrico FORTEZZA que si es soportado por SSL 3.0. Esto es debido a que TLS busca ser íntegramente público mientras que FORTEZZA es un algoritmo propietario.
- TLS utiliza un mecanismo diferente y más seguro en el cálculo del MAC.
- TLS 1.0 introduce nuevos códigos de alerta no contemplados por SSL 3.0
- TLS 1.0 introduce un nuevo mecanismo en el relleno de los bloques para frustrar ataques basados en el análisis de la longitud de los mensajes.