ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INFORMÁTICA

UNIVERSIDAD CARLOS III DE MADRID

Proyecto Fin de Carrera

Ingeniería Técnica en Informática de Gestión

" DESARROLLO DE UN PORTAL WEB SEGURO: IMPLEMENTACIÓN Y PROTECCIÓN CON TARJETAS INTELIGENTES "

AUTOR: Santiago Fernández Pinilla.

TUTOR: José Maria Sierra Cámara.

AGRADECIMIENTOS

Menciono un agradecimiento especial a mi novia Nuria y a mi familia por el apoyo y la paciencia que han tenido conmigo durante el tiempo que he empleado para desarrollar este proyecto.

Agradezco a mi tutor D. José Maria Sierra Cámara y a mi director de proyecto D. Joaquín Torres Márquez, la ayuda prestada, la flexibilidad de horarios y total disponibilidad que me han permitido mantenerme en continuo contacto para poder realizar este proyecto y compaginarlo con mi vida laboral.

Agradezco a la Universidad Carlos III los medios materiales que ha puesto a mi servicio para poder desarrollar este proyecto, tales como ordenadores, servicios web y especialmente a la biblioteca de la cual he sacado muchos conocimientos e ideas.

Agradezco al Comité Regional de Madrid de la Unión Federal de Policía y en especial a su secretario regional D. Alfredo Perdiguero y a su responsable de informática D. Carlos por el apoyo prestado y flexibilidad de ideas, que han permitido el enriquecimiento de este proyecto.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

1 - INTRODUCCIÓN

1.1 - INTRODUCCIÓN AL PROYECTO

Este proyecto es realizado para un organismo sindical, en concreto para el Comité Regional de Madrid perteneciente a la Unión Federal de Policía (U.F.P.), ya que este necesita reemplazar su página web basada en HTML, la cual tiene muy limitada su funcionalidad y protección, por un sistema web más robusto en seguridad y que proporcione a sus usuarios más funcionalidad.

La finalidad de este proyecto es, por tanto, diseñar e implementar un portal web por medio de la herramienta CMS JOOMLA, que cumpla con las expectativas de dicho sindicato, para lo cual se instalaran y configurarán en dicho portal extensiones que ofrezcan funcionalidades tales como un Foro, un Calendario de Eventos y otros, que darán solución a todos los requisitos funcionales que se demandan, los cuales quedarán perfectamente integrados en un único entorno web.

Un apartado esencial del desarrollo de este proyecto es la atención constante a la seguridad del sistema en todas sus facetas, para lo cual se configuraran de manera exhaustiva el servidor Apache y el portal JOOMLA además, se establecerán canales de comunicación segura mediante el protocolo SSL, previniendo así ataques y manteniendo la confidencialidad de la información.

1.2 - MOTIVACIÓN

La motivación de este proyecto, es definir de forma entendible la configuración necesaria para que el sistema pueda albergar una aplicación web de forma segura, de modo que siguiendo dichas descripciones se pueda implementar en el futuro un sistema seguro con capacidades similares a las descritas en el proyecto o modificar ciertos parámetros para que se amolde a nuevas características o requerimientos. Esto es posible, ya que el proyecto no se ha limitado a describir la configuración de los distintos elementos, sino a proponer alternativas de configuración que podrían ser válidas en otros contextos, de ahí que se trate de un diseño flexible.

Con respecto al diseño e implementación del portal y del foro, se ha hecho hincapié en cómo se ha creado la aplicación en base a los requisitos de usuario y software planteados y en el diseño se hace una descripción de cómo implementarlo con las tecnologías previstas, de modo que sea fácil reproducir los resultados obtenidos en este proyecto para futuros desarrollos.

Debido a que, en gran medida, este proyecto se enfoca para que un Organismo Sindical de la Policía pueda cumplir con servicios a sus afiliados y pueda ser posteriormente implantado y utilizado, es importante adjuntar una documentación completa y de fácil entendimiento sobre el manejo, a todos los niveles, de dicha aplicación. Por ello, en los apéndices se adjunta la documentación tanto de manejado a nivel de usuario como de administrador del portal web, del foro y del calendario de eventos.

1.3 - ENFOQUE DEL PROYECTO

En este proyecto se intenta plantear de forma sistemática la implementación de todo lo mencionado hasta ahora, puesto que la finalidad es que este proyecto sirva para que dicho Sindicato de Policía pueda, a partir de todo este estudio, usar, ampliar o modificar cualquier aspecto que se haya desarrollado. Por tanto, y debido a las necesidades, hay dos aspectos fundamentales a tratar: la seguridad, que debe estar presente en todas las facetas del sistema y, una segunda, que el sistema sea lo suficientemente flexible como para admitir todo tipo de ampliaciones y modificaciones.

Por ello, la configuración se ha realizado con el siguiente software libre que permite cumplir con esos requisitos:

- ✓ Apache (Servidor Web).
- ✓ MySQL (Sistema Gestor de Base de Datos).
- ✓ PhpMyAdmin (Herramienta en lenguaje PHP para facilitar la gestión de MySQL).
- ✓ OpenSSL (Paquete que nos permite crear certificados de seguridad y funcionalidades del protocolo SSL - Secure Sockets Layer).
- ✓ JOOMLA (Sistema Gestor de Contenidos).
- ✓ Cryptokit (Software desarrollado por la Fabrica Nacional de Moneda y Timbre para manejar Tarjetas Inteligentes.

Se ha usado JOOMLA y extensiones del mismo para la implementación de la aplicación web en lugar de crear una nueva desde cero, ya que estas aplicaciones de libre distribución son lo suficientemente potentes y están adecuadamente testeadas como para poder ofrecer un resultado mucho más adecuado para dicho Sindicato que cualquier implementación propia que se realizará para un proyecto, pues la implementación de un sistema similar excedería, y con mucho, los límites de este proyecto.

Para finalizar, y siguiendo con la naturaleza orientada al usuario final de este proyecto, se adjuntan unos apéndices que contienen manuales donde se describe, de forma detallada, cómo hacer cualquier operación en el portal y en el foro, de modo que cualquier usuario pueda usar esos apéndices para aprender a manejarse en dicha aplicación web.

1.4 - OBJETIVOS

Como ya se ha dicho, los objetivos son la creación de un portal web y un foro que sean seguros (de cara a la confidencialidad de la información y prevención de ataques) y que den todo el soporte necesario para mantener una comunidad dentro del Sindicato de Policía U.F.P. Desglosando un poco esos objetivos podríamos dividirlos del siguiente modo:

- **Estudio de requisitos software** planteados para el foro y el portal. Es necesario realizar un estudio de requisitos software para conocer las necesidades reales de U.F.P. (Sindicato de Policía al cual se le va a implementar el sistema). Por medio de estos requerimientos, se podrá hacer un estudio (en conjunción con las propiedades que ofrecen las herramientas a usar), para así determinar cómo debe llevarse a cabo el desarrollo del proyecto.

- **Selección de las herramientas software más adecuadas para** acometer los requisitos especificados.

  - Para el sistema: Se optará por herramientas (como se explicará más adelante) básicamente por ser software libre o gratuitas y por mostrar una calidad contrastada. Además, con el uso de ellas podrán afrontarse todos los requisitos (no funcionales en general) planteados de la forma más satisfactoria posible.

    - ✓ Un Servidor Web: ampliamente usado y que permita una configuración segura y eficiente.
    - ✓ Un Protocolo de comunicaciones seguras: por medio del cual podremos hacer comunicaciones cifradas con el sistema que implementemos.
    - ✓ Un Sistema Gestor de Bases de Datos: ampliamente usado y que permita una buena integración con aplicaciones web.
    - ✓ Un Lenguaje de Programación: ampliamente usado en el desarrollo de aplicaciones web.

  - Para la aplicación: JOOMLA es un Gestor de Contenidos (CMS) que permite crear portales web de manera relativamente rápida, sencilla y profesional. Además son muy configurables, por lo que podrán adaptarse bien a futuras necesidades. En sus respectivos apartados se comentará más sobre los porqués de estas elecciones.

- **Configuración del sistema atendiendo a la seguridad.**

  - Credenciales Seguras:

    - Uso de certificados (x509) por parte del servidor. Como credencial electrónico de las partes. Es el certificado más común usado por entidades. Es admitido por todos los navegadores Web y es la forma más segura y habitual de intercambiar claves públicas de entidades.

  - Algoritmos Seguros:

    - Uso de RSA como algoritmo de firma empleado en la autenticación. La autenticación RSA es una de las más seguras conocidas, ya que su fortaleza radica en claves enormes (se usarán de al menos 1024 bits) y la solución de un problema para el que no hay algoritmo de solución conocido, el del logaritmo discreto. Por tanto, el uso de RSA en conjunción con los certificados antes mencionados es una de las formas más seguras de conseguir una autenticación con garantías.

    - Uso de AES 256 bits para la codificación de las comunicaciones. AES es el estándar de cifrado con clave simétrica, se le considera tan seguro que el Departamento de Defensa Americano lo considera válido para cifrar su información secreta. Usaremos la versión de 128 bits (en el protocolo SSL) para garantizar así la máxima privacidad en las conexiones que nuestro servidor establezca con los distintos clientes.

    - Uso de Diffie-Hellman para el intercambio de claves privadas. Este algoritmo se basa, al igual que RSA, en el concepto de clave pública. Es la mejor forma de intercambiar claves privadas entre servidor y cliente, sin que terceros puedan reconocerla. La fortaleza del sistema radica en el mismo principio que RSA.

  - Configuraciones Seguras:

    - Crear una configuración segura para el servidor. Es importante tener claros ciertos conceptos de un servidor, para así poder usarlo de la forma más adecuada posible, de modo que tenemos que tener en cuenta temas importantes como que tipo de instalación realizar, los permisos sobre directorios y ficheros o habilitar módulos como el “mod_ssl” para capacitar al servidor para realizar conexiones cifradas y realizar otras acciones.

  - Acceso Seguro:

    - Mediante Tarjetas Inteligentes.

- **Diseño de la aplicación estandarizado.**

  - Estándar de Ingeniería del Software "ESA PSS-05-0": utilizado por la Agencia Espacial Europea. Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en la siguiente URL de Internet: http://www.esa.int/esaCP/index.html

  - Toma de Requisitos: En la fase de toma de requisitos de usuario, o fase de definición del problema, se definirán el ámbito y el alcance del sistema, es decir, lo que se espera que haga el sistema. Los requisitos de usuario son de dos tipos: de capacidad y de restricción.

  - Diagramas de Casos de Uso: Explica gráficamente la secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

  - Diagrama de Arquitectura: Muestra gráficamente la estructura y relación de las tecnologías software utilizadas en el proyecto.

  - Diagramas de secuencia: Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

  - Diseño gráfico del portal: acorde a los requerimientos. Una vez que se tiene realizado un estudio de lo que se necesita y de quién y cómo lo va a usar, se podrá realizar un diseño de la aplicación teniendo en cuenta las herramientas (y también las limitaciones) que nos ofrece el CMS JOOMLA que se usará para el desarrollo de dicha aplicación.

- **Implementación de la aplicación.**

  - Implementación del portal y del foro usando, para ello, todas las extensiones y plugins de JOOMLA que sean necesarios. La implementación del portal realizada con JOOMLA no es suficiente para satisfacer el cumplimiento íntegro de todos los requisitos funcionales, por lo que debemos recurrir a los componentes. Los componentes son pequeños programa PHP que se integran dentro del portal para así facilitarnos nuevas funciones con las que dar solución a todos los requerimientos planteados. De estos componentes, cabe destacar el foro y el calendario, gracias al cual se pueden crear citas, reuniones y cualquier tipo de evento señalado en una fecha y horario concreto.

- **Testing de la aplicación.** Comprobar la viabilidad de todas las instalaciones y configuraciones por medio de una batería de pruebas exhaustiva. Se necesita la realización de una batería de pruebas que contemple varios aspectos: primero, que garantice el funcionamiento adecuado de los elementos de la preconfiguración, como son Servidor Web, Protocolo SSL, etc. Luego, necesitamos testear el adecuado funcionamiento del portal y del foro, tanto a nivel de usuario como de administrador.

Finalmente, hay que comprobar que toda la seguridad configurada cumple adecuadamente su papel.

- **Líneas futuras de mejora e implementaciones** para añadir funcionalidades al sistema. Es un objetivo de este proyecto el plantear una serie de ampliaciones y mejoras que podrán implementarse en un futuro. Estas ideas se expondrán con más detalle en su correspondiente sección.

- **Hacer un estudio de planificación y presupuestos sobre el trabajo realizado**. Para finalizar los objetivos, planteamos la realización de un estudio (usando MS Project) con el que poder analizar las diferentes fases de las que éste ha constando y hacer una representación esquemática de la duración de esas fases. Esto es bastante útil, ya que con esa información se modelan los esfuerzos/costes que supone hacer una implementación de las características que tiene este proyecto.

1.5 - ESTRUCTURA DE LA MEMORIA

La memoria tiene tres partes diferenciadas que describiremos a continuación para así facilitar su lectura y poder entender mejor el porqué de la distribución escogida.

1) En la primera, se tratan los aspectos de configuración de los diferentes elementos que componen el sistema, tales como: el servidor web Apache, la configuración de SSL, etc. Es, por tanto, la parte con la que conseguimos dejar un sistema listo para poder implementar el portal y el foro. Principalmente se encarga de describir configuraciones de ficheros, directivas y hace especial mención al aspecto de la seguridad.

2) En la segunda parte, se tratan los aspectos de diseño e implementación del portal. Se hará, en primer, lugar un diseño por medio de diagramas UML. Luego, se hará un diseño gráfico de las aplicaciones y, finalmente, se describirá la implementación por medio del CMS JOOMLA.

3) En la tercera parte, se abordan los temas finales, que son principalmente cómo hacer copias de seguridad del portal: un estudio del desarrollo del proyecto, con el que podremos ver las diferentes fases y su duración, una batería de pruebas con la que comprobar el correcto funcionamiento de todo lo implementado y configurado, una definición de posibles líneas futuras a implementar para la continuación de este proyecto y, finalmente, un apéndice con información de todas las tecnologías usadas, definición de su uso y otros aspectos que se hayan considerado interesantes mencionar, para así poder entender mejor el proceso de desarrollo que se ha realizado.

2 - TECNOLOGÍAS EMPLEADAS

2.1 - DESCRIPCIÓN DE APACHE 2

Apache 2 es un servidor web. Esto quiere decir que puede servir contenido "web" por medio de peticiones HTTP o HTTPS a una serie de puertos preestablecidos. Pero ¿qué quiere decir exactamente servir contenido web? Es sencillo, no es nada más que transferir vía TCP documentos HTML (también hay otros tipos soportados) con la información solicitada por el cliente. Realmente Apache lo que hace es enviar el contenido del directorio solicitado por el cliente en formato HTML. Es decir, si hay una página web enviará dicha página. En caso de no haberla, enviará otro tipo de documento HTML como puede ser un aviso de error, una lista de ficheros presentes en el directorio, etc.

Cuando nosotros, como clientes, hacemos una petición web con nuestro navegador a una dirección del tipo www.dominio.extensión lo que hacemos es pedirle a Apache (si ese domino está configurado con apache claro está) que nos envíe el contenido del directorio que tiene configurado por defecto como directorio web, de modo que, si como en nuestro caso, el directorio web es /www, Apache lo que hará será enviar el contenido de ese directorio en formato HTML y si solicitásemos por ejemplo el siguiente contenido www.dominio.extensión/dirección, Apache en nuestro caso, serviría el contenido de /www/dirección.

Por medio de las directivas de configuración de Apache se puede definir cuándo queremos que apache muestre el contenido de ficheros del directorio, cuándo no o cuándo queremos que nos muestre un mensaje de error.

Apache para mostrar el contenido web "tradicional", lo que se entiende por una página web, lo que necesita que haya un documento index.html (esto se puede configurar) en el directorio solicitado. Siempre que exista ese fichero en un directorio, apache, por defecto, será lo que nos envíe. De modo que un portal web (o cualquier otro tipo de página) se construye a base de ir creando diferentes index.html para cada uno de los directorios web.

En el caso de nuestro portal, no se usan generalmente estos documentos index.html, sino index.php que se describirá en la sección de PHP qué significa y cómo funcionan.

Apache es un servidor web capaz de dar servicio en sesiones (si se activa) a gran número de usuarios. Para cada usuario el servidor lo que hace es crear un "hilo" (proceso ligero) que se encargará de atender al cliente. Por tanto de cara a la carga del servidor es conveniente sopesar la capacidad de la máquina, de la red a la que está conectada, para así evitar colapsos por medio de configuraciones erróneas.

Apache es el servidor web más popular del mundo y por medio de módulos y ampliaciones puede dar soporte a gran número de tecnologías (como PHP en nuestro caso). Además es un servidor de código abierto y totalmente compatible con la plataforma Windows. Además Apache soporta SSL, indispensable para mantener conversaciones seguras entre el servidor y los distintos clientes que a él acceden. Por todo esto se convierte en la mejor elección posible para nuestras necesidades.

Apache se compone de un núcleo (tiene toda la base del servidor) y una serie de complementos, llamados módulos, que dan soporte a ciertos aspectos no cubiertos por el servidor base.

Un ejemplo de arquitectura del servidor podría ser el siguiente:

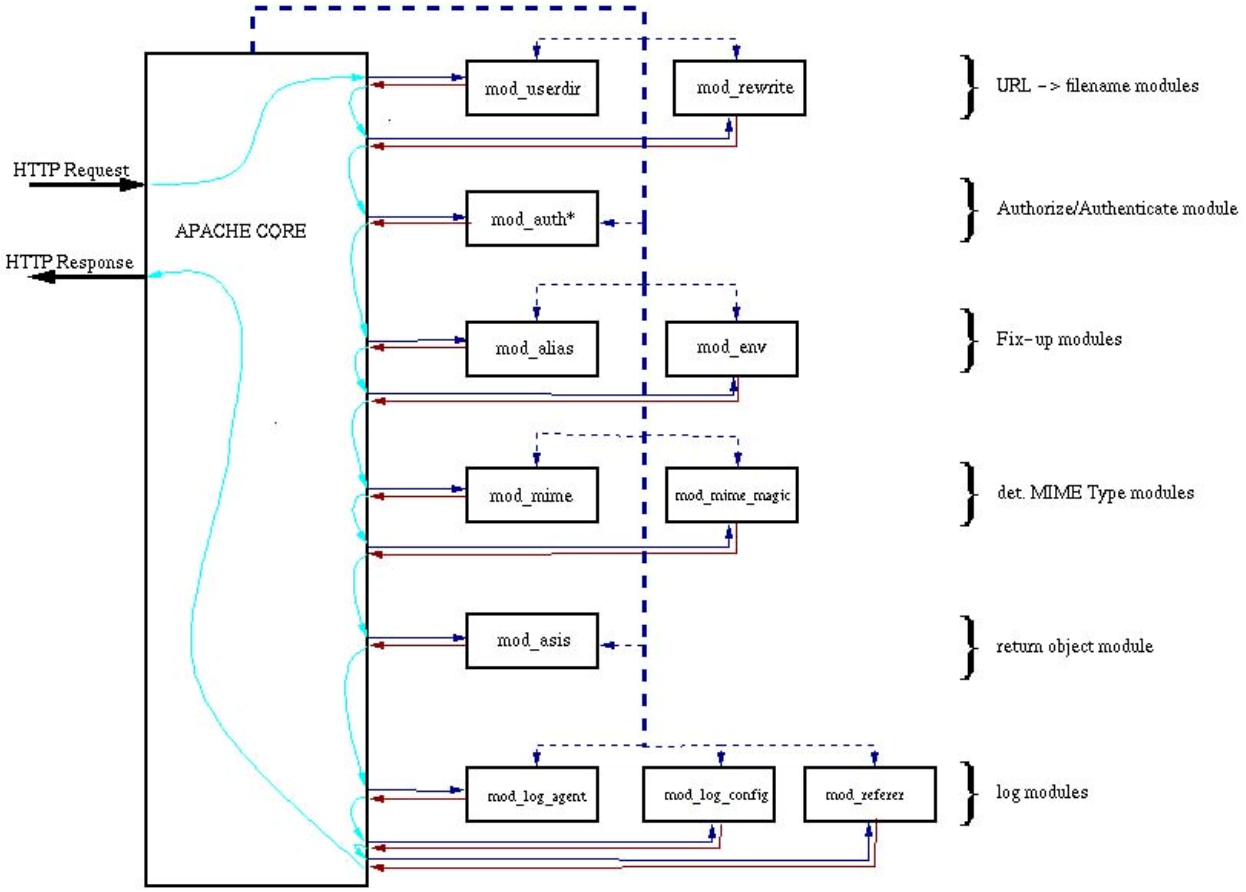


Figura 1: Esquema general del Servidor Apache

En él, se observa como es el núcleo de Apache, que se encarga de atender (escuchar / responder) todas las peticiones (tráfico HTTP). Cuando una acción específica no puede ser atendida por el núcleo del servidor, éste activa el módulo correspondiente y transmite a dicho módulo la información, para que sea él el que se encargue de atender el aspecto en cuestión. Una vez el módulo ha realizado la tarea, da la respuesta al núcleo de Apache, para que sea este quien se encargue de responder al cliente.

2.2 - PROTOCOLO SSL

SSL (o Secure Socket Layer) es un protocolo que nos permite intercambiar datos por Internet de forma segura (por medio de la criptografía). Se usará en este proyecto la implementación abierta del protocolo, que es OpenSSL, ya que además de gratuito, permite interactuar perfectamente con Apache (necesario para HTTPS).

La finalidad del protocolo SSL no es otra que cifrar los datos HTML que se intercambian entre el cliente y el servidor. Para ello, se realizan los siguientes pasos:

1) El cliente envía un "ClienteHello" con la lista de algoritmos de cifrado simétricos que soporta.

2) El servidor le contesta con un “ServerHello" indicando al cliente qué algoritmo de cifrado va a usar (selecciona el más seguro de cuantos le ofreció el cliente).

3) Luego, cliente y servidor intercambian los certificados (no es necesario que el cliente tenga certificado).

4) Una vez que el cliente tiene el certificado del servidor comprueba su autenticidad con la clave pública de la CA (Autoridad de Certificación) que lo firmó.

5) Después de acordado el algoritmo de cifrado y la aprobación de los certificados, intercambian la clave por medio del algoritmo asimétrico Diffie-Hellman (algoritmo de intercambio de claves). Esto comporta que nadie pueda interceptar la clave privada que comparten.

6) Ahora el cliente y el servidor pueden intercambiar información de forma secreta.

El esquema de la siguiente página nos muestra de forma detallada todos estos pasos que realiza el Protocolo SSL para establecer una comunicación segura.

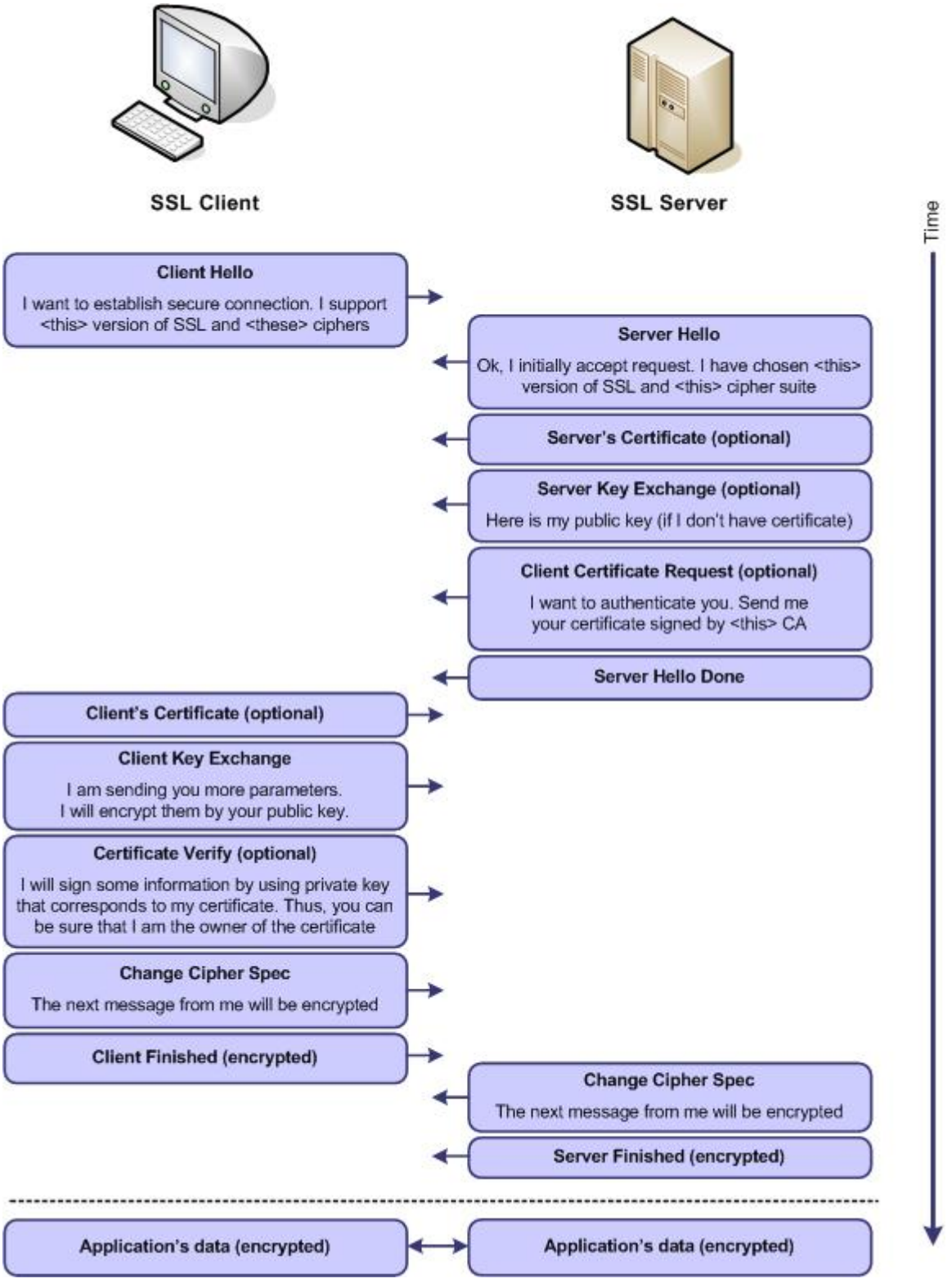


Figura 2: Negociación SSL

El anterior esquema (obtenido de la especificación del protocolo SSL) Nos muestra el intercambio de información previa a la conexión segura entre cliente y servidor (es una forma gráfica de describir lo que ya se mencionó antes).

SSL es por tanto un protocolo (que da soporte al nivel de aplicación instalándose por debajo de él).

SSL permite crear conexiones seguras para el intercambio de datos del nivel de aplicación, pero no solo para Web (que es para lo que se usa en este proyecto), sino que también permite dar soporte seguro a otros protocolos de aplicación como SMTP (correo) o NNTP (noticias).

2.3 - MYSQL

MySQL es un gestor de bases de datos relacionales multiusuario y libre (importante razón para su selección como SGBD).

MySQL nos permite realizar todas las operaciones de un sistema gestor tanto a nivel local como de forma remota (esta característica por motivos de seguridad no estará disponible en la implementación de este proyecto).

MySQL es el más importante gestor de bases de datos relacionales de soporte Web. Por tanto, se convierte en una aplicación fundamental en un proyecto como el que se realiza.

MySQL es multiusuario, requisito vital para nuestra futura aplicación y además es concurrente. Esto es importante ya que en general, habrá dos aplicaciones constantemente accediendo a la base de datos (portal y foro) y además serán muchas versiones de estas apariciones (diferentes hilos que soliciten acceso a la base de datos).

MySQL es un gestor mucho menos denso que otros existentes como Oracle, de modo que con una funcionalidad más limitada (suficiente para el uso Web) nos permite hacer ejecuciones mucho más rápidas y menos pesadas para el sistema. Por tanto lo convierten en ideal para este tipo de entornos.

Los principales motivos por los que se ha elegido este gestor de bases de datos son debidos a que la implementación del Portal Web se realizará con el CMS JOOMLA (que requiere MySQL para su instalación y uso) y también el foro, que se desarrollará con el componente denominado “FireBoard Forum” que será instalado, configurado e integrado en el mismo CMS JOOMLA (que también requiere MySQL).

Además, este sistema de bases de datos tiene total compatibilidad con las tecnologías que se usan en este proyecto, tales como Apache y principalmente PHP (incluyendo la aplicación en PHP y PHPMyAdmin).

MySQL ofrece la ventaja que tiene implementaciones para la gran mayoría de plataformas (entre ellas Linux, Windows y Mac OS).

Es importante destacar la aplicación PHPMyAdmin, ya que nos permite administrar la base de datos de manera gráfica y sencilla por medio de un navegador Web. No es algo suficientemente importante como para detentamos por este sistema gestor, pero si se convierte en un punto a su favor

Por defecto está creada la base de datos y es localhost (este dato es importante para la instalación del Portal y el Foro).

El esquema de la siguiente página nos muestra de forma detallada como esta estructurado el Sistema Gestor de Base de Datos MySQL.

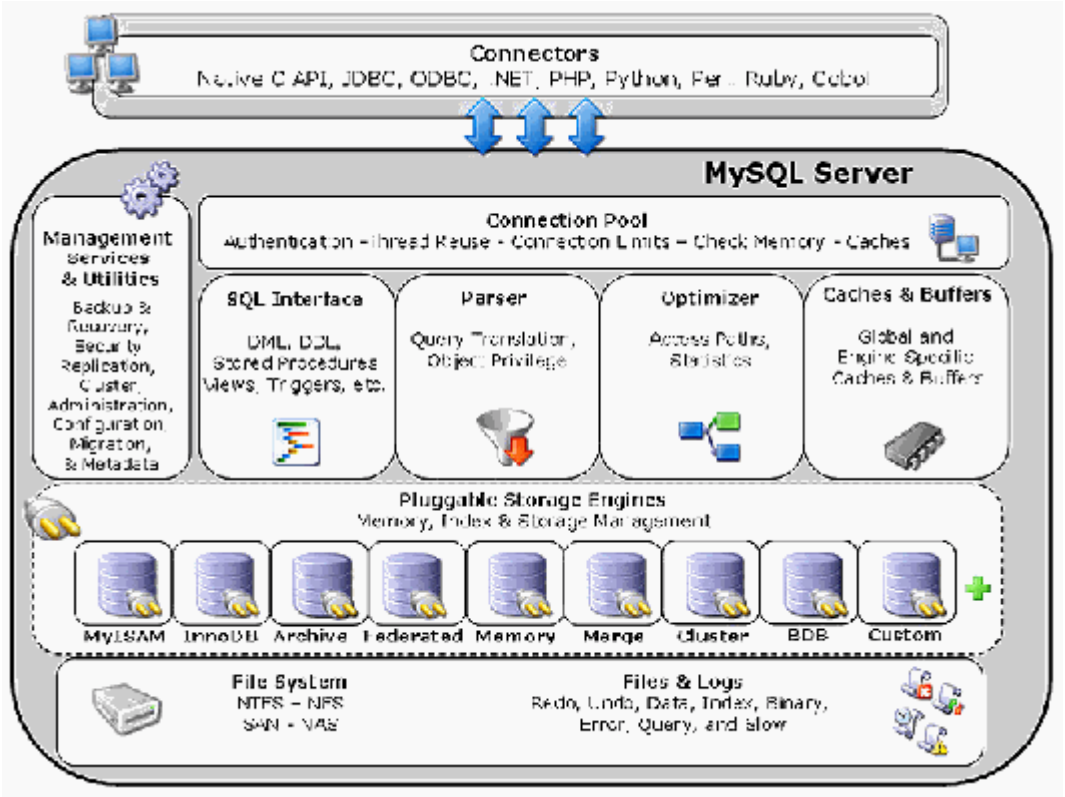


Figura 3: Esquema general MySQL

2.4 - PHP

PHP (Hypertext Pre-processor) es un lenguaje de programación interpretado (no se compila, sino que un intérprete lee línea a línea ejecutando las sentencias que encuentra), que se usa principalmente para la creación de aplicaciones Web.

PHP es lo que se denomina programación del lado del servidor, ya que el programa (contenido en un script) lo ejecuta el servidor (que contiene el intérprete de PHP) y envía el resultado de la ejecución al cliente. Por tanto, el cliente envía y recibe datos de la aplicación PHP sin necesidad de tener soporte alguno para dicha tecnología.

PHP permite una total integración con el servidor web Apache, de modo que nuestro servidor, por medio de módulos, puede interactuar con PHP para así ejecutar aplicaciones Web, enviando el resultado al cliente.

Además PHP tiene soporte completo para interactuar con el sistema gestor de bases de datos MySQL (por medio de otro módulo), por lo que entre los tres elementos citados (Apache, MySQL y PHP) tenemos el soporte suficiente para crear aplicaciones Web robustas y además poder interactuar con los clientes (por medio del servidor Apache).

PHP es software libre (como todo lo que se ha venido usando en este proyecto) que además de dar soporte a la plataforma Windows (por lo cual nos interesa) también se lo da a otras como Linux o Mac OS. También tiene soporte para otros sistemas gestores de bases de datos como Oracle, pero debido a nuestras necesidades sólo usaremos el soporte con MySQL.

Finalmente se ha decidido usar PHP en vez de otras tecnologías similares, debido a que el CMS JOOMLA (gestor de contenidos para el desarrollo de Portales Web) y el foro (implementado mediante el componente de JOOMLA “FireBoard Forum”) requieren PHP, ya que son aplicaciones que se han implementado en dicho lenguaje.

Los pasos que realiza el interprete de PHP en interacción con el servidor son los siguientes:

1) El navegador del cliente solicita una página de contenido PHP al servidor.

2) El servidor localiza dicha página y se la remite al intérprete de PHP para que este la traduzca.

3) El intérprete de PHP traduce el código PHP existente.

4) El intérprete de PHP devuelve al servidor la página como HTML.

5) El servidor devuelve el documento HTML al cliente.

El esquema de la siguiente página nos muestra de forma detallada como realiza sus funciones el intérprete de PHP, dando respuesta a las peticiones que le realiza el servidor.

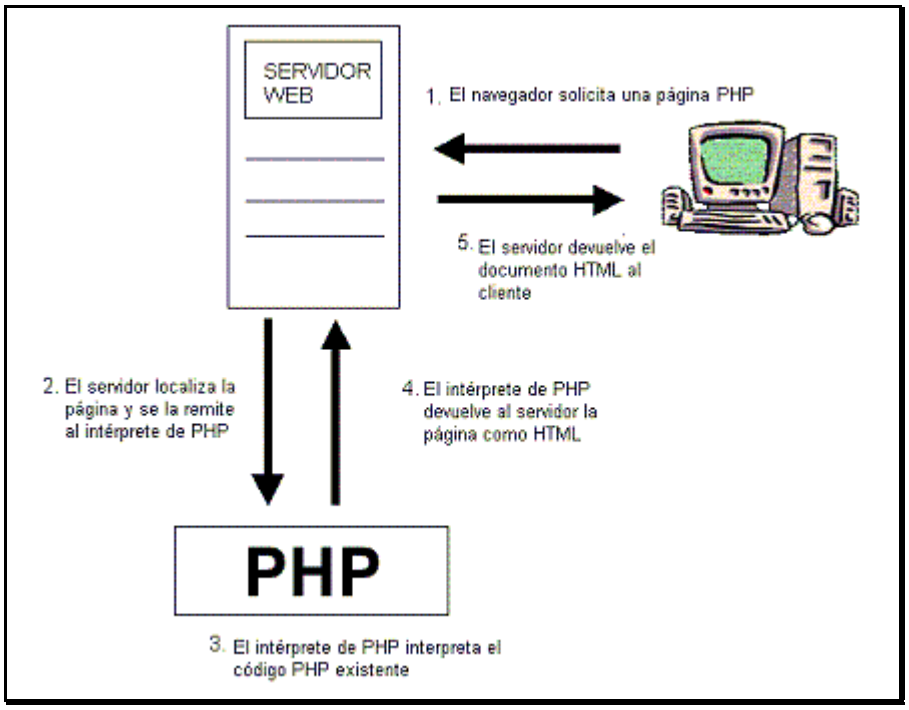


Figura 4: Diagrama del funcionamiento de PHP

2.5 - PHPMYADMIN

PhpMyAdmin es un programa de libre distribución creado mediante el lenguaje de programación PHP.

Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.

La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas.

Es por tanto un entorno gráfico y sencillo con el que podremos administrar la inmensa mayoría de los aspectos de la base de datos de forma remota y segura sin tener que estar físicamente en el propio sistema.

La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

Este último punto (Backup de la Base de Datos) será fundamental para nuestro proyecto, ya que todo el portal y el foro será desarrollado en una base de datos local, que en un futuro habrá que trasladar a un HOST perteneciente a el Sindicato de Policía U.F.P., para lo cual hará falta exportar dichos datos desde PhpMyAdmin en el ordenador local e importarlos al servidor remoto de MySQL que tenga contratado dicho sindicato.

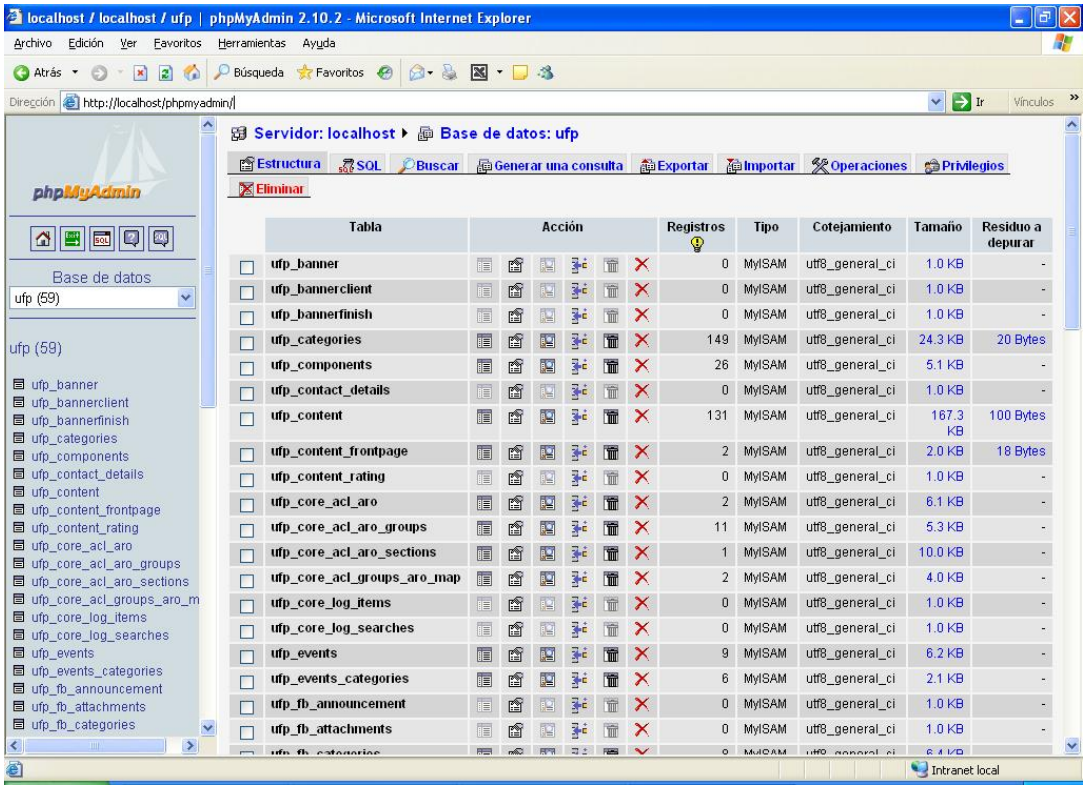


Figura 5: Pantalla de phpMyAdmin

2.6 - JOOMLA

JOOMLA es un CMS (Sistema de gestión de contenidos) implementado en PHP y que usa, como gestor de bases de datos para el contenido, MySQL.

Se ha elegido este CMS para implementar el portal por varios motivos:

- ✓ El primero, permite, de forma sencilla, hacer una implementación rápida y elegante de un portal web.
- ✓ El segundo, por medio de la instalación de nuevos componentes, se le puede añadir una gran cantidad de funcionalidades (de hecho muchos de estos componentes se usarán durante este proyecto).
- ✓ El tercero, permite una gestión del contenido basada en roles de usuario, de modo que ciertos perfiles de usuario tienen permiso para realizar determinadas acciones, mientras que el resto no.
- ✓ El cuarto, por medio de esos roles (cada uno en su medida) se permite una interactuación muy grande entre la aplicación que se monte y el usuario, algo que es vital para la implementación de un portal de un Sindicato policial, que básicamente su contenido dinámico serán noticias y artículos que van apareciendo.
- ✓ El quinto motivo por el que se ha elegido JOOMLA, es por la cantidad de plantillas disponibles que hay para la personalización del portal.
- ✓ Finalmente, se ha escogido este CMS por la gran configurabilidad que hay de cara al administrador.

Además del contenido dinámico (principalmente noticias y artículos) que se ha mencionado, JOOMLA nos permite tener una serie de contenidos estáticos dentro del portal, dándole el aspecto de una página web convencional.

Debido a que es muy personalizable, podemos crear un portal que además de ser elegante, visualmente hablando, cumpla con todos los requisitos software que se nos plantearon y todo ello de forma que sea intuitivo, sencillo y agradable al usuario.

Con JOOMLA básicamente se implementará el portal web básico con el que poder crear noticias, registros de usuarios, contenido estático, personalizar el portal, etc. y por medio de los plugins y componentes, se podrán añadir funcionalidades extra, ya sean funcionales o no funcionales, como pueden ser la posibilidad de añadir comentarios a noticias, calendarios, elementos visuales, gestores para subir contenido, etc.

La estructura básica de cara al usuario de un portal implementado en JOOMLA consiste en una serie de menús, botones y migas de pan que facilitarán la navegación por el contenido de dicho portal. En la portada de portal, se encuentran aquellos contenidos de mayor importancia, ya sean los más visitados, los más modernos o cualquier otra fórmula que el administrador del sitio haya decidido. Por tanto, usaremos esa página principal para incluir todo aquello que sea relevante: paneles de registro, acceso a los principales contenidos y resúmenes de los principales contenidos dinámicos ordenados por fecha.

De cara al administrador, se dispone de un panel (con acceso mediante contraseña) desde el que fácilmente se pueden modificar todos los aspectos del portal e incluso añadir o quitar componentes de éste.

Por motivos de seguridad, usaremos el conjunto de roles que JOOMLA ofrece, para evitar que cualquiera pueda insertar noticias o artículos en el portal, ya que podría saturarlo y entorpecer el uso comunitario de éste, sin mencionar del control legal que tiene que mantener un sindicato de policía.

Toda la información referente a cómo instalar, configurar y usar un portal web creado en JOOMLA se trata en los apartados de implementación del portal web y en los apéndices.

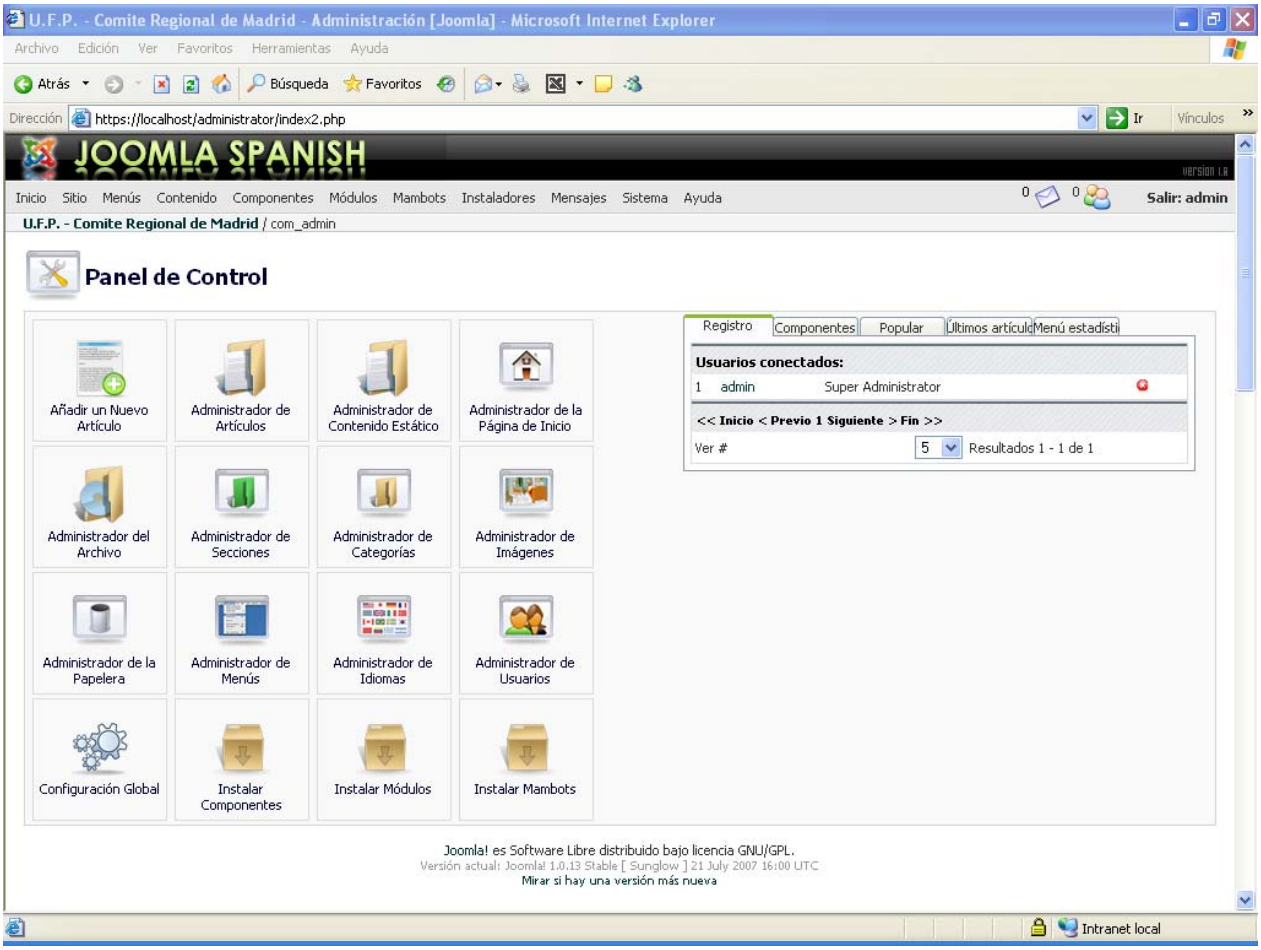


Figura 6: Back-End del CMS JOOMLA

2.7 - TARJETAS INTELIGENTES

Una tarjeta inteligente (smart card), o tarjeta con circuito integrado (TCI), es cualquier tarjeta del tamaño de un bolsillo con circuitos integrados incluidos que permitan la ejecución de cierta lógica programada.

La percepción estándar de una tarjeta inteligente es una tarjeta microprocesadora de las dimensiones de una tarjeta de crédito (o más pequeña, como por ejemplo, tarjetas SIM o GSM) con varias propiedades especiales (como por ejemplo un procesador criptográfico seguro, sistema de archivos seguro, características legibles por humanos) y es capaz de proveer servicios de seguridad (como por ejemplo confidencialidad de la información en la memoria).

Las tarjetas no contienen baterías; la energía es suministrada por los lectores de tarjetas.

Figura 7: Lector de Tarjeta Inteligente

Tipos de tarjetas según sus capacidades: Según las capacidades de su chip, las tarjetas más habituales son:

1) Memoria: tarjetas que únicamente son un contenedor de ficheros pero que no albergan aplicaciones ejecutables. Por ejemplo, MIFARE. Éstas se usan generalmente en aplicaciones de identificación y control de acceso sin altos requisitos de seguridad.

2) Microprocesadas: tarjetas con una estructura análoga a la de un ordenador (procesador, memoria volátil, memoria persistente). Éstas albergan ficheros y aplicaciones y suelen usarse para identificación y pago con monederos electrónicos.

3) Criptográficas: tarjetas microprocesadas avanzadas en las que hay módulos hardware para la ejecución de algoritmos usados en cifrados y firmas digitales. En estas tarjetas se puede almacenar de forma segura un certificado digital (y su clave privada) y firmar documentos o autenticarse con la tarjeta sin que el certificado salga de la tarjeta (sin que se instale en el almacén de certificados de un navegador web, por ejemplo) ya que es el procesador de la propia tarjeta el que realiza la firma. Un ejemplo de estas tarjetas son las emitidas por la Fábrica Nacional de Moneda y Timbre (FNMT) española para la firma digital.

Este último tipo de tarjetas criptográficas de la Fábrica Nacional de Moneda y Timbre son las que utilizaremos para desarrollar este proyecto y cuya finalidad será la de que el administrador del portal pueda identificarse contra el servidor desde cualquier ordenador que se encuentre conectado a Internet, proporcionando así una doble seguridad:

1) Primero, el administrador de este portal deberá conocer los parámetros “USUARIO” y “CONTRASEÑA”, para poder chequearse en el Back-End del portal como administrador del mismo.

2) En segundo lugar, deberá portar su tarjeta inteligente con el certificado SSL en su interior que permita identificarse y establecer una conexión segura cifrada con el servidor en el que se encuentra el portal, sin la tarjeta será imposible acceder a dicha zona de administración.

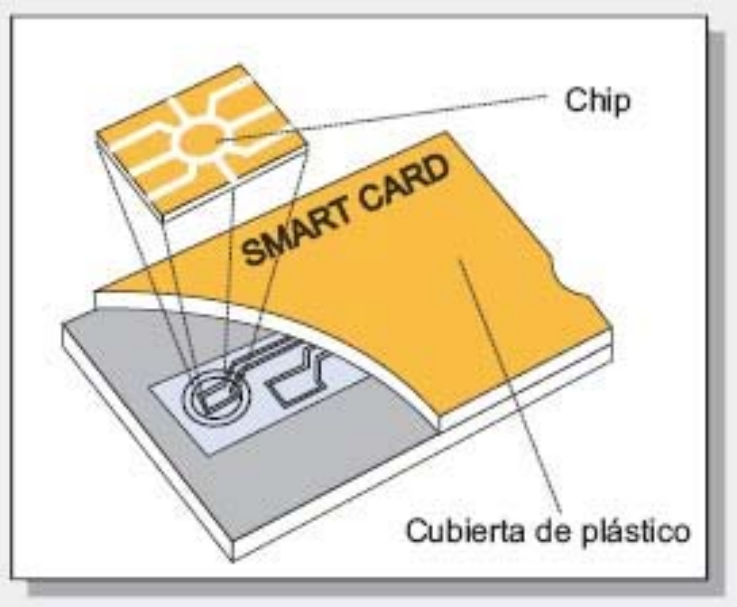


Figura 8: Descripción física de una Tarjeta Inteligente

2.8 - CRYPTOKIT

La Fábrica Nacional de Moneda y Timbre - Real Casa de la Moneda ha lanzado junto con la empresa fabricante de lectores para tarjetas inteligentes C3PO un sistema de cifrado e identificación digital, llamado Cryptokit.

Podemos definir brevemente el producto Cryptokit como una herramienta de seguridad de tipo PKI (Public Key Infrastructure = Infrestructura de Clave Pública) con soporte en tarjeta criptográfica.

El producto nace de la unión de dos elementos hardware principales: la tarjeta criptográfica y el lector de tarjetas inteligentes, que junto con el software adecuado lo convierten en una de las herramientas más seguras para el control de acceso y la identificación de usuarios, ya sea a través de Internet / Intranet o a través de redes de área local (LAN).

La creciente utilización de los medios informáticos y telemáticos en las relaciones administrativas, empresariales y comerciales se traduce en la demanda de los usuarios de servicios y mecanismos de seguridad con los que fortalecer la confianza en las transacciones electrónicas.

Además de eficacia y economía, se exige que dichos medios ayuden a preservar los derechos y obligaciones de los usuarios y consumidores, en condiciones equivalentes a las que proporciona el trámite convencional, sobre todo cuando surgen discrepancias entre las partes o se persigue el fraude.

Este sistema hace posible la seguridad de los datos, mediante el cifrado/descifrado de cualquier fichero utilizando algoritmos de alta seguridad; la seguridad en las comunicaciones, envío de correo firmado y/o cifrado de forma segura, identificación segura a través de la red y es compatible con iniciativas de la Administración Pública incluidas en el Plan Info XXI que requieran el uso de Firma Digital y/o autentificación en su operativa a través de Internet.

La tarjeta de Identificación Electrónica es un importante elemento de seguridad por su posible aplicación a numerosas gestiones, presentes y futuras, de la Administración Pública.

La tarjeta criptográfica es la opción más segura para evitar el uso fraudulento de la propia identidad y el acceso a datos confidenciales.

Tiene capacidad para generar y almacenar varios pares de claves, almacenar distintos certificados (Certificados de la FNMT-RCM, Verisign, Entrust, etc.), datos adicionales del usuario; cifrado, firma y verificación en la propia tarjeta y securización de su uso mediante un código secreto.

Figura 9: Cryptokit - Software de la Fabrica Nacional de Moneda y Timbre

Para más información sobre el uso de este software consultar en la página oficial de la Fabrica Nacional de Moneda y Timbre: http://www.cert.fnmt.es

3 - ANÁLISIS DE REQUISITOS Y DISEÑO DEL PORTAL

Para el diseño del portal web deberemos basarnos en un estándar de Ingeniería de Software que tenga unos sólidos cimientos y una fuerte estabilidad, para este proyecto haremos uso del estándar ESA PSS-05-0, utilizado por la Agencia Espacial Europea, ya que cumple con nuestras expectativas.

Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en la siguiente URL de Internet: http://www.esa.int/esaCP/index.html

Aplicando el estándar ESA PSS-05-0 seguiremos los siguientes pasos:

1) Requisitos de Usuario de Capacidad: Estos requisitos representan las capacidades que los usuarios necesitan que tenga el sistema para que pueda resolver sus problemas o cumplir sus objetivos.

2) Requisitos de Usuario de Restricción: Estos requisitos representan las restricciones que los usuarios establecen acerca de cómo se solucionarán los problemas o se lograrán los objetivos.

3) Diagramas de Casos de Uso: Estos proporcionan uno o más escenarios que indican cómo deberían interactuar el sistema con los usuarios o con otros sistemas para conseguir un objetivo específico.

4) Diagrama de Arquitectura: Muestra gráficamente la estructura y relación de las tecnologías software utilizadas en el proyecto.

5) Diagramas de Secuencia: Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

6) Trazabilidad: Capturar las relaciones de implementación y dependencia en el modelo, siguiendo un histórico de los pasos seguidos para desarrollar el software.

Finalmente se hará un estudio sobre la propia aplicación CMS JOOMLA y sus extensiones para comprobar cómo podemos dar solución a los requisitos planteados acorde a las posibilidades que ofrece dicho sistema.

3.1 - DEFINICIÓN DE REQUISITOS

En este capítulo se exponen los requisitos de usuario y requisitos software de la aplicación que se va ha desarrollar y que se utilizará para la simulación.

La definición de requisitos es fundamental para acotar el alcance de la aplicación y asegurar que se conseguirán los resultados deseados.

La definición de los requisitos de un sistema informático se realiza en dos pasos:

1) El primero consiste en entrevistar al cliente, en nuestro caso será con los responsables del Comité Regional de Madrid del Sindicato Unión Federal de Policía (U.F.P.), para concretar las ideas que éste pueda tener acerca de ***"QUÉ"*** tiene que hacer el sistema.

2) El segundo paso toma estas ideas y, gracias a la experiencia del analista informático, las transforma en la definición de ***"CÓMO"*** se construirá el sistema.

En el caso concreto de este proyecto, seguiremos el estándar de Ingeniería del Software: ***ESA PSS-05-0***, utilizado por la Agencia Espacial Europea.

Dicho estándar puede consultarse en la página web oficial de la Agencia Espacial Europea, en la siguiente URL de Internet: http://www.esa.int/esaCP/index.html

3.2 - IDENTIFICACIÓN DE REQUISITOS

La tarea de identificación de requisitos se lleva a cabo mediante entrevistas periódicas con el cliente, en nuestro caso será con los responsables del Comité Regional de Madrid del Sindicato Unión Federal de Policía (U.F.P.).

Dichas entrevistas son entrevistas abiertas (diálogo fluido y espontáneo) o semi-abiertas (combinación de diálogo abierto con inclusión de algunas preguntas preparadas por el entrevistador de antemano), entre los ingenieros del software y el cliente, tomando nota de las necesidades que éste plantee.

Para que la recogida de requisitos se realice de forma clara, sencilla y estructurada se ha definido una plantilla con las siguientes propiedades:

IDENTIFICADOR:			
Nombre:			
Descripción:			
Tipo Usuario:	☐ *Anónimo*	☐ *Registrado*	☐ *Administrador*
Prioridad:	☐ *Alta*	☐ *Media*	☐ *Baja*
Estabilidad:	☐ *Si*	☐ *No*	
Necesidad:	☐ *Esencial*	☐ *Opcional*	☐ *Conveniente*
Fuente:			

Figura 10: Plantilla para Identificación de Requisitos

A continuación explicamos brevemente el significado de los parámetros que se recogen en la plantilla:

- **Identificador:** Identifica de forma unívoca cada uno de los requisitos. Dicho identificador sigue la siguiente nomenclatura.

IDENTIFICADOR: RU+Tipo+Número

Donde:

- RU: Son las siglas de “Requisito de Usuario”.
- Tipo: Puede tomar los valores “C”, si se trata de un requisito de usuario de Capacidad, y “R” si se trata de un requisito de usuario de Restricción.
- Número: Será un número de tres cifras que empezará por 001 y se irá incrementado en una unidad por cada nuevo requisito añadido.

Ejemplos:

- RUC001: Requisito de Usuario de Capacidad, Número 1.
- RUR010: Requisito de Usuario de Restricción, Número 10.

- **Nombre:** Expresa el nombre del requisito, en pocas palabras un resumen del requisito.

- **Descripción:** Breve comentario textual del requisito.

- **Tipo Usuario:** Nos indica el tipo de usuario para el cual va dirigido el requisito. Puede tomar los siguientes valores:
  - "*Anónimo*": Usuario visitante del portal, el cual no posee ningún privilegio de acceso sobre contenidos profesionales del sector de la seguridad pública (Cuerpo Nacional de Policía).
  - "*Registrado*": Usuario afiliado al sindicato de policía U.F.P., el cual habrá sido previamente dado de alta por el administrador y posee privilegios de acceso sobre contenidos profesionales, al igual que acceso a un foro de profesionales del sector (miembros del Cuerpo Nacional de Policía).
  - "*Administrador*": Posee los privilegios para gestionar y configurar todos los aspectos del portal.

- **Prioridad:** Indica la prioridad en el desarrollo del requisito. Puede tomar los valores "Alta", "Media" o "Baja".

- **Estabilidad:** Indica la posibilidad de que el requisito cambie a lo largo del desarrollo de la aplicación. Puede tomar los siguientes valores:
  - "Si": Cuando el cliente asegura que no va a ser modificado.
  - "No": Cuando el requisito puede variar en función de las sucesivas etapas del proyecto.

- **Necesidad:** Indica el nivel de necesidad del requisito dentro del sistema final. Puede tomar los valores "Esencial", "Opcional" o "Conveniente".

- "Esencial": Cuando el cliente no acepte ninguna negociación.
- "Conveniente": Cuando el requisito se pueda negociar.
- "Opcional": Cuando su implementación puede ser eliminada.

- **Fuente:** Indica el origen a partir del cual se ha obtenido el documento. Cuando se trata de un documento externo se hace referencia a dicho documento.

3.3 - REQUISITOS DE USUARIO

En la fase de toma de requisitos de usuario, o fase de definición del problema, se definirán el ámbito y el alcance del sistema, es decir, lo que se espera que haga el sistema.

Los requisitos de usuario son de dos tipos: de capacidad y de restricción.

3.3.1 – DESCRIPCIÓN GENERAL

- ✓ **Perspectiva del Producto:**

  El sistema debe constar de dos partes bien diferenciadas:

  - El *"Front-End"*, de cara a los usuarios del portal, el cual les permitirá ver y consultar diversos contenidos que ofrece el Sindicato de Policía U.F.P.
  - El *"Back-End"*, de cara al administrador, el cual permitirá configurar y gestionar todo el portal de manera sencilla y segura.

- ✓ **Capacidades Generales:**

  El sistema ofrecerá diversas funcionalidades que podemos resumir en: visualizar contenidos e imágenes, consultar el calendario del sindicato, realizar encuestas y participar en un foro.

- ✓ **Restricciones Generales:**

  El sistema tendrá que poderse utilizar de manera remota, tanto por los usuarios como por el administrador, siendo instalado dicho sistema en un servidor y accediéndose a este mediante su URL.

  A su vez, dicho acceso debe ser, en el caso de los usuarios, por un canal convencional HTTP y el acceso del administrador, por un canal seguro HTTPS, cifrado mediante el protocolo SSL (Secure Socket Layer).

- ✓ **Características del Usuario:**

  A este sistema accederán dos tipos de usuarios:

  - Usuario Anónimo: Éste no necesita ninguna capacidad especial para acceder al contenido público del portal, tan sólo conocer la dirección de Internet para acceder desde su navegador.

- Usuario Registrado: Éste necesitará conocer los parámetros "Usuario" y "Clave", para acceder al contenido privado del portal.

✓ **Características del Administrador:**

El *"Back-End"*, de cara al administrador, permitirá configurar y gestionar todo el portal de manera sencilla y segura.

Para acceder a éste, el administrador deberá conocer los parámetros "Usuario" y "Clave". También deberá conocer la URL de acceso a la zona del administrador, "https://www.portal.com/administrador", donde "portal" se corresponderá con el dominio de Internet contratado por el Sindicato de Policía U.F.P.

El administrador deberá poseer un "*Certificado de Cliente SSL*" instalado en su navegador, que será generado en este proyecto, para que el servidor compruebe mediante su "*Certificado de Servidor SSL*" la identidad del administrador y así permitirle establecer el canal seguro "HTTPS".

El administrador deberá tener unos conocimientos básicos en el uso del CMS JOOMLA, los cuales podrá adquirir leyendo la documentación de este proyecto

✓ **Entorno de Operación:**

Todas las operaciones se realizarán de forma remota, ya que dicho portal será instalado en un servidor de Internet.

Las pruebas de funcionamiento se podrán realizar en modo local, teniendo instalado un servidor Apache junto con MySQL y el CMS JOOMLA, en una misma máquina.

3.3.2 - REQUISITOS DE USUARIO DE CAPACIDAD

Estos requisitos representan las capacidades que los usuarios necesitan que tenga el sistema para que pueda resolver sus problemas o cumplir sus objetivos.

A continuación se exponen los requisitos de capacidad:

IDENTIFICADOR: RUC001	
Nombre:	**Consultar contenidos públicos**
Descripción:	Debe permitir consultar contenidos como documentos y noticias, siempre y cuando estos tengan un carácter público, es decir, que no sean contenidos protegidos que sólo puedan ser vistos por miembros del sindicato U.F.P. y por extensión, del Cuerpo Nacional de Policía.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC002	
Nombre:	**Ver galería de imágenes**
Descripción:	Debe permitir ver una galería de imágenes con diversas fotografías que ofrece el sindicato U.F.P.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC003	
Nombre:	**Consultar calendario**
Descripción:	Debe permitir consultar el calendario del sindicato U.F.P. para conocer cuándo, dónde y a qué hora se celebran diversos eventos.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC004	
Nombre:	**Realizar encuestas**
Descripción:	Debe permitir realizar encuestas con la finalidad de recoger la opinión pública sobre ciertos temas que interesa conocer al sindicato.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC005	
Nombre:	**Realizar búsquedas sobre contenido público**
Descripción:	Debe permitir realizar búsquedas de información a través de palabras clave, sobre los contenidos, en este caso de carácter público.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC006	
Nombre:	**Autenticarse como usuario registrado**
Descripción:	Debe permitir al usuario anónimo mediante un “USUARIO” y una “CLAVE” autenticarse para pasar a tener los privilegios que posee un usuario registrado, para lo cual dicho usuario debe pertenecer al sindicato U.F.P. y haberle proporcionado éste dichos parámetros de acceso.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC007	
Nombre:	**Consultar contenidos privados**
Descripción:	Debe permitir consultar contenidos, esta vez de carácter privado, es decir, contenidos destinados a profesionales del sector de la seguridad pública como son los miembros del Cuerpo Nacional de Policía.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC008	
Nombre:	**Utilizar el foro**
Descripción:	Debe permitir el acceso a un foro de profesionales del sector de la seguridad pública como son los miembros del Cuerpo Nacional de Policía.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC009	
Nombre:	**Realizar búsquedas sobre contenido privado**
Descripción:	Debe permitir realizar búsquedas de información a través de palabras clave, sobre los contenidos, en este caso de carácter privado.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC010	
Nombre:	**Autenticarse como administrador**
Descripción:	Debe permitir al administrador mediante un "USUARIO" y una "CLAVE" autenticarse para pasar a tener los privilegios del administrador. Dicha conexión a la zona del administrador o "Back-End"debe ser por un canal seguro HTTPS.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC011	
Nombre:	**Gestión de contenidos**
Descripción:	Debe permitir añadir, modificar o eliminar cualquier clase de contenido, tanto público como privado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC012	
Nombre:	**Gestión de usuarios**
Descripción:	Debe permitir dar de alta, dar de baja o modificar los datos personales de cualquier usuario registrado (afiliado al sindicato de policía U.F.P.).
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC013	
Nombre:	**Gestión de galería de imágenes**
Descripción:	Debe permitir añadir, modificar o eliminar cualquier fotografía a la galería de imágenes del portal.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC014	
Nombre:	**Gestión del calendario**
Descripción:	Debe permitir añadir, modificar o eliminar cualquier evento que vaya a tener lugar en el calendario oficial del sindicato U.F.P.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC015	
Nombre:	**Gestión de encuestas**
Descripción:	Debe permitir añadir, modificar o eliminar cualquier encuesta que necesite realizar el sindicato U.F.P.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC016	
Nombre:	**Gestión del foro**
Descripción:	Debe permitir crear temas de discusión en el foro, modificar estos o eliminarlos, al igual que regular las normas del foro.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUC017	
Nombre:	**Instalar extensiones**
Descripción:	Debe permitir instalar cualquier tipo de ampliación o actualización del portal.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

3.3.3 - REQUISITOS DE USUARIO DE RESTRICCIÓN

Estos requisitos representan las restricciones que los usuarios establecen acerca de cómo se solucionarán los problemas o se lograrán los objetivos.

A continuación se exponen los requisitos de restricción:

IDENTIFICADOR: RUR001	
Nombre:	**Servidor Apache**
Descripción:	La versión del Servidor Apache debe ser Apache 2.2.4 para asegurarse el correcto funcionamiento del sistema desarrollado.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR002	
Nombre:	**Gestor de Bases de Datos MySQL**
Descripción:	La versión del Gestor de Bases de Datos MySQL debe ser MySQL 5.0.45 para asegurarse el correcto funcionamiento del sistema desarrollado.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR003	
Nombre:	**Intérprete de PHP**
Descripción:	La versión del Intérprete de PHP debe ser PHP 5.2.3 para asegurarse el correcto funcionamiento del sistema desarrollado.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR004	
Nombre:	**CMS JOOMLA**
Descripción:	La versión del Gestor de Contenidos JOOMLA debe ser "JOOMLA 1.0.13 Spanish" para asegurarse el correcto funcionamiento del sistema desarrollado.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR005	
Nombre:	**Resolución de la Pantalla**
Descripción:	La resolución para la cual la visualización del Gestor de Contenidos es óptima es de 1024x768.
Prioridad:	☐ *Alta* ☑ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☐ *Esencial* ☐ *Opcional* ☑ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR006	
Nombre:	**Navegador**
Descripción:	El navegador en el cuál se deben visualizar de manera óptima las interfaces del Gestor de Contenidos es para Internet Explorer 7.0.
Prioridad:	☐ *Alta* ☑ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☐ *Esencial* ☐ *Opcional* ☑ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR007	
Nombre:	**Entorno Windows**
Descripción:	El entorno donde será desarrollado y mantenido el Gestor de Contenidos será Windows XP Professional o cualquier sistema Operativo compatible con el mismo y su versión.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR008	
Nombre:	**Protocolo SSL**
Descripción:	El entorno requiere un acceso seguro para el cual se debe establecer una comunicación cifrada entre cliente y servidor, mediante el Protocolo SSL.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR009	
Nombre:	**OpenSSL**
Descripción:	El entorno requiere un acceso seguro para el cual se debe establecer una comunicación cifrada entre cliente y servidor, mediante el Protocolo SSL. Los certificados necesarios para ello serán generados con la herramienta OpenSSL versión 0.9.8h.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR010	
Nombre:	**Certificados con formato "pkcs12"**
Descripción:	El entorno requiere un acceso seguro para el cual se debe establecer una comunicación cifrada entre cliente y servidor, mediante el Protocolo SSL. Los certificados generados con OpenSSL deben ser formato pkcs12, es decir, con extensión ".p12", ya que este formato es un estándar muy extendido que reclaman los navegadores actuales para que los certificados puedan ser importados y utilizados.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR011	
Nombre:	**Módulo de Apache: "mod_ssl"**
Descripción:	El entorno requiere un acceso seguro para el cual se debe establecer una comunicación cifrada entre cliente y servidor, mediante el Protocolo SSL. El módulo del servidor apache que es imprescindible para establecer y configurar una conexión segura mediante el protocolo SSL es el "mod_ssl".
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR012	
Nombre:	**CryptoKit**
Descripción:	El entorno requiere un acceso seguro y, para posibilitar que el cliente transporte su certificado SSL de manera que pueda autenticarse desde cualquier máquina remota contra el servidor serán necesarios dos elementos: la Herramienta Software "CryptoKit" desarrollada por la Fabrica Nacional de Moneda y Timbre con la finalidad de poder manipular y configurar Tarjetas Inteligentes. Esta Herramienta nos permitirá configurar la Tarjeta Inteligente, y por tanto importar a este medio físico el "Certificado de Cliente" para poder autenticarnos en el sistema desde cualquier máquina remota.
Prioridad:	☐ *Alta* ☑ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☐ *Esencial* ☑ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR013	
Nombre:	**Tarjeta Inteligente**
Descripción:	El entorno requiere un acceso seguro y, para posibilitar que el cliente transporte su certificado SSL de manera que pueda autenticarse desde cualquier máquina remota contra el servidor serán necesarios dos elementos: la Herramienta Software "CryptoKit" desarrollada por la Fabrica Nacional de Moneda y Timbre con la finalidad de poder manipular y configurar Tarjetas Inteligentes. La Tarjeta Inteligente nos permitirá transportar fácilmente en este medio físico el "Certificado de Cliente" para poder autenticarnos en el sistema desde cualquier máquina remota.
Prioridad:	☐ *Alta* ☑ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☐ *Esencial* ☑ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR014	
Nombre:	**Extensión de JOOMLA: Componente "Events"**
Descripción:	La versión del Componente Events debe ser "com_events_1.4.2" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Componente JOOMLA nos permitirá gestionar un calendario.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR015	
Nombre:	**Extensión de JOOMLA: Componente "zOOm Media Gallery"**
Descripción:	La versión del Componente zOOm Media Gallery debe ser "zOOm_for_JOOMLA_1.5" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Componente JOOMLA nos permitirá gestionar una galería de imágenes.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR016	
Nombre:	**Extensión de JOOMLA: Componente "Fire Board Forum"**
Descripción:	La versión del Componente Fire Board Forum debe ser "FireBoard_1.0.4_Stable_CompletePackage" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Componente JOOMLA nos permitirá gestionar un foro.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR017	
Nombre:	**Extensión de JOOMLA: Componente "JOOMLAXplorer"**
Descripción:	La versión del Componente JOOMLAXplorer debe ser "com_JOOMLAxplorer_1.6.2" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Componente JOOMLA nos permitirá gestionar un explorador de archivos y carpetas del sistema.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR018	
Nombre:	**Extensión de JOOMLA: Módulo "vcnt_counter"**
Descripción:	La versión del Módulo vcnt_counter debe ser "mod_vcnt" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Módulo JOOMLA nos permitirá gestionar un contador de visitas.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR019	
Nombre:	**Extensión de JOOMLA: Módulo "mod_scroll_news"**
Descripción:	La versión del Módulo mod_scroll_news debe ser "mod_scroll_news" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Módulo JOOMLA nos permitirá gestionar un scroll dinámico que vaya mostrando noticias de nuestro portal, aprovechando así mejor el espacio en pantalla.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR020	
Nombre:	**Extensión de JOOMLA: Módulo "mod_poll"**
Descripción:	La versión del Módulo mod_poll debe ser "mod_poll" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Módulo JOOMLA nos permitirá gestionar encuestas.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

IDENTIFICADOR: RUR021	
Nombre:	**Extensión de JOOMLA: Módulo "mod_search"**
Descripción:	La versión del Módulo mod_search debe ser "mod_search" para asegurarse el correcto funcionamiento del sistema desarrollado. Este Módulo JOOMLA nos permitirá gestionar un motor de busqueda.
Prioridad:	☑ *Alta* ☐ *Media* ☐ *Baja*
Estabilidad:	☑ *Si* ☐ *No*
Necesidad:	☑ *Esencial* ☐ *Opcional* ☐ *Conveniente*
Fuente:	U.F.P. - Unión Federal de Policía (El cliente).

3.4 - DIAGRAMA DE CASOS DE USO GENERAL

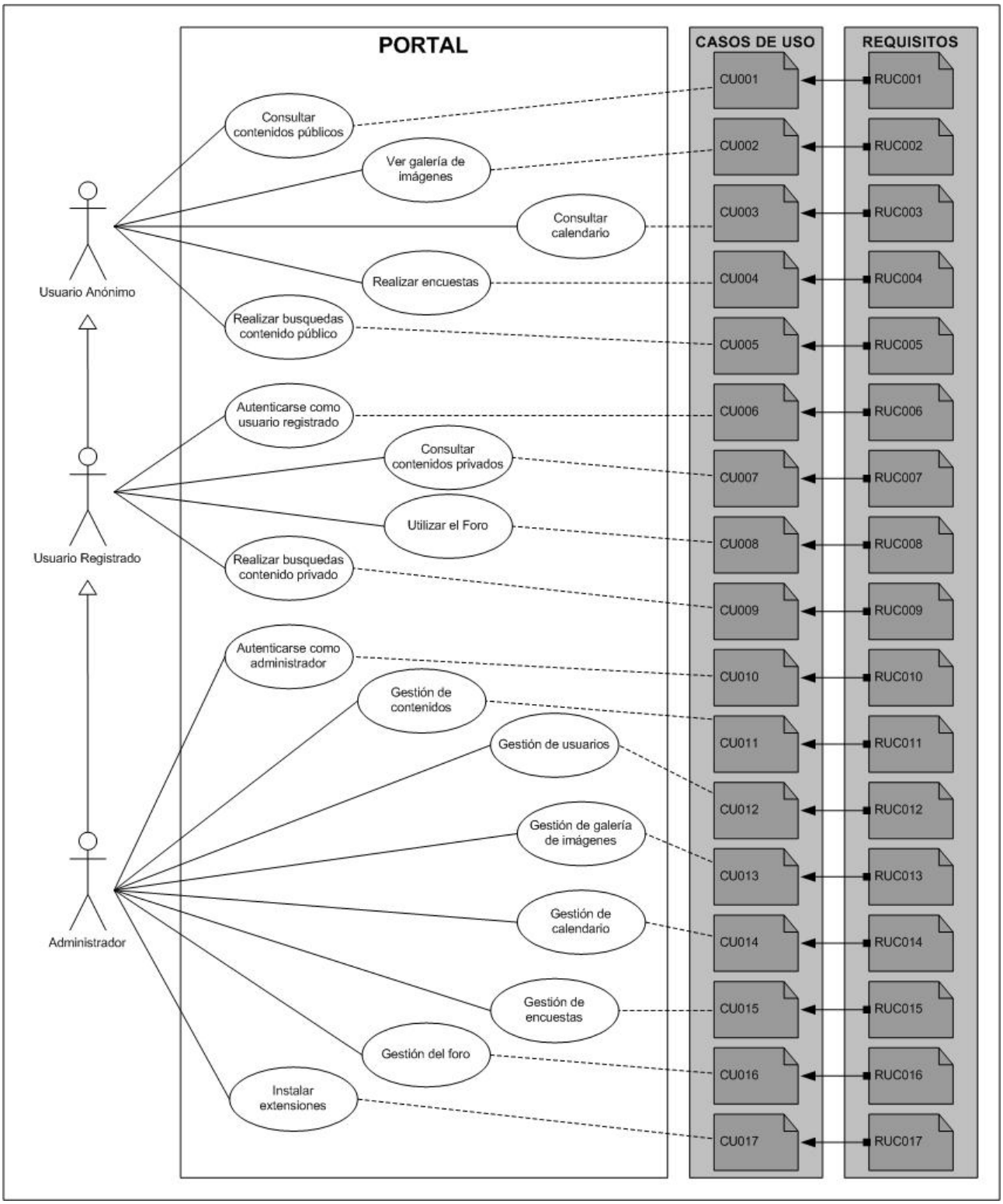


Figura 11: Diagrama de Casos de Uso General

3.4.1 - DIAGRAMA DE CASOS DE USO PARA EL USUARIO ANÓNIMO

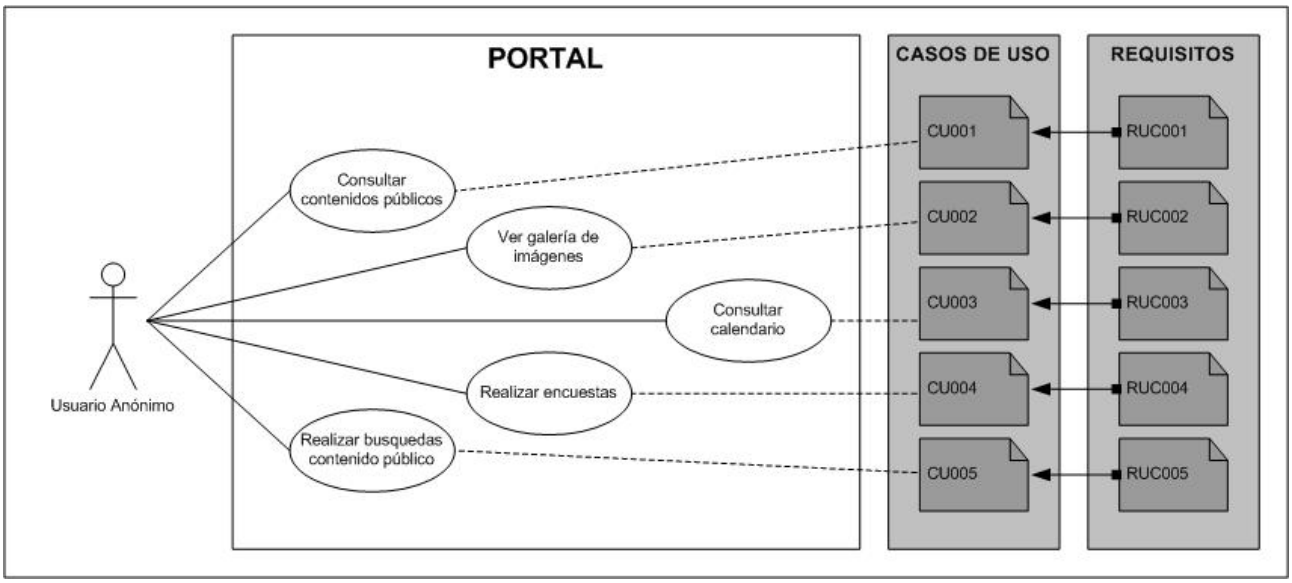


Figura 12: Diagrama de Casos de Uso para el Usuario Anónimo

IDENTIFICADOR: CU001	
Nombre:	**Consultar contenidos públicos**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC001*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC001

IDENTIFICADOR: CU002	
Nombre:	**Ver galería de imágenes**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC002*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC002

IDENTIFICADOR: CU003	
Nombre:	**Consultar calendario**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC003*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC003

IDENTIFICADOR: CU004	
Nombre:	**Realizar encuestas**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC004*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC004

IDENTIFICADOR: CU005	
Nombre:	**Realizar búsquedas contenido público**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC005*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☑ *Anónimo* ☐ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC005

3.4.2 - DIAGRAMA DE CASOS DE USO PARA EL USUARIO REGISTRADO

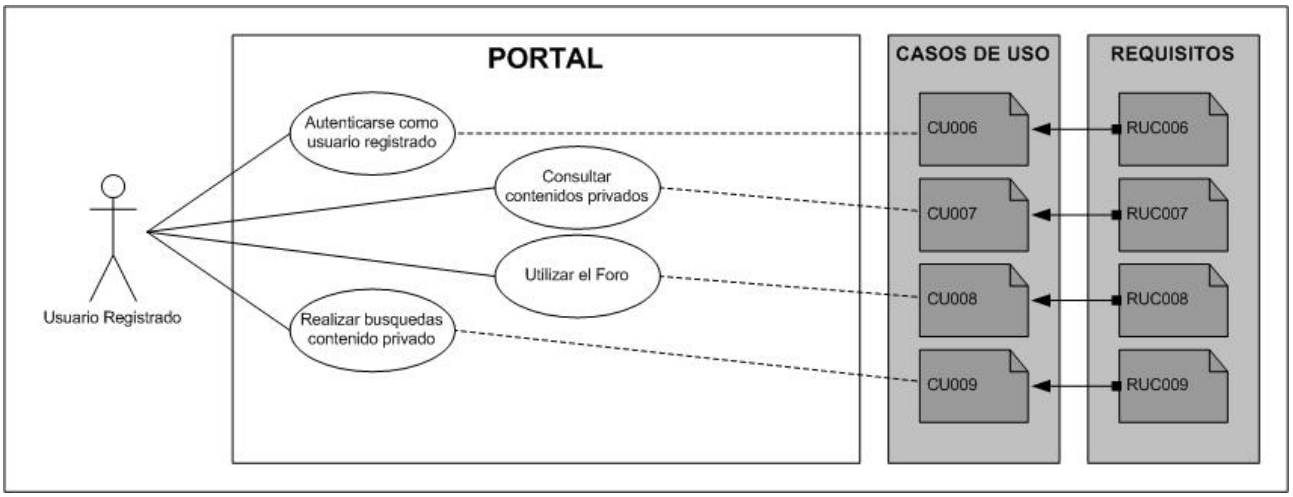


Figura 13: Diagrama de Casos de Uso para el Usuario Registrado

IDENTIFICADOR: CU006	
Nombre:	**Autenticarse como usuario registrado**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC006*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC006

IDENTIFICADOR: CU007	
Nombre:	**Consultar contenidos privados**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC007*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC007

IDENTIFICADOR: CU008	
Nombre:	**Utilizar el foro**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC008*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC008

IDENTIFICADOR: CU009	
Nombre:	**Realizar búsquedas contenido privado**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC009*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☑ *Registrado* ☐ *Administrador*
Trazabilidad:	RUC009

3.4.3 - DIAGRAMA DE CASOS DE USO PARA EL ADMINISTRADOR

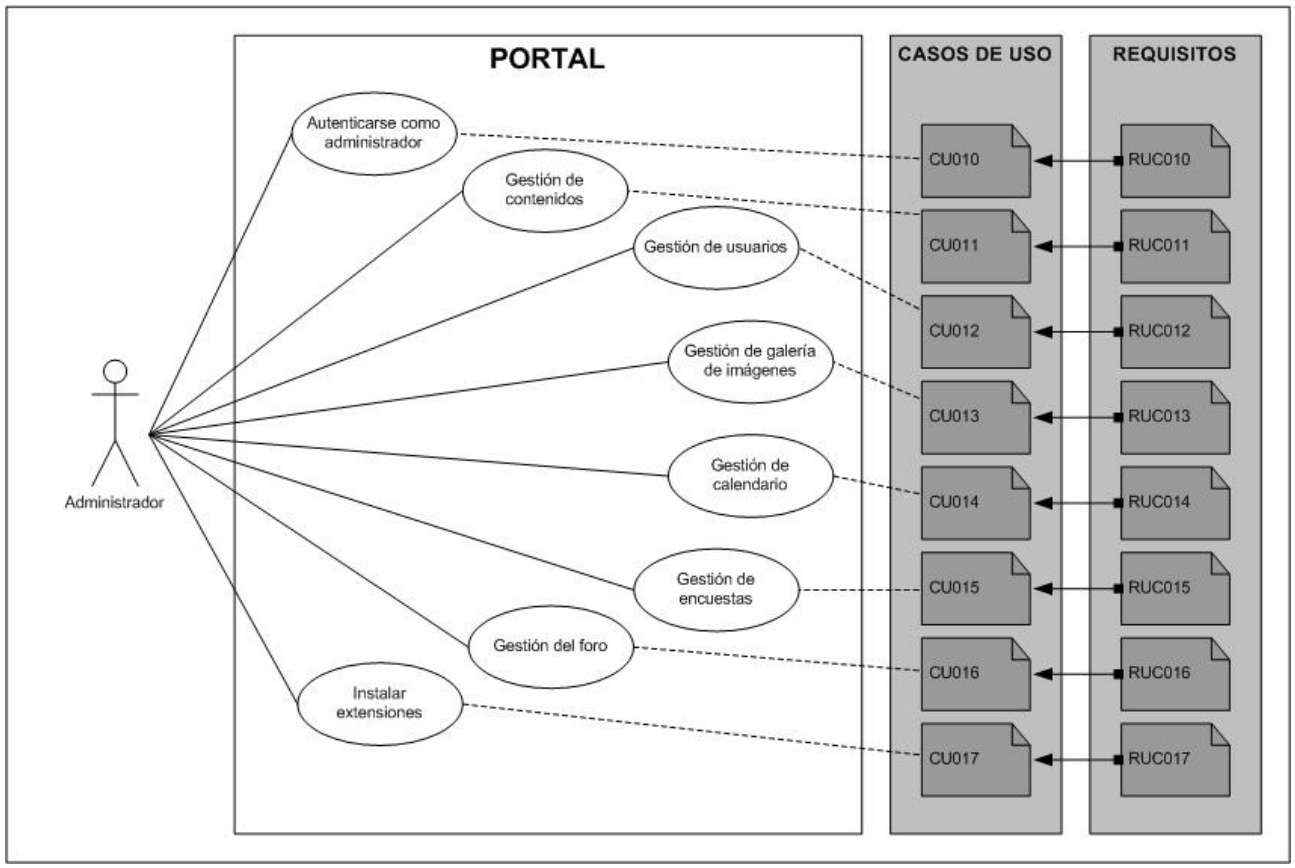


Figura 14: Diagrama de Casos de Uso para el Administrador

IDENTIFICADOR: CU010	
Nombre:	**Autenticarse como administrador**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC010*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC010

IDENTIFICADOR: CU011	
Nombre:	**Gestión de contenidos**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC011*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC011

IDENTIFICADOR: CU012	
Nombre:	**Gestión de usuarios**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC012*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC012

IDENTIFICADOR: CU013	
Nombre:	**Gestión de galería de imágenes**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC013*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC013

IDENTIFICADOR: CU014	
Nombre:	**Gestión de calendario**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC014*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC014

IDENTIFICADOR: CU015	
Nombre:	**Gestión de encuestas**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC015*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC015

IDENTIFICADOR: CU016	
Nombre:	**Gestión del foro**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC016*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC016

IDENTIFICADOR: CU017	
Nombre:	**Instalar extensiones**
Descripción:	Este caso de uso se corresponde con el Requisito de Usuario de Capacidad "*RUC017*" como se muestra, si seguimos la trazabilidad, en la figura anterior. Para conocer la funcionalidad de este caso de uso tan sólo tendremos que consultar el requisito de usuario de capacidad referenciado.
Tipo Usuario:	☐ *Anónimo* ☐ *Registrado* ☑ *Administrador*
Trazabilidad:	RUC017

3.5 - DIAGRAMA DE ARQUITECTURA

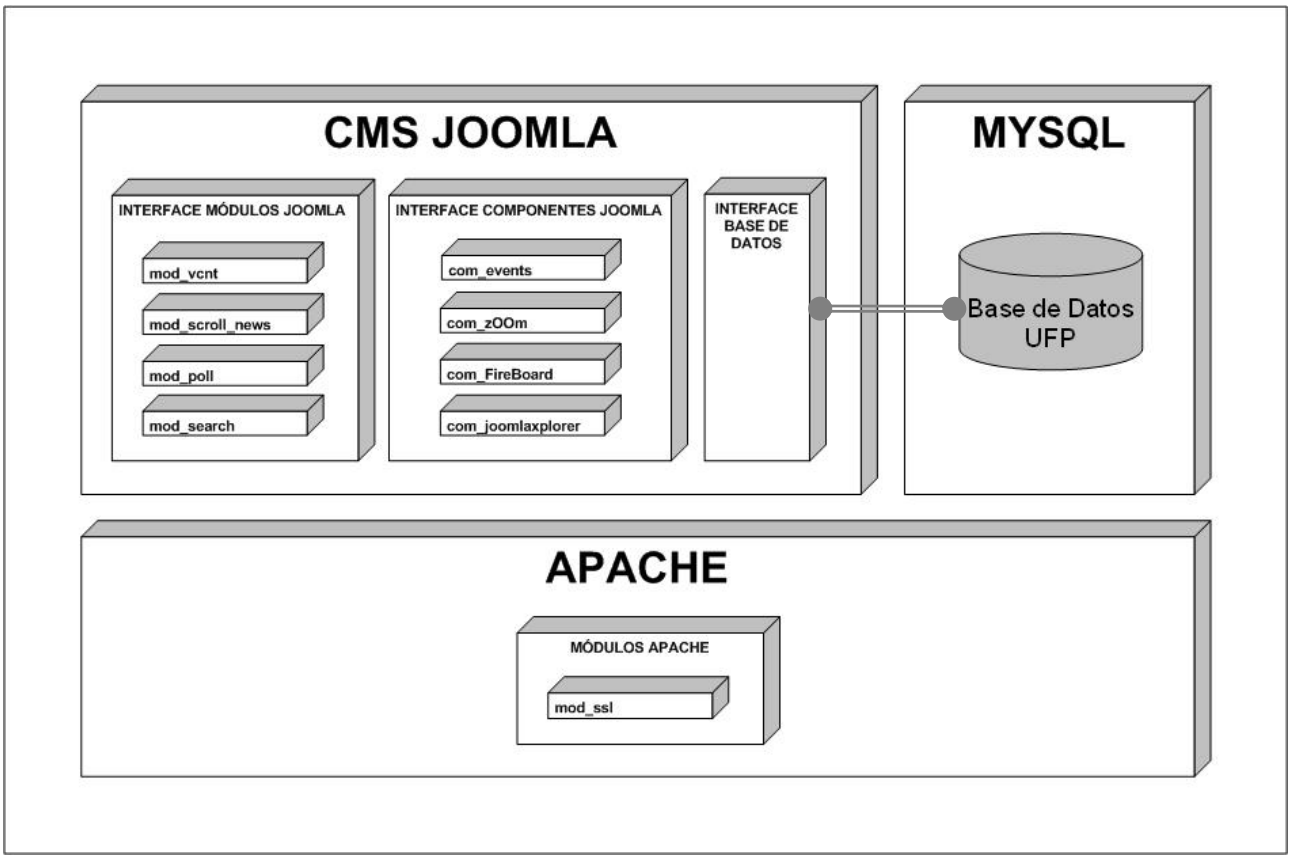


Figura 15: Diagrama de Arquitectura

En el Diagrama de Arquitectura representamos las relaciones que existen entre los distintos componentes software que configuran la plataforma de este proyecto.

El Servidor APACHE gestiona las comunicaciones seguras con la aplicación web JOOMLA através de su módulo “mod_ssl”, el cual posee la configuración del Protocolo SSL.

El Gestor de Contenidos JOOMLA interactúa através de su interfaz con la base de datos del Proyecto (MYSQL). Igualmente en JOOMLA tendremos que instalar, modificar y personalizar ciertos componentes para ampliar las funcionalidades base, que explicaremos a continuación:

- ✓ Mod_vcnt: contador.
- ✓ Mod_scroll_news: noticias rotatorias.
- ✓ Mod_poll: encuestas.
- ✓ Mod_search: motor de búsqueda.
- ✓ Com_events: calendario.
- ✓ Com_zOOm: galería de imagines.
- ✓ Com_FireBoard: foro.
- ✓ Com_joomlaxplorer: explorador de archivos.

3.6 - DIAGRAMAS DE SECUENCIA

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si tienes modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puedes "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales.

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.

Durante el análisis inicial, el modelador típicamente coloca el nombre de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la case instanciada por el objeto en la recepción final del mensaje.

3.6.01 - DIAGRAMA DE SECUENCIA “DS001” PARA EL CASO DE USO “CU001”

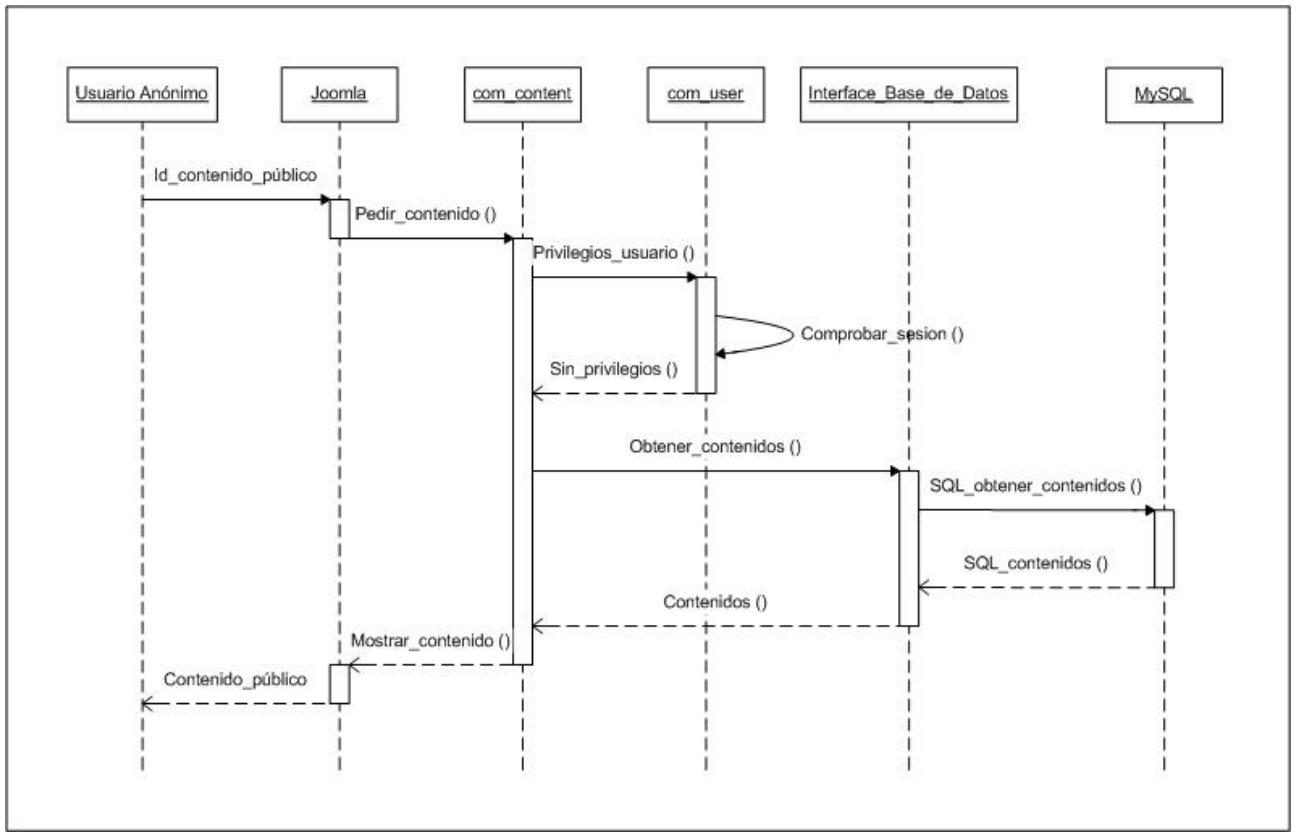


Figura 16: Diagrama de Secuencia “DS001”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU001” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario anónimo” consulta algún contenido público.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End que en el diagrama denominamos “JOOMLA” realiza una petición de contenido al componente de JOOMLA “com_content” que se encarga de gestionar los contenidos.

2) El componente de JOOMLA “com_content” se comunica con otro componente de JOOMLA denominado “com_user” para preguntarle por los privilegios que posee el usuario de la sesión actual.

3) El componente de JOOMLA “com_user” responde a “com_content” que ese usuario no posee privilegios por lo que sólo podrá ver contenido de carácter público.

4) El componente de JOOMLA "com_content", tras esta respuesta, pide al componente de JOOMLA denominado "Interface Base de Datos" que obtenga el contenido público que peticiona el usuario.

5) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

6) El Gestor de Bases de Datos "MySQL" devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

7) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_content" dándole el contenido que le había pedido.

8) El componente "com_content" a su vez muestra dicho contenido en el formato adecuado al Front-End "JOOMLA", para que pueda ser visionado por el usuario.

3.6.02 - DIAGRAMA DE SECUENCIA “DS002” PARA EL CASO DE USO “CU002”

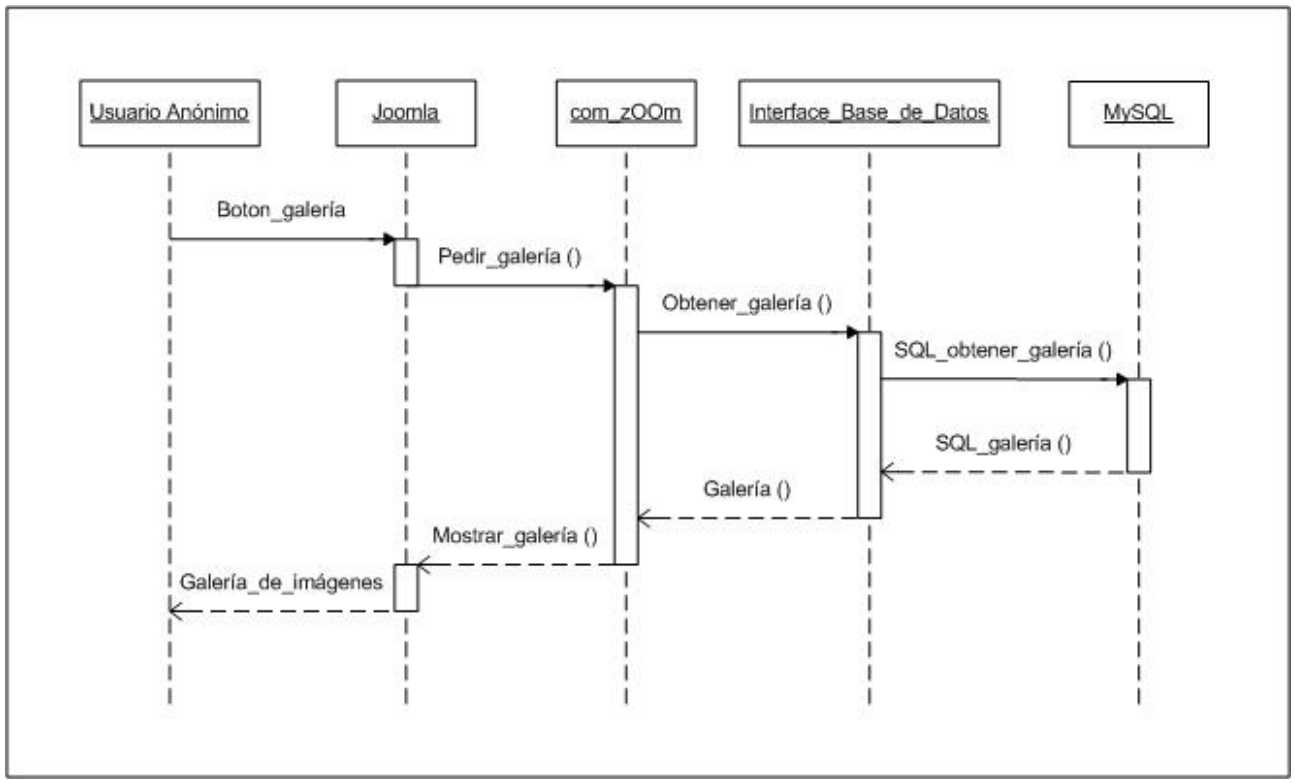


Figura 17: Diagrama de Secuencia “DS002”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU002” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario anónimo” pulsa el botón galería de imágenes.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End “JOOMLA” realiza la petición de la galería al componente de JOOMLA “com_zOOm” que se encarga de gestionar dicha galería de imágenes.

2) El componente de JOOMLA “com_zOOm”, pide al componente de JOOMLA denominado “Interface Base de Datos” que obtenga la galería que peticiona el usuario.

3) El componente de JOOMLA “Interface Base de Datos” a su vez realiza esa petición al Gestor de Bases de Datos “MySQL” esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

4) El Gestor de Bases de Datos “MySQL” devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA “Interface Base de Datos”.

5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al componente “com_zOOm” dándole la galería que le había pedido.

6) El componente “com_zOOm” a su vez muestra dicha galería de imágenes en el formato adecuado al Front-End “JOOMLA”, para que pueda ser visionado por el usuario.

3.6.03 - DIAGRAMA DE SECUENCIA “DS003” PARA EL CASO DE USO “CU003”

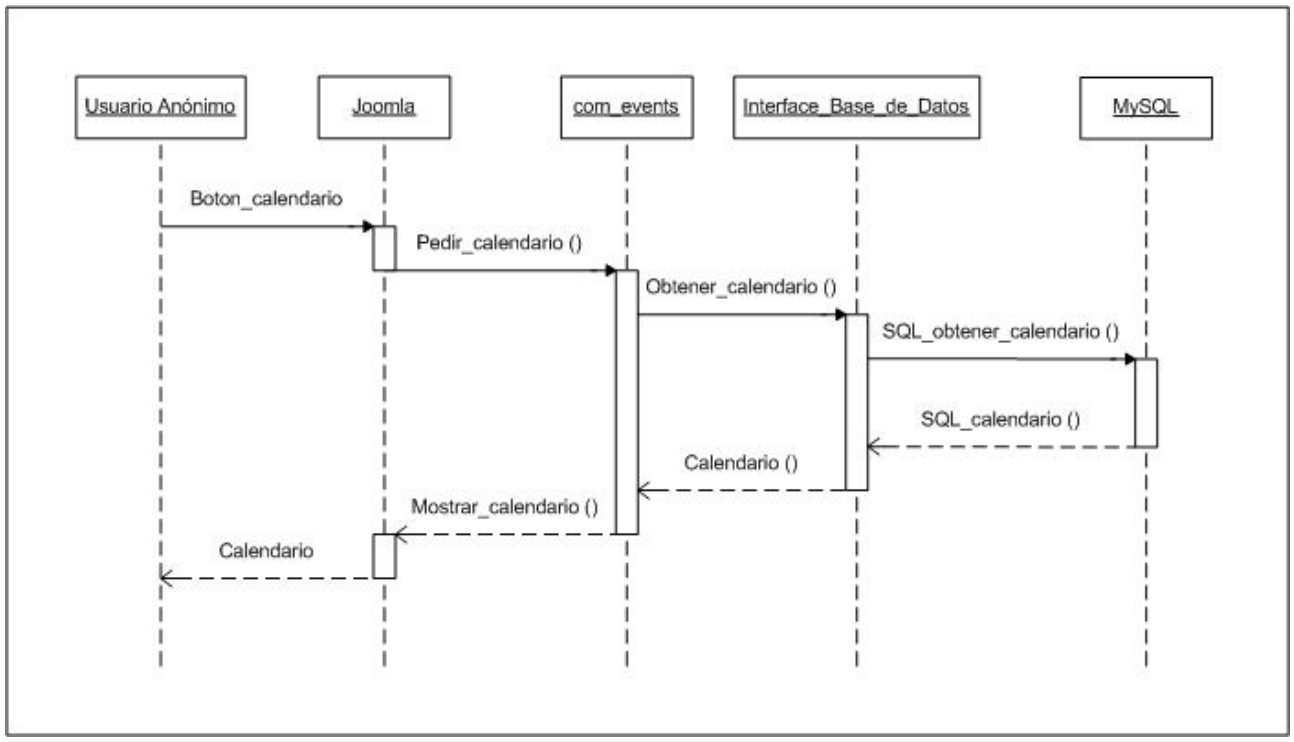


Figura 18: Diagrama de Secuencia “DS003”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU003” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario anónimo” pulsa el botón calendario.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End “JOOMLA” realiza la petición del calendario al componente de JOOMLA “com_events” que se encarga de gestionar dicho calendario.

2) El componente de JOOMLA “com_events”, pide al componente de JOOMLA denominado “Interface_Base_de_Datos” que obtenga el calendario que peticiona el usuario.

3) El componente de JOOMLA “Interface Base de Datos” a su vez realiza esa petición al Gestor de Bases de Datos “MySQL” esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

4) El Gestor de Bases de Datos “MySQL” devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA “Interface Base de Datos”.

5) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al componente “com_events” dándole el calendario que le había pedido.

6) El componente “com_events” a su vez muestra dicho calendario en el formato adecuado al Front-End “JOOMLA”, para que pueda ser visionado por el usuario.

3.6.04 - DIAGRAMA DE SECUENCIA “DS004” PARA EL CASO DE USO “CU004”

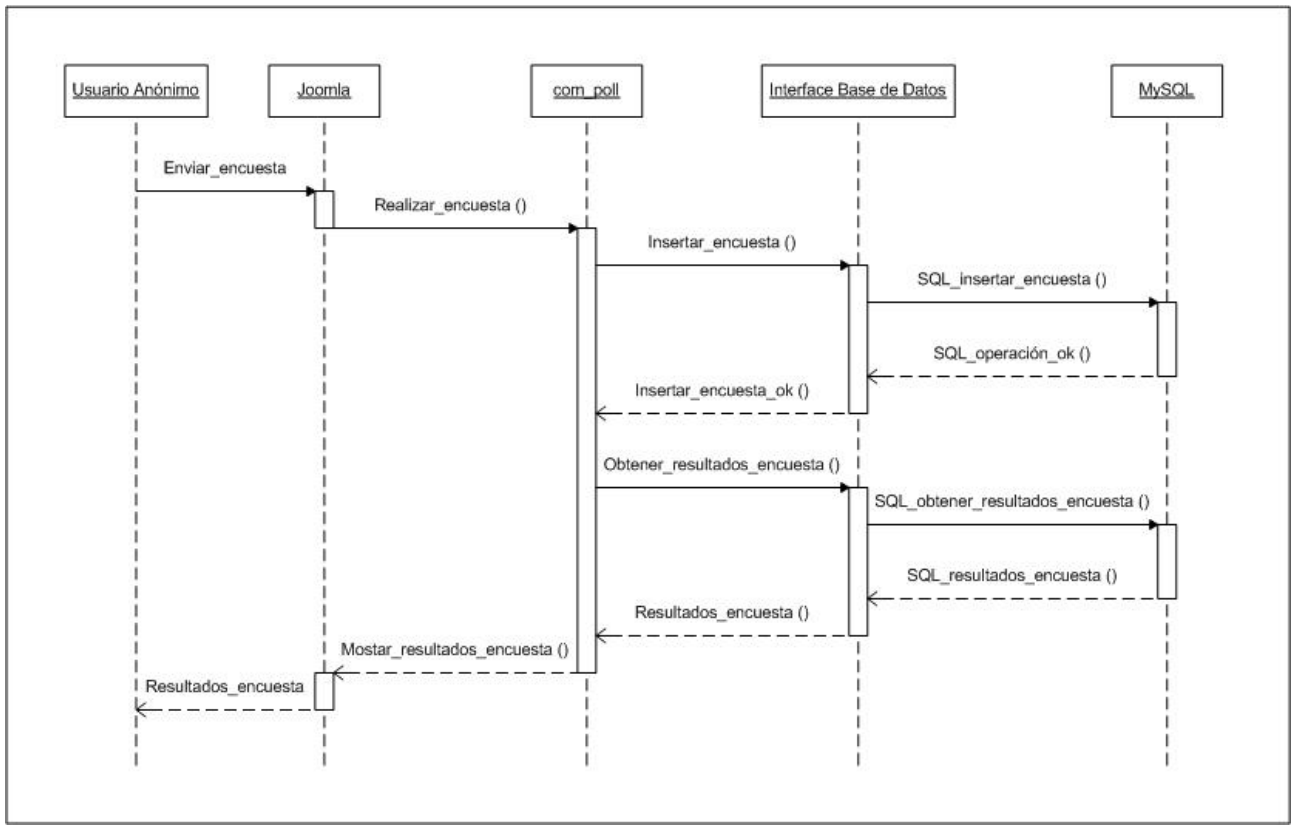


Figura 19: Diagrama de Secuencia “DS004”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU004” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario anónimo” realiza una encuesta.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End “JOOMLA” manda la encuesta realizada al componente de JOOMLA “com_poll” que se encarga de gestionar las encuestas.

2) El componente de JOOMLA “com_poll”, pide al componente de JOOMLA denominado “Interface Base de Datos” que guarde los datos de la encuesta realizada por el usuario.

3) El componente de JOOMLA “Interface Base de Datos” a su vez realiza esa petición al Gestor de Bases de Datos “MySQL” esta vez en lenguaje SQL, para insertar en ésta, los datos que proporciona el usuario.

4) El Gestor de Bases de Datos "MySQL" devuelve que dicha operación ha sido realizada con éxito, en lenguaje SQL, al componente de JOOMLA "Interface Base de Datos".

5) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_poll" diciéndole que dicha operación se ha realizado con éxito.

6) El componente de JOOMLA "com_poll", a su vez, vuelve a pedir al componente de JOOMLA denominado "Interface Base de Datos" que obtenga los resultados de la encuesta realizada por el usuario para posteriormente informarle de los resultados actuales de la misma.

7) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

8) El Gestor de Bases de Datos "MySQL" devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

9) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_poll" dándole los resultados de dicha encuesta.

10) El componente "com_poll" a su vez muestra dichos resultados en el formato adecuado al Front-End "JOOMLA", para que pueda ser visionado por el usuario.

3.6.05 - DIAGRAMA DE SECUENCIA “DS005” PARA EL CASO DE USO “CU005”

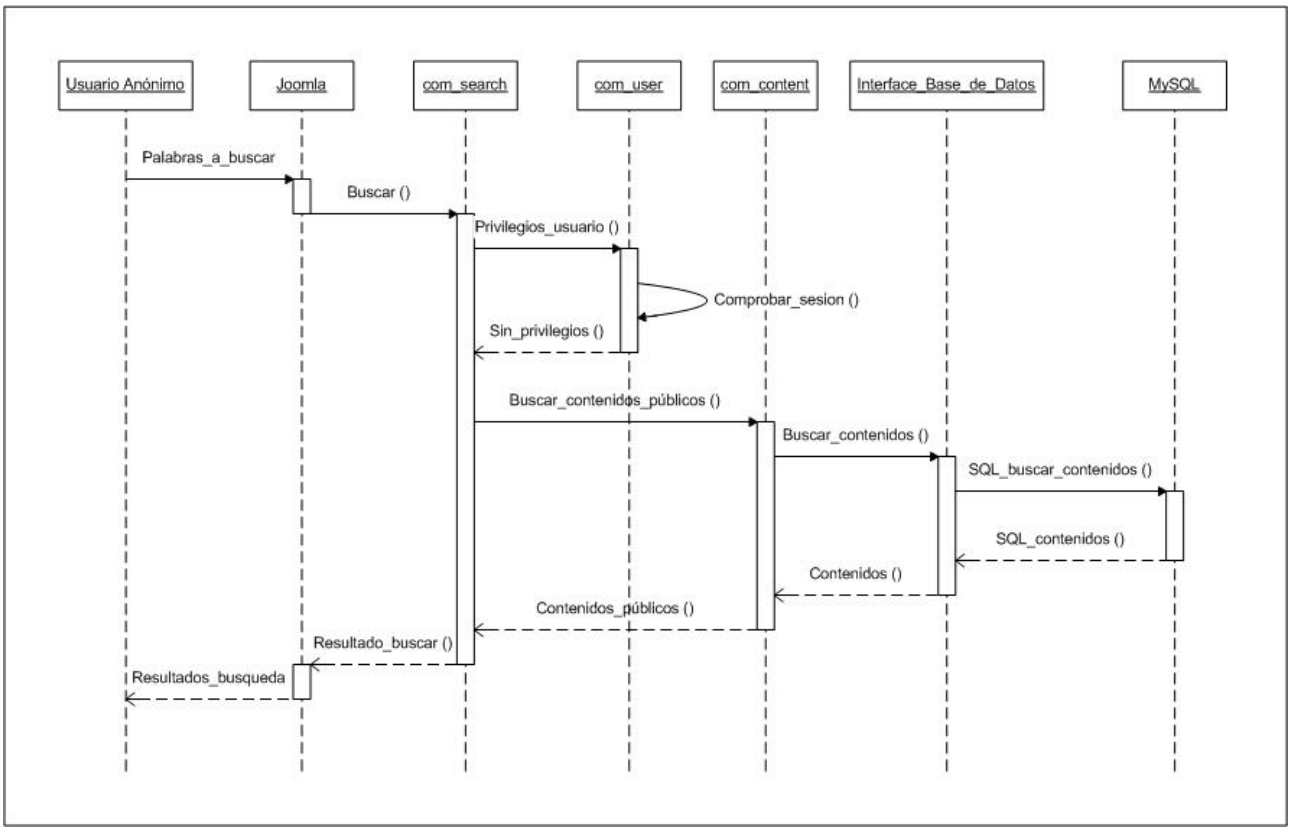


Figura 20: Diagrama de Secuencia “DS005”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU005” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario anónimo” utiliza el motor de búsqueda del portal para buscar un contenido público.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End “JOOMLA” realiza la petición de buscar por los criterios y palabras introducidos por el usuario al componente de JOOMLA “com_search” que es el motor de búsquedas que se encarga de gestionarlas dentro del portal.

2) El componente de JOOMLA “com_search” se comunica con otro componente de JOOMLA denominado “com_user” para preguntarle por los privilegios que posee el usuario de la sesión actual.

3) El componente de JOOMLA “com_user” responde a “com_search” que ese usuario no posee privilegios por lo que sólo podrá ver contenido de carácter público.

4) El componente de JOOMLA "com_search", tras esta respuesta, pide al componente de JOOMLA "com_content" que obtenga el contenido público que concuerde con los criterios del usuario, ya que este se encarga de gestionar los contenidos del portal.

5) El componente de JOOMLA "com_content", a su vez, pide al componente de JOOMLA denominado "Interface Base de Datos" que obtenga el contenido público que peticiona el usuario.

6) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

7) El Gestor de Bases de Datos "MySQL" devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

8) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_content" dándole los contenidos públicos que le había pedido.

9) El componente "com_content" a su vez, devuelve dichos contenidos al motor de búsqueda "com_search".

10) El componente "com_search" muestra dichos contenidos públicos en el formato adecuado al Front-End "JOOMLA", para que pueda ser visionado por el usuario.

3.6.06 - DIAGRAMA DE SECUENCIA “DS006” PARA EL CASO DE USO “CU006”

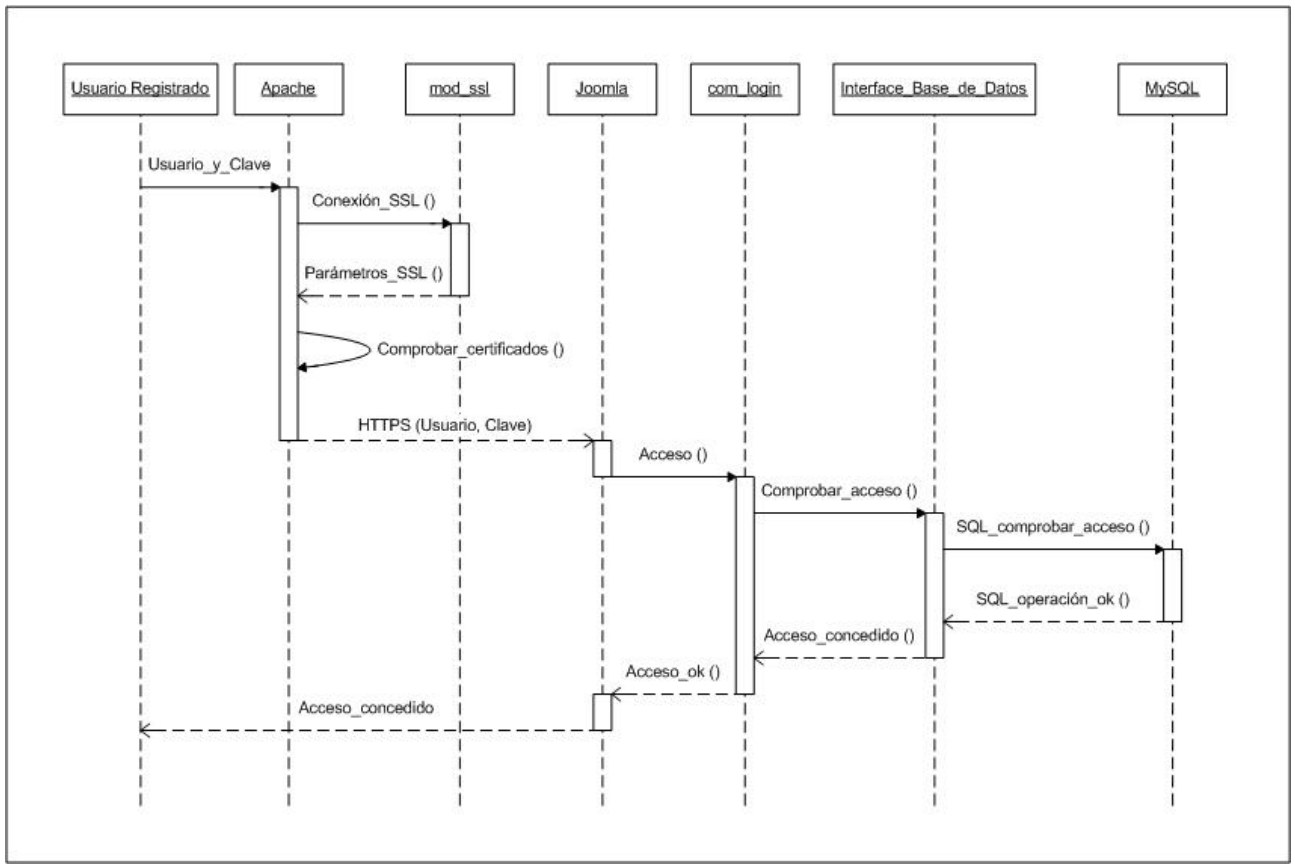


Figura 21: Diagrama de Secuencia “DS006”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU006” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” se autentica en el portal mediante un formulario de acceso.

Esta acción será una premisa para que este usuario pueda realizar cualquier otra acción con privilegios dentro de dicho portal.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el usuario pulsa el botón enviar en el formulario de acceso del portal, dicha acción será redirigida a un canal seguro SSL para que los datos “usuario” y “clave” no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor “Apache” antes de establecer dicha conexión segura SSL, comprueba que tanto el propio “servidor” como el “cliente” posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor “Apache” envía por un canal seguro SSL los parámetros “usuario” y “clave” al Front-End “JOOMLA”, interpretado en el diagrama mediante “HTTPS (Usuario, Clave)”.

5) El Front-End “JOOMLA” realiza la petición de acceso con estos parámetros al componente de JOOMLA “com_login” que se encarga de gestionar los accesos en el portal.

6) El componente de JOOMLA “com_login”, a su vez, pide al componente de JOOMLA denominado “Interface Base de Datos” que consulte dichos parámetros de acceso.

7) El componente de JOOMLA “Interface Base de Datos” a su vez realiza esa petición al Gestor de Bases de Datos “MySQL” esta vez en lenguaje SQL, para comprobar en ésta, los datos que proporciona el usuario.

8) El Gestor de Bases de Datos “MySQL” devuelve que dicha operación ha sido realizada con éxito, en lenguaje SQL, al componente de JOOMLA “Interface Base de Datos”.

9) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al componente “com_login” diciéndole que el acceso ha sido concedido.

10) El componente “com_login” muestra dicho acceso concedido en el formato adecuado al Front-End “JOOMLA”, para que pueda ser visionado por el usuario.

3.6.07 - DIAGRAMA DE SECUENCIA “DS007” PARA EL CASO DE USO “CU007”

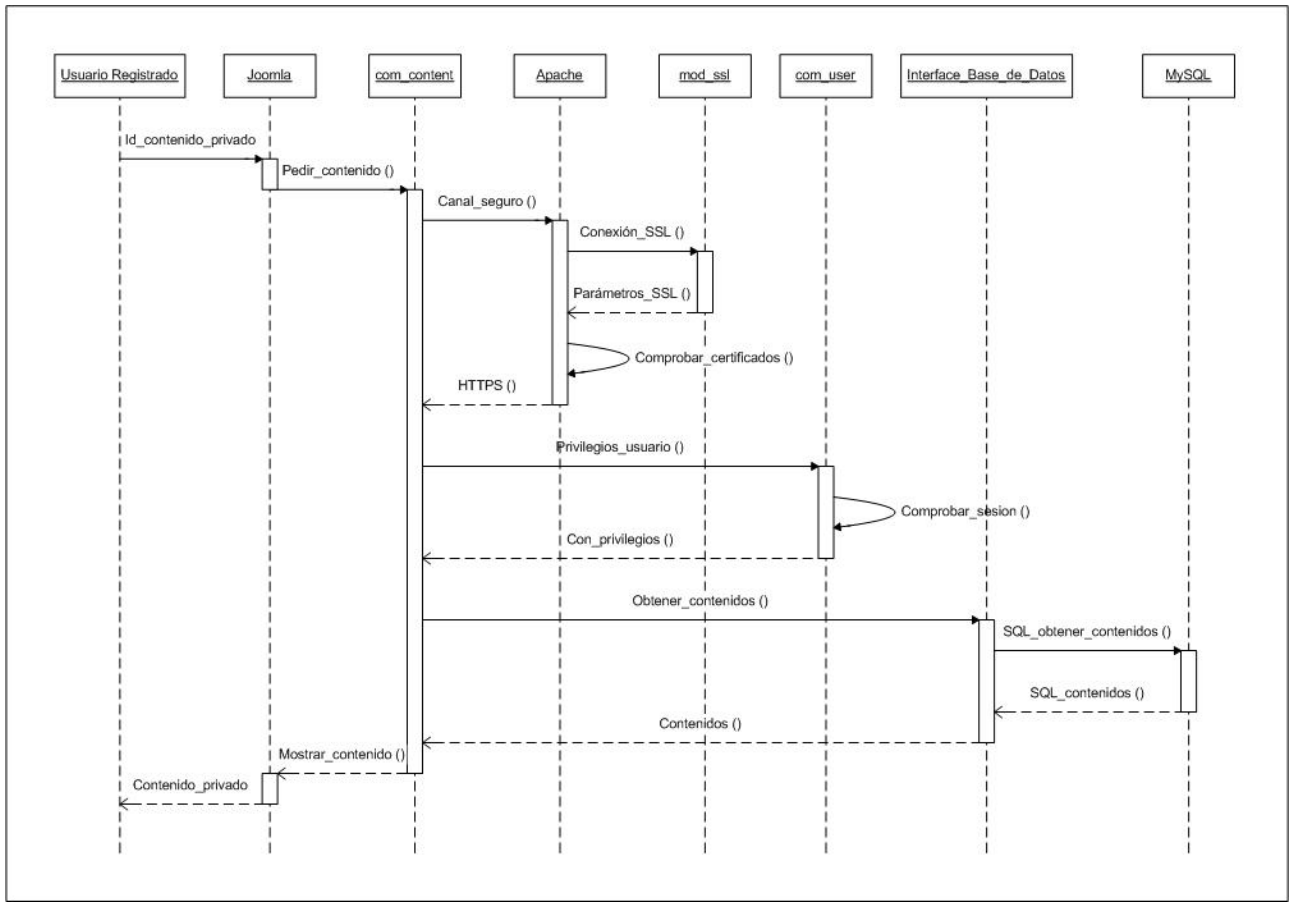


Figura 22: Diagrama de Secuencia “DS007”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU007” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” consulta algún contenido de carácter privado.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End “JOOMLA” realiza una petición de contenido al componente de JOOMLA “com_content” que se encarga de gestionar los contenidos.

2) El servidor “Apache” cuando detecta que se esta realizando una petición de contenido privado en el portal actuará de manera que, dicha acción será redirigida a un canal seguro SSL para que esos datos sensibles no puedan ser interceptados en un posible ataque.

3) El servidor "Apache" pide establecer un canal seguro mediante el protocolo SSL a su módulo "mod_ssl".

4) El módulo de apache "mod_ssl" responde a "Apache" enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

5) El servidor "Apache" antes de establecer dicha conexión segura SSL, comprueba que tanto el propio "servidor" como el "cliente" posean un certificado válido que haga posible establecer dicho canal seguro, una vez comprobado se establece dicho canal, interpretado en el diagrama mediante "HTTPS ()".

6) El componente de JOOMLA "com_content" se comunica con otro componente de JOOMLA denominado "com_user" para preguntarle por los privilegios que posee el usuario de la sesión actual.

7) El componente de JOOMLA "com_user" responde a "com_content" que ese usuario posee privilegios por lo que podrá ver contenidos tanto públicos como privados.

8) El componente de JOOMLA "com_content", tras esta respuesta, pide al componente de JOOMLA denominado "Interface Base de Datos" que obtenga el contenido privado que peticiona el usuario.

9) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

10) El Gestor de Bases de Datos "MySQL" devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

11) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_content" dándole el contenido que le había pedido.

12) El componente "com_content" a su vez muestra dicho contenido en el formato adecuado al Front-End "JOOMLA", para que pueda ser visionado por el usuario.

3.6.08 - DIAGRAMA DE SECUENCIA “DS008” PARA EL CASO DE USO “CU008”

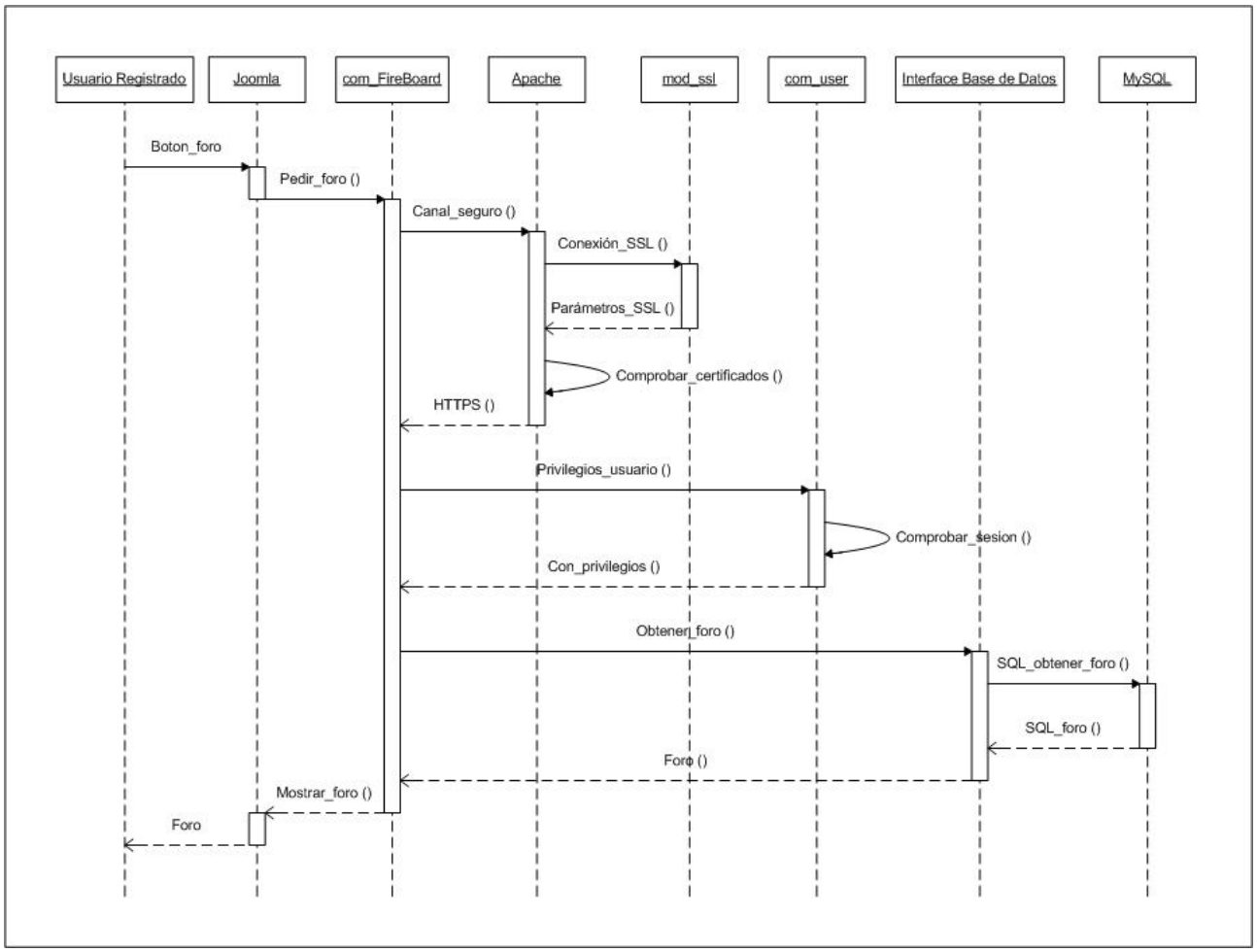


Figura 23: Diagrama de Secuencia “DS008”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU008” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” utiliza el foro.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End “JOOMLA” realiza una petición del foro al componente de JOOMLA “com_FireBoard” que se encarga de gestionar el foro.

2) El servidor “Apache” cuando detecta que se esta realizando una petición del foro en el portal actuará de manera que, dicha acción será redirigida a un canal seguro SSL para que esos datos sensibles no puedan ser interceptados en un posible ataque.

3) El servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

4) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

5) El servidor “Apache” antes de establecer dicha conexión segura SSL, comprueba que tanto el propio “servidor” como el “cliente” posean un certificado válido que haga posible establecer dicho canal seguro, una vez comprobado se establece dicho canal, interpretado en el diagrama mediante “HTTPS ()”.

6) El componente de JOOMLA “com_FireBoard” se comunica con otro componente de JOOMLA denominado “com_user” para preguntarle por los privilegios que posee el usuario de la sesión actual.

7) El componente de JOOMLA “com_user” responde a “com_FireBoard” que ese usuario posee privilegios por lo que podrá utilizar el foro.

8) El componente de JOOMLA “com_FireBoard”, tras esta respuesta, pide al componente de JOOMLA denominado “Interface Base de Datos” que obtenga el foro que peticiona el usuario.

9) El componente de JOOMLA “Interface Base de Datos” a su vez realiza esa petición al Gestor de Bases de Datos “MySQL” esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

10) El Gestor de Bases de Datos “MySQL” devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA “Interface Base de Datos”.

11) El componente “Interface_Base_de_Datos” interpreta el lenguaje SQL y responde al componente “com_FireBoard” dándole el foro que le había pedido.

12) El componente “com_FireBoard” a su vez muestra dicho foro en el formato adecuado al Front-End “JOOMLA”, para que pueda ser visionado por el usuario.

3.6.09 - DIAGRAMA DE SECUENCIA “DS009” PARA EL CASO DE USO “CU009”

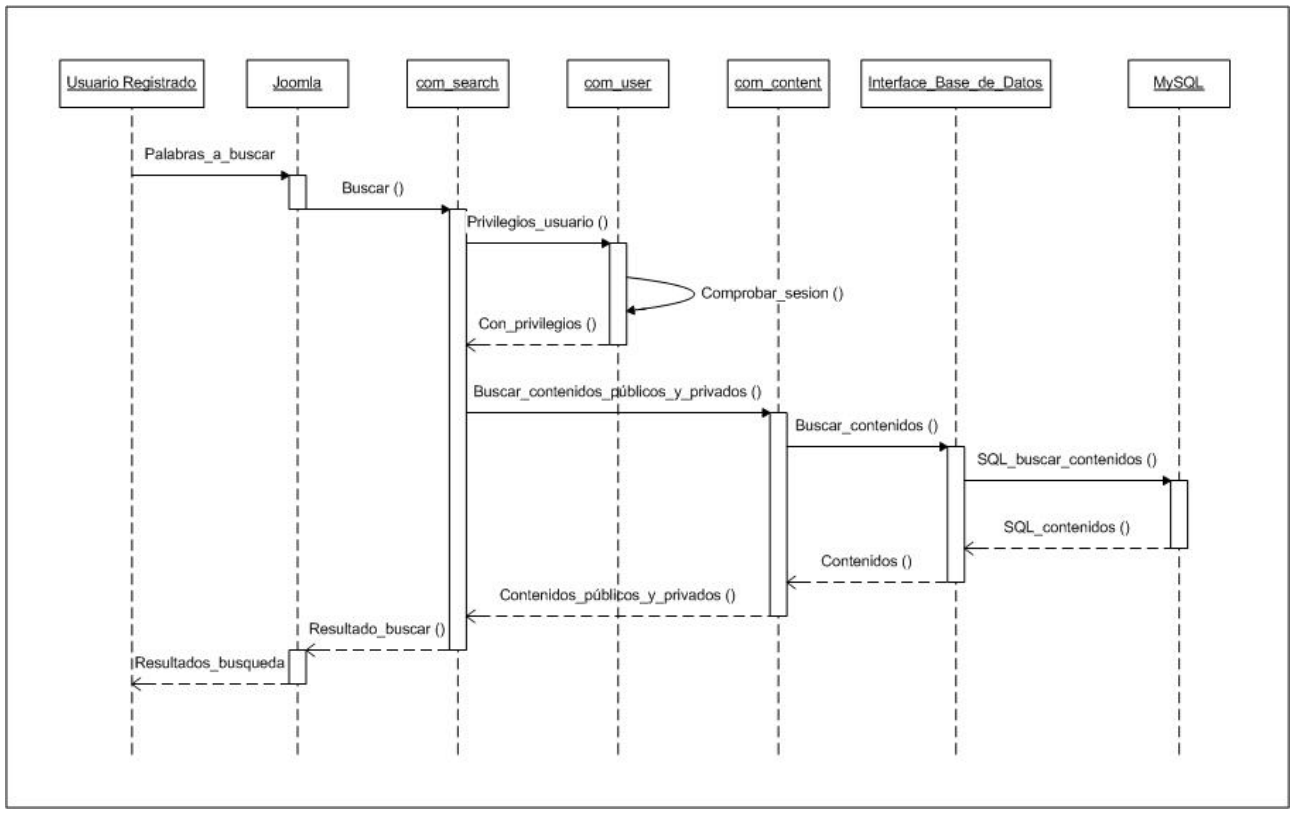


Figura 24: Diagrama de Secuencia “DS009”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU009” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando un “usuario registrado” utiliza el motor de búsqueda del portal para buscar un contenido privado.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) El Front-End “JOOMLA” realiza la petición de buscar por los criterios y palabras introducidos por el usuario al componente de JOOMLA “com_search” que es el motor de búsquedas que se encarga de gestionarlas dentro del portal.

2) El componente de JOOMLA “com_search” se comunica con otro componente de JOOMLA denominado “com_user” para preguntarle por los privilegios que posee el usuario de la sesión actual.

3) El componente de JOOMLA “com_user” responde a “com_search” que ese usuario posee privilegios por lo que podrá ver contenidos de carácter público y privado.

4) El componente de JOOMLA "com_search", tras esta respuesta, pide al componente de JOOMLA "com_content" que obtenga el contenido público y privado que concuerde con los criterios del usuario, ya que este se encarga de gestionar los contenidos del portal.

5) El componente de JOOMLA "com_content", a su vez, pide al componente de JOOMLA denominado "Interface Base de Datos" que obtenga el contenido público y privado que peticiona el usuario.

6) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para buscar en ésta, los datos que peticiona el usuario.

7) El Gestor de Bases de Datos "MySQL" devuelve los datos pedidos en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

8) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_content" dándole los contenidos públicos y privados que le había pedido.

9) El componente "com_content" a su vez, devuelve dichos contenidos al motor de búsqueda "com_search".

10) El componente "com_search" muestra dichos contenidos públicos y privados en el formato adecuado al Front-End "JOOMLA", para que pueda ser visionado por el usuario.

3.6.10 - DIAGRAMA DE SECUENCIA “DS010” PARA EL CASO DE USO “CU010”

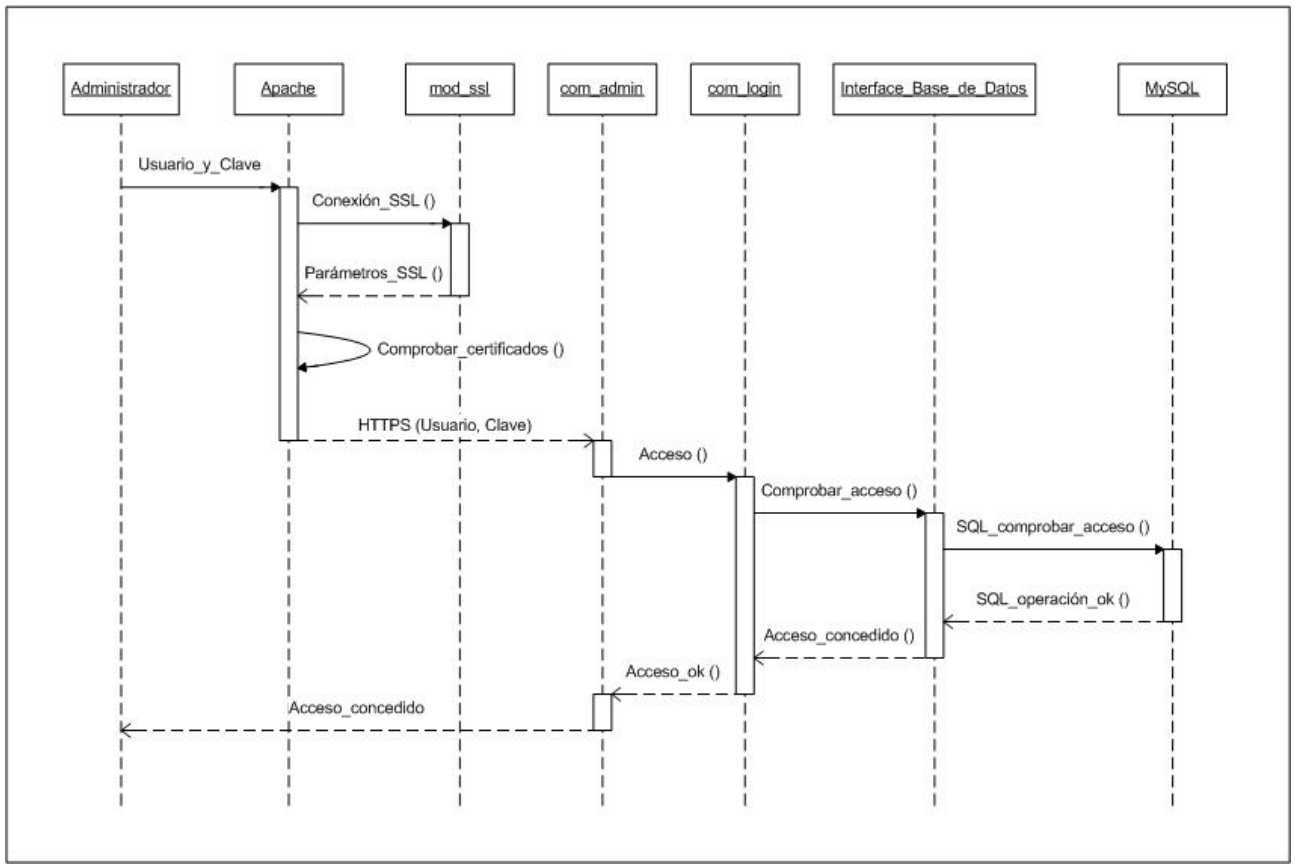


Figura 25: Diagrama de Secuencia “DS010”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU010” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” se autentica en el “Back-End” mediante un formulario de acceso.

Esta acción será una premisa para que el administrador pueda realizar cualquier otra acción con privilegios dentro del “Back-End”.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador pulsa el botón enviar en el formulario de acceso del “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos “usuario” y “clave” no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache "mod_ssl" responde a "Apache" enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor "Apache" antes de establecer dicha conexión segura SSL, comprueba que tanto el propio "servidor" como el "cliente" posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor "Apache" envía por un canal seguro SSL los parámetros "usuario" y "clave" al Back-End "com_admin", interpretado en el diagrama mediante "HTTPS (Usuario, Clave)".

5) El Back-End "com_admin" realiza la petición de acceso con estos parámetros al componente de JOOMLA "com_login" que se encarga de gestionar los accesos en el portal.

6) El componente de JOOMLA "com_login", a su vez, pide al componente de JOOMLA denominado "Interface Base de Datos" que consulte dichos parámetros de acceso.

7) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para comprobar en ésta, los datos que proporciona el administrador.

8) El Gestor de Bases de Datos "MySQL" devuelve que dicha operación ha sido realizada con éxito, en lenguaje SQL, al componente de JOOMLA "Interface Base de Datos".

9) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_login" diciéndole que el acceso ha sido concedido.

10) El componente "com_login" muestra dicho acceso concedido en el formato adecuado al Back-End "com_admin", para que pueda ser visionado por el administrador.

3.6.11 - DIAGRAMA DE SECUENCIA “DS011” PARA EL CASO DE USO “CU011”

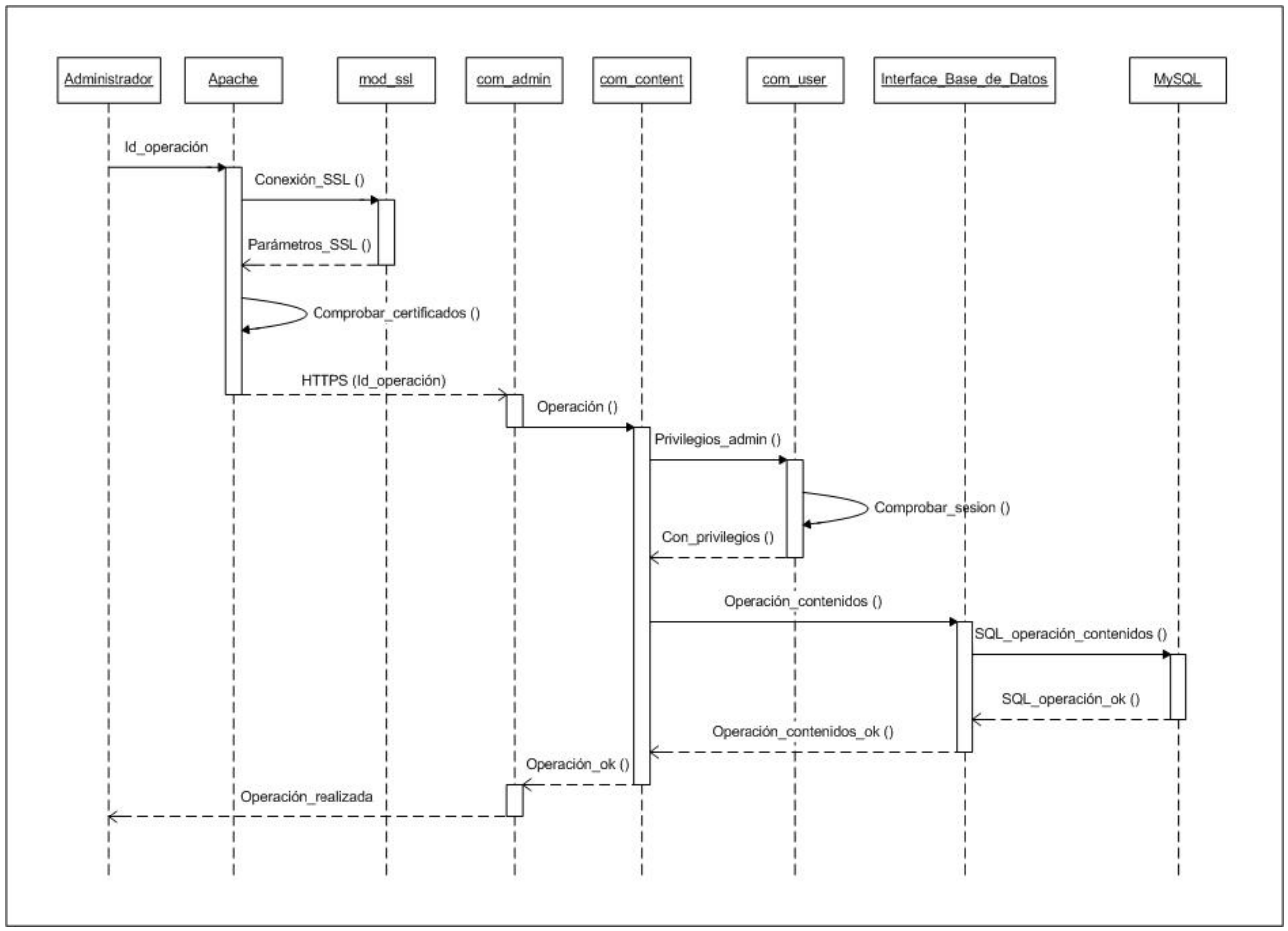


Figura 26: Diagrama de Secuencia “DS011”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU011” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” realiza alguna operación sobre algún contenido, pudiendo ser este de carácter tanto público como privado, ya sea para añadir algún contenido, borrarlo o modificarlo.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador va ha realizar una operación cualquiera sobre el “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor "Apache" antes de establecer dicha conexión segura SSL, comprueba que tanto el propio "servidor" como el "cliente" posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor "Apache" envía por un canal seguro SSL el parámetro identificador de la operación a realizar "Id_operación" al Back-End "com_admin", interpretado en el diagrama mediante "HTTPS (Id_operación)".

5) El Back-End "com_admin" realiza una petición de operación sobre contenido al componente de JOOMLA "com_content" que se encarga de gestionar los contenidos.

6) El componente de JOOMLA "com_content" se comunica con otro componente de JOOMLA denominado "com_user" para preguntarle por los privilegios que posee el administrador de la sesión actual.

7) El componente de JOOMLA "com_user" responde a "com_content" que ese administrador posee privilegios por lo que podrá gestionar contenidos tanto públicos como privados.

8) El componente de JOOMLA "com_content", tras esta respuesta, pide al componente de JOOMLA denominado "Interface Base de Datos" que realice la operación sobre el contenido que desea el administrador.

9) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para realizar en ésta, la operación que peticiona el administrador.

10) El Gestor de Bases de Datos "MySQL" devuelve respuesta sobre la operación realizada en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

11) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_content" diciéndole si la operación sobre el contenido ha sido realizada con éxito.

12) El componente "com_content" a su vez muestra el resultado de dicha operación sobre el contenido en el formato adecuado al Back-End "com_admin", para que pueda ser visionado por el administrador.

3.6.12 - DIAGRAMA DE SECUENCIA “DS012” PARA EL CASO DE USO “CU012”

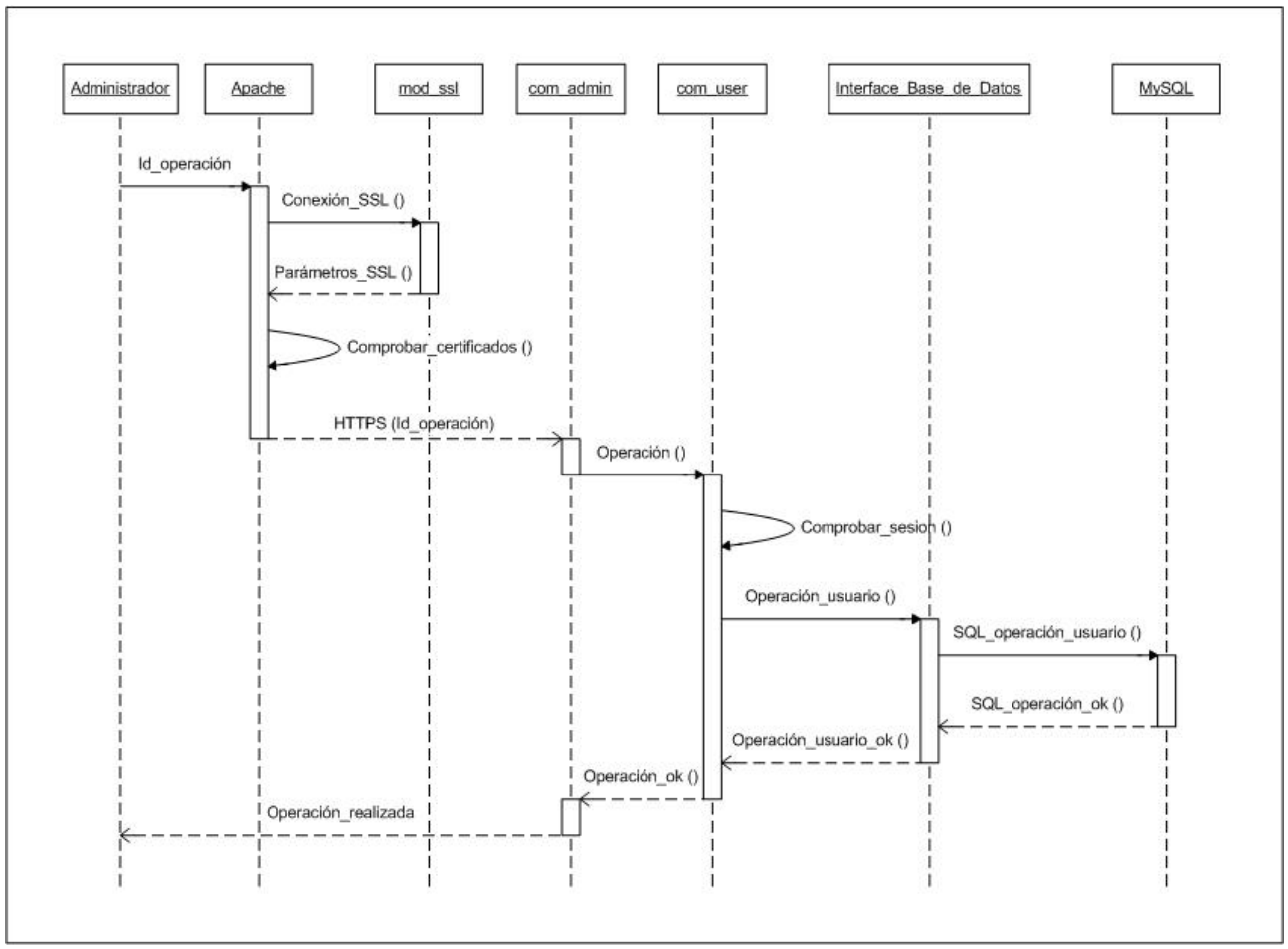


Figura 27: Diagrama de Secuencia “DS012”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU012” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” realiza alguna operación sobre algún usuario, ya sea para darle de alta, darle de baja o modificar sus datos.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador va ha realizar una operación cualquiera sobre el “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor “Apache” antes de establecer dicha conexión segura SSL, comprueba que tanto el propio “servidor” como el “cliente” posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor “Apache” envía por un canal seguro SSL el parámetro identificador de la operación a realizar “Id_operación” al Back-End “com_admin”, interpretado en el diagrama mediante “HTTPS (Id_operación)”.

5) El Back-End “com_admin” realiza una petición de operación sobre usuarios al componente de JOOMLA “com_user” que se encarga de gestionar los usuarios.

6) El componente de JOOMLA “com_user” comprueba los privilegios que posee el administrador de la sesión actual.

7) El componente de JOOMLA “com_user”, tras esta respuesta, pide al componente de JOOMLA denominado “Interface Base de Datos” que realice la operación sobre el usuario que desea el administrador.

8) El componente de JOOMLA “Interface Base de Datos” a su vez realiza esa petición al Gestor de Bases de Datos “MySQL” esta vez en lenguaje SQL, para realizar en ésta, la operación que peticiona el administrador.

9) El Gestor de Bases de Datos “MySQL” devuelve respuesta sobre la operación realizada en lenguaje SQL al componente de JOOMLA “Interface Base de Datos”.

10) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al componente “com_user” diciéndole si la operación sobre el usuario ha sido realizada con éxito.

11) El componente “com_user” a su vez muestra el resultado de dicha operación sobre el usuario en el formato adecuado al Back-End “com_admin”, para que pueda ser visionado por el administrador.

3.6.13 - DIAGRAMA DE SECUENCIA “DS013” PARA EL CASO DE USO “CU013”

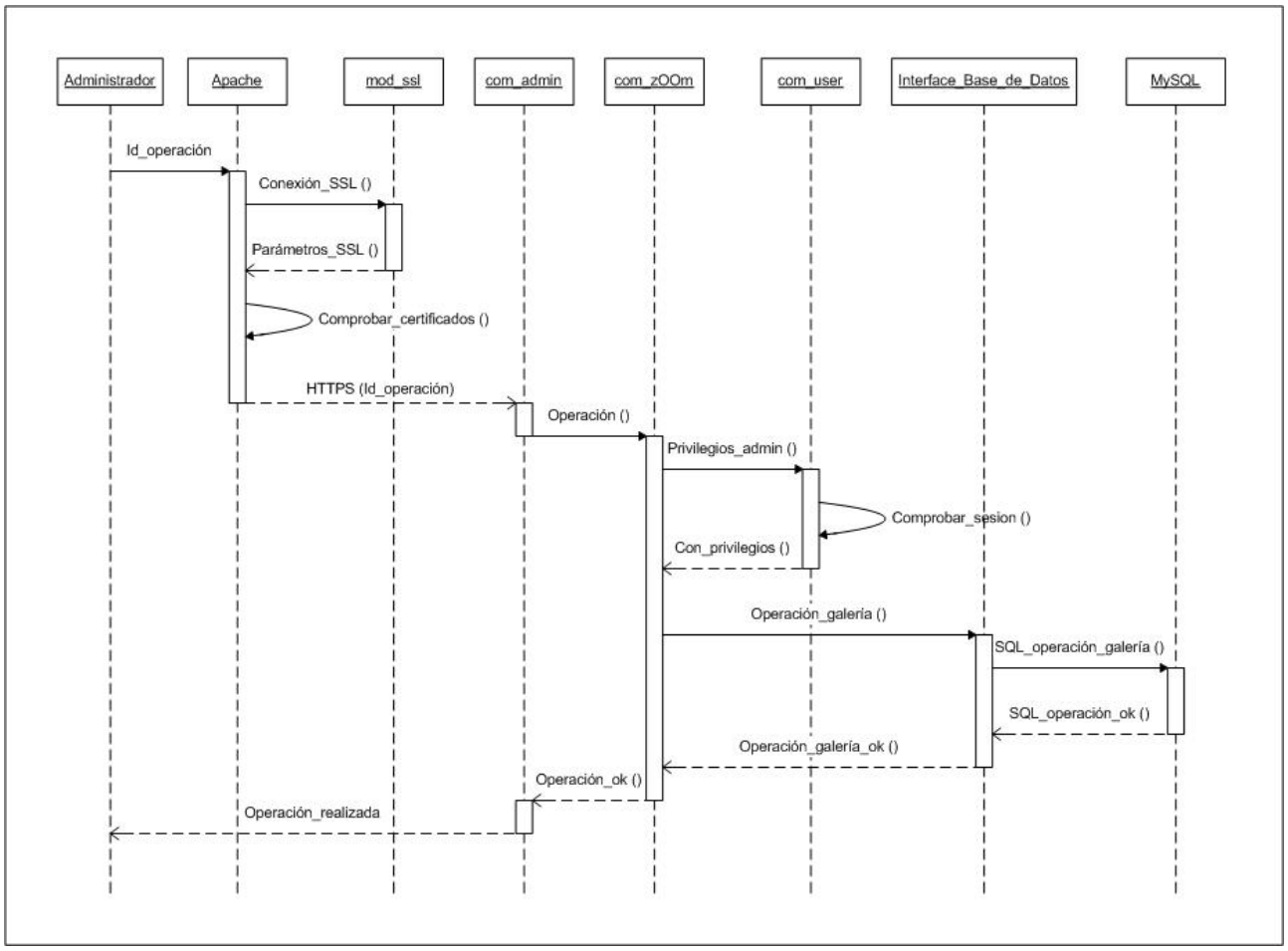


Figura 28: Diagrama de Secuencia “DS013”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU013” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” realiza alguna operación sobre la galería de imágenes, ya sea para añadir alguna imagen, borrarla o modificarla.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador va ha realizar una operación cualquiera sobre el “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor “Apache” antes de establecer dicha conexión segura SSL, comprueba que tanto el propio “servidor” como el “cliente” posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor “Apache” envía por un canal seguro SSL el parámetro identificador de la operación a realizar “Id_operación” al Back-End “com_admin”, interpretado en el diagrama mediante “HTTPS (Id_operación)”.

5) El Back-End “com_admin” realiza una petición de operación sobre la galería de imágenes al componente de JOOMLA “com_zOOm” que se encarga de gestionar dicha galería.

6) El componente de JOOMLA “com_zOOm” se comunica con otro componente de JOOMLA denominado “com_user” para preguntarle por los privilegios que posee el administrador de la sesión actual.

7) El componente de JOOMLA “com_user” responde a “com_zOOm” que ese administrador posee privilegios por lo que podrá gestionar la galería de imágenes.

8) El componente de JOOMLA “com_zOOm”, tras esta respuesta, pide al componente de JOOMLA denominado “Interface Base de Datos” que realice la operación sobre la galería que desea el administrador.

9) El componente de JOOMLA “Interface Base de Datos” a su vez realiza esa petición al Gestor de Bases de Datos “MySQL” esta vez en lenguaje SQL, para realizar en ésta, la operación que peticiona el administrador.

10) El Gestor de Bases de Datos “MySQL” devuelve respuesta sobre la operación realizada en lenguaje SQL al componente de JOOMLA “Interface Base de Datos”.

11) El componente “Interface Base de Datos” interpreta el lenguaje SQL y responde al componente “com_zOOm” diciéndole si la operación sobre la galería ha sido realizada con éxito.

12) El componente “com_zOOm” a su vez muestra el resultado de dicha operación sobre la galería de imágenes en el formato adecuado al Back-End “com_admin”, para que pueda ser visionado por el administrador.

3.6.14 - DIAGRAMA DE SECUENCIA “DS014” PARA EL CASO DE USO “CU014”

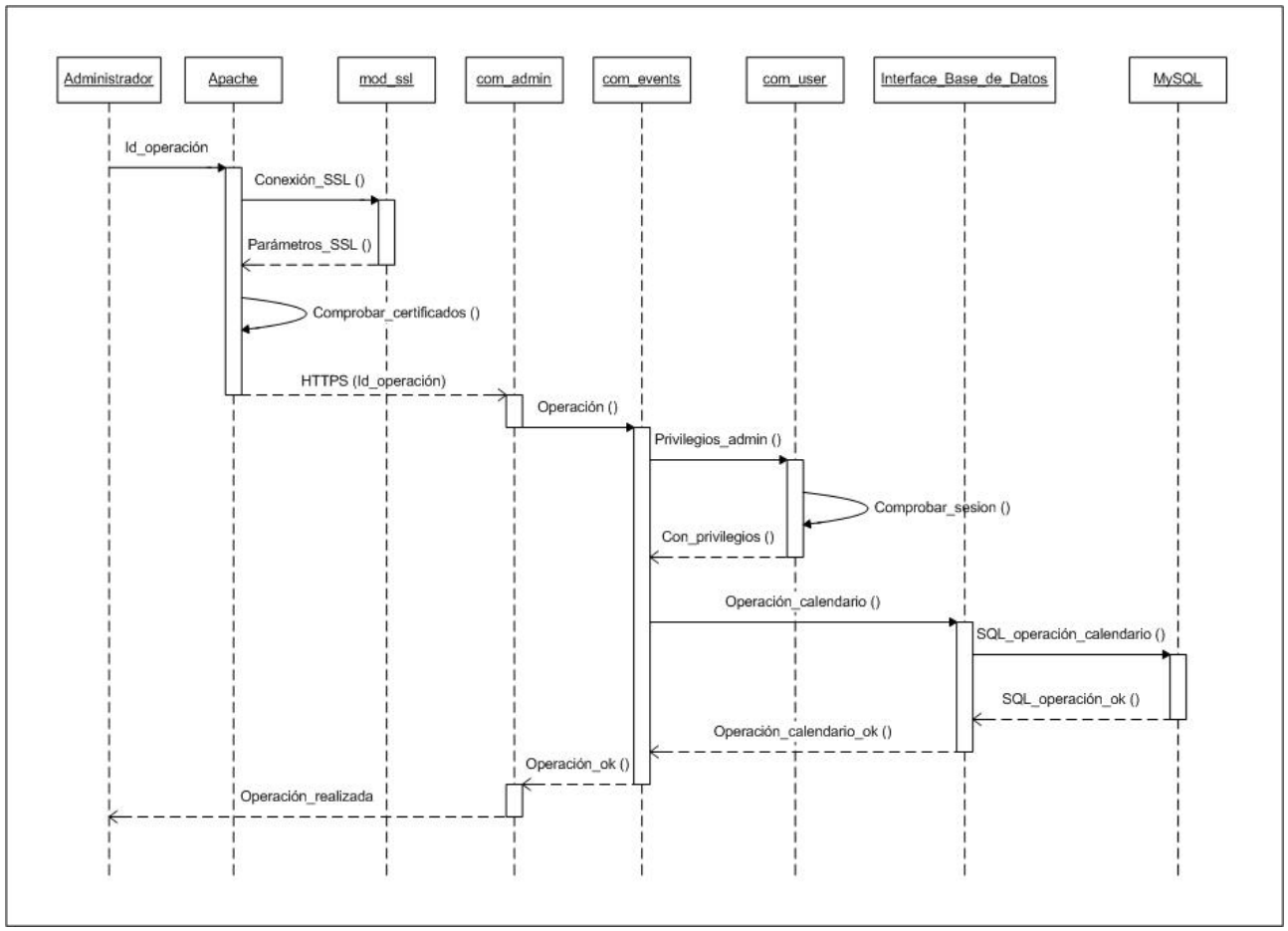


Figura 29: Diagrama de Secuencia “DS014”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU014” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” realiza alguna operación sobre el calendario, ya sea para añadir algún evento, borrarlo o modificarlo.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador va ha realizar una operación cualquiera sobre el “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor "Apache" antes de establecer dicha conexión segura SSL, comprueba que tanto el propio "servidor" como el "cliente" posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor "Apache" envía por un canal seguro SSL el parámetro identificador de la operación a realizar "Id_operación" al Back-End "com_admin", interpretado en el diagrama mediante "HTTPS (Id_operación)".

5) El Back-End "com_admin" realiza una petición de operación sobre el calendario al componente de JOOMLA "com_events" que se encarga de gestionar el calendario.

6) El componente de JOOMLA "com_events" se comunica con otro componente de JOOMLA denominado "com_user" para preguntarle por los privilegios que posee el administrador de la sesión actual.

7) El componente de JOOMLA "com_user" responde a "com_events" que ese administrador posee privilegios por lo que podrá gestionar el calendario.

8) El componente de JOOMLA "com_events", tras esta respuesta, pide al componente de JOOMLA denominado "Interface Base de Datos" que realice la operación sobre el calendario que desea el administrador.

9) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para realizar en ésta, la operación que peticiona el administrador.

10) El Gestor de Bases de Datos "MySQL" devuelve respuesta sobre la operación realizada en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

11) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_events" diciéndole si la operación sobre el calendario ha sido realizada con éxito.

12) El componente "com_events" a su vez muestra el resultado de dicha operación sobre el calendario en el formato adecuado al Back-End "com_admin", para que pueda ser visionado por el administrador.

3.6.15 - DIAGRAMA DE SECUENCIA “DS015” PARA EL CASO DE USO “CU015”

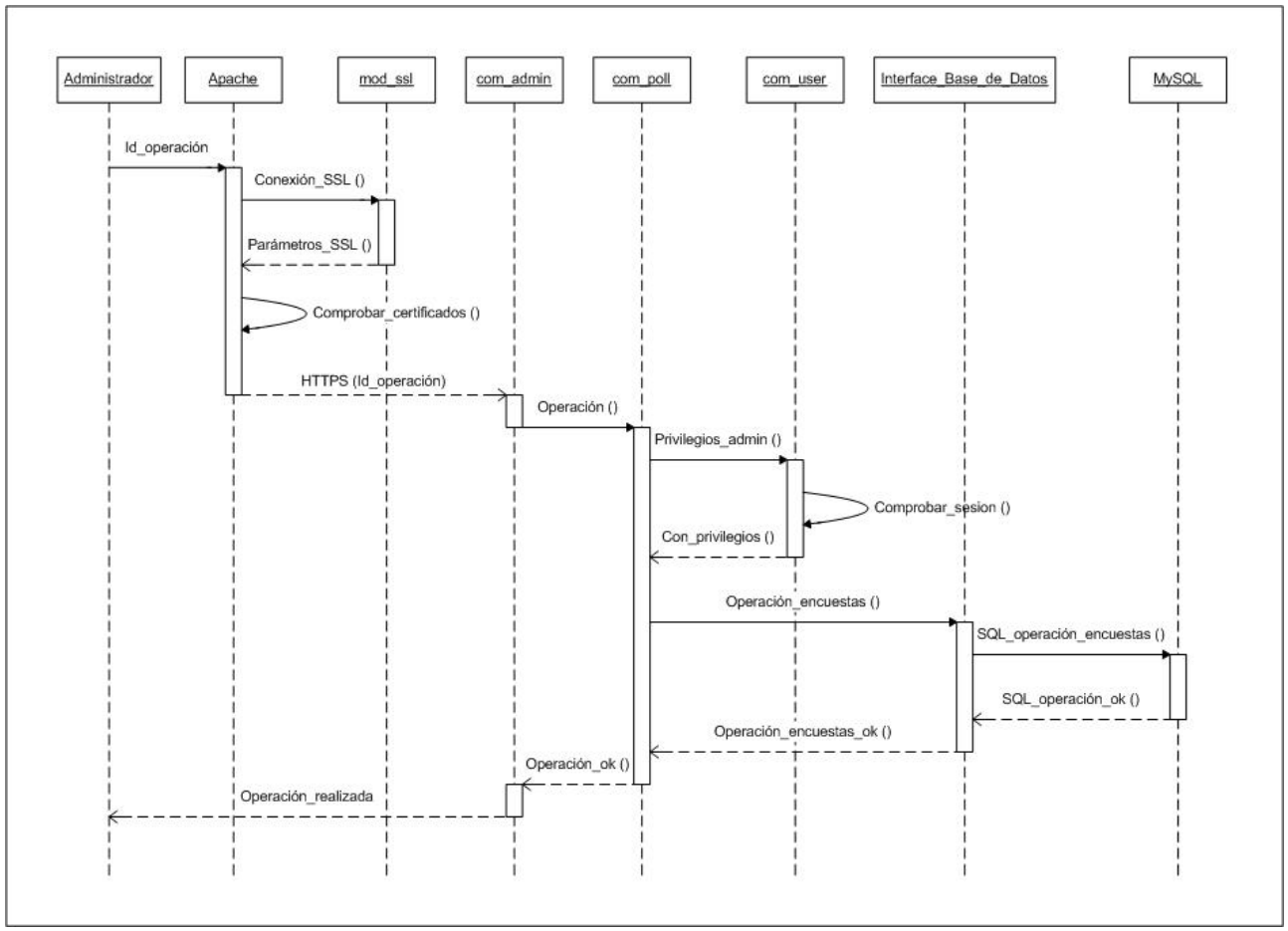


Figura 30: Diagrama de Secuencia “DS015”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU015” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” realiza alguna operación sobre las encuestas, ya sea para añadir alguna, borrarla o modificarla.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador va ha realizar una operación cualquiera sobre el “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor "Apache" antes de establecer dicha conexión segura SSL, comprueba que tanto el propio "servidor" como el "cliente" posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor "Apache" envía por un canal seguro SSL el parámetro identificador de la operación a realizar "Id_operación" al Back-End "com_admin", interpretado en el diagrama mediante "HTTPS (Id_operación)".

5) El Back-End "com_admin" realiza una petición de operación sobre las encuestas al componente de JOOMLA "com_poll" que se encarga de gestionar dichas encuestas.

6) El componente de JOOMLA "com_poll" se comunica con otro componente de JOOMLA denominado "com_user" para preguntarle por los privilegios que posee el administrador de la sesión actual.

7) El componente de JOOMLA "com_user" responde a "com_poll" que ese administrador posee privilegios por lo que podrá gestionar las encuestas.

8) El componente de JOOMLA "com_poll", tras esta respuesta, pide al componente de JOOMLA denominado "Interface Base de Datos" que realice la operación sobre las encuestas que desea el administrador.

9) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para realizar en ésta, la operación que peticiona el administrador.

10) El Gestor de Bases de Datos "MySQL" devuelve respuesta sobre la operación realizada en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

11) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_poll" diciéndole si la operación sobre las encuestas ha sido realizada con éxito.

12) El componente "com_poll" a su vez muestra el resultado de dicha operación sobre las encuestas en el formato adecuado al Back-End "com_admin", para que pueda ser visionado por el administrador.

3.6.16 - DIAGRAMA DE SECUENCIA “DS016” PARA EL CASO DE USO “CU016”

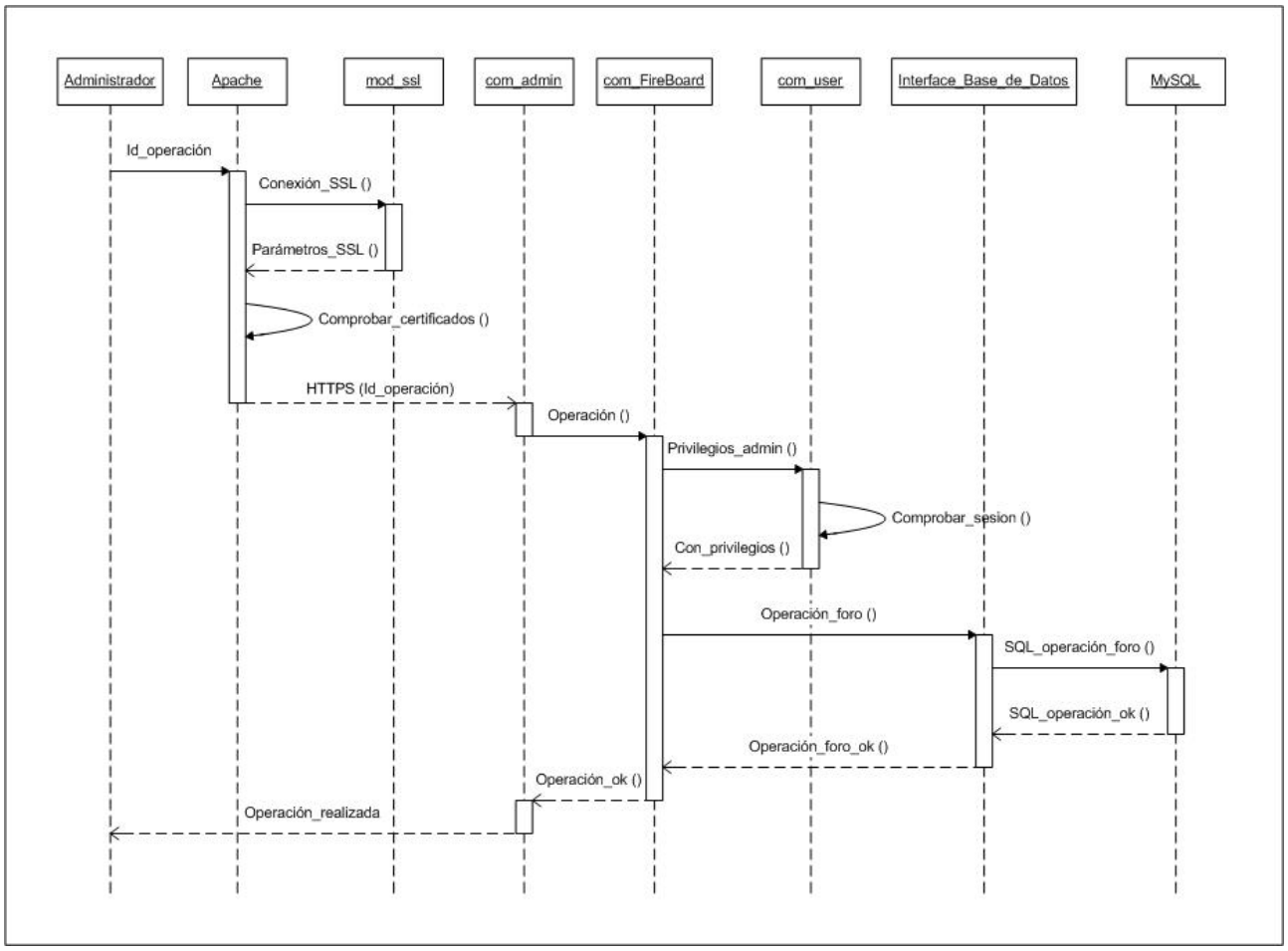


Figura 31: Diagrama de Secuencia “DS016”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU016” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” realiza alguna operación sobre el foro, ya sea para añadir algún tema de discusión, borrarlo o modificarlo.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador va ha realizar una operación cualquiera sobre el “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache “mod_ssl” responde a “Apache” enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor "Apache" antes de establecer dicha conexión segura SSL, comprueba que tanto el propio "servidor" como el "cliente" posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor "Apache" envía por un canal seguro SSL el parámetro identificador de la operación a realizar "Id_operación" al Back-End "com_admin", interpretado en el diagrama mediante "HTTPS (Id_operación)".

5) El Back-End "com_admin" realiza una petición de operación sobre el foro al componente de JOOMLA "com_FireBoard" que se encarga de gestionar el foro.

6) El componente de JOOMLA "com_FireBoard" se comunica con otro componente de JOOMLA denominado "com_user" para preguntarle por los privilegios que posee el administrador de la sesión actual.

7) El componente de JOOMLA "com_user" responde a "com_FireBoard" que ese administrador posee privilegios por lo que podrá gestionar el foro.

8) El componente de JOOMLA "com_FireBoard", tras esta respuesta, pide al componente de JOOMLA denominado "Interface Base de Datos" que realice la operación sobre el foro que desea el administrador.

9) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para realizar en ésta, la operación que peticiona el administrador.

10) El Gestor de Bases de Datos "MySQL" devuelve respuesta sobre la operación realizada en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

11) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_FireBoard" diciéndole si la operación sobre el foro ha sido realizada con éxito.

12) El componente "com_FireBoard" a su vez muestra el resultado de dicha operación sobre el foro en el formato adecuado al Back-End "com_admin", para que pueda ser visionado por el administrador.

3.6.17 - DIAGRAMA DE SECUENCIA “DS017” PARA EL CASO DE USO “CU017”

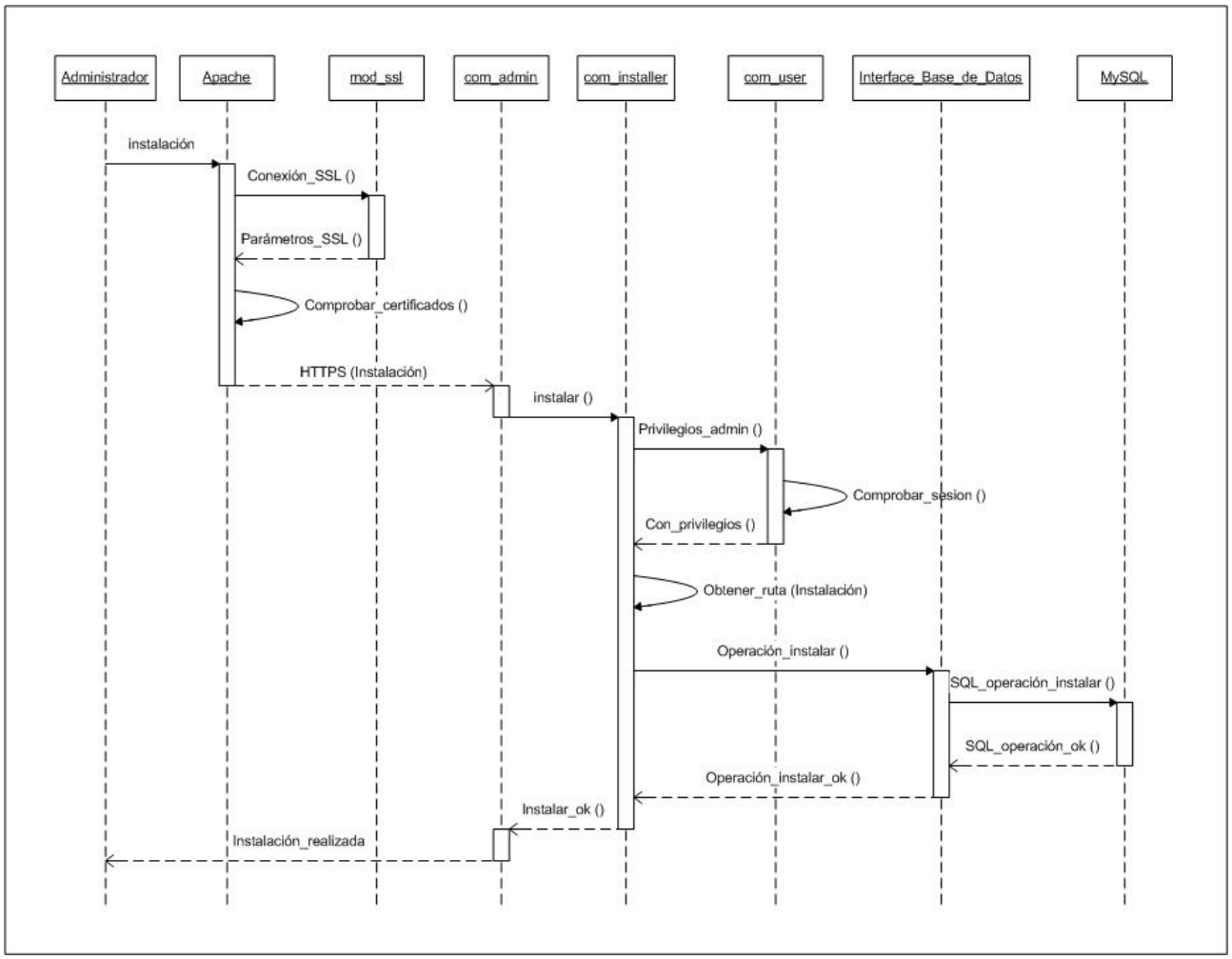


Figura 32: Diagrama de Secuencia “DS017”

Este diagrama de secuencia se corresponde con la acción que realiza el sistema para el caso de uso “CU017” y sirve para mostrar la secuencia de pasos y métodos a utilizar en cada una de las distintas capas que atraviesa la ejecución del programa cuando el “administrador” realiza la instalación de alguna extensión de software sobre el CMS JOOMLA, pudiendo ser estas módulos, componentes o plugins compatibles con JOOMLA, permitiendo añadir dicha extensión, borrarla o modificarla.

A continuación describimos los pasos que realiza este diagrama de secuencia de manera detallada:

1) Cuando el administrador va ha realizar una instalación o actualización cualquiera sobre el “Back-End”, dicha acción será redirigida a un canal seguro SSL para que los datos no puedan ser interceptados en un posible ataque, para lo cual, el servidor “Apache” pide establecer un canal seguro mediante el protocolo SSL a su módulo “mod_ssl”.

2) El módulo de apache "mod_ssl" responde a "Apache" enviándole los parámetros necesarios para poder establecer un canal seguro mediante el protocolo SSL.

3) El servidor "Apache" antes de establecer dicha conexión segura SSL, comprueba que tanto el propio "servidor" como el "cliente" posean un certificado válido que haga posible establecer dicho canal seguro.

4) Una vez realizadas todas estas comprobaciones, el servidor "Apache" envía por un canal seguro SSL los parámetros de la instalación sobre la extensión que se va ha realizar al Back-End "com_admin", interpretado en el diagrama mediante "HTTPS (Instalación)".

5) El Back-End "com_admin" realiza una petición de instalación de una extensión al componente de JOOMLA "com_installer" que se encarga de gestionar dichas instalaciones y actualizaciones de extensiones sobre el CMS JOOMLA.

6) El componente de JOOMLA "com_installer" se comunica con otro componente de JOOMLA denominado "com_user" para preguntarle por los privilegios que posee el administrador de la sesión actual.

7) El componente de JOOMLA "com_user" responde a "com_installer" que ese administrador posee privilegios por lo que podrá instalar extensiones.

8) El componente de JOOMLA "com_installer" comprobará la ruta donde se encuentre el paquete de instalación de la extensión, dicha operación se representa en este diagrama mediante la expresión: "Obtener_ruta (Instalación)".

9) El componente de JOOMLA "com_installer", tras estas comprobaciones, pide al componente de JOOMLA denominado "Interface Base de Datos" que realice la instalación que desea el administrador.

10) El componente de JOOMLA "Interface Base de Datos" a su vez realiza esa petición al Gestor de Bases de Datos "MySQL" esta vez en lenguaje SQL, para realizar en ésta, la instalación que peticiona el administrador.

11) El Gestor de Bases de Datos "MySQL" devuelve respuesta sobre la operación realizada en lenguaje SQL al componente de JOOMLA "Interface Base de Datos".

12) El componente "Interface Base de Datos" interpreta el lenguaje SQL y responde al componente "com_installer" diciéndole si la operación de instalación ha sido realizada con éxito.

13) El componente "com_installer" a su vez muestra el resultado de dicha operación de instalación en el formato adecuado al Back-End "com_admin", para que pueda ser visionado por el administrador.

3.7 - TRAZABILIDAD

Trazabilidad significa capturar las relaciones de implementación y dependencia en el modelo, siguiendo un histórico de los pasos seguidos para desarrollar el software.

Por ejemplo:

1) Un proceso de negocio requerirá estudiarlo para poder obtener del cliente que nos reclama dicho software una serie de "*REQUISITOS*".

2) A partir de dichos requisitos definiremos las principales funcionalidades del sistema "*CASOS DE USO*" para implementar las funciones del proceso.

3) De estudiar, dividiendo en pasos y tareas estos casos de uso, surgirán más tarde los "*DIAGRMAS DE SECUENCIA*".

El proceso anterior nos hace ver una serie de pasos que van avanzando desde un lenguaje cercano a nuestro (lenguaje natural) hacia un lenguaje más cercano al software.

En el caso del software desarrollado en este proyecto, un ejemplo de la trazabilidad desde un requisito hasta su diagrama de secuencia correspondiente seria el siguiente:

REQUISITO (**RUC001**) → CASO DE USO (**CU001**) → DIAGRAMA DE SECUENCIA (**DS001**)

3.7.1 - TABLA DE TRAZABILIDAD

TABLA DE TRAZABILIDAD		
REQUISITOS DE CAPACIDAD	**CASOS DE USO**	**DIAGRAMAS DE SECUENCIA**
RUC001	CU001	DS001
RUC002	CU002	DS002
RUC003	CU003	DS003
RUC004	CU004	DS004
RUC005	CU005	DS005
RUC006	CU006	DS006
RUC007	CU007	DS007
RUC008	CU008	DS008
RUC009	CU009	DS009
RUC010	CU010	DS010
RUC011	CU011	DS011
RUC012	CU012	DS012
RUC013	CU013	DS013
RUC014	CU014	DS014
RUC015	CU015	DS015
RUC016	CU016	DS016
RUC017	CU017	DS017

3.8 - DISEÑO DEL PORTAL

El siguiente diagrama muestra la jerarquía de diseño planeada para generar nuestro Portal web desde el punto de vista del usuario final o Front-End. En la esquina inferior derecha se referencia el punto de esta documentación en donde se amplia información acerca del "item" del diagrama.

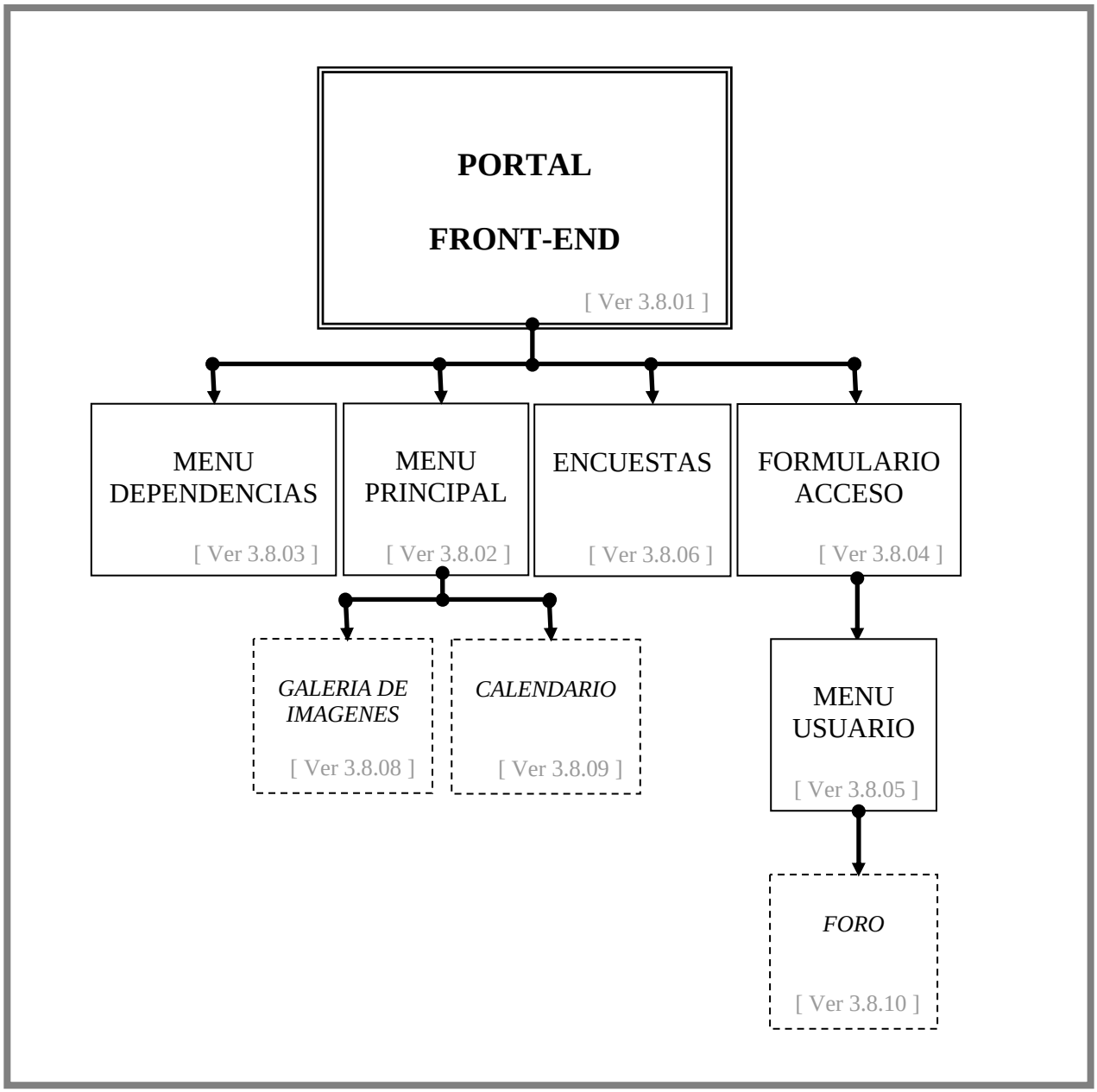


Figura 33: Diseño del Portal

A continuación, vamos a hacer una descripción gráfica (las imágenes han salido del portal ya implementado) del diseño del portal, de modo que se vaya viendo, paso a paso, cómo se ha decidido diseñar cada aspecto del portal acorde a los requisitos funcionales especificados.

3.8.01 - DISEÑO DEL FRONT-END

Figura 34: Diseño del Front-End

Como podemos observar en la captura de la portada, vemos que la distribución del portal esta dividida por zonas, siguiendo este esquema:

- ✓ En la parte superior: Un logo dinámico desarrollado en tecnología flash para dicho sindicato y el motor de búsqueda del portal.
- ✓ En la parte central: Contenidos estáticos y dinámicos.
- ✓ En el lateral izquierdo: El menú principal con sus opciones y el menú de dependencias que sirve como directorio de teléfonos, faxes y direcciones.
- ✓ En el lateral derecho: El formulario de acceso a los afiliados, un formulario de encuestas y un contador de visitas que incluye estadísticas por días.

En la parte superior, se muestra el logo del Sindicato de Policía U.F.P. al que se le realiza la implementación de la aplicación web, este ha sido desarrollado en este proyecto mediante tecnología flash con la finalidad de ofrecer un toque de dinamismo al portal.

Figura 35: Logo U.F.P. - Secuencia animada en Flash

Como se muestra en la secuencia anterior de imágenes, la animación representa un tornado que deposita las letras del logo sobre el fondo corporativo de dicho sindicato.

Se ha realizado una plantilla CSS (plantilla de apariencia) de modo que todo sea visible fácilmente, con la finalidad de que se diferencien bien las secciones y que el usuario no se vea desbordado por un sinfín de colores y fuentes.

El color predominante será el azul corporativo, ya que este es el color del Cuerpo Nacional de Policía, y el fondo de pantalla se basa en tonalidades claras con la finalidad de que la vista del usuario no se canse tanto como con tonalidades más fuertes y a la vez se tenga una visión clara de los contenidos.

Uno de los principales objetivos del portal es que esos menús y opciones laterales estén siempre a la vista, de modo que al navegar sólo cambie el contenido central (la información en cuestión).

Ese contenido central en la portada será de dos tipos:

1) Contenido Estático: El contenido central superior será un mensaje estático de bienvenida al portal de dicho sindicato.

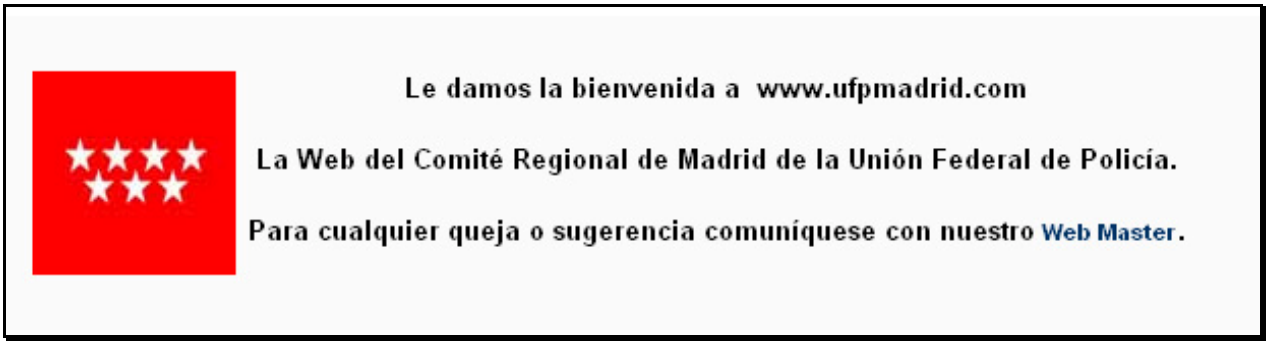


Figura 36: Portada - Contenido Estático

2) Contenido Dinámico: El contenido central inferior será un scroll o desplazamiento dinámico de texto e imágenes, el cual irá mostrando las últimas noticias agregadas al portal.

Figura 37: Portada - Contenido Dinámico

3.8.02 - DISEÑO DEL MENÚ PRINCIPAL

El menú principal, como ya dijimos, es la principal herramienta de navegación por el portal. Todos los elementos del menú conducen a la sección del portal que indican; pero algunos, además de eso, despliegan sobre si mismos un submenú con nuevos lugares del portal relacionados con el elemento principal seleccionado, como se muestra en la “Figura 39”.

Figura 38: Menú Principal

Figura 39: Menú Principal desplegado

Las secciones Galería de imágenes y Calendario no muestran contenidos (ni dinámicos ni estáticos), sino que nos enviarán a esas respectivas secciones, que, aunque están dentro del portal, son funcionalidades añadidas a la aplicación mediante sus correspondientes extensiones.

El resto de los elementos del menú y los submenús nos enviarán al distinto contenido estático y dinámico del portal.

En el diseño gráfico de los menús (plantilla CSS de estilos realizada para este proyecto) se ha elegido una gama de tonalidades azules para seguir con el estilo del portal y con el color corporativo del Cuerpo Nacional de Policía.

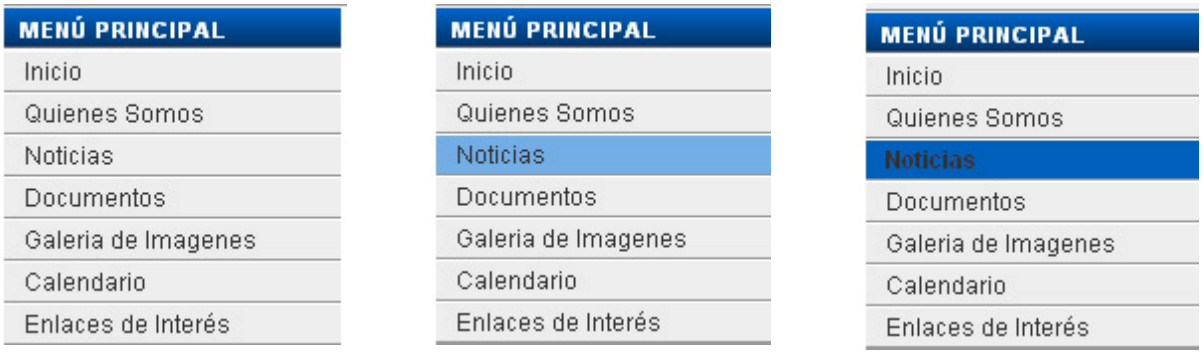
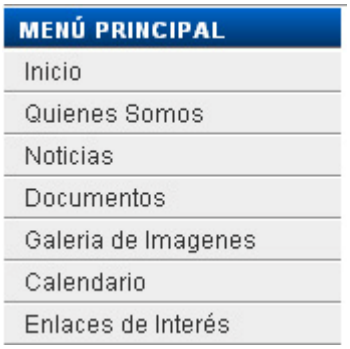
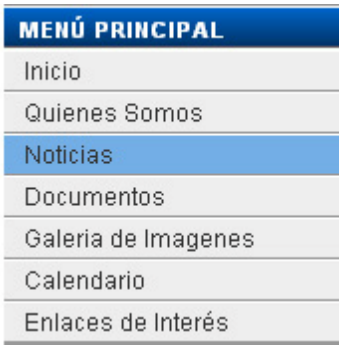


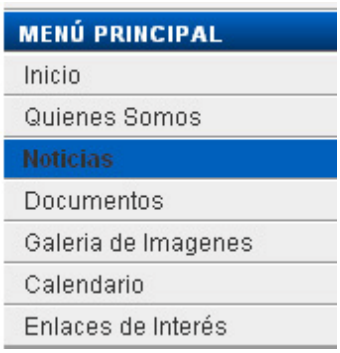
Figura 40: Menú Principal - Secuencia de selección de una opción



- Las opciones del menú tendrán un fondo claro como ya se comento anteriormente por motivos de claridad y descanso de la vista del usuario del portal, como se muestra en la primera representación gráfica de la "Figura 40".



- Al pasar el puntero del ratón sobre las opciones del menú, estas resaltaran sobre un fondo, esta vez azul claro, siguiendo con esta gama de colores suaves y corporativos de la policía, como se muestra en la segunda representación gráfica de la "Figura 40".



- Por último, al haber seleccionado la opción deseada, ésta se verá resaltada sobre fondo azul corporativo para que sea fácilmente identificable por el usuario saber en que zona del portal web se encuentra en dicho momento, como se muestra en la tercera representación gráfica de la "Figura 40".

3.8.03 - DISEÑO DEL MENÚ DEPENDENCIAS

El menú de dependencias sirve como herramienta rápida de navegación por un directorio de teléfonos, faxes y direcciones de las distintas dependencias del Cuerpo Nacional de Policía, en particular de dependencias de ámbito estatal y de la Comunidad de Madrid, ya que el cliente de este proyecto es el Comité Regional de Madrid del Sindicato policial U.F.P que se encarga de gestionar dicho territorio.

Figura 41: Menú Dependencias

Todos los elementos del menú conducen a la sección del portal que indican; pero algunos, además de eso, despliegan sobre si mismos un submenú con nuevos lugares del portal relacionados con el elemento principal seleccionado, como se muestra en la “Figura 42”.

Figura 42: Menú Dependencias desplegado

3.8.04 - DISEÑO DEL FORMULARIO DE ACCESO

El formulario de acceso de usuarios nos permite autenticamos ante el sistema por medio de nuestro Usuario y Clave, como muestra la “Figura 43”.

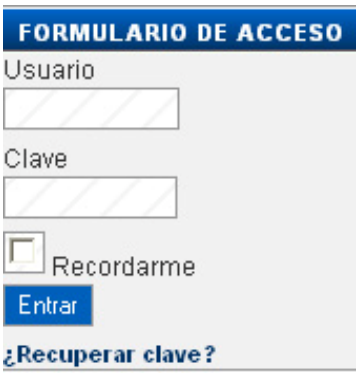


Figura 43: Formulario de Acceso

Una vez que nos hemos autenticado en el portal, se nos ofrecen nuevas opciones en el menú, disponibles sólo para los usuarios.

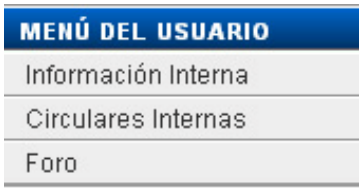
Figura 44: Pantalla Usuario

3.8.05 - DISEÑO DEL MENU DE USUARIO

A continuación se describirá el menú completo propio del usuario, en la figura 45 se muestra con más detalle como cambian el "Formulario de Acceso" y como aparece un nuevo menú, denominado "Menú de Usuario", tras la autenticación anterior.

Figura 45: Acceso concedido

- Una vez estemos autenticados, el formulario se convierte en el botón de salir del sistema, como muestra la "Figura 45".



- Una vez autenticados en el sistema, nos aparecerá el menú de usuarios, como muestra la "Figura 45", y podremos interactuar de forma mucho más global sobre el sistema, mediante el Foro del sindicato. Además podremos consultar contenidos de carácter privado.

3.8.06 - DISEÑO DE ENCUESTAS

El recuadro de las encuestas que se muestra en la pantalla principal del portal nos muestra siempre la última pregunta que se publicó como encuesta, todas las posibles alternativas de respuesta y finalmente las dos opciones de las que se dispone: votar y resultados.

Figura 46: Encuestas

Para votar, seleccionaremos la opción por la que nos decantemos (opción de marcaje) y pulsaremos sobre votar.

Para ver los resultados se pulsa directamente sobre resultados. Decir que una vez se ha votado siempre se muestran los resultados de la encuesta.

Figura 47: Resultados de encuestas

En los resultados de la encuesta, podemos ver principalmente las estadísticas de cuántos han votado por cada opción (se muestra el número de votos y el porcentaje).

En la parte inferior se aprecia cuándo se realizaron el primer y el último voto y en la superior hay un menú desplegable donde podemos ver los resultados de las anteriores encuestas planteadas.

3.8.07 - DISEÑO DEL CONTADOR DE VISITAS

El recuadro contador de visitas que se muestra en la pantalla principal del portal nos muestra siempre una estadística de las visitas que ha recibido el portal.

Figura 48: Contador de Visitas

Como se muestra en la figura anterior, se muestran datos de las visitas recibidas hoy, de las recibidas la última semana, de las recibidas el último mes y del total de visitas recibidas.

3.8.08 - DISEÑO DE LA GALERÍA DE IMÁGENES

Las galerías de imágenes son un contenido estático más. Son un contenido visual disponible para todo el mundo.

El contenido de la galería es gestionado por el administrador desde el panel de administración. Es una funcionalidad un tanto secundaria; pero que fue sugerida por el Sindicato de policía U.F.P. y, por tanto, ha sido diseñada e implementada.

Figura 49: Galería de Imágenes

El funcionamiento es sencillo, se muestran las distintas galerías existentes mostrando una imagen de las que contienen.

Figura 50: Navegador de Imágenes de la Galería

Al pinchar sobre la galería en cuestión, se nos mostrarán todas las imágenes que contiene, desde donde se podrán visionar dichas imágenes mediante un navegador contenido en dicha herramienta y se podrá hacer uso de una lupa para ver con más detalles las fotografías.

3.8.09 - DISEÑO DEL CALENDARIO

En el calendario podemos ver un desglose del mes con las citas que hay anotadas. La vista puede ser cambiada con los botones superiores a diaria, semanal, anual (en vez de la mensual que está definida por defecto).

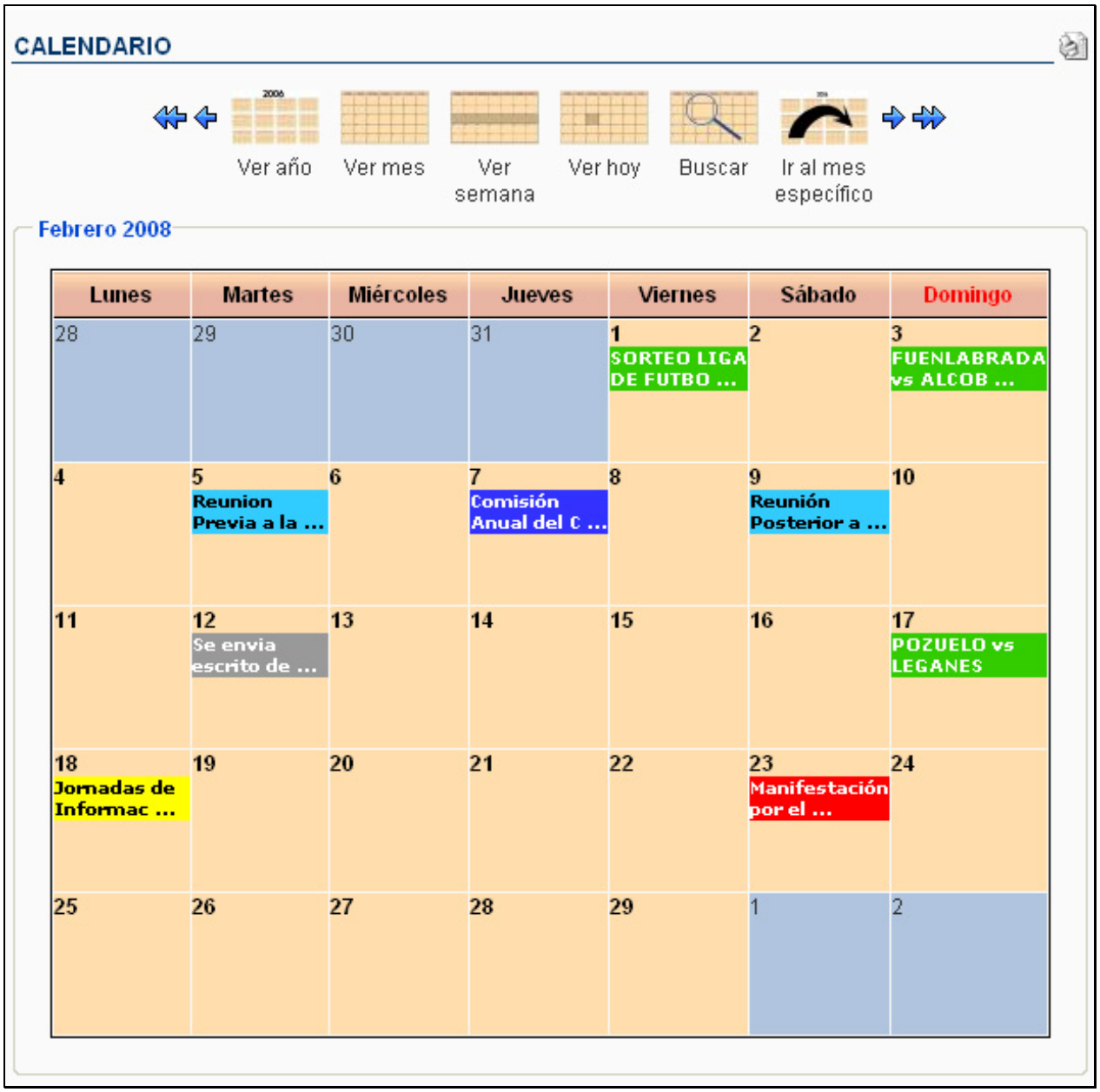


Figura 51: Calendario

Cuando pulsamos sobre una cita en cuestión, podemos ver información más detallada sobre ella.

Se nos muestra una descripción del acontecimiento, una fecha, una duración del evento, una localización y un contacto para la misma.

Finalmente se muestra el icono de impresión, para poder imprimir la cita automáticamente sin tener que copiar nada.

Figura 52: Evento del Calendario

3.8.10 - DISEÑO DEL FORO

Al Foro sólo podrán acceder usuarios registrados, una vez abierto desde el botón correspondiente del menú usuario se mostrará la imagen que muestra la figura 53. Dicho foro se divide en cuatro partes bien diferenciadas.

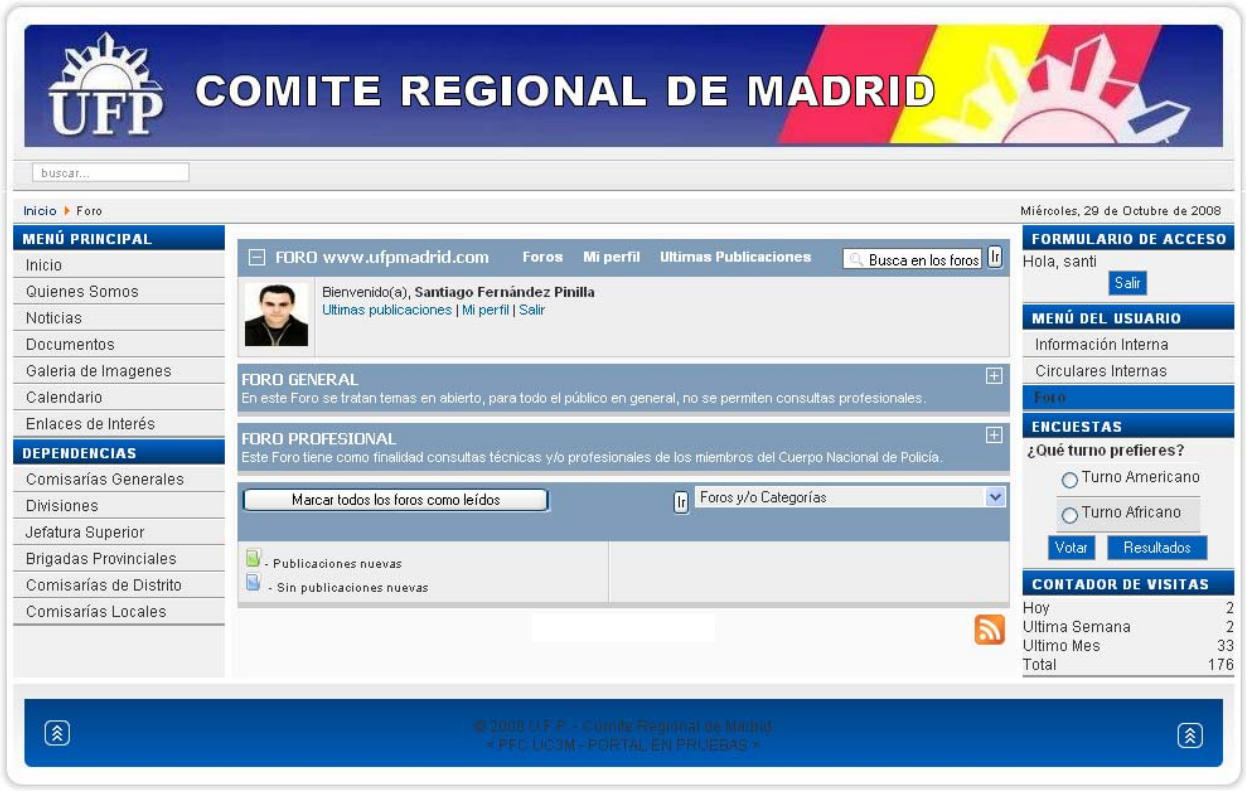


Figura 53: Foro

En la parte superior los datos personales y foto del usuario del foro, que coincidirá con el usuario del portal y un motor de búsqueda de contenidos en los foros.

A continuación los dos foros, el foro general y el foro profesional, con un listado de las temáticas que ofrece cada uno de ellos.

Por último una leyenda de los significados de los distintos iconos del foro y un campo desplegable, para navegar hasta el tema que se desee de manera directa.

Figura 54: Partes del Foro

El diseño (apariencia) se ha pretendido que sea en tonos azulados, manteniendo el aspecto y color corporativos del Cuerpo Nacional de Policía.

Se ha decidido (ya que el foro lo permite) que los foros estén marcados cuando hay mensajes o temas nuevos que el usuario no ha leído, de modo que el icono del foro pasa de ser de color azulado a ser de color verdoso.

3.9 - DISEÑO DEL PORTAL DE ADMINISTRACIÓN

El siguiente diagrama muestra la jerarquía de diseño planeada para generar nuestro Portal web desde el punto de vista del administrador o Back-End. En la esquina inferior derecha se referencia el punto de esta documentación en donde se amplia información acerca del "item" del diagrama.

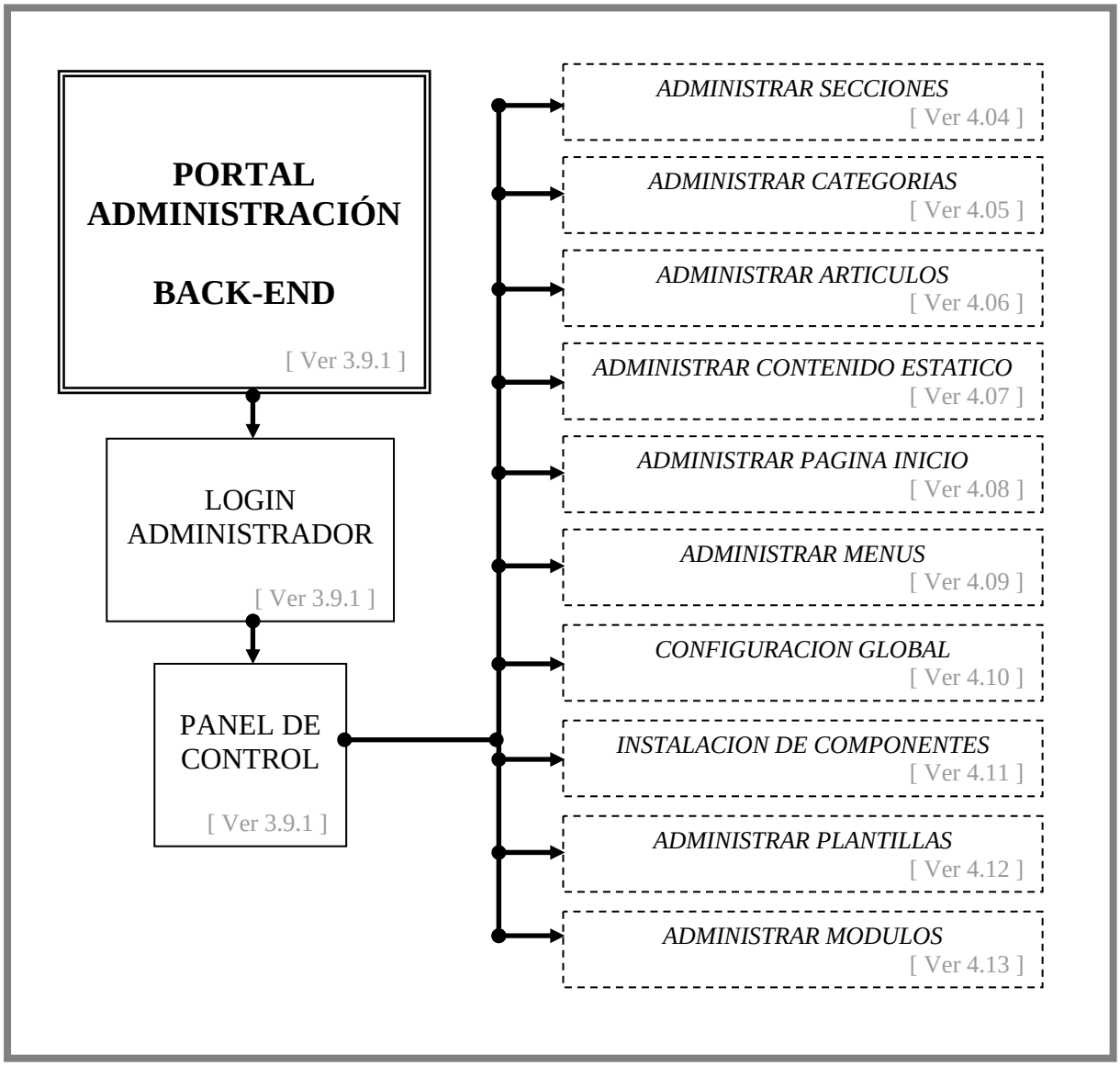


Figura 55: Diseño del Portal de Administración

A continuación, vamos a hacer una descripción gráfica (las imágenes han salido del portal ya implementado) del diseño del portal, de modo que se vaya viendo, paso a paso, cómo se ha decidido diseñar cada aspecto del portal acorde a los requisitos funcionales especificados.

3.9.1 - DISEÑO DEL BACK-END

Al panel de administración se accede desde un canal seguro cifrado por el protocolo SSL, mediante “Nombre de Usuario” y “Contraseña”, y sólo está disponible para aquellos usuarios que tengan el rango de administrador y que posean el certificado de seguridad de cliente adecuado, el cual estará instalado en una tarjeta inteligente o en el navegador del administrador.

Todo lo anterior proporciona un nivel de seguridad muy alto, ya que se unen tres requisitos de seguridad:

1) Conocer los parámetros: “Nombre de Usuario” y “Contraseña”.

2) Poseer una Tarjeta Inteligente con el certificado de seguridad de cliente adecuado, que junto con el certificado del servidor hacen posible establecer ese canal seguro SSL. Dicha tarjeta cumple dos funciones: nos proporciona confianza en la identidad del administrador y a su vez permite a este conectarse desde cualquier ordenador y así poder realizar el mantenimiento del portal desde cualquier punto.

3) El canal seguro mediante el protocolo de comunicaciones SSL.

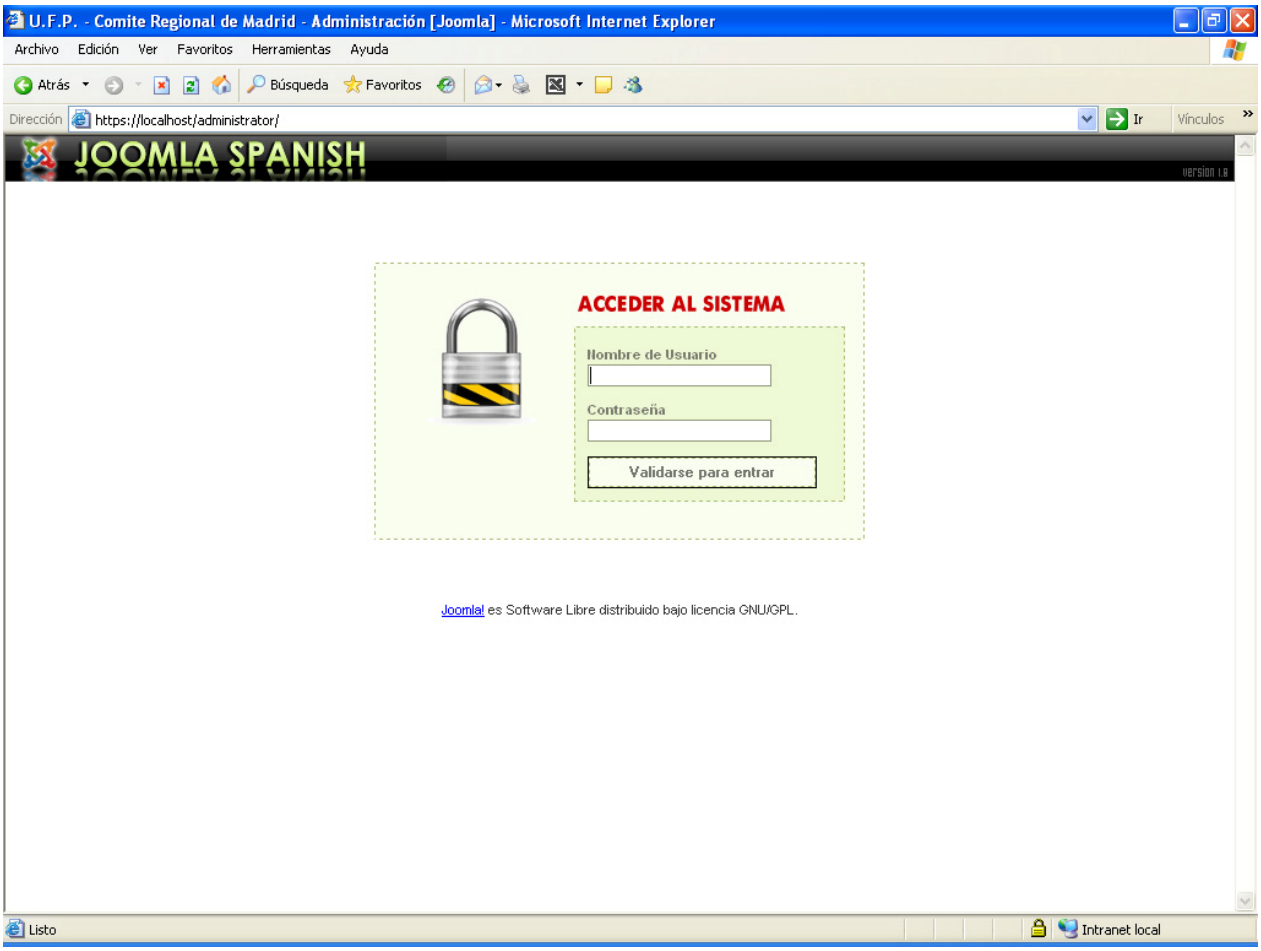


Figura 56: Login Administrador

Desde este panel se puede controlar absolutamente todo el portal tal y como se ve en los distintos iconos (acciones a realizar) que se muestra en la siguiente figura “Panel de Control”. Este apartado se comentará plenamente en la sección de implementación del portal, ya que aquí es donde se tratan todos los temas estructurales y de diseño del portal.

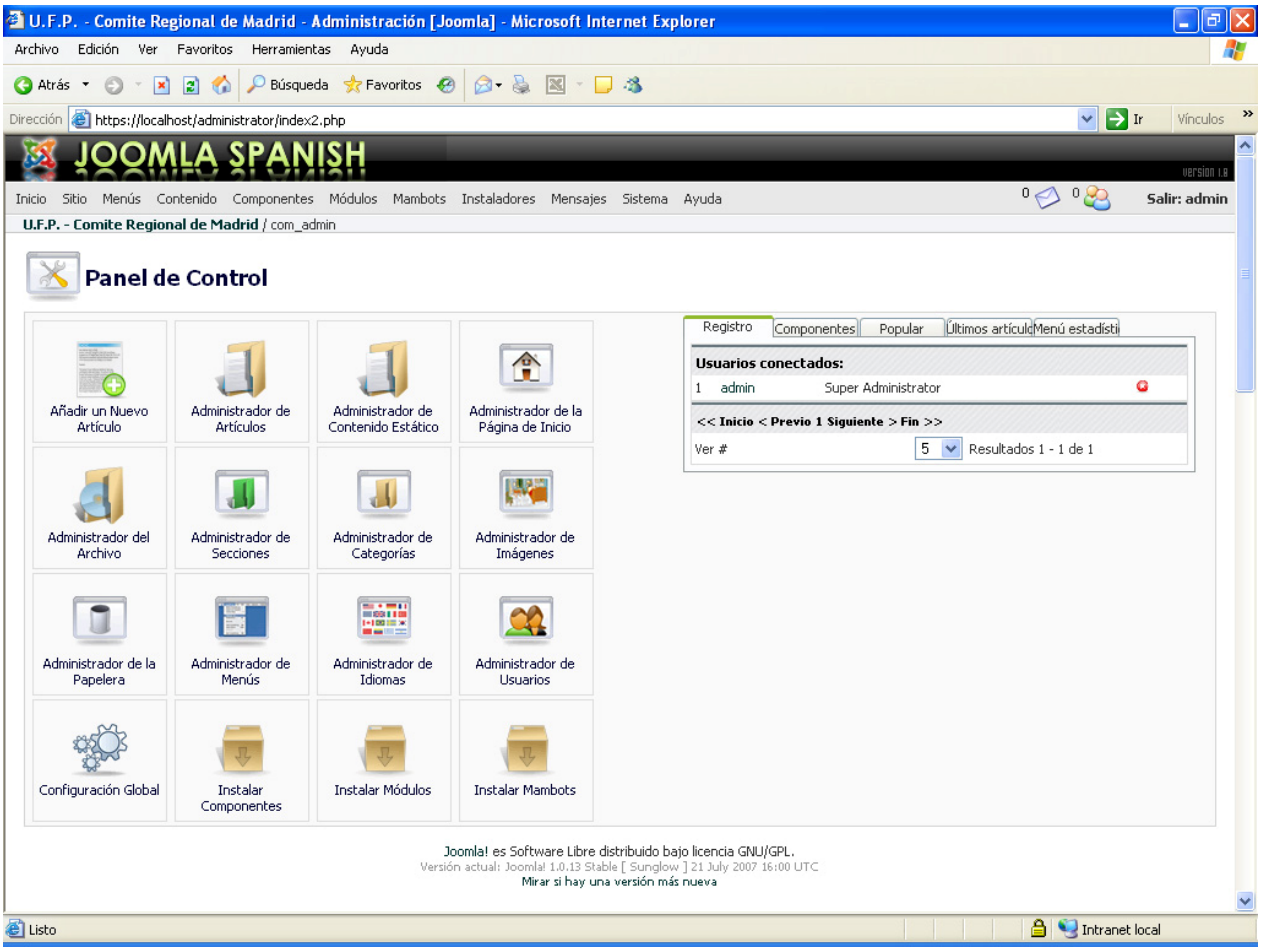


Figura 57: Panel de Control

Los principales controles que se nos ofrecen son: la configuración del sistema, la creación de categorías, de foros, de grupos de usuarios, la selección de la apariencia y la modificación de cualquier parámetro ya administrado o creado con anterioridad.

4 - IMPLEMENTACIÓN DEL PORTAL

En esta sección, vamos a ir describiendo, paso a paso, las acciones que se han ido realizando desde la instalación del portal hasta la configuración de nuevos componentes que añaden nuevas funcionalidades. Con estas indicaciones, que así se plantean, y siguiendo el manual de JOOMLA para aquellos detalles que no quedasen totalmente claros, se puede reconstruir el portal implementado desde cero.

4.01 - INSTALACIÓN DE JOOMLA

Antes de cualquier operación sobre el CMS JOOMLA debemos instalarlo, para una correcta puesta a punto del sistema deberemos consultar el "*APÉNDICE C*" denominado "*INSTALACIÓN JOOMLA 1.0.13*" y a partir de este punto comenzar a implementar nuestro portal.

4.02 ACCESO AL PANEL DE CONTROL DE JOOMLA

Para acceder al panel de administración de JOOMLA, desde el que generaremos todo el portal, debemos escribir en el navegador "http://localhost/administrator", a continuación, nos pedirá el nombre de usuario y la contraseña (dicho usuario debe tener permisos como administrador para poder entrar en el panel).

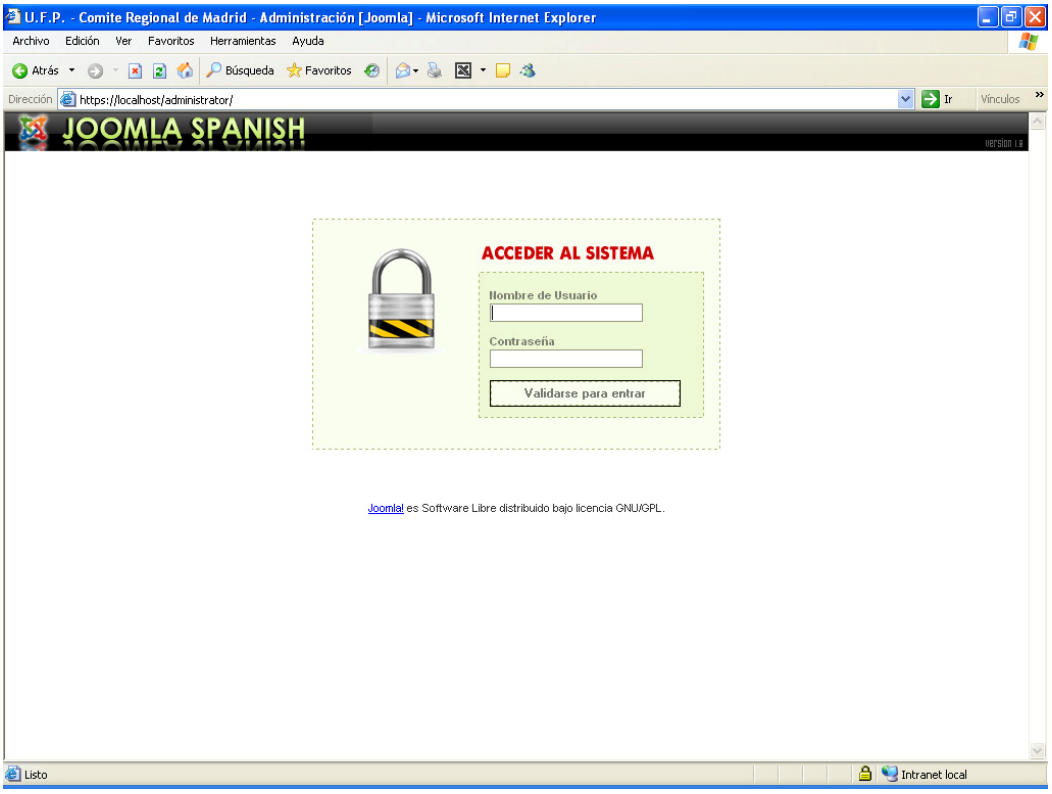


Figura 58: Login Administrador

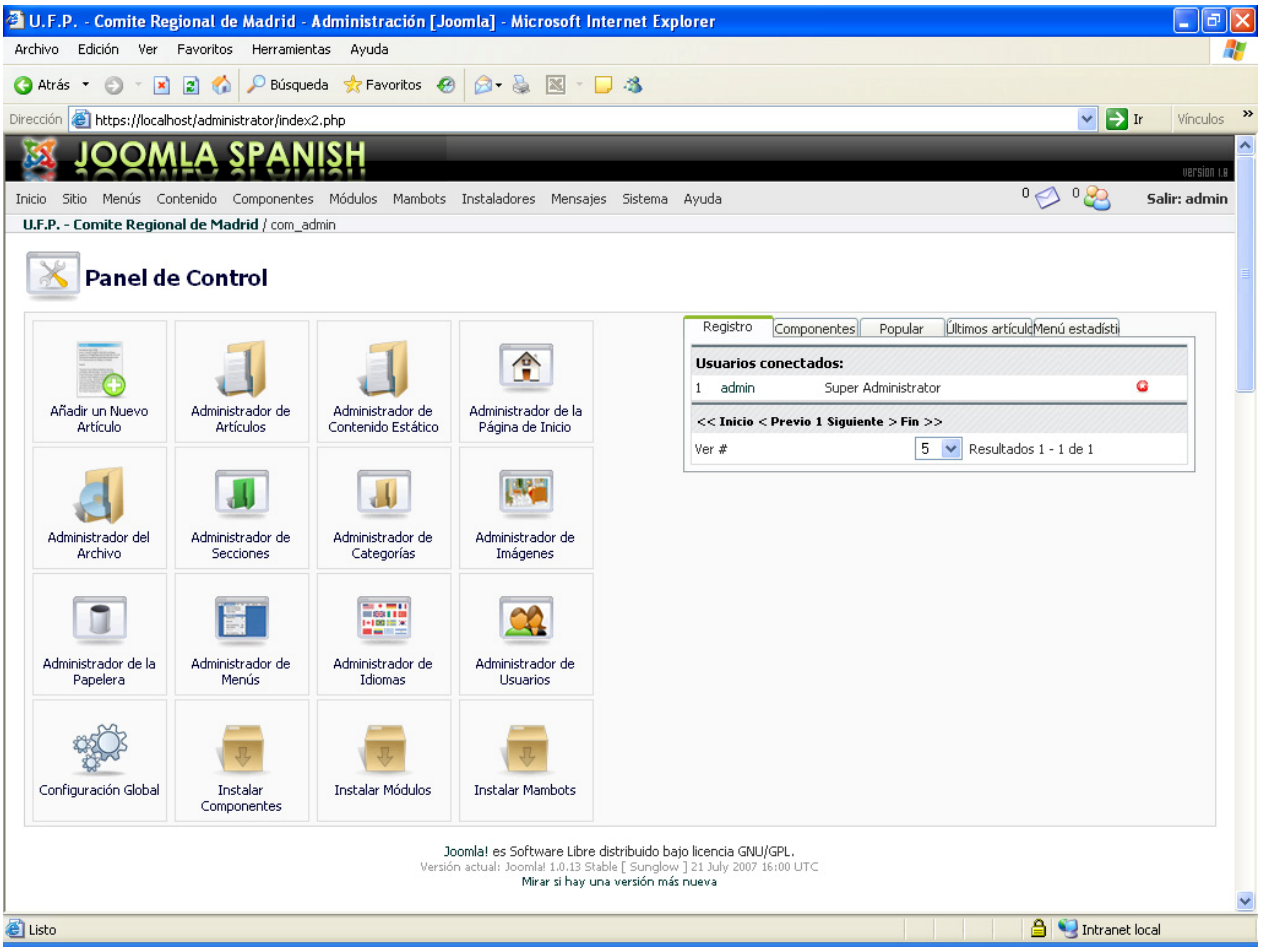


Figura 59: Panel de Control

En la parte superior, se nos muestra un menú con las diferentes opciones de configuración (se usará puntualmente), pero la inmensa mayoría de las opciones de confirmación están en los iconos centrales de configuración.

4.03 - ESTRUCTURA DEL PORTAL DESARROLLADO EN ESTE PROYECTO

4.03.1 - ESTRUCTURA DEL MENU PRINCIPAL

MENU PRINCIPAL	SECCIONES	CATEGORIAS	ARTÍCULOS
Inicio			*(Se mostraran las noticias más actuales)*
Quienes Somos			Quienes Somos
Noticias			*(Se mostraran todas las noticias)*
Documentos			
	Estatutos UFP		Texto de los Estatutos U.F.P.
	Legislación		
		Constitución Española	
			Texto de la Constitución Española.
		Constitución Europea	
			Texto de la Constitución Europea.
			Extracto de la Constitución Europea.
		Legislación sobre Riesgos Laborales	
			Ley 31/1995, de Prevención de Riesgos Laborales.
			Real Decreto 39/1997, Reglamento de Servicios de Prevención.
			Real Decreto 1488/1998, sobre adaptación de las L.P.R.L. a la A.G.E.
			Real Decreto 2/2006, sobre L.P.R.L. en la actividad policial.
			Plan de Prevención de Riesgos Laborales de la Dirección General de la Policía.
		Legislación Policial	
			Novedades de la Instrucción 12.2007 sobre tratamientos a detenidos.
			Modelo y ejemplo de ayuda pública de delito violento.
			Reconocimiento de ayudas a las víctimas de los delitos violentos y contra la libertad sexual.
			Modificaciones del permiso de paternidad en el artículo 30.1 de la Ley de la Función Pública establecidas por la LO 2/2007, de 22 de Marzo de igualdad de mujeres y hombres.
			Estatuto básico del Empleado Público (Ley 7/2007 de 12 de Abril).
			Impreso solicitud de Habeas Corpus en Dependencias Policiales.
			Tratamiento a Detenidos.
	Incrementos Salariales		
			Real Decreto 5/2007 de 12 de Enero por el que se modifican determinados complementos retributivos de las Fuerzas y Cuerpos de Seguridad del Estado.
			Mejoras retributivas de los Funcionarios del Cuerpo Nacional de Policía.
			Orden General Extraordinaria 16.01.2007 sobre retribuciones.
			Circular informativa sobre las Retribuciones del año 2007.
			Negociación salarial Año 2007.
			Complementos de destino según nivel.
			Cuadro de Retribuciones para el año 2007.

MENU PRINCIPAL	SECCIONES	CATEGORIAS	ARTÍCULOS
			Cálculo de retribuciones 2ª Actividad para la categoría de Comisario.
			Cálculo de retribuciones 2ª Actividad para la categoría de Inspector - Jefe.
			Cálculo de retribuciones 2ª Actividad para la categoría de Inspector.
			Cálculo de retribuciones 2ª Actividad para la categoría de Subinspector.
			Cálculo de retribuciones 2ª Actividad para la categoría de Oficial de Policía.
			Cálculo de retribuciones 2ª Actividad para la categoría de Policía.
			Acuerdos económicos entre el Ministerio del Interior y Sindicatos del C.N.P.
			Escrito dirigido al Director General sobre Acuerdos Ministerio del Interior y Sindicatos del C.N.P. de fecha 06.11.2007.
			Acta síntesis reunión Ministerial - Sindicatos CNP 28.12.2006.
	Riesgos Laborales		
			Unidad 01.- Sistema de Gestión de Prevención de Riesgos Laborales.
			Unidad 02.- Conceptos Básicos de Prevención de Riesgos Laborales.
			Unidad 03.- Prevención de Riesgos Laborales en el Cuerpo Nacional de Policía.
			Unidad 04.- Sistemas Elementales de Control.
			Como proceder en primeros Auxilios.
			Conceptos sobre enfermedad común y accidente profesional.
			Formulario de Prevención de Riesgos Laborales.
			Escrito de Prevención de Riesgos Laborales.
			Estudio del Plomo como factor de riesgo laboral.
			La cultura de la Prevención avanza muy lentamente en el CNP.
			Informe sobre trabajo nocturno.
Galeria de Imagenes			*(EXTENSION INSTALADA EN JOOMLA)*
Calendario			*(EXTENSION INSTALADA EN JOOMLA)*
Enlaces de Interes			Enlaces de Interés

4.03.2 - ESTRUCTURA DEL MENU DEPENDENCIAS

MENU DEPENDENCIAS	SECCIONES	SECCIONES	CATEGORIAS	ARTICULOS
Dependencias				
	Jefatura Superior			
			Contacto	*(Dirección – Teléfono – Fax)*
	Brigadas Provinciales			
		B.P.S.C.		
			Contacto	*(Dirección – Teléfono – Fax)*
		B.P.P.J.		
			Contacto	*(Dirección – Teléfono – Fax)*
		B.P.P.C.		
			Contacto	*(Dirección – Teléfono – Fax)*
		B.P.I.		
			Contacto	*(Dirección – Teléfono – Fax)*
		B.P.Ex.D.		
			Contacto	*(Dirección – Teléfono – Fax)*
	Comisarias de Distrito			
		Arganzuela		
			Contacto	*(Dirección – Teléfono – Fax)*
		Barajas		
			Contacto	*(Dirección – Teléfono – Fax)*
		Carabanchel		
			Contacto	*(Dirección – Teléfono – Fax)*
		Centro		
			Contacto	*(Dirección – Teléfono – Fax)*
		Chamartín		
			Contacto	*(Dirección – Teléfono – Fax)*
		Chamberí		
			Contacto	*(Dirección – Teléfono – Fax)*
		Ciudad Lineal		
			Contacto	*(Dirección – Teléfono – Fax)*
		Fuencarral-El Pardo		
			Contacto	*(Dirección – Teléfono – Fax)*
		Hortaleza-Barajas		
			Contacto	*(Dirección – Teléfono – Fax)*
		Latina		
			Contacto	*(Dirección – Teléfono – Fax)*

MENU DEPENDENCIAS	SECCIONES	SECCIONES	CATEGORIAS	ARTICULOS
		Moncloa-Aravaca		
			Contacto	*(Dirección – Teléfono – Fax)*
		Moratalaz		
			Contacto	*(Dirección – Teléfono – Fax)*
		Puente de Vallecas		
			Contacto	*(Dirección – Teléfono – Fax)*
		Retiro		
			Contacto	*(Dirección – Teléfono – Fax)*
		Salamanca		
			Contacto	*(Dirección – Teléfono – Fax)*
		San Blas		
			Contacto	*(Dirección – Teléfono – Fax)*
		Tetuán		
			Contacto	*(Dirección – Teléfono – Fax)*
		Usera		
			Contacto	*(Dirección – Teléfono – Fax)*
		Villa de Vallecas		
			Contacto	*(Dirección – Teléfono – Fax)*
	Comisarias Locales			
		Alcalá de Henares		
			Contacto	*(Dirección – Teléfono – Fax)*
		Alcobendas-S.S. Reyes		
			Contacto	*(Dirección – Teléfono – Fax)*
		Alcorcón		
			Contacto	*(Dirección – Teléfono – Fax)*
		Aranjuez		
			Contacto	*(Dirección – Teléfono – Fax)*
		Coslada-San Fernando		
			Contacto	*(Dirección – Teléfono – Fax)*
		Fuenlabrada		
			Contacto	*(Dirección – Teléfono – Fax)*
		Getafe		
			Contacto	*(Dirección – Teléfono – Fax)*
		Leganés		
			Contacto	*(Dirección – Teléfono – Fax)*
		Móstoles		
			Contacto	*(Dirección – Teléfono – Fax)*

MENU DEPENDENCIAS	SECCIONES	SECCIONES	CATEGORIAS	ARTICULOS
		Parla		
			Contacto	*(Dirección – Teléfono – Fax)*
		Pozuelo de Alarcón		
			Contacto	*(Dirección – Teléfono – Fax)*
		Torrejón de Ardoz		
			Contacto	*(Dirección – Teléfono – Fax)*
	Comisarias Generales			
		C.G.S.C.		
			Contacto	*(Dirección – Teléfono – Fax)*
		C.G.P.J.		
			Contacto	*(Dirección – Teléfono – Fax)*
		C.G.P.C.		
			Contacto	*(Dirección – Teléfono – Fax)*
		C.G.I.		
			Contacto	*(Dirección – Teléfono – Fax)*
		C.G.Ex.D		
			Contacto	*(Dirección – Teléfono – Fax)*
	Divisiones			
		División de Personal		
			Contacto	*(Dirección – Teléfono – Fax)*
		División de Formación		
			Contacto	*(Dirección – Teléfono – Fax)*
		División Coord. E. y T.		
			Contacto	*(Dirección – Teléfono – Fax)*

4.03.3 - ESTRUCTURA DEL MENU DEL USUARIO

MENU USUARIO	SECCIONES	CATEGORIAS	ARTÍCULOS
Información Interna	Información Interna		*(Se mostraran las noticias privadas o profesionales del Cuerpo Nacional de Policía, de manera restringida, sólo para usuarios registrados).*
Circulares Internas	Circulares Internas		*(Se mostraran las circulares o B.O.E. privados o profesionales del Cuerpo Nacional de Policía, de manera restringida, sólo para usuarios registrados).*
Foro			*(EXTENSION INSTALADA EN JOOMLA)*

4.04 - SECCIONES

Figura 60: Administrador de Secciones

Para generar las secciones del portal (las que ya definimos en el diseño y en el apartado anterior “4.3”), debemos ir a Administrador de secciones y pulsar en nueva.

De las opciones que se ven con las tres secciones creadas es importante destacar el orden, ya que, aunque aquí no es importante, generalmente, describirá la precedencia de los contenidos que se muestren.

En la columna de publicado, podemos elegir entre dejarla publicada (accesible) o deshabilitarla para que nadie pueda acceder a ella.

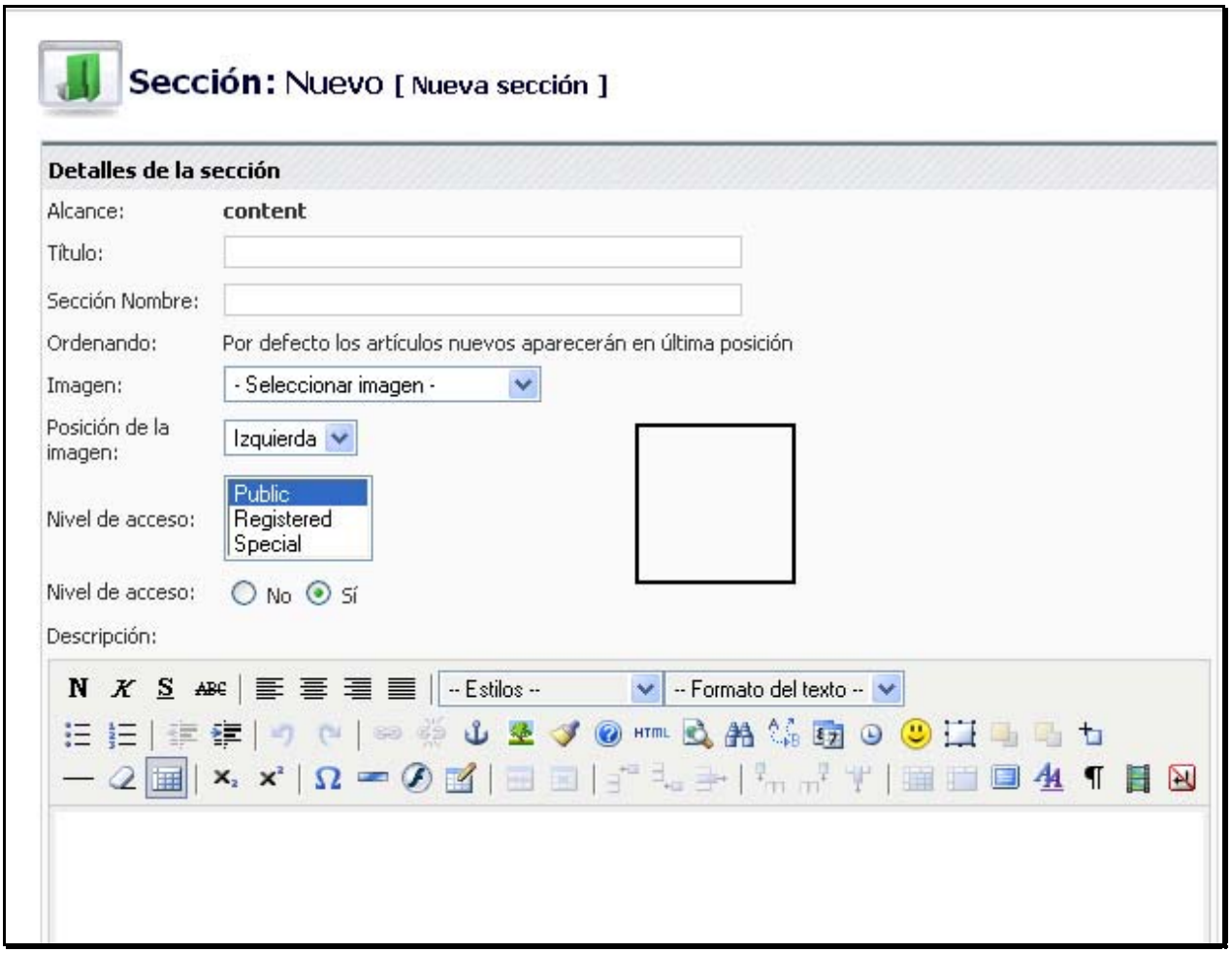


Figura 61: Crear Sección

Para crear una sección, definimos su título, el nombre de la sección y adjuntar, si queremos, un HTML con el editor que servirá de resumen de la sección.

Es importante destacar el Nivel de acceso, que se repetirá mucho durante la generación del portal: "*Public*" es que todo el mundo podrá ver esa sección, "*Registered*" que sólo usuarios registrados podrán ver la sección y, finalmente, "*Special*" que define que sólo usuarios con capacidad de edición (publicadores y administradores) tendrán acceso a dicha sección.

4.05 - CATEGORÍAS

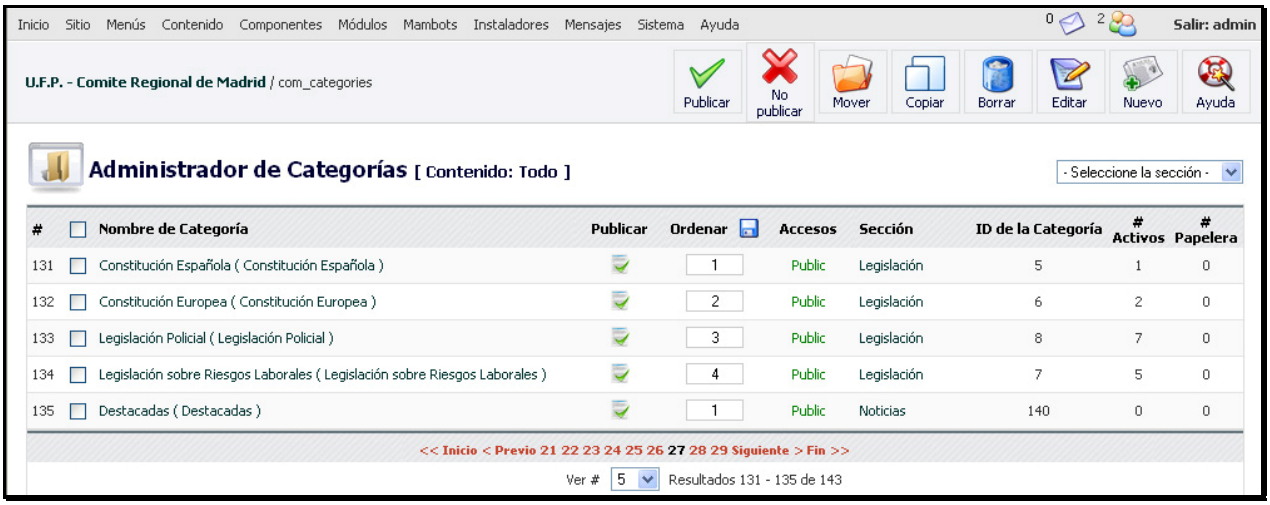


Figura 62: Administrador de Categorías

Para generar categorías, vamos al icono de Administrador de categorías desde donde podremos crear y administrar las categorías, al igual que las secciones.

Las categorías son divisiones de las secciones, de modo que toda categoría que creemos deberá colgar de una sección ya definida (tal y como se definió en el diseño del portal).

Pulsaremos en Nueva para crear una categoría.

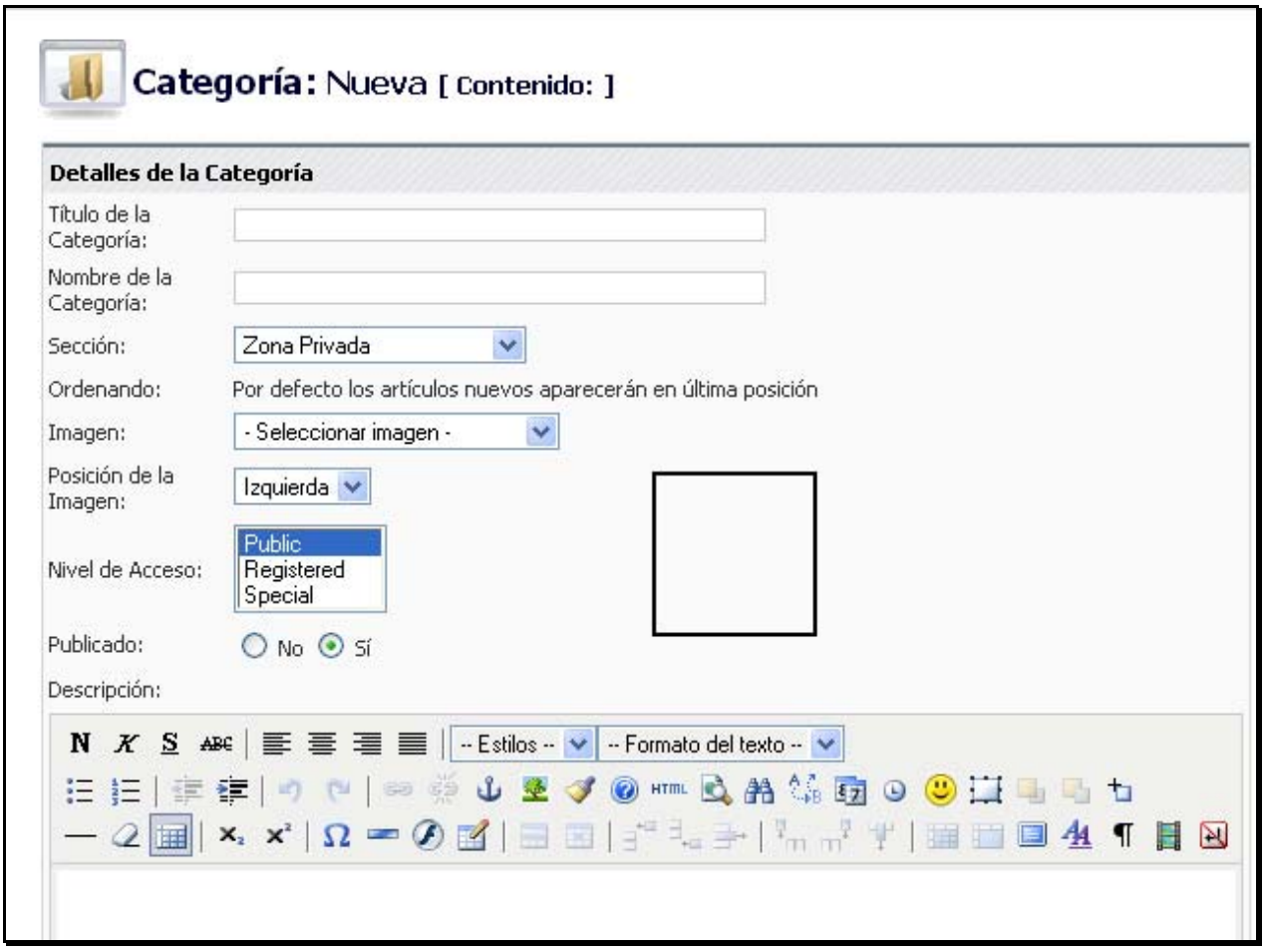


Figura 63: Crear Categoría

La creación de una categoría es idéntica en todos los aspectos a lo ya mencionado en las secciones, sólo que aquí hay que expresar a qué sección pertenece la categoría que estamos creando.

Como las categorías son divisiones de las secciones, nos servirán para organizar el contenido del portal (tanto dinámico como estático), ya que todo contenido debe estar enmarcado en su correspondiente categoría (y por extensión en una sección).

No se volverá a repetir lo ya dicho en el diseño sorbe las categorías. Aquí se hará una implementación de esa especificación ya analizada.

4.06 - ARTÍCULOS

Figura 64: Administrador de Artículos

A través de Administrar Artículos (también desde Añadir un nuevo artículo), podemos generar contenido dinámico para el portal.

Es importante destacar que en esta sección es donde están presentes todos los contenidos dinámicos del portal. La referencia de una nueva columna en la “Página de inicio”, permite que el artículo en cuestión aparezca en la página de inicio del portal (esto tiene su propia sección de configuración).

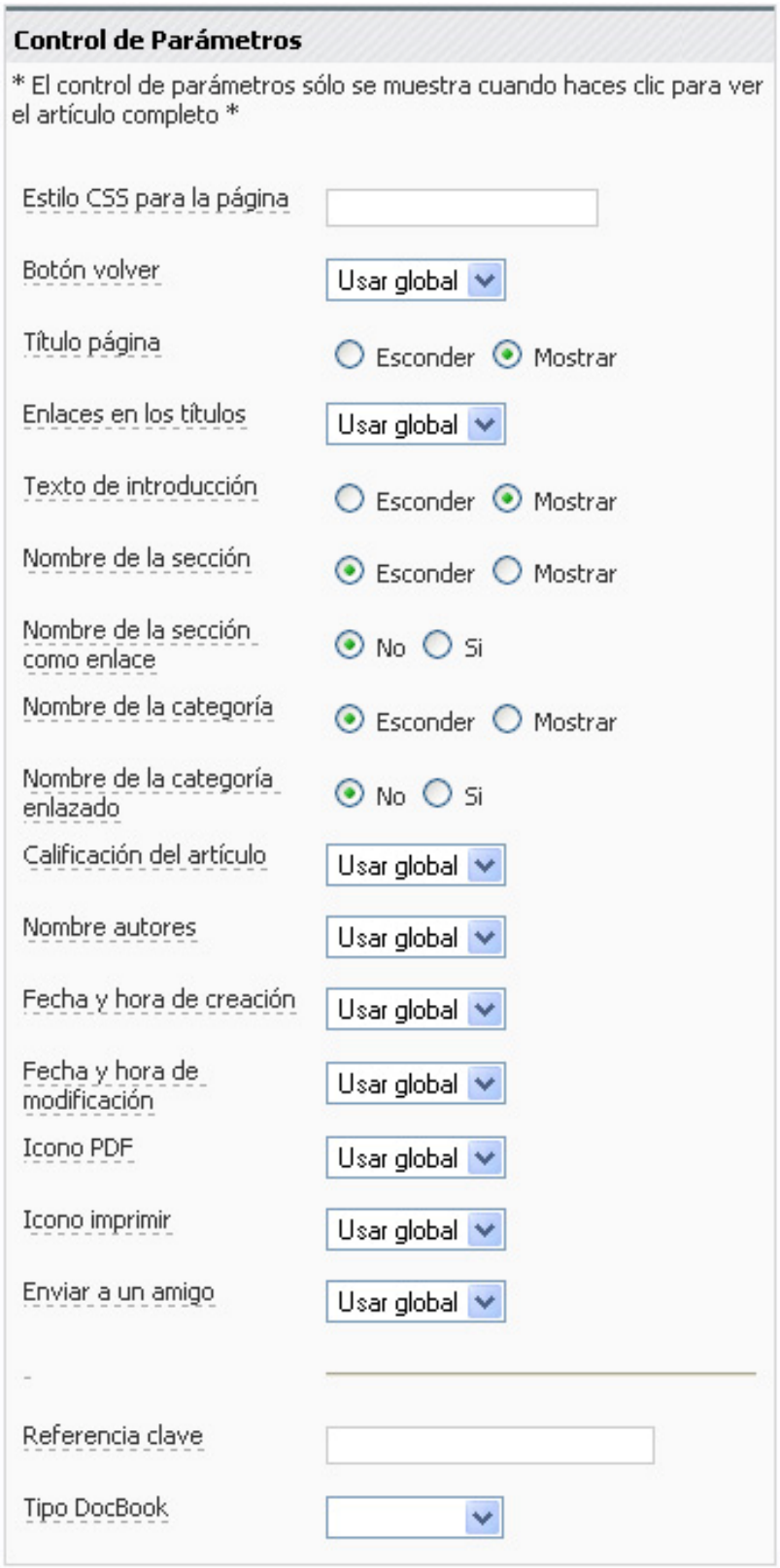


Figura 65: Parámetros del Contenido

En la pestaña de Parámetros del contenido (que está una vez hemos accedido a crear un nuevo artículo) tenemos un número importante de opciones que describirán cómo se verá el artículo en sí una vez publicado.

- ✓ Título página: Sirve para ver el título del artículo en la página.
- ✓ Nombre de la sección: sirve para mostrar o no la sección a la que pertenece este artículo.
- ✓ Nombre de la categoría enlazado: sirve para que el nombre de la categoría sea un enlace a la propia categoría en sí.

El resto de opciones pueden establecerse con varios parámetros: Usar global significa que se aplicarán las configuraciones globales (se verán más adelante) a este contenido, Esconder significa que no se mostrará la opción en cuestión y Mostrar que sí se mostrará. Por tanto, las dejaremos todas como globales y será en esa configuración donde determinemos su valor (así conseguimos ahorrar tiempo a la hora de implementar los contenidos).

De este modo, añadimos los artículos que deseemos al portal, aunque la forma más habitual es que los editores y publicadores sean quienes se encarguen de añadir este contenido dinámico, pero eso no se trata en la sección de implementación del portal.

En la pestaña Publicación se puede expresar directamente que rango de visibilidad tendrá el artículo (del mismo modo que se describió en categorías).

4.07 - CONTENIDO ESTATICO

El contenido estático se genera exactamente igual que el dinámico, sólo que el estático solamente lo puede crear el administrador. Está orientado para generar contenidos inmutables en el tiempo. En el caso del Sindicato Policial U.F.P., para describir todo su funcionamiento interno y aspectos similares.

Se va a generar todo el contenido estático que se mencionó en el apartado de diseño del menú con carácter público para que todo el mundo tenga acceso a él y los parámetros del contenido, los definiremos manualmente. Esto se debe a que en la configuración global no se hace diferenciación entre tipos de contenidos y por tanto hacemos una configuración para los contenidos dinámicos, ya que queremos evitar que se muestren datos como la hora de la creación. Puesto que el contenido estático lo genera el administrador y es poco en comparación con el dinámico, seremos nosotros manualmente quienes definamos esos parámetros, en vez de usar configuraciones manuales para los dinámicos y globales para los estáticos.

Figura 66: Parámetros de Contenido Estático

4.08 - PAGINA DE INICIO

Figura 67: Parámetros de Contenido Estático

En el apartado Administrador de la Página de Inicio, podemos administrar el contenido dinámico que previamente se añadió en su sección a la página de inicio.

Nuevamente debemos seleccionar el orden en el que aparecen los artículos presentes y esta vez es importantísimo, ya que ahora, en la portada, se nos mostrarán los artículos con el orden que aquí quede establecido. También se puede dejar el contenido sin publicar temporalmente, por lo que no aparecerá en la página de inicio.

4.09 - MENUS

Figura 68: Menús generales

Hay cuatro menús por defecto, anquen se pueden añadir más si se desea, los cuatro menús existentes son:

- ✓ mainmenu: sirve para generar el menú de navegación del portal Es el más importante de todos.
- ✓ othermenu: este tipo de menú no se ha usado en la implementación.
- ✓ topmenu: sirve para definir el menú que se mostrará en la parte superior del portal.
- ✓ usermenu: este menú es un menú adicional (al manimenu) que sólo se mostrará a aquellos navegantes que estén autenticados en el sistema.

En nuestra implementación hemos desarrollado los siguientes menus:

- ✓ MENU PRINCIPAL → Creado a partir del "*mainmenu*".
- ✓ MENU USUARIO → Creado a partir del "*usermenu*".
- ✓ MENU DEPENDENCIAS → Creado de cero, para este proyecto.

Para generar el contenido de los menús, pulsaremos sobe el botón de Artículos del menú. Primero, vamos a crear el mainmenu (Menú principal).

Figura 69: Mainmenu

En el menú principal, vemos que se pueden crear menús y varios niveles de submenús (ahora veremos cómo se consigue este resultado).

En esta pantalla (una vez creados los elementos) se eligen básicamente los elementos que son públicos (aquellos que son visibles) y algo muy importante: el orden en que aparecerán los elementos en el menú.

El orden va más allá de ser una mera cuestión de orden, sobre todo en lo que se refiera al primer puesto del menú, ya que JOOMLA muestra como página de inicio lo que referenciamos con la primera posición en el mainmenu.

Por defecto, este menú tiene creado el acceso a Inicio y ése ha de ocupar siempre la primera posición.

Para crear nuevos elementos del menú, damos en el botón Nuevo.

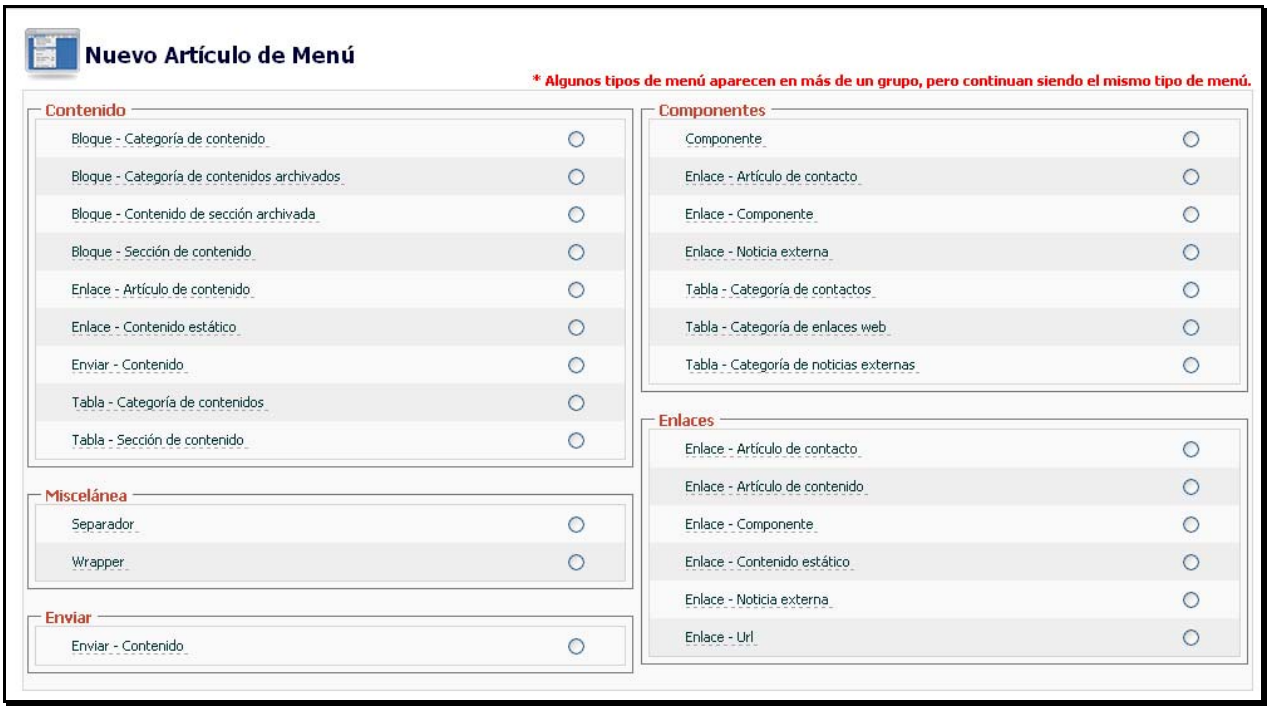


Figura 70: Tipo de Menú

Lo primero que debemos elegir es el tipo de contenido que referenciará nuestro enlace: si se enlaza con una sección, con una categoría, con una URL, con un contenido estático o con un componente en cuestión.

Por tanto, creamos los enlaces principales del menú apuntando al tipo de componente exacto al que referenciarán (esto ya se definió en el diseño del portal).

No definiremos cómo se han creado todos los enlaces, porque son muchos; pero, a modo de ejemplo, trataremos cómo crear un enlace a una categoría de contenidos (incluido el caso de tratarse de un submenú).

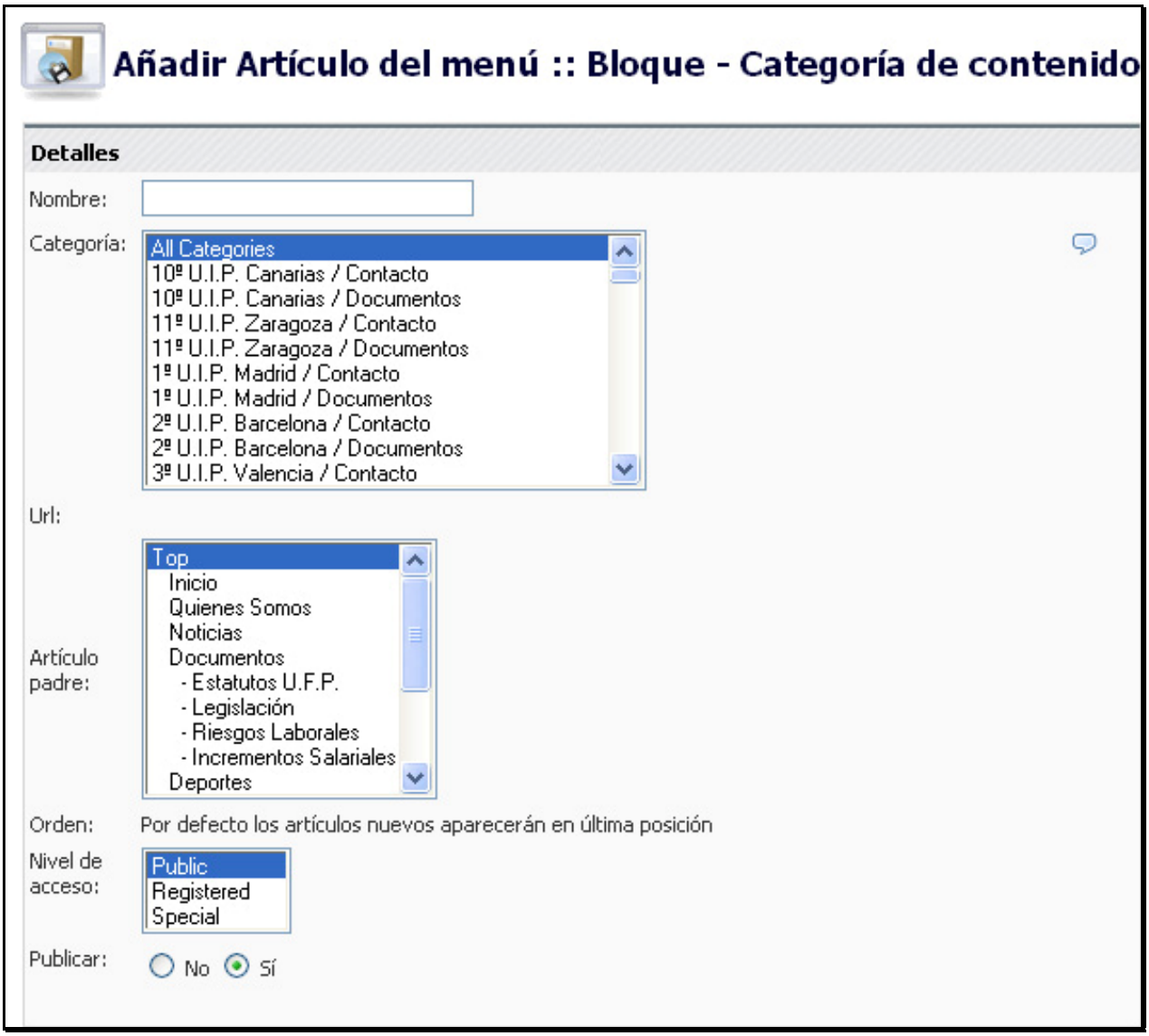


Figura 71: Crear elemento de menú

Primero, seleccionamos el nombre del elemento del menú. A continuación, la categoría en cuestión que referenciará (esto variaría de tratarse de un enlace con otro tipo de contenido). Luego, se define el artículo padre. Esto es muy importante, porque sirve para definir los submenús.

Si escogemos Top el elemento será principal y aparecerá en el menú, si escogemos otro elemento de la lista, entonces el elemento que creamos será un elemento hijo del seleccionado y se mostrará desplegando el menú.

Al final, seleccionamos la visibilidad del elemento (como ya se ha definido en otras secciones).

Seguiremos este proceso para dar cobertura a todo el contenido dado por el Sindicato Policial U.F.P. para hacer que el menú nos sirva para acceder a todo el portal.

El funcionamiento del menú de usuario es idéntico, sólo que el contenido se mostrará sólo cuando el visitante esté autenticado y el del menú dependencias es, al igual que el menú principal, de carácter público.

En algunos menús, en el panel de la derecha (no hay capturas, pero es el mismo que el de los artículos) aparece una opción interesante, que es el número de columnas que queremos para ese menú. Esto significa cuántas columnas saldrán para mostrar la información (o cuántos elementos aparecerán por fila). Nosotros escogeremos siempre 1 para que el contenido aparezca uno encima del otro.

4.10 - CONFIGURACIÓN GLOBAL

En este apartado, es donde definimos los aspectos de configuración más destacables del nuestro portal.

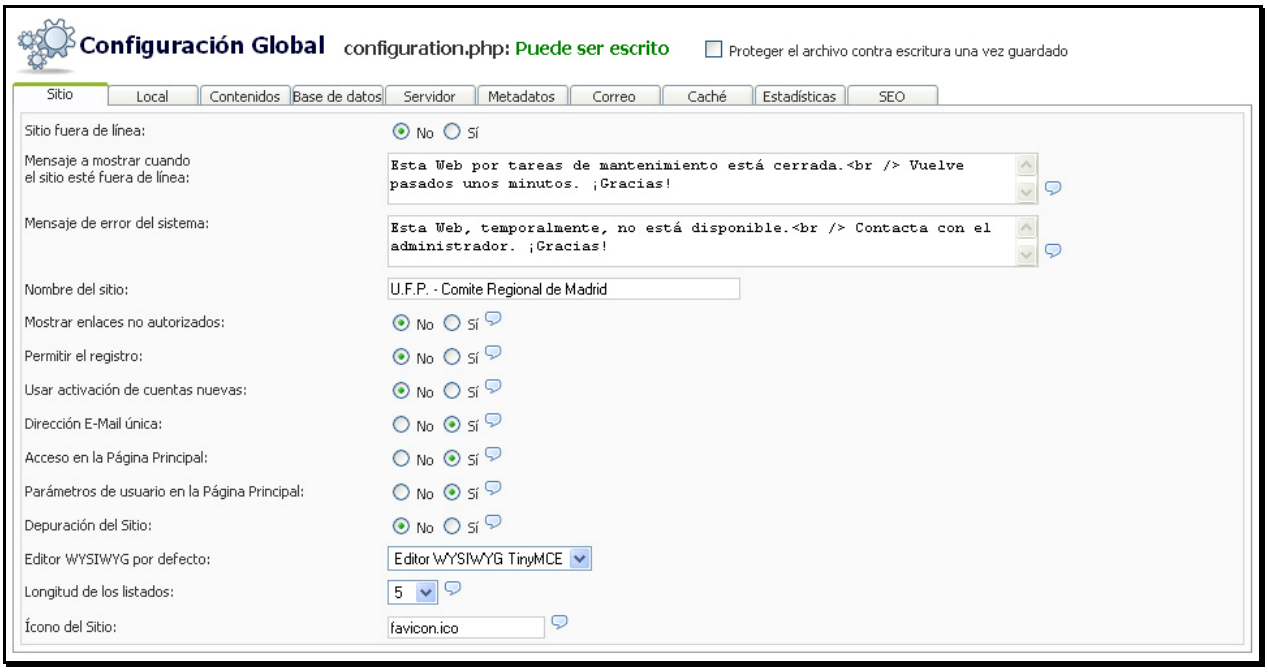


Figura 72: Configuración del sitio

Aquí lo más importante a destacar en la opción registro es que diremos que las direcciones de e-mail han de ser únicas, para que, cuando tengamos que comunicarnos con los usuarios, no haya repeticiones de correos.

Marcamos acceso a la página principal, para que sea está la que se use como página de inicio.

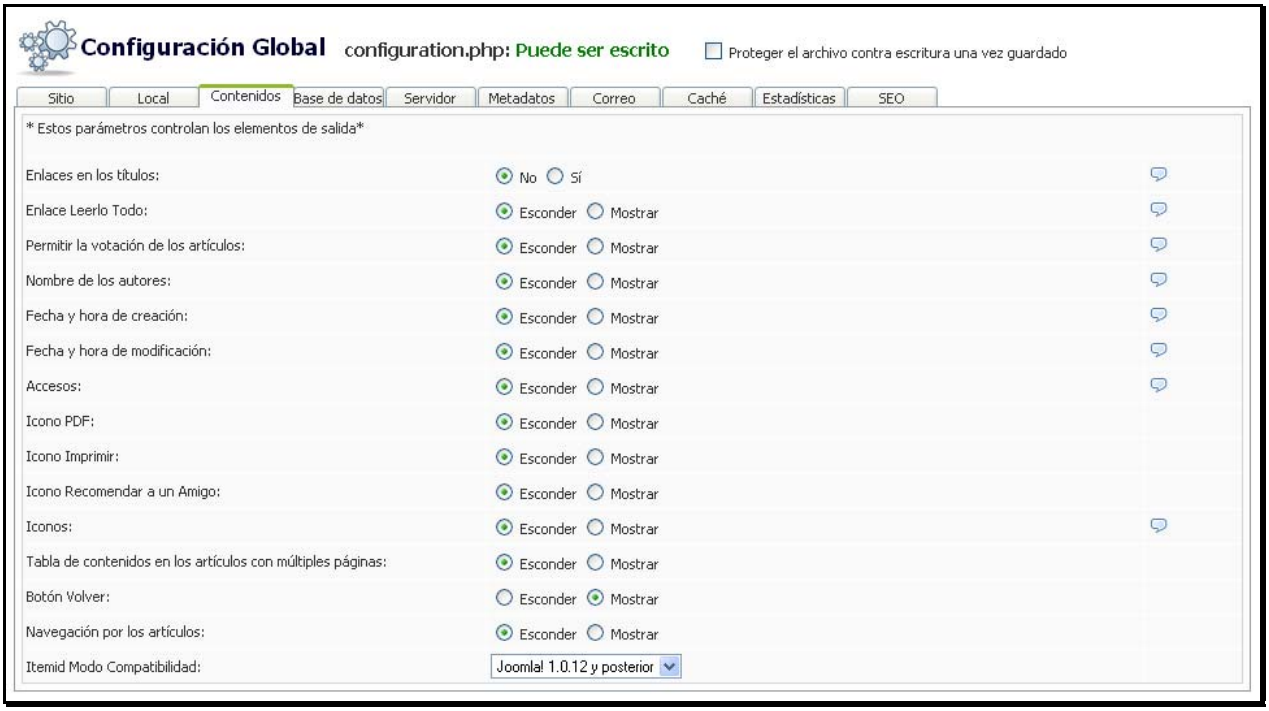


Figura 73: Configuración de contenidos

En la configuración de contenidos, es donde seleccionamos los parámetros por defecto que se aplicarán al contenido (como ya mencionamos antes, nosotros lo usaremos como propiedades del contenido dinámico). Las distintas opciones son muy básicas y no merecen mayor comentario, así que las definiremos tal cual se aprecien en la imagen Configuración de contenidos.

En la siguiente pestaña de bases de datos, no deberemos tocar nada, pues ahí están los datos introducidos durante la instalación que son los únicos válidos.

El resto de pestañas pueden quedarse con sus valores por defecto, ya que no son importantes para la implementación de nuestro portal.

4.11 - INSTALACIÓN DE COMPONENTES

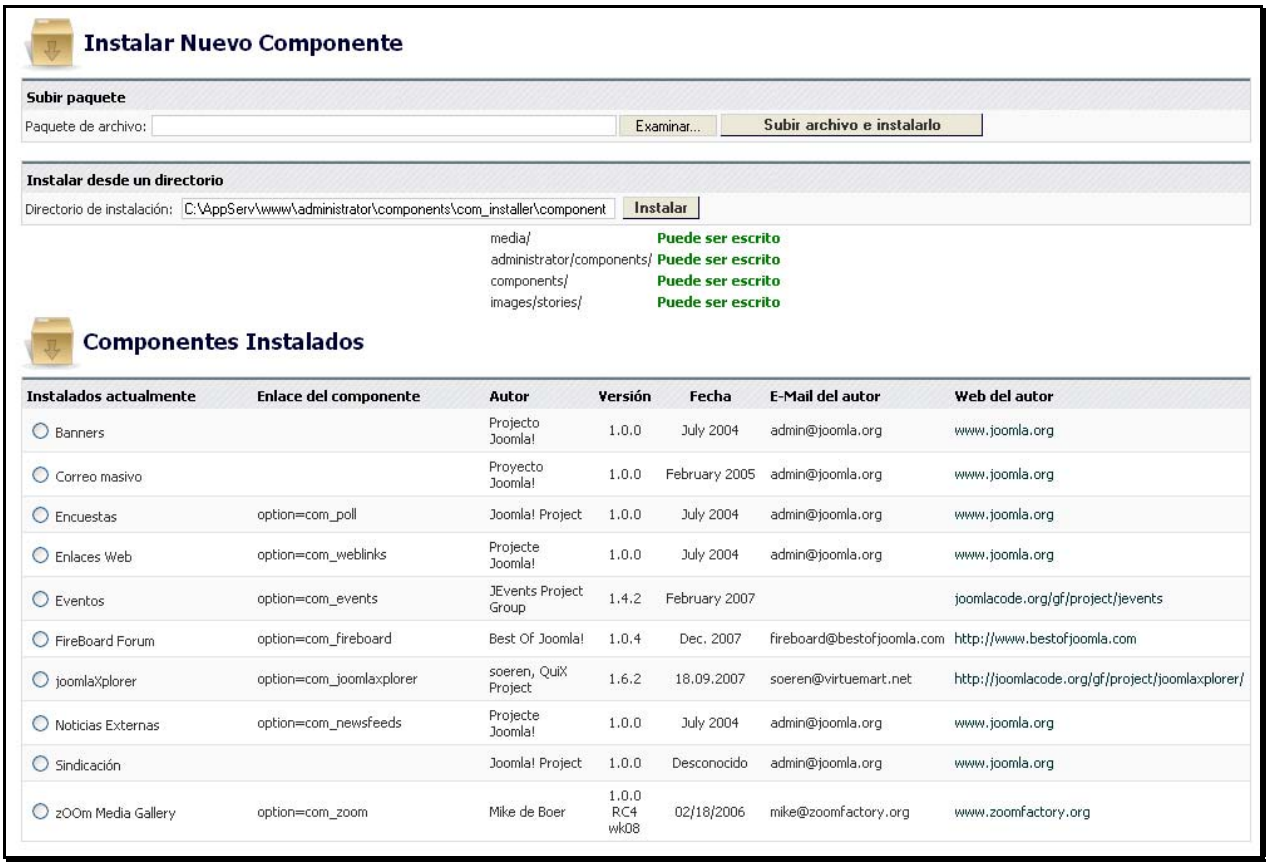


Figura 74: Instalar Componentes

Desde este panel es desde donde instalamos nuevos complementos y desinstalamos los que ya están en el sistema, pero que no son útiles. Mencionar que todos los complementos que se han añadido a JOOMLA se han descargado desde la página oficial del proyecto http://extensions.JOOMLA.org/.

El proceso de añadir un nuevo componente es sencillo. Primero, descargamos el archivo comprimido del componente (un fichero de tipo ZIP). Luego, desde la barra de Subir paquete, seleccionamos el fichero que contiene el componente y pulsamos en Subir e instalar. Una vez hecho, se instalará el componente y aparecerá en la lista de componentes del sistema. Para quitar un componente instalado, basta con seleccionarlo y pulsar en el botón Desinstalar.

No todos los componentes que se ven en esa lista son instalaciones adicionales. Un gran número de ellos son componentes que se instalan, por defecto, en la instalación inicial del sistema.

Los componentes que hemos instalado básicamente han sido:

- ✓ JOOMLAXplorer: Extensión que facilita la navegación por los ficheros del sistema.
- ✓ zOOm Media Gallery: Extensión que nos da la opción de crear galerías de imágenes.
- ✓ Events: Extensión que nos da la opción de crear y gestionar un calendario.
- ✓ FIRE Board Forum: Extensión que nos permite crear y gestionar un Foro.
- ✓ VCNT Counter: Extensión que nos permite crear y gestionar un contador de visitas.
- ✓ Scroll News: Extensión que nos permite crear un mostrador de noticias rotatorio.

4.12 - APARIENCIA DEL PORTAL

Figura 75: Administrador de Plantillas

En el menú superior del panel de administración, en la sección Sito > Administración de plantillas > Plantillas del sitio, podemos cambiar la plantilla del sistema por otra nueva que hayamos descargado de internet (se recomienda usar las del sitio oficial o sucursales de éste).

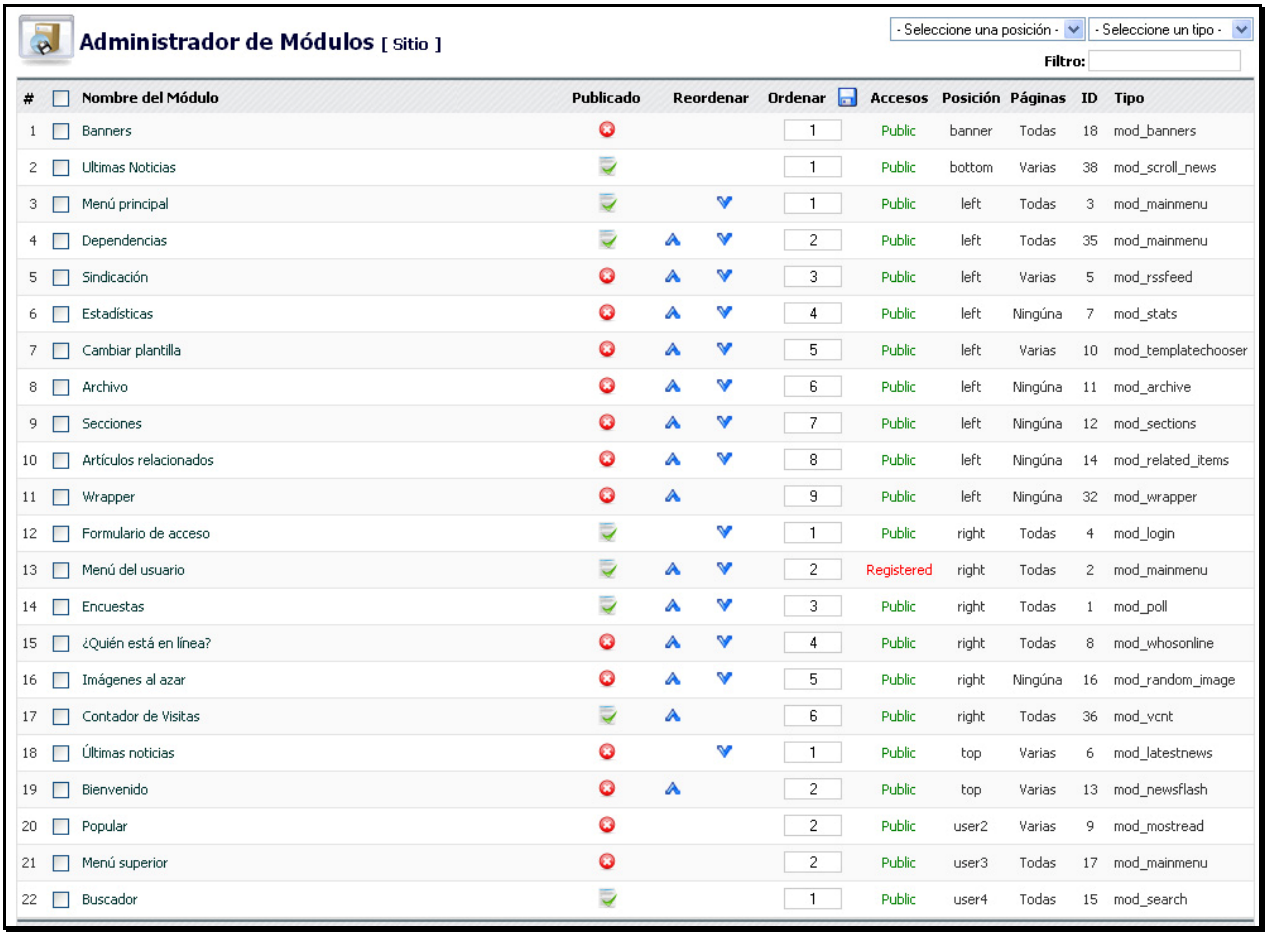
Para instalar una nueva plantilla, pulsamos en el botón Nuevo y seguimos los mismos pasos que para instalar un nuevo componente.

Las plantillas son las que otorgan al portal su apariencia. Por tanto, debemos escoger una que se amolde a nuestras necesidades visuales.

Existen otros dos botones importantes en esta sección, que son el de CSS y el de HTML, que sirven para editar la hoja de estilos del tema y su código fuente, respectivamente, operación que habrá que realizar para personalizar el portal hasta conseguir la apariencia deseada.

4.13 - DISPOSICIÓN DE LOS ELEMENTOS

Para decidir qué elementos aparecerán en nuestro portal como funcionalidades que rodean al contenido y en qué posición aparecerán, usaremos la opción del menú superior del panel de administración Módulos > Módulos del sitio.



Administrador de Módulos [Sitio]

- Seleccione una posición - | - Seleccione un tipo - | Filtro:

#	Nombre del Módulo	Publicado	Reordenar	Ordenar	Accesos	Posición	Páginas	ID	Tipo
1	Banners	✗		1	Public	banner	Todas	18	mod_banners
2	Ultimas Noticias	✓		1	Public	bottom	Varias	38	mod_scroll_news
3	Menú principal	✓	▼	1	Public	left	Todas	3	mod_mainmenu
4	Dependencias	✓	▲ ▼	2	Public	left	Todas	35	mod_mainmenu
5	Sindicación	✗	▲ ▼	3	Public	left	Varias	5	mod_rssfeed
6	Estadísticas	✗	▲ ▼	4	Public	left	Ningúna	7	mod_stats
7	Cambiar plantilla	✗	▲ ▼	5	Public	left	Varias	10	mod_templatechooser
8	Archivo	✗	▲ ▼	6	Public	left	Ningúna	11	mod_archive
9	Secciones	✗	▲ ▼	7	Public	left	Ningúna	12	mod_sections
10	Artículos relacionados	✗	▲ ▼	8	Public	left	Ningúna	14	mod_related_items
11	Wrapper	✗	▲	9	Public	left	Ningúna	32	mod_wrapper
12	Formulario de acceso	✓	▼	1	Public	right	Todas	4	mod_login
13	Menú del usuario	✓	▲ ▼	2	Registered	right	Todas	2	mod_mainmenu
14	Encuestas	✓	▲ ▼	3	Public	right	Todas	1	mod_poll
15	¿Quién está en línea?	✗	▲ ▼	4	Public	right	Todas	8	mod_whosonline
16	Imágenes al azar	✗	▲ ▼	5	Public	right	Ningúna	16	mod_random_image
17	Contador de Visitas	✓	▲	6	Public	right	Todas	36	mod_vcnt
18	Últimas noticias	✗	▼	1	Public	top	Varias	6	mod_latestnews
19	Bienvenido	✗	▲	2	Public	top	Varias	13	mod_newsflash
20	Popular	✗		2	Public	user2	Varias	9	mod_mostread
21	Menú superior	✗		2	Public	user3	Todas	17	mod_mainmenu
22	Buscador	✓		1	Public	user4	Todas	15	mod_search

Figura 76: Colocación de elementos

Aquí se nos muestran todos los módulos y componentes que trae JOOMLA por defecto y aquellos que hemos instalado nosotros posteriormente.

En la columna de publicado, elegiremos los elementos que queremos que aparezcan y, en qué posición, seleccionaremos la posición relativa que queremos que ocupe el elemento en esa posición (si queremos que sea el primero de la izquierda o el segundo de la derecha, por ejemplo).

Al pulsar sobre un elemento en cuestión, se nos permite seleccionar su posición (hay muchas) de entre las que elegiremos una de las cuatro principales, que son: left (izquierda), rigth (derecha), top (a lo alto) y user x (distintas localizaciones reservadas para espacios de usuarios).

En esa pantalla también seleccionamos qué visibilidad le queremos dar al elemento en cuestión. Por ejemplo, si es para todo el mundo o sólo para usuarios registrados.

Con esto, logramos que el portal adquiera el formato deseado y finalizamos además la implementación de nuestro portal.

5 - CONFIGURACIÓN SEGURA DEL SISTEMA

5.1 - SEGURIDAD SSL

5.1.01 - PROTOCOLO SSL

- ✓ INTRODUCCIÓN:

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

  Existen pequeñas diferencias entre SSL 3.0 y TLS 1.0, pero el protocolo permanece sustancialmente igual.

- ✓ DESCRIPCIÓN:

  SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía.

  Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes.

  Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas (eavesdropping), la falsificación de la identidad del remitente (phishing) y mantener la integridad del mensaje.

  - SSL implica una serie de fases básicas:

    1) Negociar entre las partes el algoritmo que se usará en la comunicación.

    2) Intercambio de claves públicas y autenticación basada en certificados digitales.

    3) Cifrado del tráfico basado en cifrado simétrico.

  - Durante la primera fase, el cliente y el servidor negocian qué algoritmos criptográficos se van a usar. Las implementaciones actuales proporcionan las siguientes opciones:

1) Para criptografía de clave pública: RSA, Diffie-Hellman, DSA (Digital Signature Algorithm) o Fortezza;

2) Para cifrado simétrico: RC2, RC4, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard), Triple DES o AES (Advanced Encryption Standard);

3) Con funciones hash: MD5 o de la familia SHA.

✓ CÓMO FUNCIONA:

El protocolo SSL intercambia registros; opcionalmente, cada registro puede ser comprimido, cifrado y empaquetado con un código de autentificación del mensaje (MAC).

Cada registro tiene un campo de content_type que especifica el protocolo de nivel superior que se está usando.

Cuando se inicia la conexión, el nivel de registro encapsula otro protocolo, el protocolo handshake, que tiene el content_type 22.

- El cliente envía y recibe varias estructuras handshake:

1) Envía un mensaje ClientHello especificando una lista de conjunto de cifrados, métodos de compresión y la versión del protocolo SSL más alta permitida. Éste también envía bytes aleatorios que serán usados más tarde (llamados Challenge de Cliente o Reto). Además puede incluir el identificador de la sesión.

2) Después, recibe un registro ServerHello, en el que el servidor elige los parámetros de conexión a partir de las opciones ofertadas con anterioridad por el cliente.

3) Cuando los parámetros de la conexión son conocidos, cliente y servidor intercambian certificados (dependiendo de las claves públicas de cifrado seleccionadas). Estos certificados son actualmente X.509, pero hay también un borrador especificando el uso de certificados basados en OpenPGP.

4) El servidor puede requerir un certificado al cliente, para que la conexión sea mutuamente autenticada.

5) Cliente y servidor negocian una clave secreta (simétrica) común llamada master secret, posiblemente usando el resultado de un intercambio Diffie-Hellman, o simplemente cifrando una clave secreta con una clave pública que es descifrada con la clave privada de cada uno. Todos los datos de claves restantes son derivados a partir de este master secret (y los valores aleatorios generados en el cliente y el servidor), que son pasados a través una función pseudoaleatoria cuidadosamente elegida.

- TLS/SSL poseen una variedad de medidas de seguridad:

1) Numerando todos los registros y usando el número de secuencia en el MAC.

2) Usando un resumen de mensaje mejorado con una clave (de forma que solo con dicha clave se pueda comprobar el MAC). Esto se especifica en el RFC 2104).

3) Protección contra varios ataques conocidos (incluyendo ataques man-in-the-middle), como los que implican un degradado del protocolo a versiones previas (por tanto, menos seguras), o conjuntos de cifrados más débiles.

4) El mensaje que finaliza el protocolo handshake (Finished) envía un hash de todos los datos intercambiados y vistos por ambas partes.

5) La función pseudo aleatoria divide los datos de entrada en 2 mitades y las procesa con algoritmos hash diferentes (MD5 y SHA), después realiza sobre ellos una operación XOR. De esta forma se protege a sí mismo de la eventualidad de que alguno de estos algoritmos se revelen vulnerables en el futuro.

5.1.02- OPENSSL Y AUTORIDADES DE CERTIFICACION

OpenSSL es un proyecto de software desarrollado por los miembros de la comunidad Open Source para libre descarga y está basado en SSLeay desarrollado por Eric Young y Tim Hudson.

Consiste en un robusto paquete de herramientas de administración y librerías relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenSSH y navegadores web (para acceso seguro a sitios HTTPS).

Estas herramientas ayudan al sistema a implementar el Secure Sockets Layer (SSL), así como otros protocolos relacionados con la seguridad, como el Transport Layer Security (TLS).

Este paquete de software es importante para cualquiera que esté planeando usar cierto nivel de seguridad en su máquina.

OpenSSL también nos permite crear certificados digitales y generar autoridades de certificación, que podremos aplicar a nuestro servidor, en nuestro caso “Apache”.

Con openssl generaremos certificados (con el formato x509) y claves privadas (en formato RSA) que serán usadas por Apache para autenticar al servidor por medio del protocolo SSL (la base de HTTPS).

5.1.03 - CONFIGURACIÓN DE OPENSSL

Una vez descargado el software OpenSSL de la web oficial http://www.openssl.org/ y seguidos los pasos para instalar el mismo, procederemos a describir las operaciones realizadas en el desarrollo de este proyecto.

Nuestros objetivos son conseguir crear una autoridad de certificación y dos certificados, uno que vamos a instalar en nuestro servidor y el otro en nuestro cliente (ya sea este último, instalado en el navegador del ordenador cliente o en su tarjeta inteligente), estos dos certificados además deberan ir firmados por la autoridad de certificación creada anteriormente para ofrecer un mayor nivel de confianza en los mismos.

Para dichos objetivos vamos a crear tres scripts que nos permitiran automatizar la creación tanto de la autoridad de certificación como de los certificados de cliente y servidor, y otros tres scripts para realizar de manera automática los borrados de dichos certificados:

- ✓ Para crear la Autoridad de Certificación → ***crearCA.cmd***
- ✓ Para crear y firmar el Certificado del Servidor → ***SERVIDOR.cmd***
- ✓ Para crear y firmar el Certificado del Cliente → ***CLIENTE.cmd***
- ✓ Para borrar la Autoridad de Certificación → ***borrarCA.cmd***
- ✓ Para borrar el Certificado del Servidor → ***borrarServidor.cmd***
- ✓ Para borrar el Certificado del Cliente → ***borrarCliente.cmd***

La secuencia lógica de pasos a seguir será crear en primer lugar la “Autoridad de Certificación” y a continuación los Certificados de “Servidor” y de “Cliente”, ya que estos deben ser firmados por la autoridad anteriormente creada.

5.1.04 - FICHERO “crearCA.cmd”

Este fichero es un script que nos permite automatizar la creación de nuestra “Autoridad de Certificación”, la cual es necesaria para realizar la firma de los “Certificados” del “Servidor” y del “Cliente”, aumentando asi el nivel de seguridad y la confianza en dichos certificados.

CÓDIGO DEL FICHERO “crearCA.cmd”

```
set PATH=C:\OpenSSL\bin

pause

mkdir demoCA
mkdir demoCA\certs
mkdir demoCA\crl
mkdir demoCA\newcerts
mkdir demoCA\private
mkdir demoCA\p12
mkdir demoCA\der

echo 01 > demoCA\serial

copy nul demoCA\index.txt > nul

openssl req -config openssl2.cnf -x509 -newkey rsa:1024 -out demoCA\cacert.pem
-outform PEM

pause

openssl pkcs12 -export  -inkey demoCA\private\cakey.pem -in demoCA\cacert.pem
-out demoCA\p12\cacert.p12 -cacerts

pause

openssl x509 -inform PEM -outform DER -in demoCA\cacert.pem -out
demoCA\der\cacert.der
```

5.1.05 - FICHERO “SERVIDOR.cmd”

Este fichero es un script que nos permite automatizar la creación de nuestro “Certificado del Servidor” y la firma del mismo, mediante la “Autoridad de Certificación” creada con anterioridad.

CÓDIGO DEL FICHERO “SERVIDOR.cmd”

```
set PATH=C:\OpenSSL\bin

mkdir SERVIDOR
mkdir SERVIDOR\CLAVEPRIV

openssl req -config openssl2.cnf -new -keyout SERVIDOR\CLAVEPRIV\servidor.key
-out SERVIDOR\CLAVEPRIV\servidor.key -subj "/CN=localhost/OU=Servidor_U.F.P."

openssl ca -config openssl2.cnf  -policy policy_anything -out
SERVIDOR\servidor.crt -infiles SERVIDOR\CLAVEPRIV\servidor.key

pause

openssl rsa -in SERVIDOR\CLAVEPRIV\servidor.key -out
SERVIDOR\CLAVEPRIV\servidor2.key

openssl pkcs12 -export -in SERVIDOR\servidor.pem -inkey
SERVIDOR\CLAVEPRIV\servidor.pem -out SERVIDOR\servidor.p12
```

5.1.06 - FICHERO “CLIENTE.cmd”

Este fichero es un script que nos permite automatizar la creación de nuestro “Certificado del Cliente” y la firma del mismo, mediante la “Autoridad de Certificación” creada con anterioridad.

CÓDIGO DEL FICHERO “CLIENTE.cmd”

```
set PATH=C:\OpenSSL\bin

mkdir CLIENTE
mkdir CLIENTE\CLAVEPRIV

openssl req -config openssl2.cnf -newkey rsa:1024 -new -keyout
CLIENTE\CLAVEPRIV\cliente.key -out CLIENTE\CLAVEPRIV\cliente.key -subj
"/CN=Cliente_U.F.P./OU=Cliente_U.F.P."

openssl ca -config openssl2.cnf  -policy policy_anything -out
CLIENTE\cliente.pem -infiles CLIENTE\CLAVEPRIV\cliente.key

pause

openssl rsa -in CLIENTE\CLAVEPRIV\cliente.key -out
CLIENTE\CLAVEPRIV\cliente2.key

openssl pkcs12 -export -in CLIENTE\cliente.pem -inkey
CLIENTE\CLAVEPRIV\cliente.key -out CLIENTE\cliente.p12
```

5.1.07 - FICHERO “borrarCA.cmd”

Este fichero es un script que nos permite automatizar el borrado de nuestra “Autoridad de Certificación” creada con anterioridad.

CÓDIGO DEL FICHERO “borrarCA.cmd”

```
rmdir /s/q demoCA > nul
```

5.1.08 - FICHERO “borrarServidor.cmd”

Este fichero es un script que nos permite automatizar el borrado de nuestro “Certificado del Servidor” creado con anterioridad.

CÓDIGO DEL FICHERO “borrarServidor.cmd”

```
rmdir /s/q SERVIDOR > nul
```

5.1.09 - FICHERO “borrarCliente.cmd”

Este fichero es un script que nos permite automatizar el borrado de nuestro “Certificado del Cliente” creado con anterioridad.

CÓDIGO DEL FICHERO “borrarCliente.cmd”

```
rmdir /s/q CLIENTE > nul
```

5.2 - CONFIGURACIÓN SEGURA DEL SERVIDOR APACHE

5.2.1 - INTRODUCCIÓN AL SERVIDOR APACHE

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), 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 eligió ese nombre porque 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, mensajes de error 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. (Estadísticas históricas y de uso diario proporcionadas por Netcraft).

5.2.2 - CONFIGURANDO APACHE 2.2

Aquí trataremos la configuración básica de Apache 2.2. Por tanto, no se explicarán ni las directivas relacionadas con SSL (*mod ssl*) ni las relacionadas con reescritura de URLs (*mod rewrite*).

Lo primero que hay que decir es que Apache es un proceso en ejecución en el sistema y que ese proceso está cargado en memoria con una serie de características específicas. Esas características son las que se le expresan en el fichero *"httpd.conf"*, presente en el caso especifico de este proyecto, en el directorio *"C:\AppServ\Apache2.2\conf\"*. Por tanto, cada vez que el programa arranca, leerá la información contenida en dicho fichero y si no hay errores, arrancará en base a esos parámetros establecidos.

Por tanto, será el fichero *"httpd.conf"* el que tendremos que configurar para adaptar Apache 2.2 a nuestros gustos o necesidades.

Todas las directivas de configuración se encuentran encapsuladas dentro de "módulos", que no son más que complementos que añaden funcionalidad a las directivas básicas (las del núcleo).

Los módulos que están cargados en Apache (los que se pueden usar) se encuentran referenciados y activados desde el archivo "httpd.conf" y los que están disponibles, pero no en uso se encuentran en la ruta *"C:\AppServ\Apache2.2\modules\"*.

En este proyecto, se usarán dos módulos disponibles pero que están sin activar, el de SSL (*mod ssl*) y el de reescritura de URLs (*mod rewrite*). Más adelante veremos dicha descripción y cómo activarlos.

El formato básico de ese fichero es el de directivas y bloques:

- ✓ "Directivas" quiere decir que se escribe una palabra reservada (comando) con una serie de opciones. Este comando con sus respectivas opciones tendrá un efecto que determinará el comportamiento del servidor.

- ✓ "Bloques" significa que se puede definir el ámbito de efecto de un grupo de directivas, esto significa que mientras no se exprese lo contrario una directiva afectará al comportamiento global del servidor, pero por medio de bloques se puede lograr que afecte sólo a una parte del servidor o bien que afecte al servidor sólo en determinadas circunstancias.

Vamos a crear en este punto el directorio raíz que contendrá toda la infraestructura del portal y el foro, lo crearemos y referenciaremos como *"C:\AppServ\www\"*.

5.2.3 - FICHERO “httpd.conf”

Aquí se expone el fichero de configuración básico “httpd.conf” con varios comentarios dentro del mismo código para hacer más comprensible dichas directivas.

El directorio que contendrá este fichero de configuración básica en nuestro proyecto lo podremos encontrar en la siguiente ruta: *“C:\AppServ\Apache2.2\conf\”*.

A continuación examinaremos las principales modificaciones sobre el fichero base, para más información sobre este fichero consultar el (*APÉNDICE “G” - FICHERO “httpd.conf”*) de esta documentación.

5.2.3.1 - Directorio Root

- ✓ Referencia al directorio raiz del Servidor Apache en nuestro sistema.

```
ServerRoot "C:/AppServ/Apache2.2"
```

5.2.3.2 - Configuración Segura de Puertos

- ✓ El puerto 80 se utiliza para que el servidor escuche por un puerto no seguro.
- ✓ El puerto 443 se utiliza para que el servidor escuche por un puerto seguro, que usaremos con el protocolo SSL.

```
Listen 80
Listen 443
```

5.2.3.3 - Archivo con reglas de seguridad JOOMLA

- ✓ Nombramos el archivo que en JOOMLA contendrá las reglas de seguridad para nuestro portal web.

```
AccessFileName .htaccess.txt
```

5.2.3.4 - Activamos el Módulo “mod_rewrite”

- ✓ Activamos el Módulo “mod_rewrite” para que el archivo “.htaccess” contenido dentro de nuestro Portal web JOOMLA sea tenido en cuenta y se interpreten sus reglas internas de seguridad.

```
LoadModule rewrite_module modules/mod_rewrite.so
```

5.2.3.5 - Activamos el Módulo “mod_ssl”

- ✓ Activamos el Módulo “mod_ssl” para que sea posible utilizar el protocolo de comunicaciones seguras SSL con el Servidor Apache.

```
LoadModule ssl_module modules/mod_ssl.so
```

5.2.3.6 - Correo del Administrador

- ✓ La cuenta de correo del Administrador del Servidor Apache.

```
ServerAdmin 100030106@alumnos.uc3m.es
```

5.2.3.7 - El nombre del Servidor

- ✓ El nombre del Servidor, en nuestro caso actual será “localhost”, aunque en un futuro esta regla deberá cambiar para corresponderse con la URL de internet donde se aloje el Portal web, como por ejemplo “http://www.ufpmadrid.com”.

```
ServerName localhost
```

5.2.3.8 - Ficheros de configuración para Protocolo SSL

- ✓ En esta regla hacemos uso del fichero de configuración por defecto y del fichero de configuración del Protocolo de comunicaciones seguras SSL.

```
Include conf/extra/httpd-default.conf
Include conf/extra/httpd-ssl.conf
```

5.2.4 - FICHERO “httpd-ssl.conf”

Aquí se expone el fichero de configuración del protocolo de seguridad SSL “httpd-ssl.conf” con varios comentarios dentro del mismo código para hacer más comprensible dichas directivas.

El directorio que contendrá este fichero de configuración de seguridad en nuestro proyecto lo podremos encontrar en la siguiente ruta: *“C:\AppServ\Apache2.2\conf\extra\”*.

Este fichero a diferencia del anterior ha sido implementado por completo para adaptarlo a este proyecto por lo que no se encontraran directivas por defecto.

A continuación examinaremos su implementación, para más información sobre este fichero consultar el (*APÉNDICE “H” - FICHERO “httpd-ssl.conf”*) de esta documentación.

5.2.4.1 - Servidor Virtual Seguro SSL

- ✓ Sevidor Virtual: El canal SSL se establecera en el puerto seguro, 443.

```
<VirtualHost localhost:443>
```

5.2.4.2 - Configuración Servidor SSL

- ✓ Servidor: Habilitamos el Certificado y la Clave Privada del Servidor.

```
SSLCertificateFile conf/SERVIDOR/servidor.crt
SSLCertificateKeyFile conf/SERVIDOR/CLAVEPRIV/servidor2.key
```

5.2.4.3 - Configuración Autoridad de Certificación SSL

- ✓ CA: Habilitamos a la Autoridad de Certificacion.

```
SSLCACertificatePath conf/demoCA
SSLCACertificateFile conf/demoCA/cacert.pem
```

5.2.4.4 - Configuración Cliente SSL

- ✓ Cliente: Forzamos a que el Cliente tenga Certificado para establecer el canal SSL.

```
SSLVerifyClient require
```

5.2.4.5 - Redireccionando a modo Seguro al Usuario

- ✓ Forzamos la comunicación segura: Mediante el Protocolo SSL cuando accedemos a ciertas areas de nuestro Portal web JOOMLA con información sensible, es decir, nuestra comunicación se realiza de manera cifrada, desde http forzamos a https.

```
<Directory "C:/AppServ/www">
RewriteEngine on
```

- ✓ Condiciones: Si no es un acceso por el puerto seguro 443 y se accede a algún área del Portal JOOMLA que contenga información sensible, se redirecciona la comunicación a un canal seguro.

```
## CONDICION: Si no es un acceso por el puerto 443 ...
RewriteCond %{SERVER_PORT} !443$

## CONDICION: Y cuando la URL contenga cualquiera de las siguientes cadenas:

# FORMULARIO DE ACCESO: Boton entrar.
RewriteCond %{QUERY_STRING} (option=login)+ [OR]

# FORMULARIO DE ACCESO: Opcion Recuperar Clave.
RewriteCond %{QUERY_STRING} (option=com_registration)+ [OR]

# MENU USUARIO.
RewriteCond %{QUERY_STRING} (task=blogcategory)+ [OR]

# FORO USUARIO.
RewriteCond %{QUERY_STRING} (com_fireboard)+
```

- ✓ Regla: Si se cumplen las condiciones anteriores, automaticamente cambiamos de http a https esa misma URL. Si la URL, no contiene a ninguna de dichas cadenas, o el puerto por el que atendiamos ya era el 443, no se ejecutara dicha regla.

```
RewriteRule ^(.*)$ https://localhost/$1 [L,R]
```

5.2.4.6 - Redireccionando a modo Seguro al Administrador

- ✓ Forzamos la comunicación segura: Mediante el Protocolo SSL cuando accedemos al área del Administrador o “Back-End”, se redirecciona la comunicación a un canal seguro, es decir, nuestra comunicación se realiza de manera cifrada, desde http forzamos a https.

```
<Directory "C:/AppServ/www/administrator">
RewriteEngine on
```

- ✓ Condiciones: Si no es un acceso por el puerto seguro 443 y se accede a algún área del Administrador o “Back-End”, se redirecciona la comunicación a un canal seguro.

```
## CONDICION: Si no es un acceso por el puerto 443 ...
RewriteCond %{SERVER_PORT} !443$
```

- ✓ Regla: Si se cumplen las condiciones anteriores, automaticamente cambiamos de http a https esa misma URL. Si el puerto por el que atendiamos ya era el 443, no se ejecutara dicha regla.

```
RewriteRule ^(.*)$ https://localhost/administrator/$1 [L,R]
```

5.2.5 - FICHERO ".htaccess"

Aquí se expone el fichero de configuración ".htaccess" con varios comentarios dentro del mismo código para hacer más comprensible dichas directivas.

El directorio que contendrá este fichero de configuración en nuestro proyecto lo podremos encontrar en la ruta raiz del portal: *"C:\AppServ\www\"*.

A continuación examinaremos las principales modificaciones sobre el fichero base, para más información sobre este fichero consultar el (*APÉNDICE "I" - FICHERO ".htaccess"*) de esta documentación.

5.2.5.1 - Protección del Archivo

- ✓ Protegemos el archivo ".htaccess" de posibles ataques, denegando el acceso al mismo.

```
<Files .htaccess>
    order allow,deny
    deny from all
</Files>
```

5.2.5.2 - Desactivamos "register_globals"

- ✓ Desactivamos register_globals para prevenir posibles ataques contra JOOMLA.

```
php_value register_globals 0
```

5.3 - ACCESO AUTENTICADO MEDIANTE TARJETAS INTELIGENTES

Como ya se expresó en la sección de Apache, necesitamos el Certificado del Cliente, el Certificado del Servidor y la Autoridad de certificación para poder establecer conexiones HTTPS (basadas en SSL).

Por tanto, como ya se definió en su momento, debemos mover dichos ficheros al directorio correspondiente del servidor Apache, en nuestro caso:

- ✓ AUTORIDAD DE CERTIFICACIÓN: "C:\AppServ\Apache2.2\conf\demoCA\"

Que tendrá la siguiente estructura de directorios:

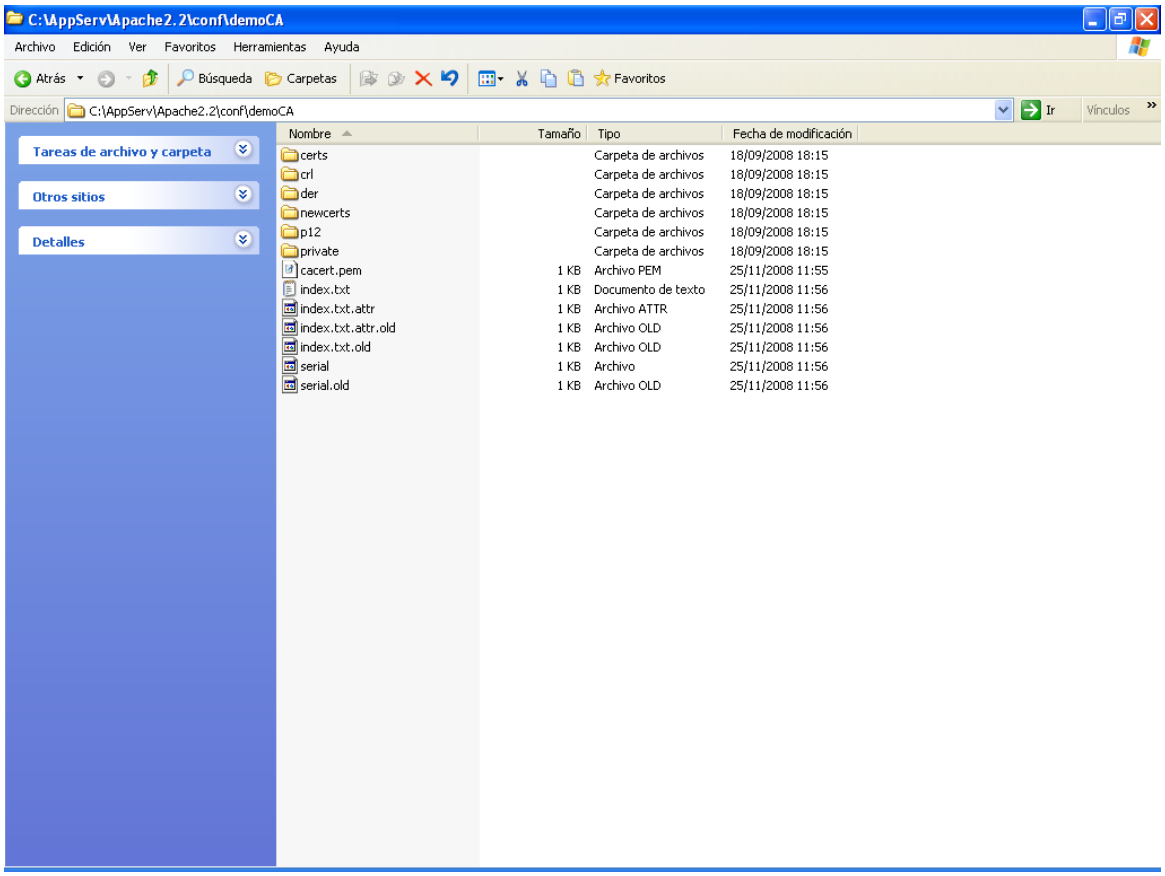


Figura 77: Directorio "C:\AppServ\Apache2.2\conf\demoCA\"

✓ CERTIFICADO DEL SERVIDOR: "C:\AppServ\Apache2.2\conf\SERVIDOR\"

Que tendrá la siguiente estructura de directorios:

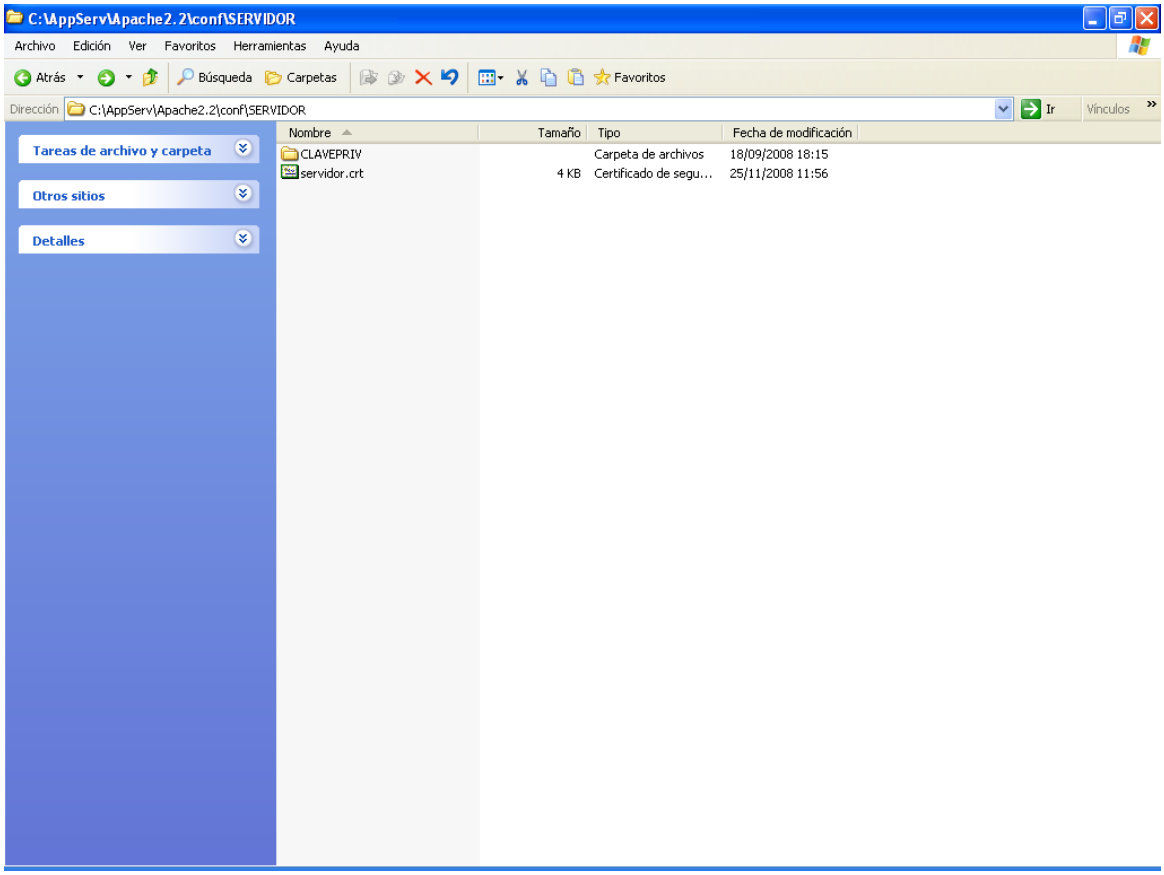


Figura 78: Directorio "C:\AppServ\Apache2.2\conf\SERVIDOR\"

✓ CERTIFICADO DEL CLIENTE:

El Certificado del Cliente será un caso aparte, ya que podrá ser instalado de dos maneras distintas dependiendo de las necesidades:

1) INSTALACIÓN EN EL NAVEGADOR:

El primer caso y más sencillo será instalando dicho certificado en el navegador web que vaya a utilizar el cliente, para lo cual deberá realizar los siguientes pasos:

1.1) Primero debemos ejecutar el fichero "*CLIENTE.cmd*" que como se explico en el punto (*4.2.06*) de esta documentación, sirve para crear de manera automática toda la estructura necesaria para el Certificado del Cliente.

1.2) Una vez creada dicha estrutura, como muestra la siguiente figura:

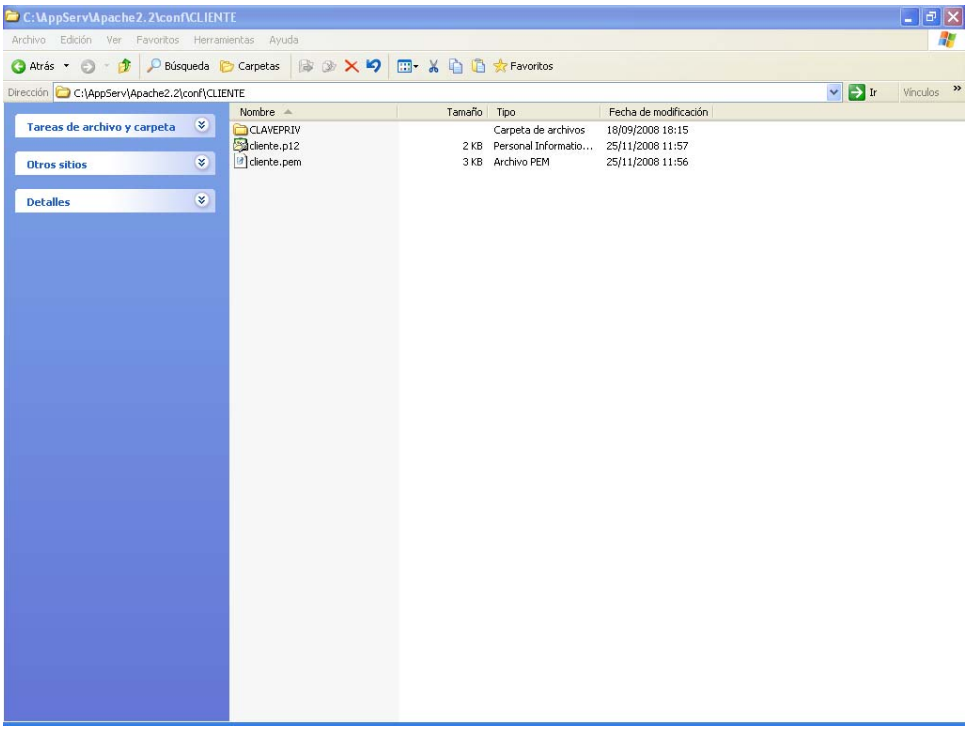


Figura 79: Directorio "C:\AppServ\Apache2.2\conf\CLIENTE\"

Deberemos instalar el Certificado del Cliente mediante el formato estandar "*.P12*" en nuestro navegador web.

1.3) Para dicha operación debemos abrir nuestro navegador, en nuestro caso "*Internet Explorer*".

A continuación abriremos la siguiente ruta en el menú del navegador: *"Herramientas"* → *"Opciones de Internet"* → *"Contenido"*

Aparecerá la sigiente ventana en la que seleccionaremos la opción "*Certificados*", como muestra la siguiente figura:

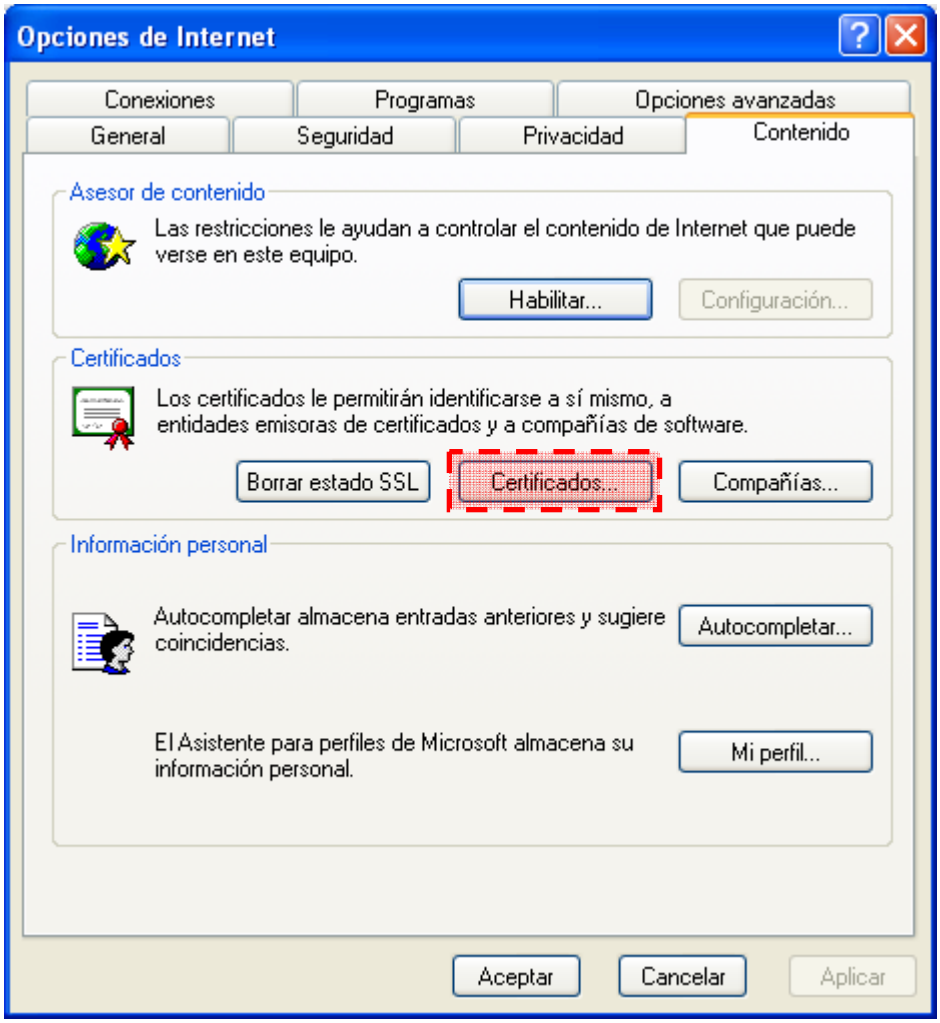


Figura 80: Ventana "Opciones de Internet"

1.4) Al pulsar dicho botón nos aparecerá la sigiente pantalla:

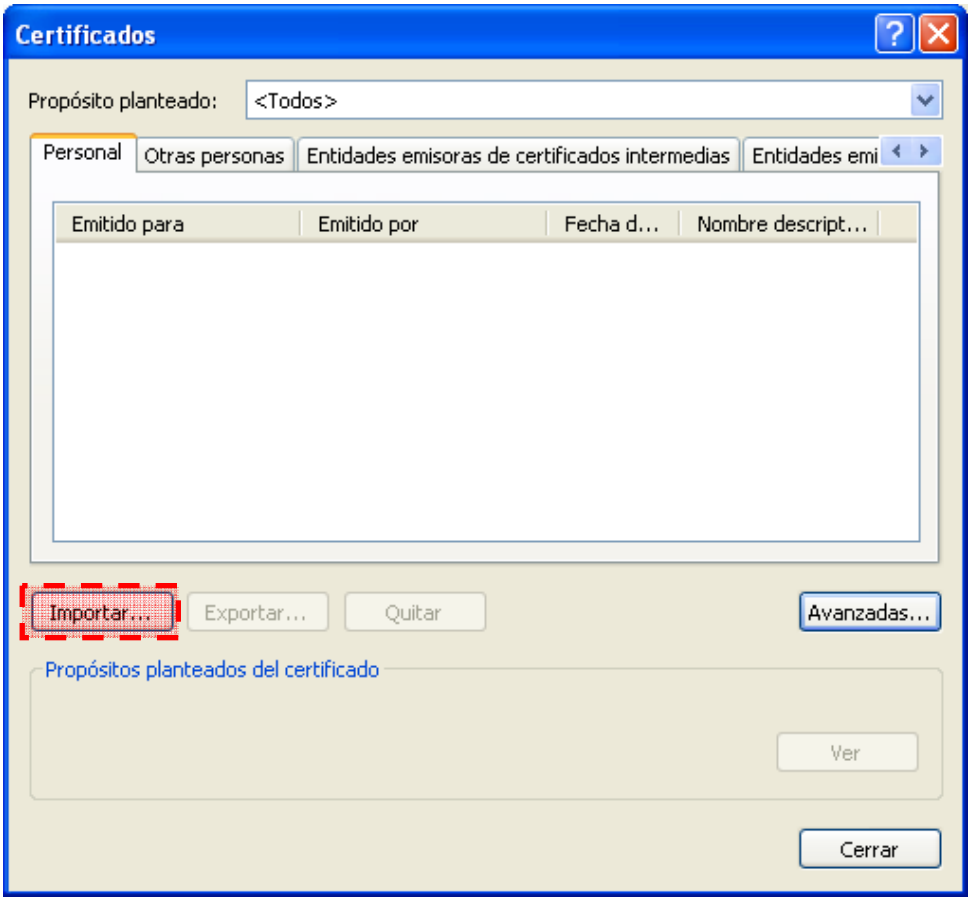


Figura 81: Ventana "Certificados"

Pulsaremos el botón "*Importar*", que realiza la instalación del Certificado del Cliente desde el archivo "*cliente.p12*", ya que este se encuentra en formato estandar y puede ser importado.

1.5) Al pulsar dicho botón nos aparecerá la sigiente pantalla:

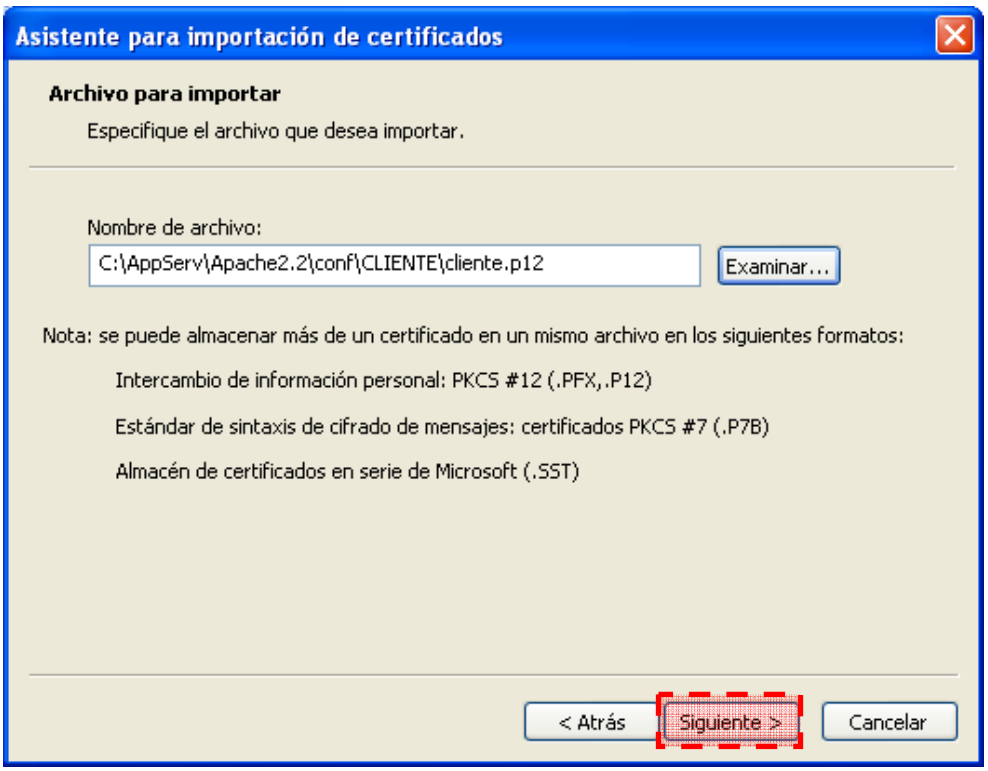


Figura 82: Importando “cliente.p12”

A continuación nos pedirá la “*clave privada*” de ese certificado.

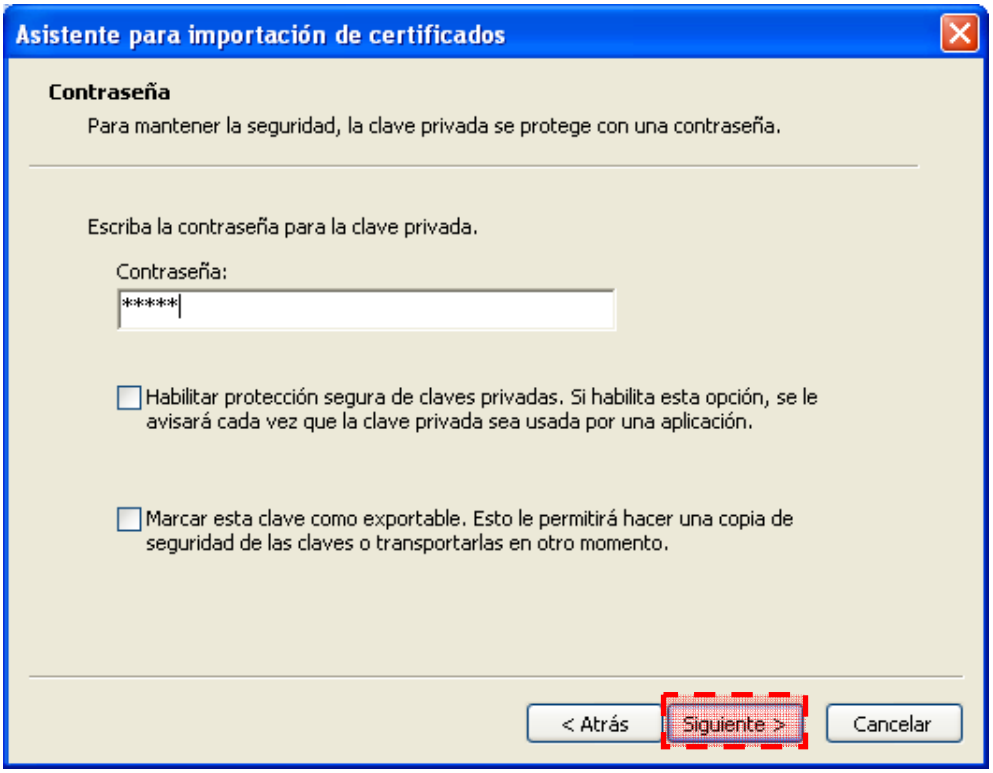


Figura 83: Clave Privada “cliente.p12”

1.6) Sin la clave privada no podriamos instalar dicho certificado.

En la nueva pantalla de "*Opciones de Internet - Certificados*" podremos ver como nuestro "*Certificado del Cliente*" esta instalado correctamente.

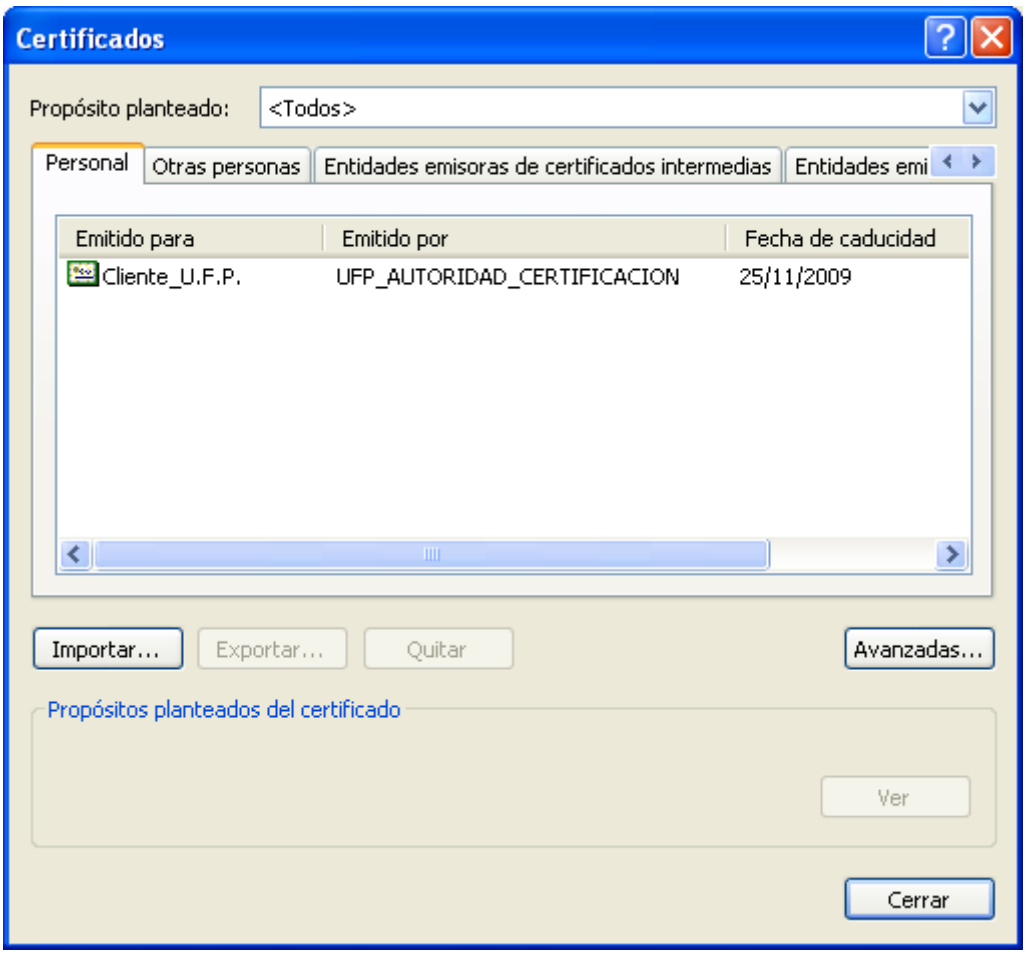


Figura 84: "cliente.p12" instalado

Una vez completados todos estos pasos podremos establecer una conexión segura mediante el protocolo SSL con nuestro proyecto (Portal JOOMLA) desde el navegador donde este instalado este certificado de cliente.

NOTA OPCIONAL (Instalación de la Autoridad de Certificación en el Navegador):

Para evitar que el navegador desconfie de la "*Autoridad de Certificación*" creada en el desarrollo de este proyecto, ésta deberá ser instalada en el navegador web.

Este requisito no es necesario ya que la conexión será realizada igualmente, pero permitirá más comodidad al usuario final, evitando mensajes del navegador que advierten sobre la desconfianza en dicha Autoridad de Certificación.

Para desarrollar dicha tarea se deberan seguir los mismos pasos que para instalar el "*Certificado de Cliente*" en el navegador, con la salvedad que en la pantalla "*Certificados*" habrá que seleccionar la pestaña "*Entidades Emisoras Raiz de Confianza*" e "*Importar*" en este espacio el archivo "*cacert.p12*" relativo a la "*Autoridad de Certificación*" que hemos creado para este proyecto.

El resultado de estos pasos se muestra en la siguiente figura:

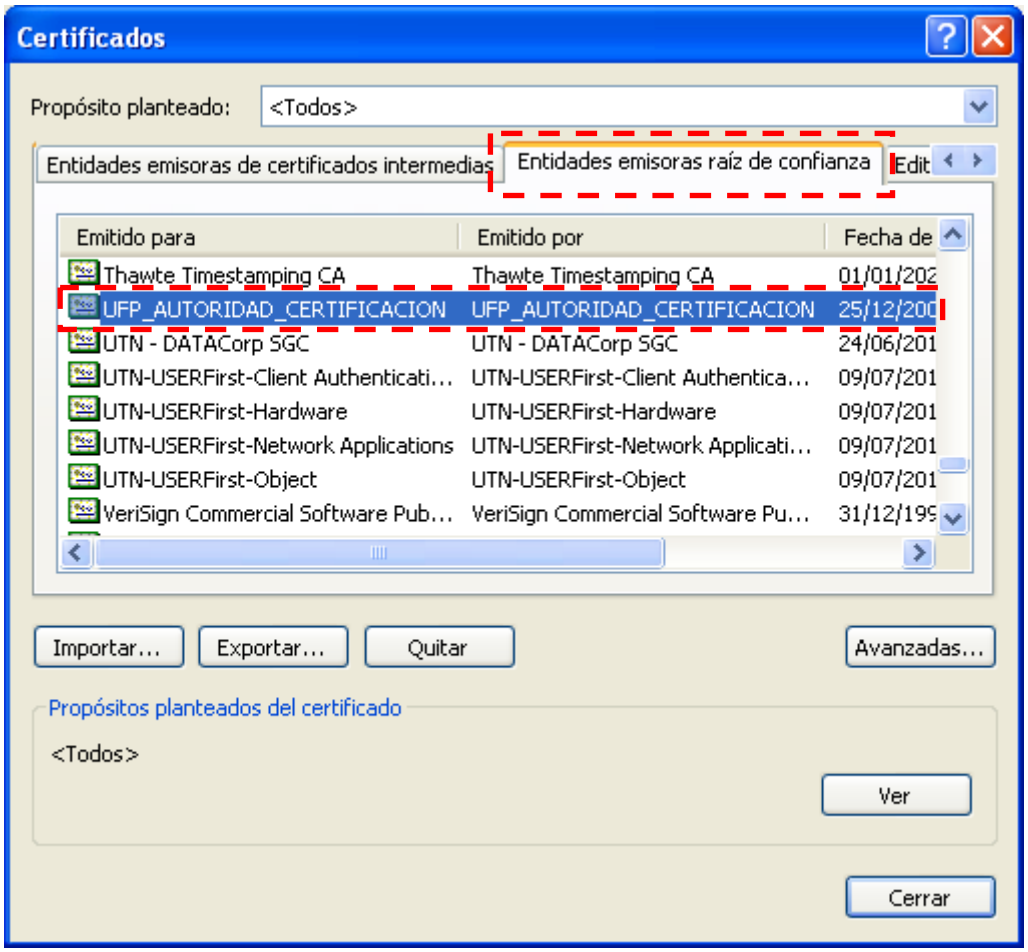


Figura 85: "cacert.p12" instalado

2) INSTALACIÓN EN TARJETA INTELIGENTE:

Este segundo caso es ligeramente más complejo, pero a la vez más funcional porque al instalar el "Certificado del Cliente" en una Tarjeta Inteligente obtendremos más libertad para acceder a nuestro portal.

La explicación a este hecho es que podremos conectarnos desde cualquier terminal con acceso a Internet y chequearnos con nuestra Tarjeta Inteligente contra el Portal sin necesidad de instalar dicho certificado en todos esos terminales, ya que lo llevaremos instalado en nuestra Tarjeta Inteligente.

Los pasos a seguir para instalar el "Certificado de Cliente" en nuestra Tarjeta Inteligente son los siguientes:

2.1) Primero debemos ejecutar el fichero "*CLIENTE.cmd*" que como se explico en el punto (*4.2.06*) de esta documentación, sirve para crear de manera automática toda la estructura necesaria para el Certificado del Cliente.

2.2) Debemos tener instalada la herramienta software "*Cryptokit*" de la Fabrica Nacional de Moneda y Timbre, explicada en el punto (*2.8*) de esta documentación, para más información: http://www.cert.fnmt.es

2.3) Debemos tener un lector de tarjetas, similar al que se muestra en la siguiente figura.

Figura 86: Lector de Tarjeta Inteligente

2.4) Abrimos la aplicación "*Importador de Certificados*" del software "*Cryptokit*", en ese momento aparecerá la siguiente pantalla:

Figura 87: Asistente para la Importación de Certificados

2.5) En la siguiente pantalla seleccionamos el certificado a importar, realizando así la instalación del "*Certificado del Cliente*" desde el archivo "*cliente.p12*", ya que este se encuentra en formato estandar y puede ser importado.

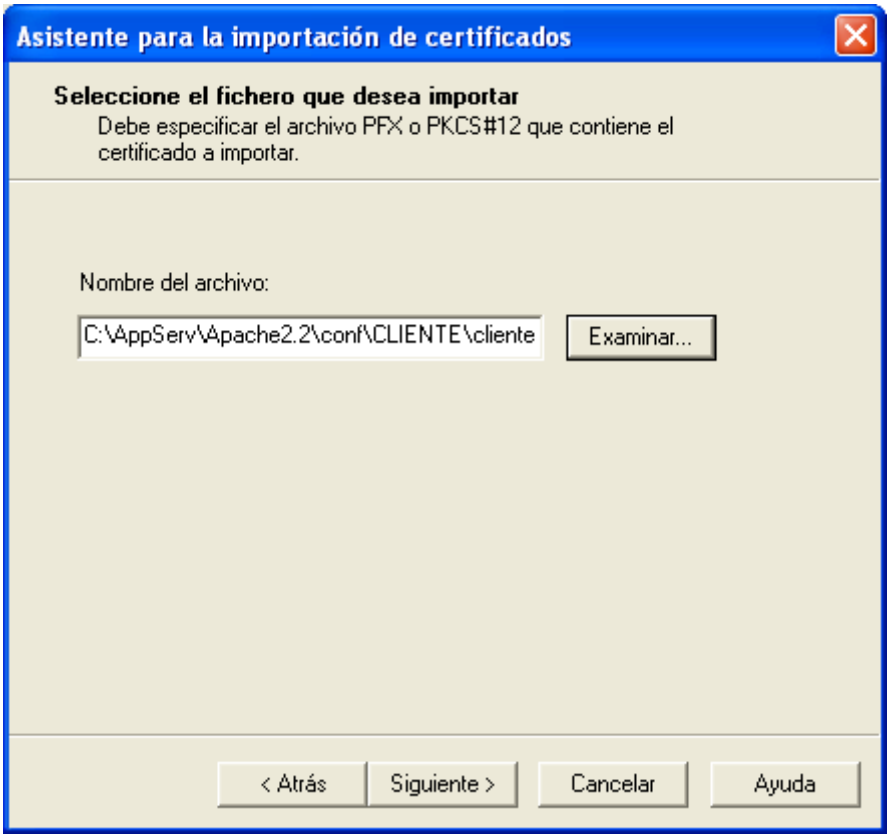


Figura 88: Importación del Certificado "cliente.p12" en Tarjeta Inteligente

2.6) A continuación nos pedirá la "*clave privada*" de ese certificado.

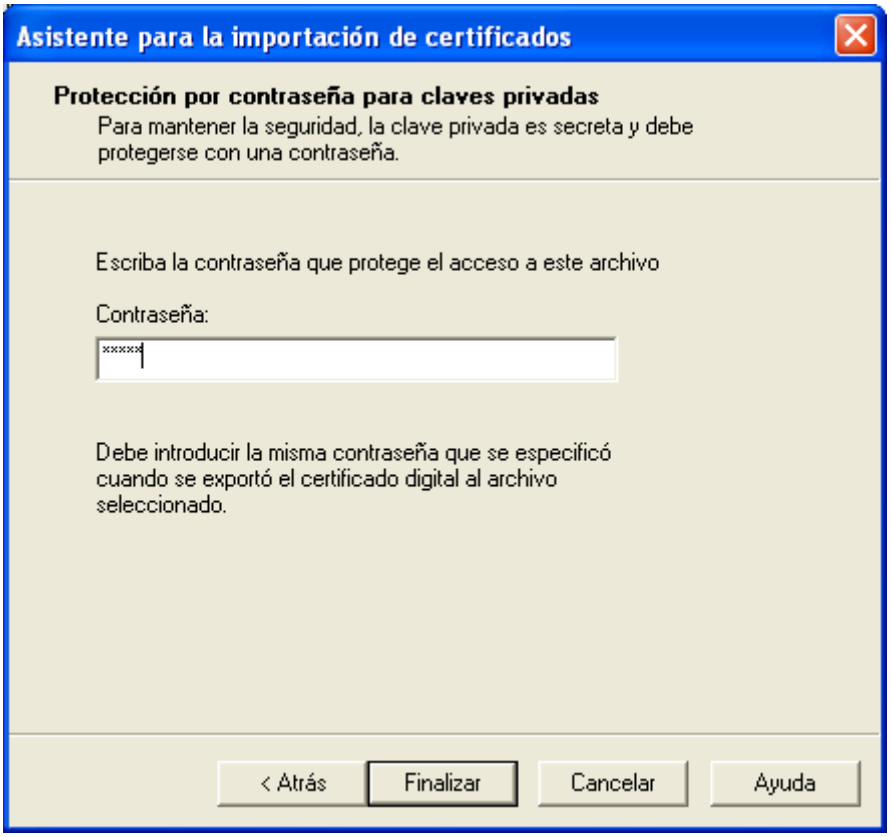


Figura 89: Clave privada del Certificado "cliente.p12" en Tarjeta Inteligente

2.7) Tras estas operaciones tendremos instalado dicho certificado en nuestra Tarjeta Inteligente.

Figura 90: "cliente.p12" instalado en Tarjeta Inteligente

5.4 - COPIAS DE SEGURIDAD

5.4.1 - CREAR COPIAS DE SEGURIDAD

Para realizar las copias de seguridad debemos seguir dos pasos fundamentales:

1) El primero es hacer una copia fisica del directorio del portal.

2) El segundo es hacer una copia de la base de datos MySQL (o al menos de las tablas relacionadas con el portal y el foro).

Debemos hacer una copia fisica de los directorios, ya que en ellos se almacenan: imágenes de usuarios, contenidos subidos al servidor, componentes instalados posteriores a la instalación base, etc.

De modo que, si queremos que esos componentes, información, etc. sigan presentes en el futuro y además no se vuelva inestable la base de datos (muchos de esos componentes crean sus propias tablas dentro de la base de datos), debemos realizar esa copia física de los directorios.

La segunda parte consiste en hacer la copia de seguridad de la base de datos. Dicha copia se puede realizar muy fácilmente desde el administrador PHPMyAdmin.

5.4.2 - RESTAURAR COPIAS DE SEGURIDAD

Si queremos restaurar las copias de seguridad hechas con anterioridad, lo primero que debemos hacer es llevar a cabo la configuración del sistema tal cual se ha descrito en este proyecto (en caso de ser necesario).

Luego, debemos descomprimir los ficheros que contienen los directorios del portal en el DocumentRoot del sistema (para que sean accesibles vía web).

Con lo que nuevamente obtenemos la disposición antigua de dichos directorios. Ahora es el momento de recuperar la base de datos, para ello debe estar bien configurada (tener preparado el usuario root y su password).

Ahora se deben dar los permisos oportunos a todos los ficheros y directorios del portal (son los mismos de la instalación y alguno extra que identificaremos rápidamente por errores en la ejecución del portal).

Una vez que está todo listo, hay que cerciorarse de que los parámetros de los ficheros de configuración del portal (*configuration.php*) tengan los valores adecuados:

Configuration.php

$mosConfig absolute_path debe referenciar el directorio de instalación del portal.

$mosConfig_cachepath debe referenciar el directorio de cache del portal (dentro del portal).

$mosConfig db debe referenciar el nombre de la base de datos del portal (no el servidor).

$mosConfig dbprefix debe referenciar el prefijo de las tablas (ufp_ por defecto).

$mosConfig host debe referenciar a la base de datos (por defecto localhost).

$mosConfig live site debe referenciar la URL de acceso al servidor (el dominio).

$mosConfig_password debe referenciar el password del usuario en la base de datos.

$mosConfig user debe referenciar al usuario de la base de datos.

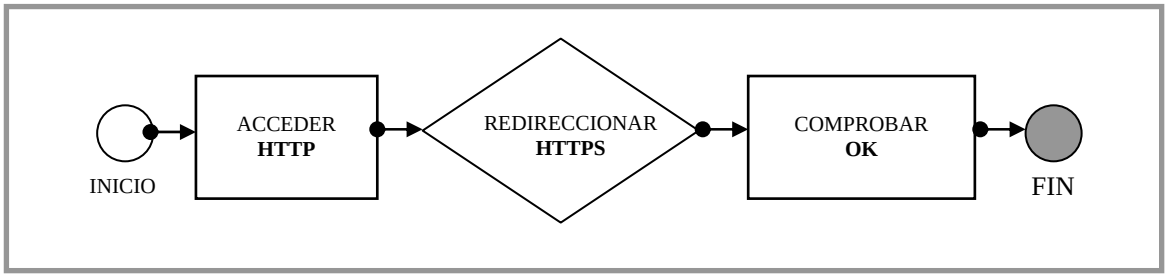
6 - TESTING DEL SISTEMA

Vamos a realizar dos tipos de comprobaciones en esta batería de pruebas: por un lado, que la configuración (sobre todo en el aspecto de seguridad) funciona adecuadamente y por otro, será verificar el correcto funcionamiento de las aplicaciones implementadas en sí.

6.1 - TESTING DE LA CONFIGURACIÓN

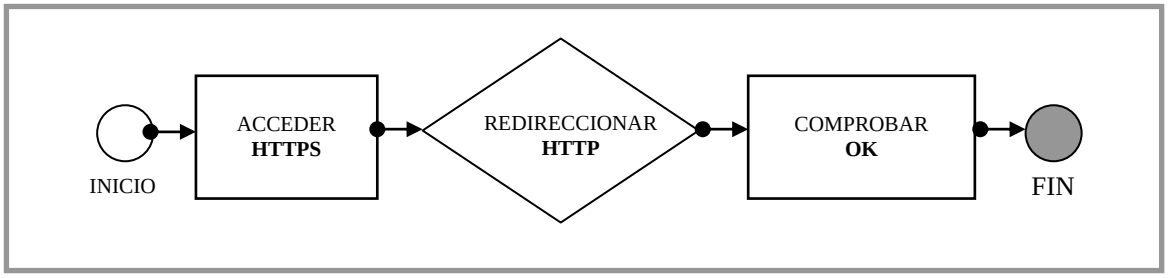
Prueba 1: Será comprobar si Apache redirige, de forma automática, el tráfico del puerto 80 al 443 (HTTPS), cuando es necesario (se intenta acceder a una zona restringida, es decir, zona del administrador o zona de usuario registrado).

- ✓ Obtenemos que, de forma automática, esta dirección http://localhost/administrator/ se redirige a https://localhost/administrator/. Igualmente cuando accedemos como usuario registrado. De modo que la redirección a tráfico seguro funciona correctamente.

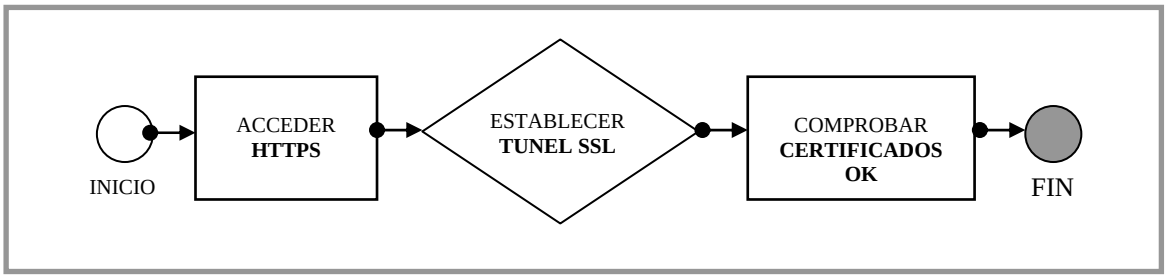


Prueba 2: Será comprobar si Apache redirige, de forma automática, el tráfico del puerto 443 (HTTPS) al 80, cuando es necesario (se sale de una zona restringida, es decir, zona del administrador o zona de usuario registrado).

- ✓ Obtenemos que, de forma automática, esta dirección https://localhost/ se redirige a http://localhost/, cuando pulsamos el botón salir en las zonas del administrador o del usuario registrado. De modo que la redirección a tráfico no seguro funciona correctamente.



Prueba 3: Comprobaremos ahora el uso de SSL para cifrar las conexiones Web y que usa el mejor cifrado posible.



✓ En primer lugar, se accede vía HTTPS. Además, se nos muestra el candado en la barra de dirección y en la parte inferior del navegador, como se muestra en la siguiente figura.

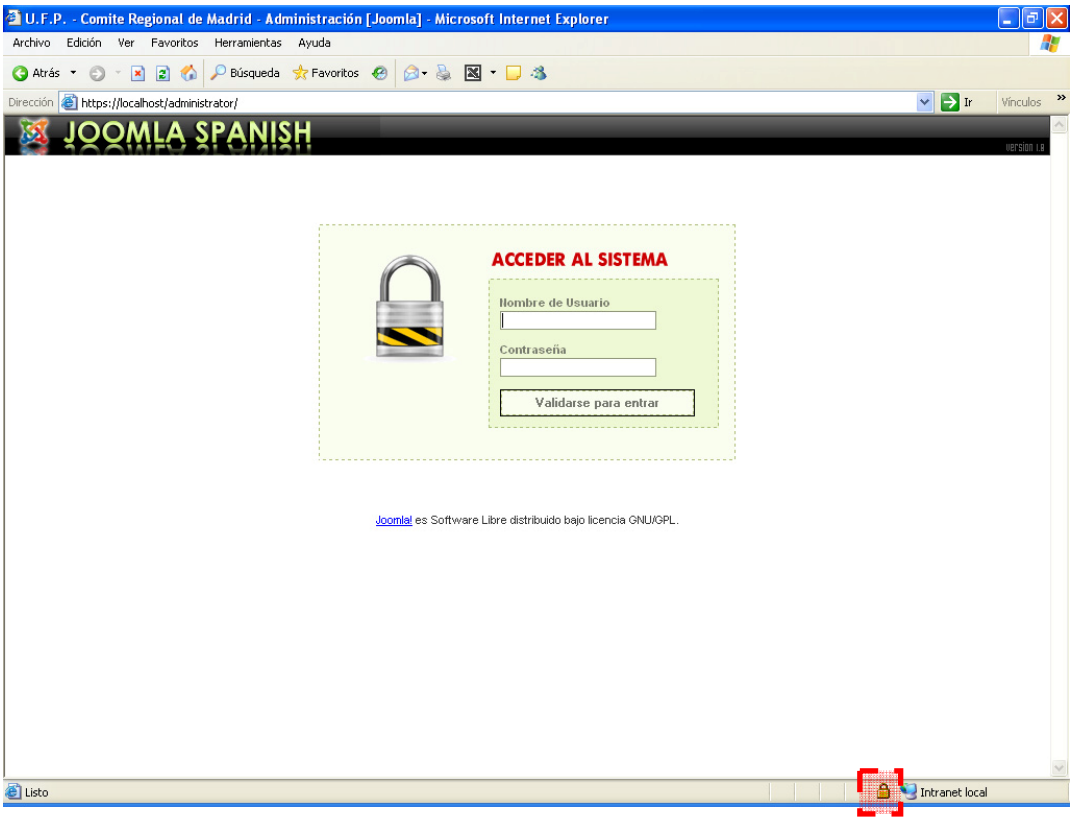


Figura 91: https://localhost/administrator/

- ✓ Además, si pulsamos sobre dicho "candado" obtendremos los datos del certificado.

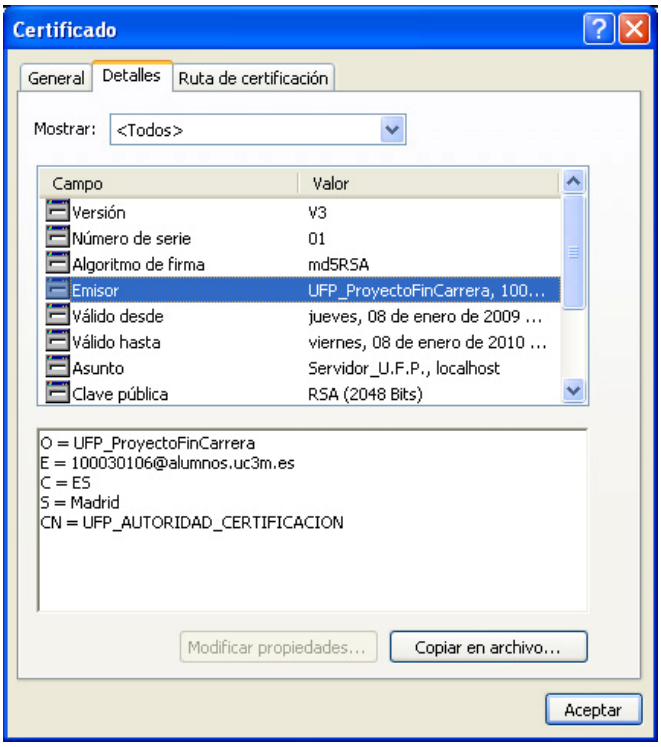
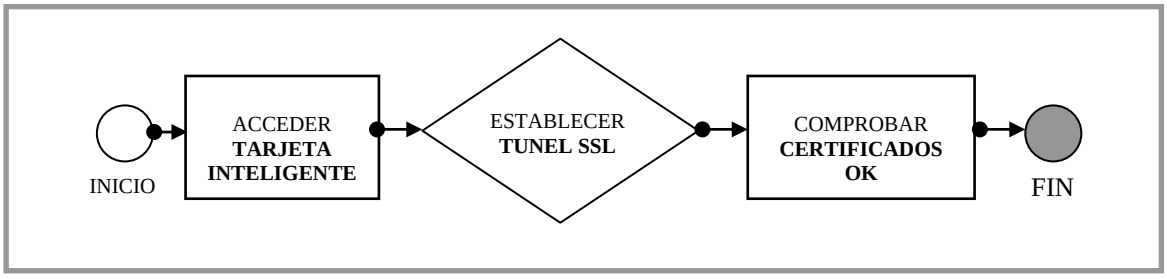


Figura 92: Datos Certificado

Prueba 4: Comprobaremos ahora el uso de SSL para cifrar las conexiones Web, esta vez con acceso mediante tarjetas inteligentes.

- ✓ La conexión segura SSL se realiza correctamente siempre que se tenga la tarjeta metida en el lector y con el certificado de cliente instalado en la misma.



Prueba 5: Comprobar que el firewall no entorpece los servicios del sistema:

- ✓ La navegación se realiza por el contenido de la aplicación de forma adecuada, por lo que el firewall no bloquea el tráfico del servidor.

Prueba 6: Para comprobar que MySQL y PHP funcionan correctamente basta con cargar la página de inicio del portal. En caso de salir todo correctamente, es que funcionan bien, sino, es que hay algún error.

- ✓ Al acceder al portal, obtenemos la página principal sin problema. Esto significa que funcionan ambos adecuadamente.

Prueba 7: Para comprobar que el Servidor Apache funciona correctamente basta con peticionar cualquier recurso que gestione, como cargar la página de inicio del portal. En caso de salir todo correctamente, es que funcionan bien, sino, es que hay algún error.

- ✓ Al acceder al portal, obtenemos la página principal sin problema. Esto significa que funciona adecuadamente. Podemos parar el servicio de Apache y comprobar como ahora nos es imposible acceder al portal JOOMLA.

6.2. - TESTING DE LA APLICACIÓN WEB

Para las aplicaciones no vamos a exponer la prueba y resultado, ya que fueron todos satisfactorios. Por tanto expondremos una lista de acciones acometidas con éxito (todas cuantas se probaron):

Prueba 1: Navegación por el portal y el foro.

- ✓ Comprobamos que se puede navegar perfectamente por las diferentes aplicaciones web implementadas en el sistema. Esto demuestra también que los servidores funcionan adecuadamente y que las medidas de seguridad no entorpecen el tráfico.

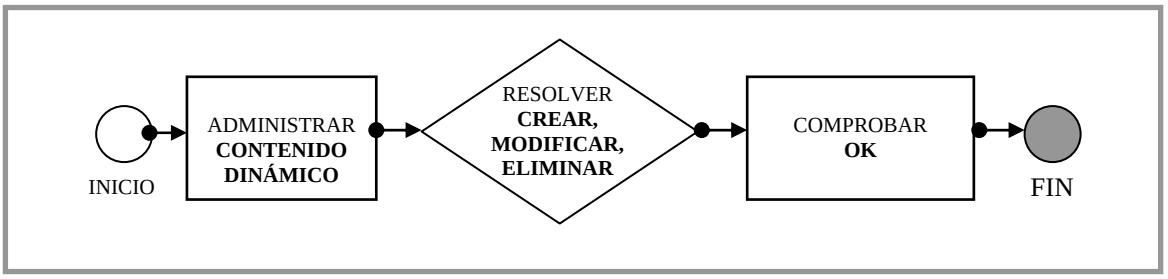
Prueba 2: Registro de usuarios.

- ✓ Comprobamos que podemos registrar usuarios, y que los mismos son dados de alta tanto en el portal como en el foro, siendo estos usuarios válidos en todo el conjunto de la aplicación, quedando perfectamente integrado.

- ✓ Comprobamos, también, que se cumplen las reglas de registro de usuarios, tales como que no haya e-mails de usuarios repetidos en el sistema, que nadie tiene nombres de usuario prohibidos (como admin, por ejemplo).

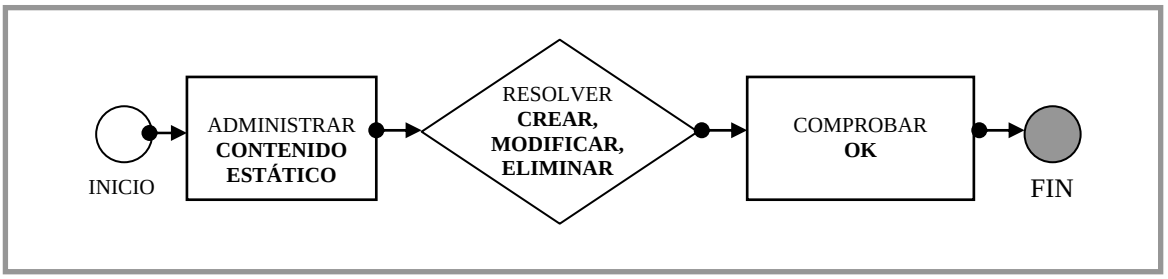
Prueba 3: Comprobaciones sobre el contenido dinámico.

- ✓ Comprueba que el administrador, pueden crear, modificar y eliminar, el contenido dinámico presente en el portal.



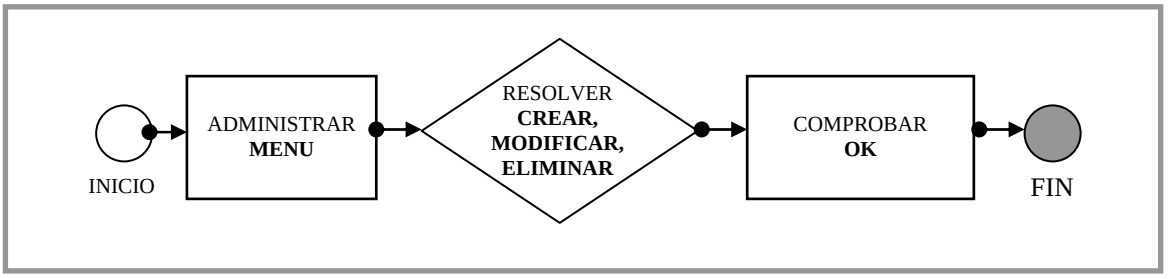
Prueba 4: Contenido estático del portal.

- ✓ Comprueba que el administrador, pueden crear, modificar y eliminar, el contenido estático presente en el portal.



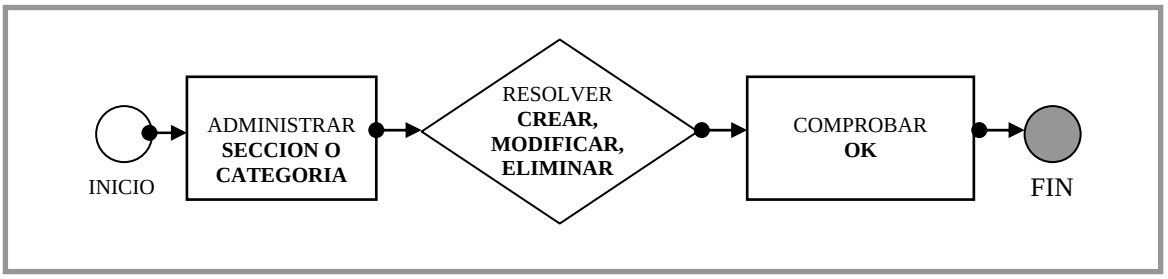
Prueba 5: Menús del portal.

- ✓ Comprobamos que el administrador puede generar todo tipo de menús (los definidos en su correspondiente sección), hacer modificaciones sobre ellos, cambiar sus posiciones y definir la visibilidad.



Prueba 6: Crear secciones y categorías.

- ✓ Comprobamos que el administrador puede manejar las diferentes secciones del portal (crear, eliminar y modificar) y también que puede gestionar las distintas categorías adscritas a alguna de las secciones existentes. En caso de no haber secciones, no se permite generar nuevas categorías.

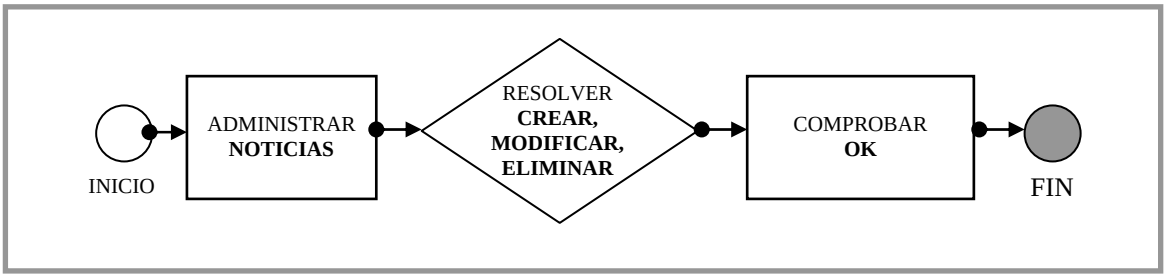


Prueba 7: Distribución del contenido.

- ✓ Comprobamos que el administrador puede decidir qué contenido aparece en la página principal y también la disposición que cada elemento tendrá en dicha página.

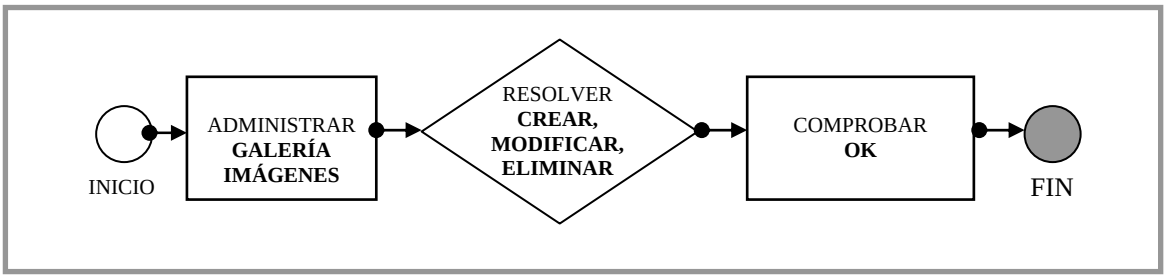
Prueba 8: Noticias.

- ✓ Comprobamos que el administrador puede generar todo tipo de notocias (los definidos en su correspondiente sección), hacer modificaciones sobre ellos, cambiar sus posiciones y definir la visibilidad.



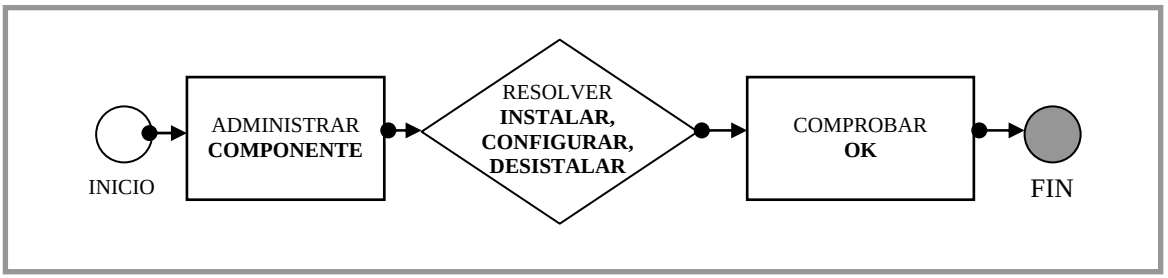
Prueba 9: Galería de imágenes.

- ✓ Comprobamos que los usuarios pueden ver las diferentes imágenes de las galerías.
- ✓ Comprobamos también que el administrador puede gestionar el contenido de dichas galerías.



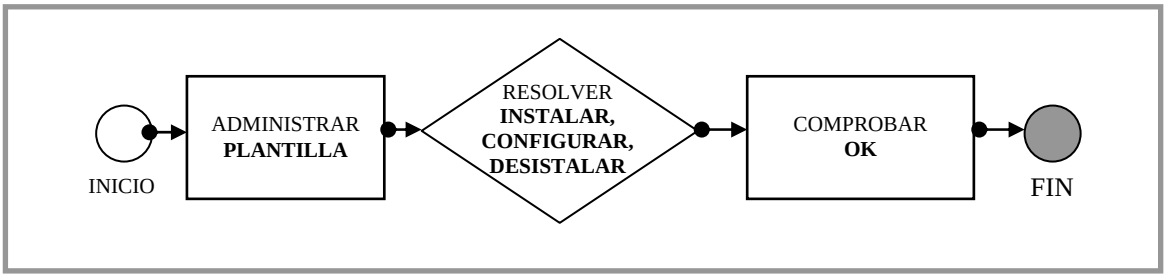
Prueba 10: Componentes del portal.

- ✓ Comprobamos que el administrador puede instalar, desinstalar y configurar todos los módulos del portal.
- ✓ Además, comprobamos que, una vez instalados y configurados, son funcionales.
- ✓ En caso de desinstalar un componente, se comprueba que verdaderamente deja de estar presente en el sistema.



Prueba 11: Cambiar la apariencia del portal.

- ✓ Comprobamos que el administrador del portal puede administrar la apariencia de la aplicación con nuevas plantillas de diseño.

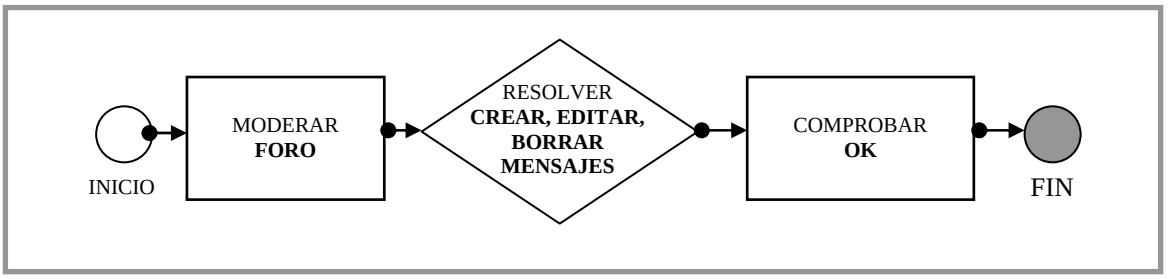


Prueba 12: Escribir mensajes en el foro.

- ✓ Comprobamos que, una vez un usuario está registrado en el foro, puede crear comentarios.

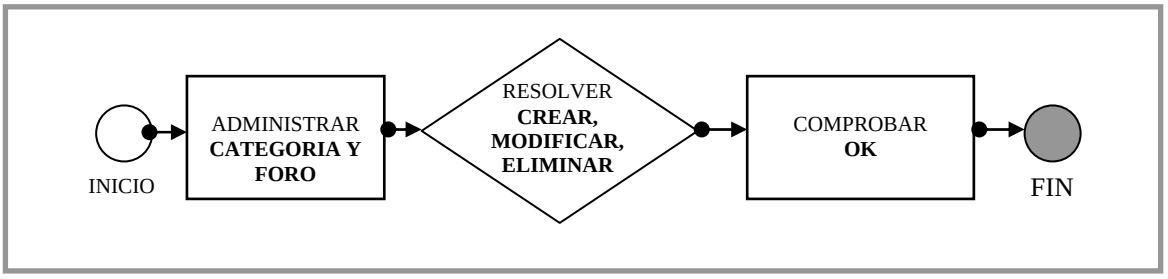
Prueba 13: Moderar mensajes.

- ✓ Comprobamos que el administrador del portal tiene capacidad para moderar los foros, y puede editar o borrar cualquier mensaje que esté presente en él.



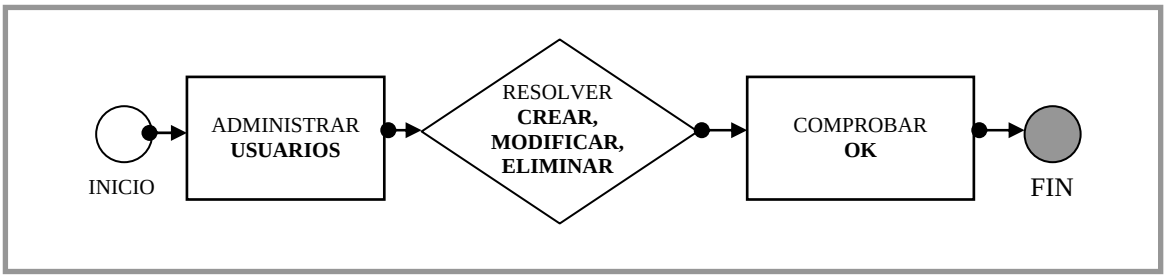
Prueba 14: Crear categorías y foros.

- ✓ Comprobamos que el administrador puede administrar las categorías y los foros.
- ✓ Puede, también, cambiar la disposición jerárquica existente entre las distintas categorías y sus foros.



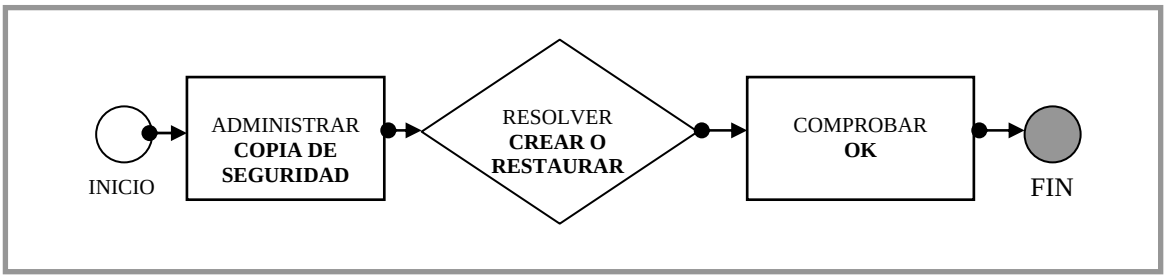
Prueba 15: Administración de usuarios.

- ✓ Comprobamos que el administrador puede cambiar el perfil con el que se crearán las nuevas cuentas de usuarios.
- ✓ Además, comprobamos que puede asignar el perfil, por defecto, con el que se registrarán nuevos usuarios.



Prueba 16: Hacer copias de seguridad.

- ✓ Comprobamos que se pueden hacer copias de seguridad de la base de datos MySQL y de los contenidos fisicos del portal y del foro.
- ✓ Además, comprobamos que esos contenidos pueden ser restaurados de forma satisfactoria.



7 - CONCLUSIONES Y FUTUROS DESARROLLOS

7.1 - LÍNEAS FUTURAS

Hay muchas posibilidades de añadir funcionalidad al portal, pero eso no serían ampliaciones futuras como tal, ya que no suponen ningún cambio de filosofia con respecto a lo que se ha planteado hasta ahora, puesto que siempre se ha intentado dar una configuración y proponer alternativas para futuros usos.

Por tanto, y aunque se pudiera, no vamos a contemplar el añadir nuevos componente al portal como líneas futuras de desarrollo, vamos a intentar ir un poco más allá para buscar elementos de mayor calado, que realmente aportasen algo nuevo de verdad al sistema.

7.1.1 - ACCESIBILIDAD DE CONTENIDOS

Puesto que los contenidos, tanto del portal como del foro, están almacenados en la base de datos MySQL, podría implementarse una nueva página web en PHP (una subcategoría del portal por decirlo de una forma más exacta) cuyo funcionamiento fuera el que a continuación se describe.

- ✓ La nueva aplicación web lo que haría sería crear un menú en texto (HTML lo más plano posible), emulando al menú de contenidos del portal.
- ✓ Cada uno de esos encales HTML, lo que haría sería enlazar contenido PHP, cuya función sería leer el contenido de la base de datos, para mostrarlo en formato de texto al usuario.
- ✓ Siguiendo esta filosofia de reducción de contenido visual, podría implementarse una nueva aplicación que podría ser usada por gente, por ejemplo, con discapacidades visuales.
- ✓ Esto se podría extender de forma similar al foro, ya que los mensajes se almacenan en MySQL de forma similar.
- ✓ Este sistema podría ponerse en marcha por medio de una página alternativa, mediante una subpágina de la página principal usando suscripciones.
- ✓ Este sistema de suscripción consistiría en usar Apache para dirigir ese nuevo tráfico a los hosts especificados (los suscritos) mientras que al resto se les seguiría mostrando el contenido normal de la aplicación.
- ✓ Como colofón final una aplicación instalada en el ordenador local del cliente podría interpretar esta web alternativa e ir leyendo dichos caracteres de texto plano a la persona con problemas visuales.

7.1.2 - CONFIGURACIÓN DE UN SERVIDOR DE CORREO CON SENDMAIL

El CMS usado (JOOMLA) permite la activación de cuentas mediante el acceso a un link determinado que se envía al correo del que solicita la cuenta.

Nosotros no disponemos de una configuración para Sendmail en nuestro sistema, por lo que los registros son inmediatos. Esto es un peligro para la seguridad, ya que los bots podrían saturar el sistema de falsos registros. Dicho problema de seguridad no afecta a nuestro portal, ya que el administrador es el que crea las cuentas a los usuarios registrados (requisito del cliente U.F.P. Unión Federal de policía).

Para ello, habría que configurar un servidor Sendmail de SMTP para que nuestro sistema pudiera enviar correos y hacer uso del mencionado servicio.

Por tanto, se contempla esta futura línea de implementación para mejorar la seguridad del sistema, la configuración de un servidor de correo SMTP con Sendmail, ya que es el que usa JOOMLA.

7.2 - CONCLUSIONES

Se puede decir que los CMS son unas herramientas potentes, sencillas y que permiten la generación y gestión de contenidos de una forma rápida y profesional. Por tanto, los CMS (como JOOMLA) son aplicaciones de un gran interés no sólo para aquellos usuarios poco experimentados con el mundo de la programación de aplicaciones Web, sino también para aquellas personas que necesitan tener resultados en un corto espacio de tiempo, sin por ello renunciar a unos mínimos de calidad exigibles.

Puesto que el contenido de este proyecto va dirigido al Sindicato de Policía U.F.P., que será el que, en un futuro, use y gestione el portal y el foro implementados, hace que sea mucho más valioso el uso de este tipo de herramientas, ya que a una organización de estas características les sería muy dificil gestionar otro tipo de aplicación Web (al menos una que ofreciera una funcionalidad similar), puesto que los conocimientos informáticos que serían necesarios excederían lo que la mayoría de las personas tienen. Por tanto, podemos llegar a la conclusión de que el uso de estas herramientas es una opción no sólo válida para este caso, sino acertadísima.

Otra conclusión importante a la que se ha llegado durante el desarrollo de este proyecto, es la importancia de la seguridad en el ámbito de las aplicaciones Web. En un caso como el que tratamos, con bases de datos que podrían contener información sensible, se hace indispensable el uso de toda la seguridad que esté a nuestro alcance, por lo que se ha usado SSL intentando crear configuraciones con un nivel de seguridad elevado.

El protocolo SSL, por medio del uso de los diferentes algoritmos de encritpación y autenticación, nos brinda una plataforma ideal para así poder dar solución al grave problema que es mantener la confidencialidad de los datos en los trasiegos de información entre cliente y servidor.

Otra conclusión importante es ver la vulnerabilidad de los servidores que albergan estas aplicaciones. Un ataque podría hacer mucho daño en un servidor de estas características. Por tanto, no sólo se necesita seguridad en las comunicaciones, sino que se necesita una seguridad física de mayor envergadura que evite, en la medida de lo posible, los ataques malintencionados contra el servidor. Se ha tratado desde Apache, haciendo uso de configuraciones que no permitan la saturación del servidor.

Para concluir, se puede decir, que este proyecto ha servido para enfocar todos los problemas que se plantean al hacer uso de tecnologías Web y el uso de servidores de información y cómo podemos enfocar las posibles soluciones.

BIBLIOGRAFÍA

1)	**Título:**	Desarrollo Web con PHP, Apache y MySQL.
	Publicación:	Madrid: Anaya Multimedia, [2004] (imp. 2007).
	Autor:	Glass, Michael.

2)	**Título:**	La biblia de Servidor Apache 2.
	Publicación:	Madrid: Anaya Multimedia, [2003].
	Autor:	Kabir, Mohammed J.

3)	**Sitio Web:**	Sitio Web Oficial del Servidor Apache.
	URL:	http://httpd.apache.org/

4)	**Sitio Web:**	Sitio Web Oficial del Paquete AppServ (Apache + MySQL + PHP).
	URL:	http://www.appservnetwork.com/

5)	**Título:**	Network security with OpenSSL.
	Publicación:	Beijing [etc.]: O'Reilly, 2002.
	Autor:	Viega, John.

6)	**Sitio Web:**	Sitio Web Oficial de OpenSSL.
	URL:	http://www.openssl.org/

7)	**Título:**	Domine PHP y MySQL: (programación dinámica en el lado del servidor).
	Publicación:	Paracuellos del Jarama (Madrid): Ra-Ma, [2006].
	Autor:	López Quijado, José.

8)	**Sitio Web:**	Sitio Web Oficial del Sistema Gestor de Bases de Datos MySQL.
	URL:	http://www.mysql.com/

9)	**Sitio Web:**	Sitio Web Oficial del Lenguaje de Programación PHP.
	URL:	http://www.php.net/

10)	**Sitio Web:**	Sitio Web Oficial de PhpMyAdmin.
	URL:	http://www.phpmyadmin.net/

11)	**Título:**	Building websites with Joomla!
	Publicación:	Birmingham: Packt, 2006.
	Autor:	Graf, Hagen.

12)	**Sitio Web:**	Sitio Web Oficial del C.M.S. JOOMLA.
	URL:	http://www.joomla.org/

13)	**Sitio Web:**	Sitio Web Oficial de JOOMLA (en idioma castellano).
	URL:	http://www.joomlaspanish.org/

14)	**Título:**	Tarjetas inteligentes.
	Publicación:	Madrid: Paraninfo, [1999].
	Autor:	Sandoval González, Juan Domingo.

15)	**Sitio Web:**	Sitio Web Oficial de La Fábrica Nacional de Moneda y Timbre - (Cryptokit).
	URL:	http://www.cert.fnmt.es

APÉNDICES

APÉNDICE “A” - PLANIFICACIÓN Y PRESUPUESTO

En el diagrama de Project que se adjunta, se puede ver la división en tiempos y tareas que se ha hecho de la implementación desde cero, que comenzó el 17 de Diciembre de 2008.

Hay, básicamente, ocho fases, dieciocho subfases (fases y subfases se describirán a continuación) y cinco hitos, los cuales marcan el inicio y fin del proyecto y tres momentos importantes durante el desarrollo.

A.1 - DIAGRAMA DE GANTT

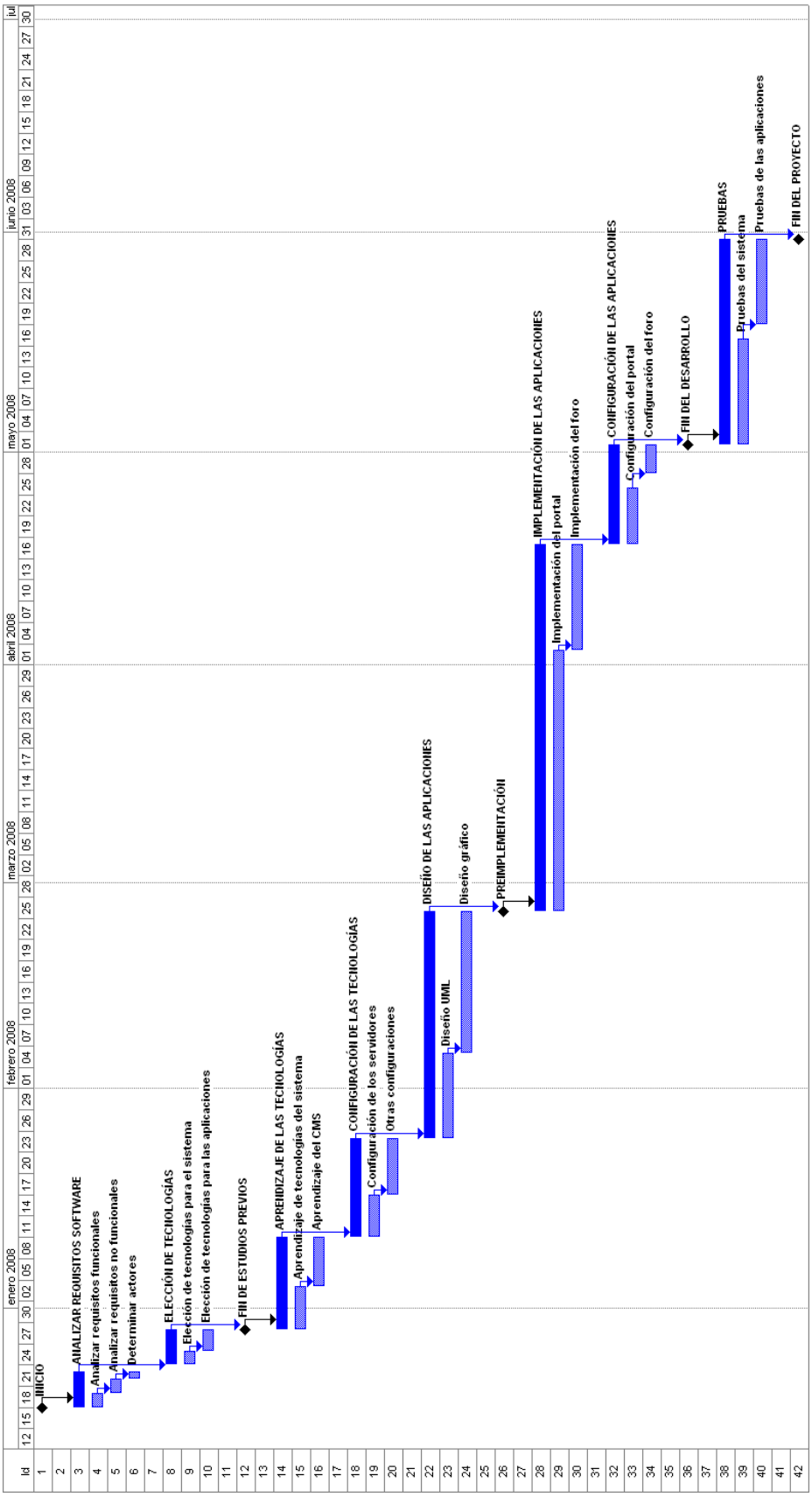


Figura 93: Diagrama de GANTT del proyecto

A.2 - ESTUDIO DEL DIAGRAMA DE GANTT

- INICIO: Es el hito de comienzo del proyecto, no representa nada más que el símbolo de iniciar el trabajo.

- ANALIZAR REQUISITOS SOFTWARE: En esta fase se analizaron los requerimientos del Sindicato del Cuerpo Nacional de Policía U.F.P. para determinar qué quieren realizar exactamente.

  - ✓ Analizar requisitos funcionales: Aquí se hará el estudio sobre las necesidades "funcionales" que necesitan. Todo aquello que sea motivo de ejecutarse en la aplicación debe estar aquí detallado.

  - ✓ Analizar requisitos no funcionales: Aquí se analizan todas las necesidades que van a hacer que los requisitos funcionales se puedan ejecutar. Aspectos como la organización, la seguridad y temas similares son los que se abordarán aquí.

  - ✓ Determinar actores: Una vez ya se sabe lo que se necesita implementar, necesitamos saber quién podrá hacer qué cosa. Para eso, debemos hacer un estudio que determine los actores del sistema y sus funciones.

- ELECCIÓN DE LAS TECNOLOGÍAS: En esta fase se ha analizado qué herramientas se iban a usar para acometer el proyecto, herramientas de configuración del sistema y herramientas para implementar las aplicaciones en sí.

  - ✓ Elección de tecnologías para el sistema: Aquí se trató el apartado de qué elementos usar para configurar un sistema que pudiera albergar de forma segura servidores y aplicaciones web. Por tanto, se escogieron: el sistema operativo a usar, los distintos tipos de servidores, el gestor de bases de datos y todo sistema necesario para crear la base del proyecto.

  - ✓ Elección de tecnologías para las aplicaciones: En esta subfase se trató que tecnologías usar para la creación de las aplicaciones. Finalmente, se decidió el uso de CMS sobre todo por su relación calidad de la aplicación / tiempo de desarrollo y por el cliente al que iba encaminada dicha aplicación (un Sindicato del Cuerpo Nacional de Policía).

- FIN DE ESTUDIOS PREVIOS: Este hito marca el fin de las consideraciones previas del proyecto y marca el inicio del auténtico proceso de desarrollo.

- APRENDIZAJE DE LAS TECNOLOGÍAS: En esta fase se hizo la recopilación y estudio del material referente al uso de todas las tecnologías que iban a ser usadas para el desarrollo del proyecto (tanto para el sistema como para las aplicaciones).

- ✓ Aprendizaje de tecnologías del sistema: En este apartado se realizó el aprendizaje de todas las tecnologías necesarias para configurar el sistema. Puesto que se tenía un conocimiento previo de la mayoría de ellas, sólo fue necesario realizar el aprendizaje de generación de espacios seguros con SSL.

- ✓ Aprendizaje de CMS: En esta subfase se acometió el familiarizarse con los CMS JOOMLA. El aprender a usar estos CMS es relativamente sencillo para gente habituada a manejar distintos sistemas informáticos y, además, la documentación oficial disponible en los diferentes sitios oficiales, en gran medida, a realizar un aprendizaje rápido.

- CONFIGURACIÓN DE LAS TECNOLOGÍAS: En esta fase se trató todo lo relacionado con la configuración del sistema (nada relacionado con las aplicaciones web).

  - ✓ Configuración de los servidores: Aquí se acometió la configuración del servidor web (Apache) acorde a las necesidades planteadas. Teniendo especial cuidado de la configuración de cara a la seguridad.

  - ✓ Otras configuraciones: En este apartado se realizó el resto de configuraciones no relacionadas con los servidores en sí, tales como configurar autoridades de certificación, protocolo SSL o el sistema gestor de bases de datos.

- DISEÑO DE LAS APLICACIONES: En esta fase se acometió el diseño de las aplicaciones mediante el uso de distintos tipos de diagramas y finalmente la realización del diseño gráfico de éstas.

  - ✓ Diseño UML: En esta subfase se crearon los diagramas UML necesarios. Estos diagramas permitieron ver de forma visual cómo interactuaban los actores con las funcionalidades del sistema, facilitando las tareas de división de tareas en la implementación de las aplicaciones.

  - ✓ Diseño gráfico: En esta sección se realizó un diseño gráfico de cómo deberían quedar las aplicaciones cumpliendo con todos los requerimientos planteados. Esto es útil para poder tener una idea del resultado antes de la implementación, cosa que facilitará mucho dicha implementación.

- PREIMPLEMENTACIÓN: Este hito marca el final del proceso de diseño de la aplicación, de manera que todo lo que se haga a continuación tenga que ver con la implementación de las aplicaciones.

- IMPLEMENTACIÓN DE LAS APLICACIONES: En esta fase se acometió la implementación del portal y el foro, mediante el CMS JOOMLA (el auténtico desarrollo físico de las aplicaciones).

- ✓ Implementación del portal: En esta subfase se trató el desarrollo del portal con el gestor de contenidos JOOMLA. Se trata más bien de una implementación básica dejándolo todo listo, pero sin profundizar demasiado en configuraciones específicas

- ✓ Implementación del foro: En esta fase se trató la implementación del foro web con el uso del módulo de JOOMLA "FireBoard", que permite una integración perfecta entre portal y foro.

- CONFIGURACIÓN DE LAS APLICACIONES: En esta fase se acometió la configuración exhaustiva del portal y del foro para que las aplicaciones cumplieran con lo analizado en las primeras fases del proyecto.

  - ✓ Configuración del portal: Aquí se trató la configuración pormenorizada de todos los complementos del portal, que eran necesarios para satisfacer todos los requisitos planteados.

  - ✓ Configuración del foro: En esta fase se trataron pequeños aspectos de configuración del foro. Como es menos complejo que el portal, la mayor parte del trabajo se realizó durante la implementación.

- FIN DEL DESARROLLO: Este hito marca el fin del proceso de desarrollo de las aplicaciones. Hasta aquí ya se ha terminado el trabajo de desarrollo, pero no el proyecto.

- PRUEBAS: En esta fase se hicieron todas las pruebas necesarias para garantizar la funcionalidad de todo lo implementado y configurado.

  - ✓ Pruebas del sistema: En esta subfase se realizaron las pruebas que garantizaron el adecuado funcionamiento de los servidores y los aspectos de seguridad del sistema.

  - ✓ Pruebas de las aplicaciones: Aquí se garantizó el adecuado funcionamiento de todo el proceso típico de uso de las aplicaciones, tratando los temas de usuarios y administradores.

- FIN DEL PROYECTO: Este hito marca la finalización definitiva del proyecto una vez concluidas las pruebas.

A.3 - ESTUDIO DETALLADO DE FASES Y TIEMPOS

Se ha añadido el detalle de las fases, subfases e hitos, destacando los días de duración de cada uno (los hitos son siempre 0) y las fechas de inicio y de fin de cada uno.

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	**INICIO**	0 días	mar 18/12/07	mar 18/12/07	
2					
3	**ANALIZAR REQUISITOS SOFTWARE**	5 días	mar 18/12/07	sáb 22/12/07	1
4	Analizar requisitos funcionales	2 días	mar 18/12/07	mié 19/12/07	
5	Analizar requisitos no funcionales	2 días	jue 20/12/07	vie 21/12/07	4
6	Determinar actores	1 día	sáb 22/12/07	sáb 22/12/07	5
7					
8	**ELECCIÓN DE TECNOLOGÍAS**	5 días	lun 24/12/07	vie 28/12/07	3
9	Elección de tecnologías para el sistema	2 días	lun 24/12/07	mar 25/12/07	
10	Elección de tecnologías para las aplicaciones	3 días	mié 26/12/07	vie 28/12/07	9
11					
12	**FIN DE ESTUDIOS PREVIOS**	0 días	sáb 29/12/07	sáb 29/12/07	8
13					
14	**APRENDIZAJE DE LAS TECNOLOGÍAS**	10 días	sáb 29/12/07	jue 10/01/08	12
15	Aprendizaje de tecnologías del sistema	5 días	sáb 29/12/07	jue 03/01/08	
16	Aprendizaje del CMS	5 días	vie 04/01/08	jue 10/01/08	15
17					
18	**CONFIGURACIÓN DE LAS TECNOLOGÍAS**	10 días	vie 11/01/08	jue 24/01/08	14
19	Configuración de los servidores	4 días	vie 11/01/08	mié 16/01/08	
20	Otras configuraciones	6 días	jue 17/01/08	jue 24/01/08	19
21					
22	**DISEÑO DE LAS APLICACIONES**	22 días	vie 25/01/08	lun 25/02/08	18
23	Diseño UML	8 días	vie 25/01/08	mar 05/02/08	
24	Diseño gráfico	14 días	mié 06/02/08	lun 25/02/08	23
25					
26	**PREIMPLEMENTACIÓN**	0 días	mar 26/02/08	mar 26/02/08	22
27					
28	**IMPLEMENTACIÓN DE LAS APLICACIONES**	38 días	mar 26/02/08	jue 17/04/08	26
29	Implementación del portal	27 días	mar 26/02/08	mié 02/04/08	
30	Implementación del foro	11 días	jue 03/04/08	jue 17/04/08	29
31					
32	**CONFIGURACIÓN DE LAS APLICACIONES**	10 días	vie 18/04/08	jue 01/05/08	28
33	Configuración del portal	6 días	vie 18/04/08	vie 25/04/08	
34	Configuración del foro	4 días	lun 28/04/08	jue 01/05/08	33
35					
36	**FIN DEL DESARROLLO**	0 días	vie 02/05/08	vie 02/05/08	32
37					
38	**PRUEBAS**	21 días	vie 02/05/08	vie 30/05/08	36
39	Pruebas del sistema	11 días	vie 02/05/08	vie 16/05/08	
40	Pruebas de las aplicaciones	10 días	lun 19/05/08	vie 30/05/08	39
41					
42	**FIN DEL PROYECTO**	0 días	sáb 31/05/08	sáb 31/05/08	38

Figura 94: Fases y Tiempos del Diagrama de GANTT

A.4 - PRESUPUESTO

RECURSOS	DURACIÓN		COSTE/HORA (€)	TOTAL SIN IVA (€)	IVA (16%)	TOTAL (€)
	DIAS	HORAS				
HARDWARE:						
Ordenador Portátil	121	968	0,18	174,24	27,88	202,12
Sistema de Almacenamiento	121	968	0,04	38,72	6,2	44,92
Lector de Tarjetas Inteligentes	95	760	0,08	60,8	9,73	70,53
Tarjeta Inteligente	95	760	0,02	15,2	2,43	17,63
SUB-TOTAL (HW):				**288,96**	**46,24**	**335,2**
SOFTWARE:						
Microsoft Windows XP	121	968	0,04	38,72	6,2	44,92
Microsoft Office XP	42	336	0,04	13,44	2,16	15,6
AppServ (Apache + MySQL + PHP)	121	968	0	0	0	0
Cryptokit	95	760	0	0	0	0
OpenSSL	95	760	0	0	0	0
C.M.S. JOOMLA	121	968	0	0	0	0
SUB-TOTAL (SW):				**52,16**	**8,36**	**60,52**
PERFILES:						
Analista-Programador	121	968	16,5	15972	2555,52	18527,52
SUB-TOTAL (Recuros Humanos):				**15972**	**2555,52**	**18527,52**
TOTAL DEL PROYECTO:						**18923,24**

APÉNDICE "B" - INSTALACIÓN APPSERV 2.5.9

Desde la siguiente URL podremos descargar el paquete de herramientas "AppServ", el cual contiene:

- ✓ El Servidor "*Apache*".
- ✓ El Sistema Gestor de Bases de Datos "*MySQL*".
- ✓ La Herramienta de Administración de Bases de Datos "*PhpMyAdmin*".
- ✓ Interprete del Lenguaje PHP.

http://www.appservnetwork.com/index.php?newlang=spanish

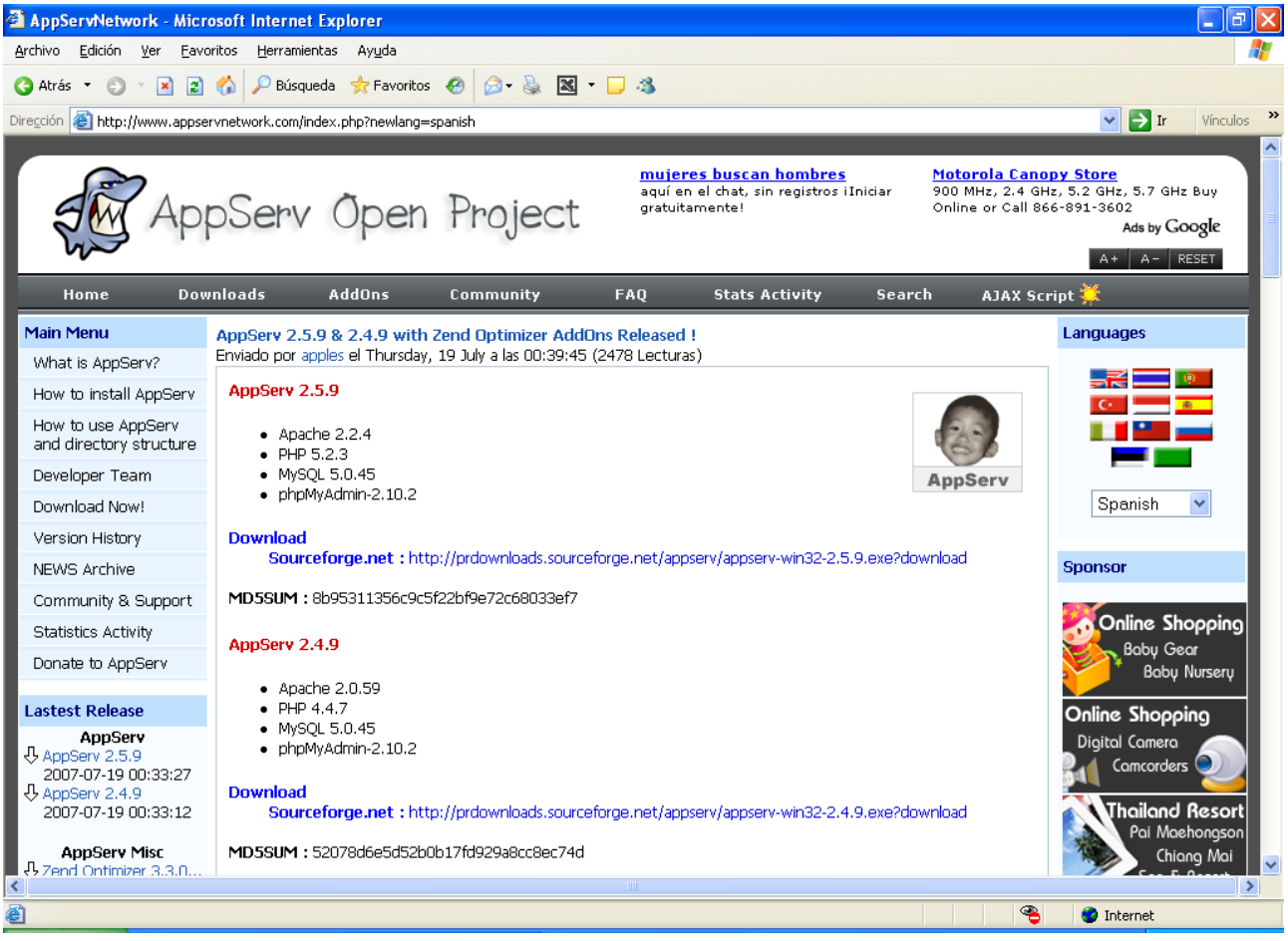


Figura 95: URL "http://www.appservnetwork.com"

Dicha URL nos deriva a otra URL, que se referencia a continuación, desde donde escogeremos el "Mirror" desde el cual descargarnos el software.

http://sourceforge.net/project/downloading.php?groupname=appserv&filename=appserv-win32-2.5.9.exe&use_mirror=puzzle

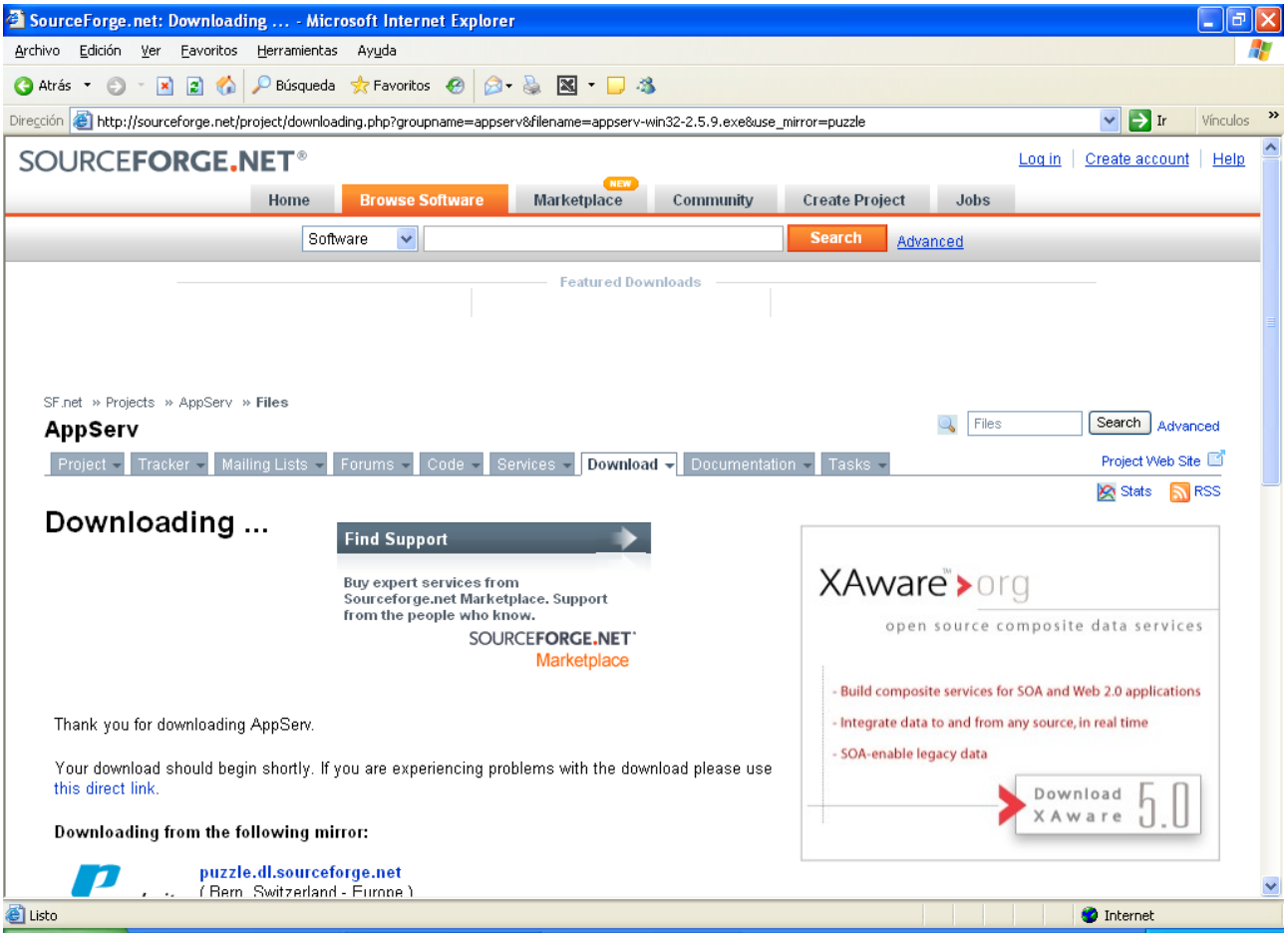


Figura 96: URL "http://sourceforge.net/project/downloading.php"

A continuación y una vez descargado el paquete de instalación: “appserv-win32-2.5.9.exe”, lo ejecutaremos para iniciar la instalación.

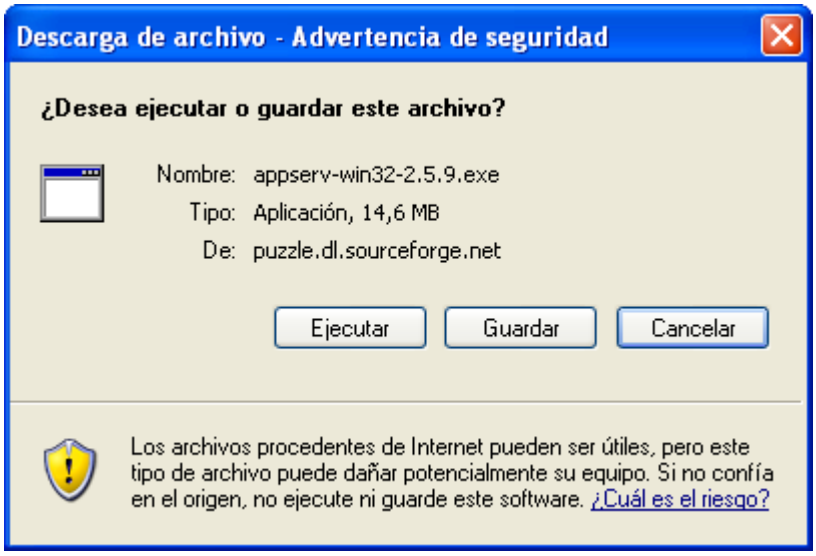


Figura 97: “appserv-win32-2.5.9.exe”

En la siguiente pantalla pulsaremos el botón “*Next*”.

Figura 98: “AppServ 2.5.9 Setup”

En la siguiente pantalla pulsaremos el botón "*I Agree*" tras leer el texto de licencia.

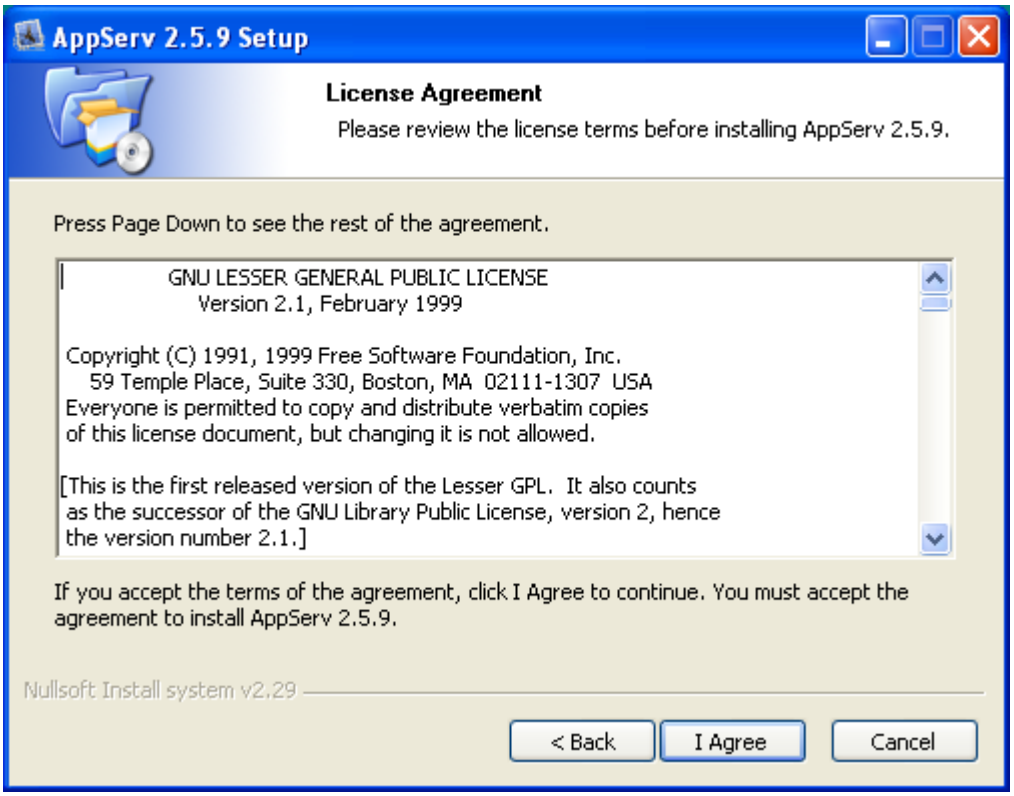


Figura 99: "AppServ 2.5.9 License"

Y a continuación seleccionaremos la ruta de instalación de nuestra herramienta de trabajo.

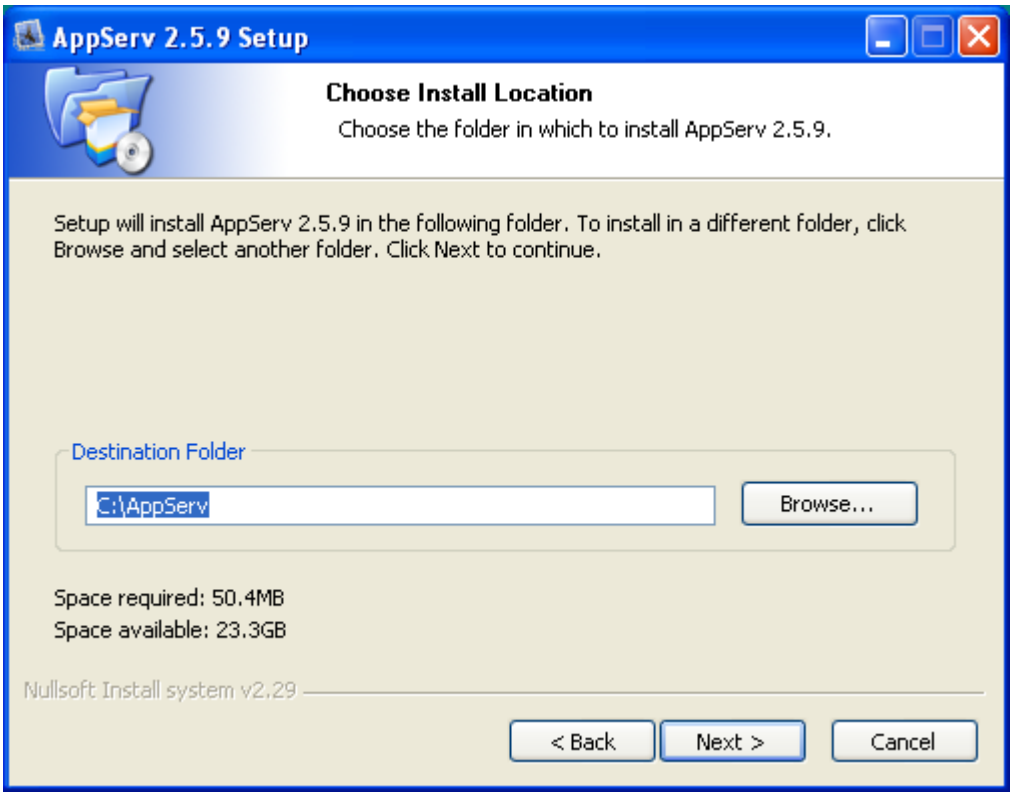


Figura 100: "AppServ 2.5.9 Install Location"

Seleccionamos los componentes que necesitaremos para nuestro proyecto, que en nuestro caso seran todos ellos, y pulsamos "*Next*".

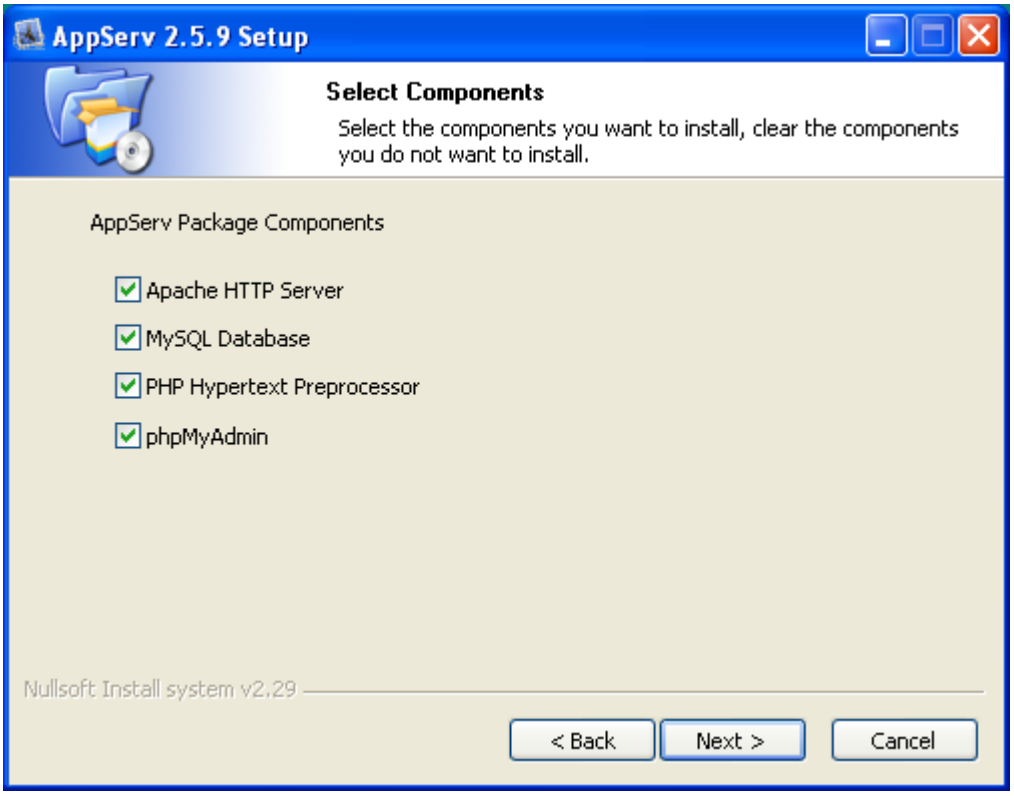


Figura 101: "AppServ 2.5.9 Select Components"

A continuación tendremos que introducir varios parámetros relativos al Servidor "Apache", para ello seleccionamos el nombre del servidor, el email del administrador y el puerto principal de trabajo, y pulsamos "*Next*".

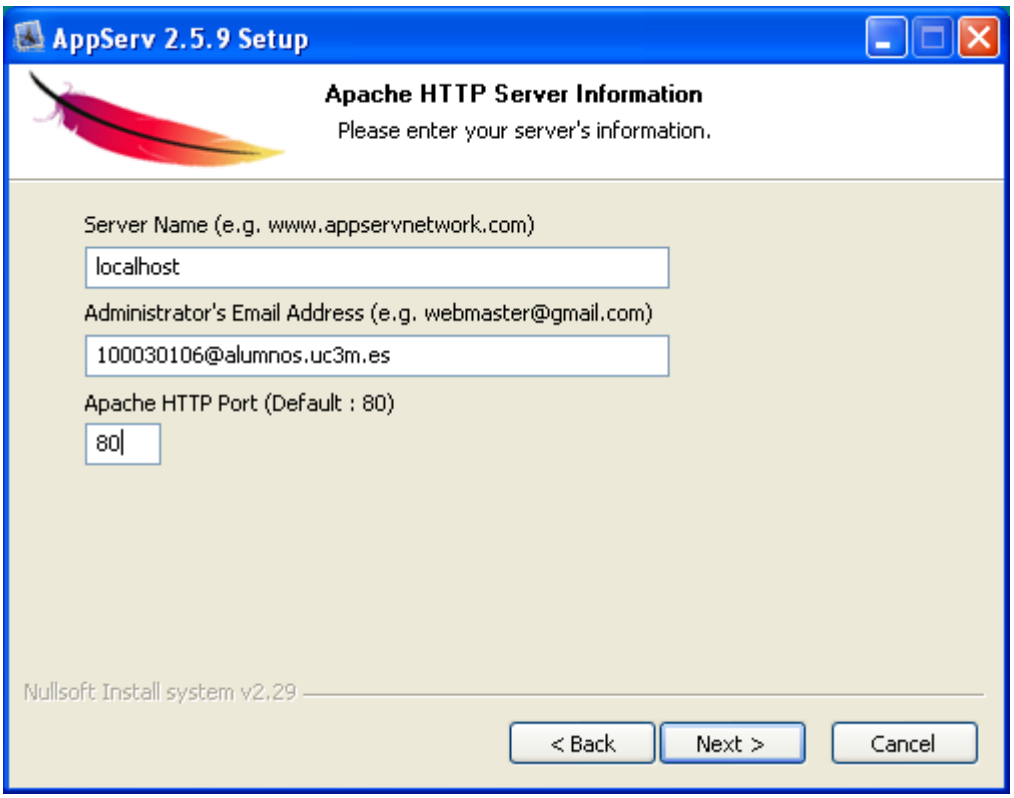


Figura 102: "AppServ 2.5.9 Apache"

Seguidamente tendremos que introducir varios parámetros relativos al Sistema Gestor de Bases de Datos “MySQL”, para ello introducimos el password del usuario “root” de la base de datos, repitiendo dicha operación dos veces, y seleccionamos el formato de carácter a usar, y pulsamos “*Install*”.

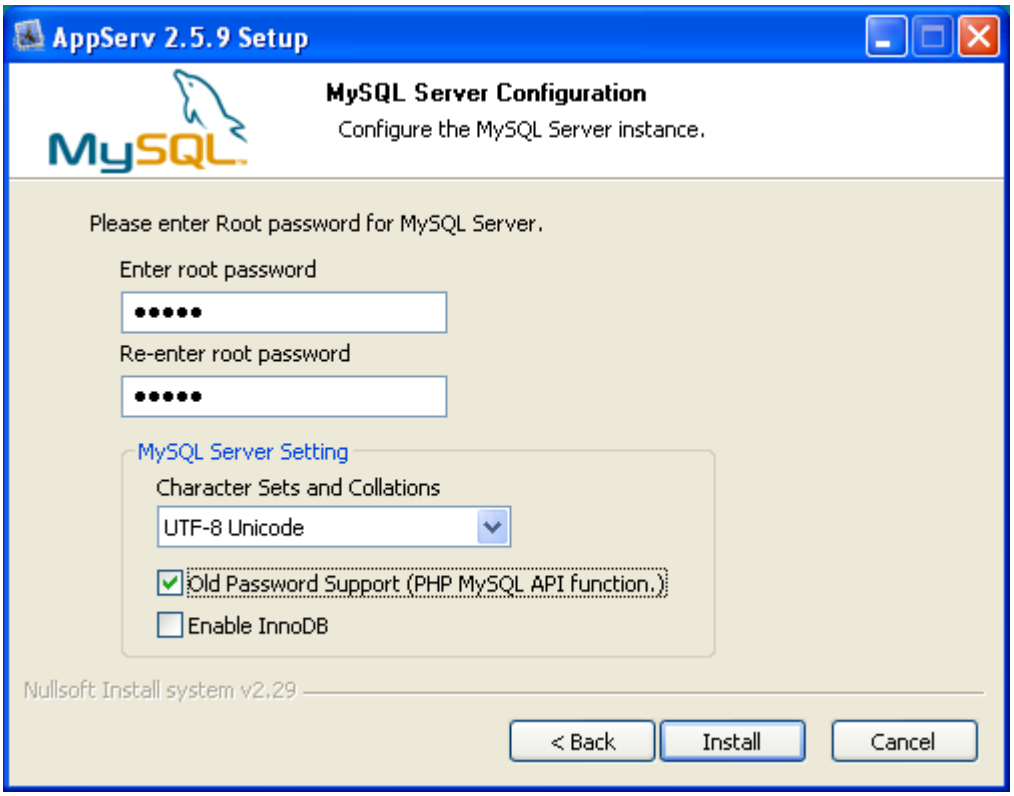


Figura 103: “AppServ 2.5.9 MySQL”

En este momento se lleva acabo la instalación de todos los complementos seleccionados anteriormente, esta operación puede tardar varios minutos.

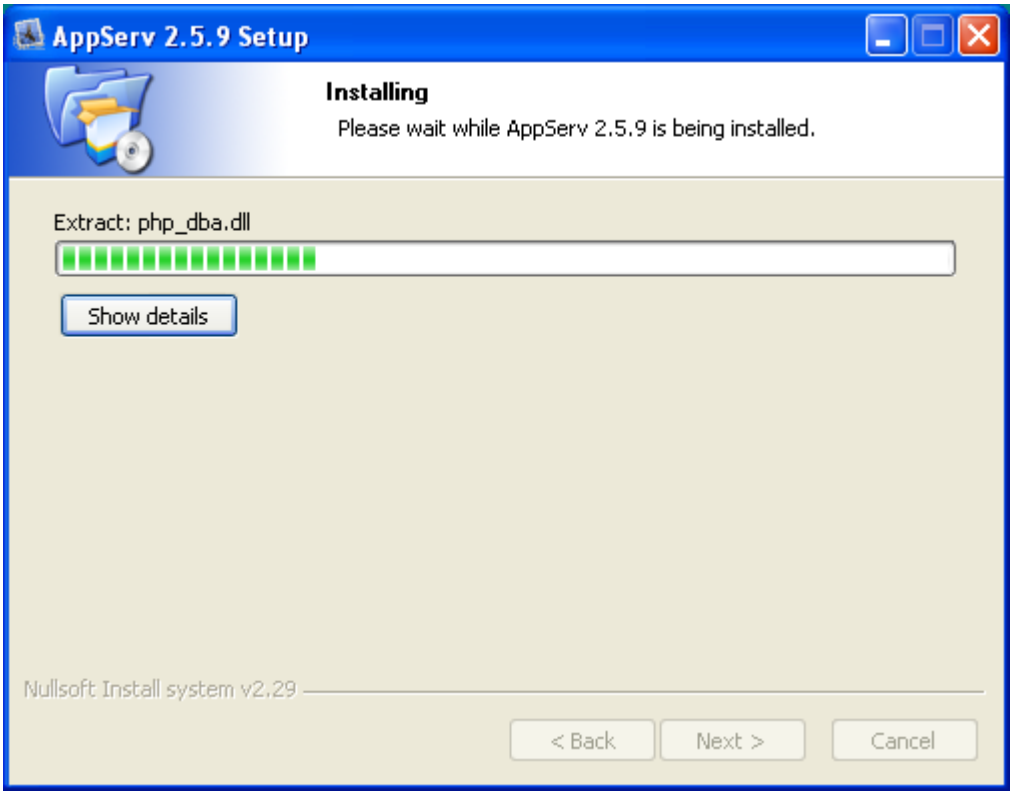


Figura 104: "AppServ 2.5.9 Installing"

Tras la operación mostrada en la anterior figura quedará instalada nuestra herramienta de trabajo “AppServ”.

Figura 105: “AppServ 2.5.9 Completing”

En este momento tendremos nuestra herramienta perfectamente operativa, al escribir en nuestro navegador la URL "http://localhost" se nos mostrará la página por defecto del servidor que acabamos de instalar, dicho documento nos dará información de las versiones de los componentes instalados en los pasos anteriores.

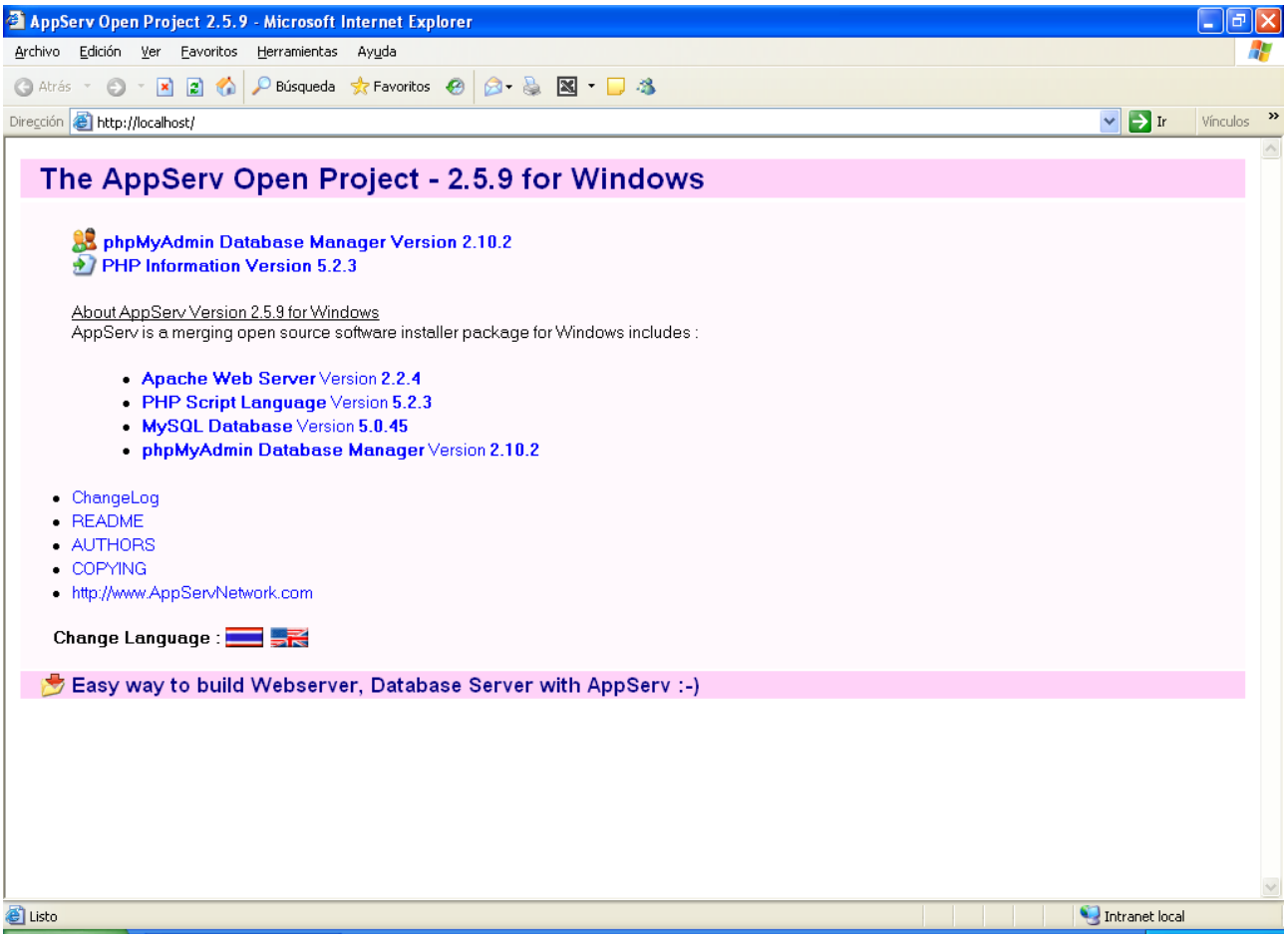


Figura 106: Página por defecto de "http://localhost"

APÉNDICE "C" - INSTALACIÓN JOOMLA 1.0.13

Primero accederemos desde la siguiente URL a la página oficial del proyecto de código abierto CMS JOOMLA en idioma español.

http://www.JOOMLAspanish.org/

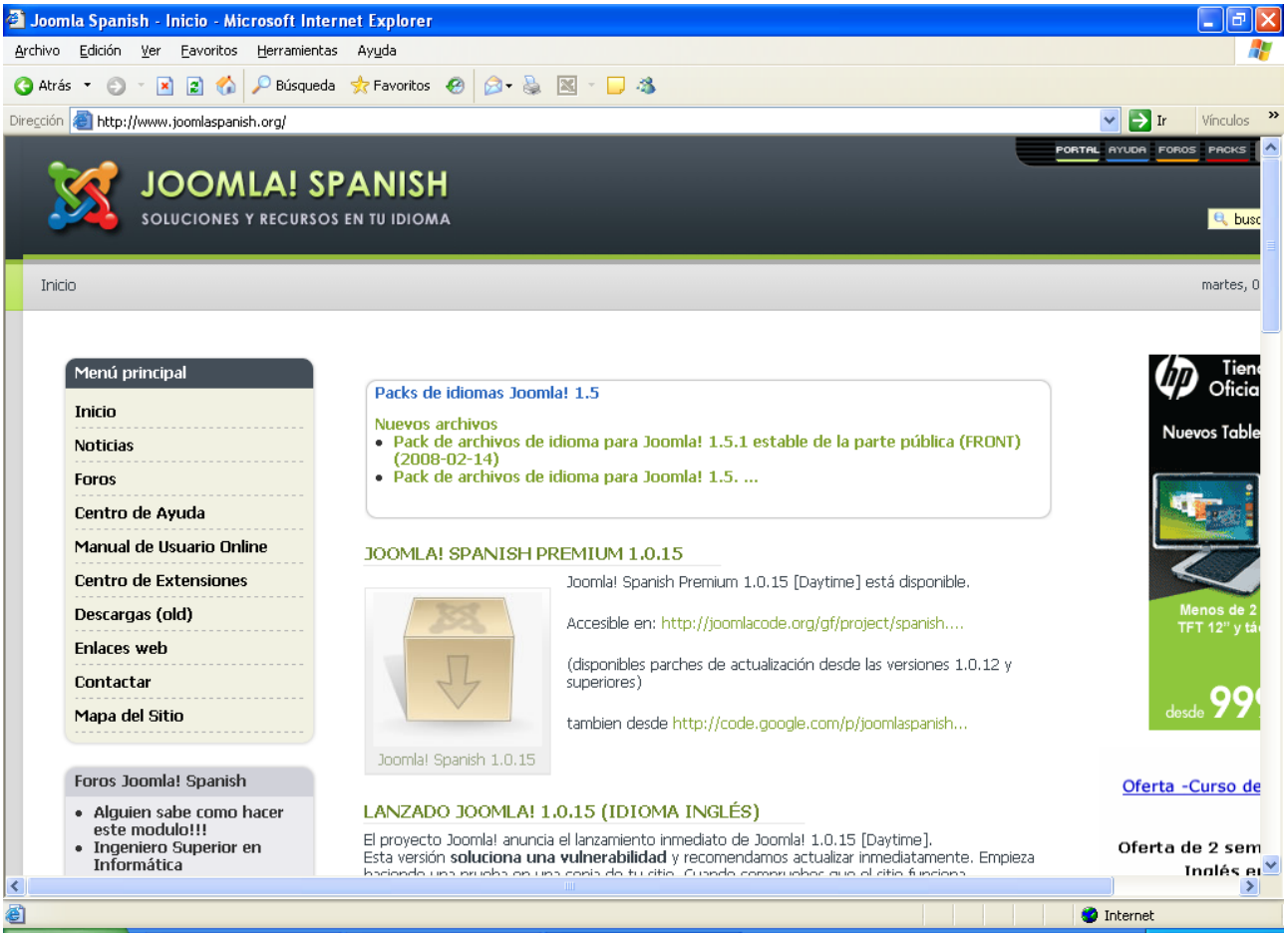


Figura 107: "http://www.JOOMLAspanish.org"

Desde esta página descargamos el paquete de instalación del CMS JOOMLA desde su centro de extensiones:

http://extensiones.JOOMLAspanish.org/

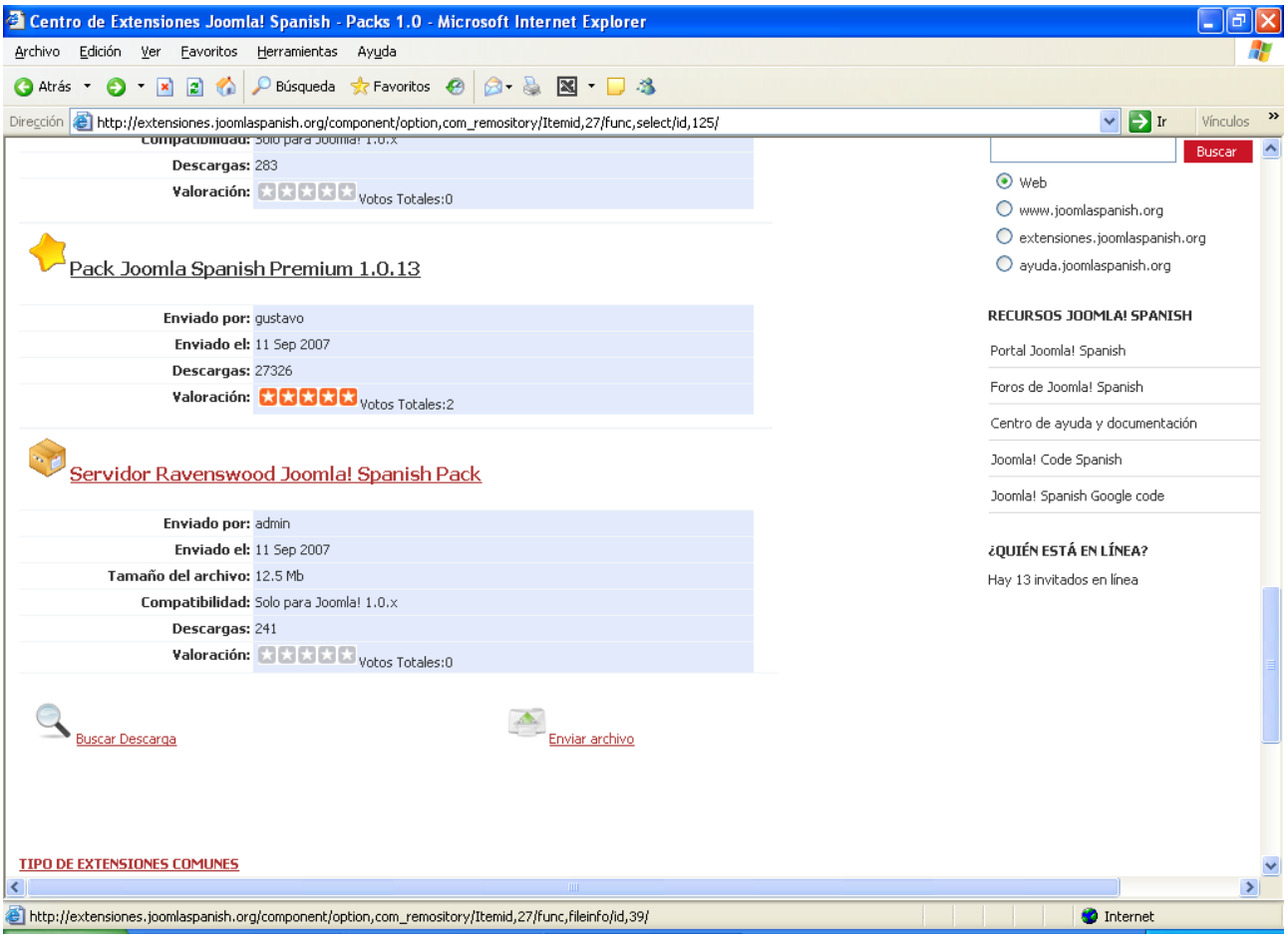


Figura 108: "http://extensiones.JOOMLAspanish.org/"

Seleccionamos la versión, en nuestro caso la 1.0.13 por ser la versión más estable y probada en el momento de inicio del proyecto, y procedemos a descargarla.

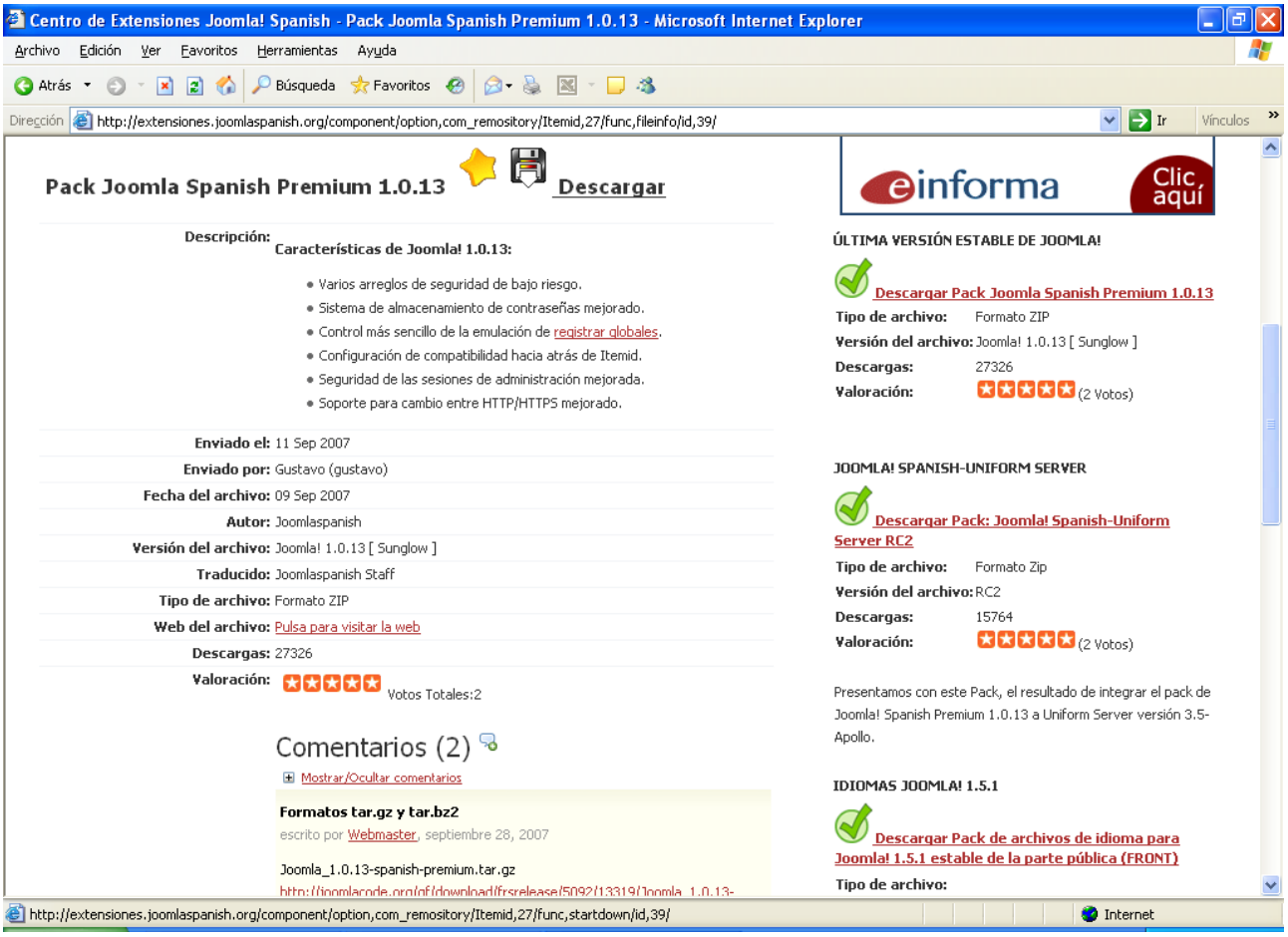


Figura 109: Versión 1.0.13 desde "http://extensiones.JOOMLAspanish.org/"

El contenido de este archivo comprimido en ".ZIP", se descomprime y se sitúa en la carpeta del Servidor AppServ, en la siguiente ruta: "C:\AppServ\www"

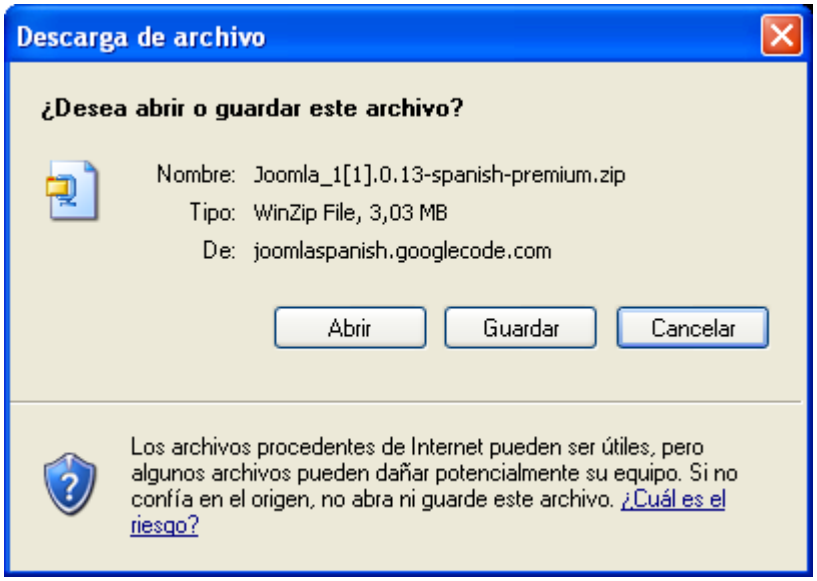


Figura 110: "JOOMLA_1.0.13-spanish-premium.zip"

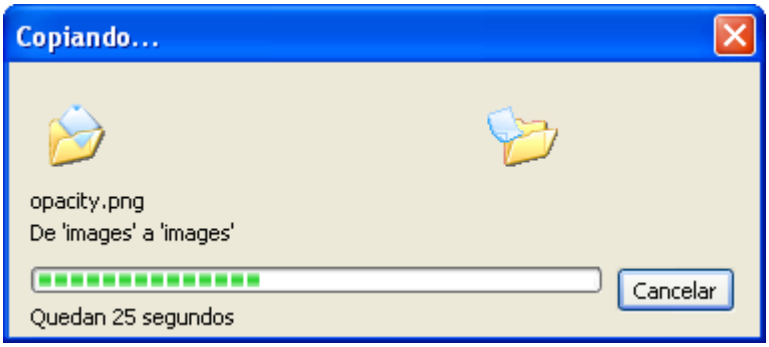


Figura 111: Descomprimiendo "JOOMLA_1.0.13-spanish-premium.zip"

El contenido de esta carpeta quedará tras esta operación como se muestra en la siguiente figura:

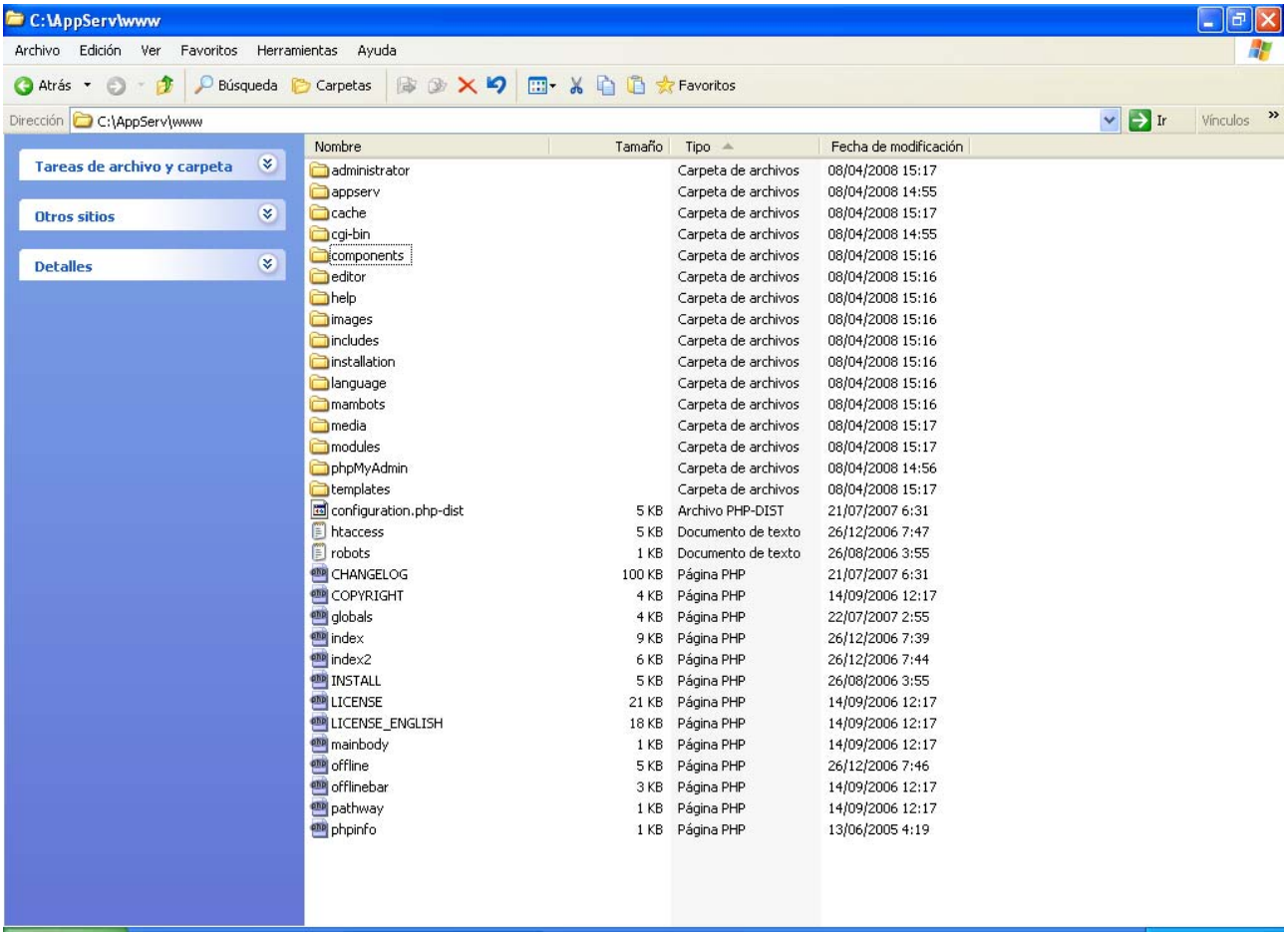


Figura 112: Contenido de "C:\AppServ\www"

La instalación de “JOOMLA” continúa abriendo con un navegador la dirección de “localhost” o en su defecto la IP “127.0.0.1”, como muestra la siguiente figura:

http://localhost

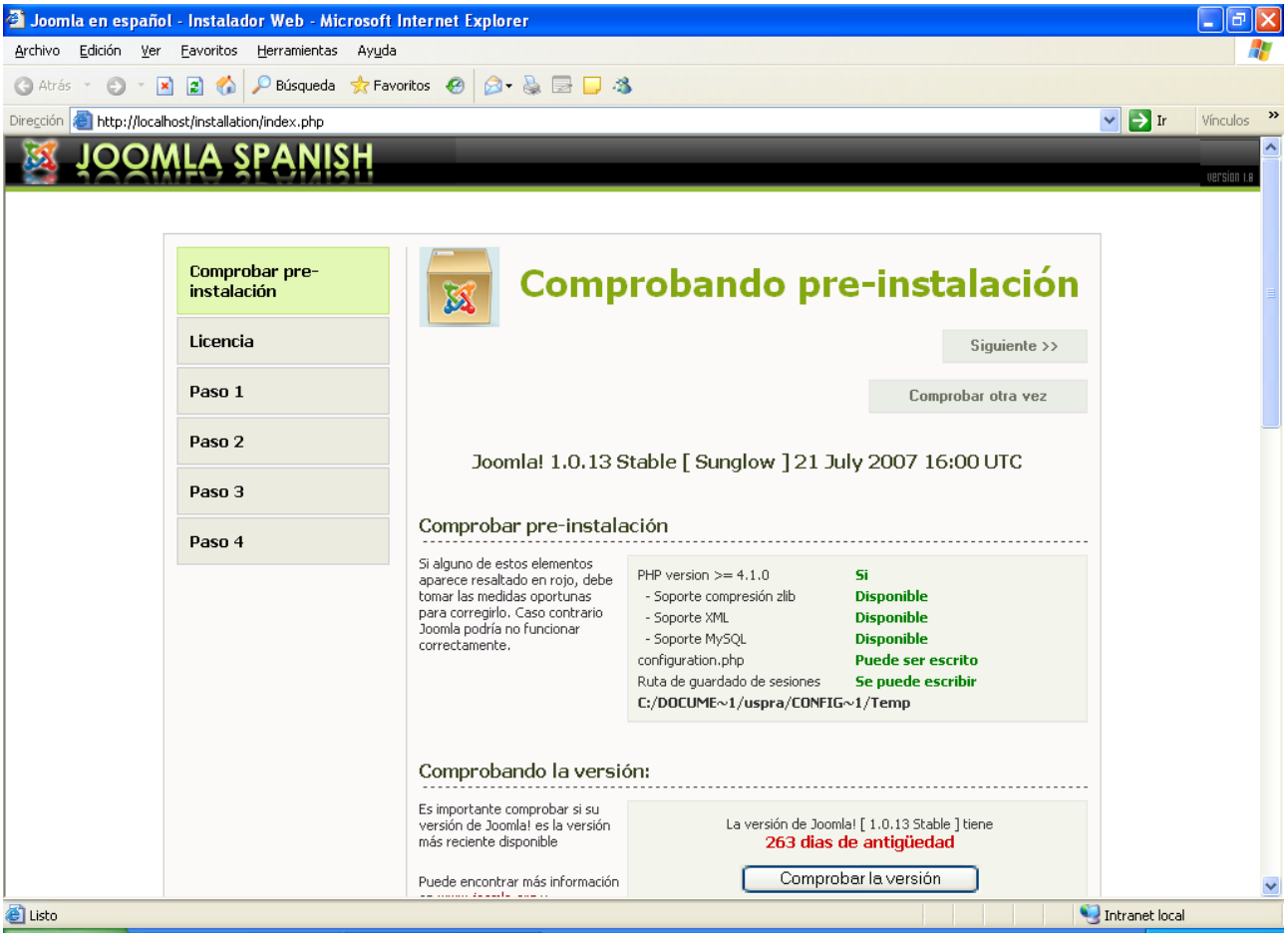


Figura 113: Pre-instalación - JOOMLA

Pulsamos el botón “*Siguiente*”.

Leemos la licencia GNU/GPL, que se corresponde con una licencia de Software Libre y pulsamos el botón "*Siguiente*".

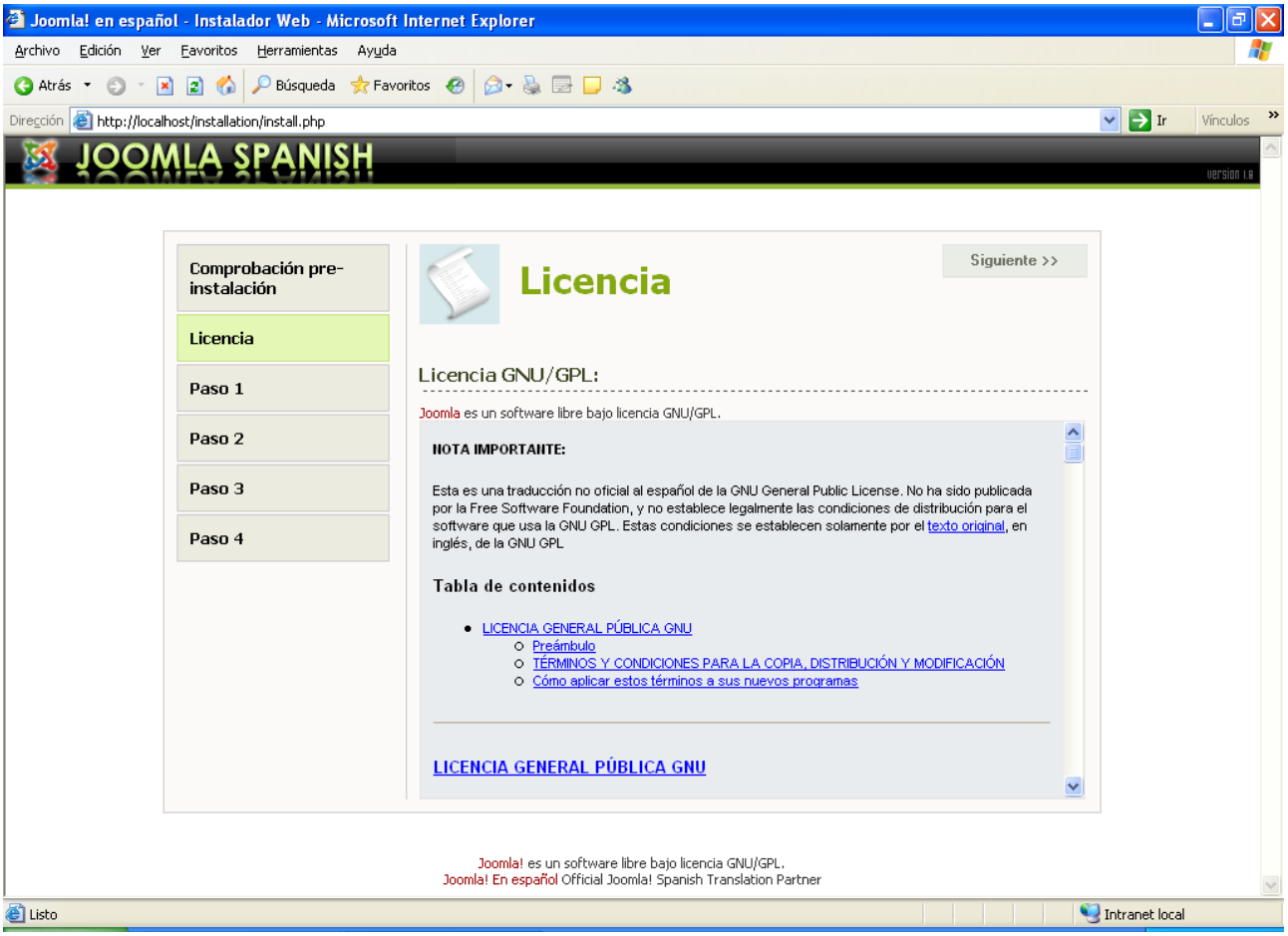


Figura 114: Licencia - JOOMLA

- ✓ **PASO UNO DE INSTALACIÓN:**

En el primer paso de la instalación, debemos definir los aspectos relacionados con la base de datos, así como el nombre del servidor, especificaremos “localhost”, ya que es ese el nombre de nuestro servidor MySQL.

Como usuario de la base de datos, estableceremos “root”, que es el único usuario con permisos que tenemos definido. También debemos expresar la contraseña del usuario que se omite por motivos de seguridad.

Los dos últimos parámetros hacen referencia al nombre de la base de datos, que en nuestro caso en MySQL será “ufp” (haciendo referencia al cliente “Unión Federal de Policía”) y qué prefijo tendrán las tablas que cree el CMS, que, en nuestro caso, será “ufp_”.

Es importante mencionar que debemos desmarcar la opción Insertar datos de ejemplo para evitar que JOOMLA nos llene de datos basura el portal.

A continuación exponemos los datos de estos parámetros para el caso concreto de nuestro proyecto:

Nombre del servidor localhost	*Usualmente 'localhost'*
Nombre del usuario MySQL root	*EL nombre de usuario de la base de datos en tu hosting*
Contraseña MySQL **********	*Por seguridad es obligatorio una contraseña mysql*
Nombre de la base de datos MySQL ufp	*Algunos servidores sólo permiten un número limitado de bases de datos por sitio Web. Usa un prefijo de tabla diferente para JOOMLA.*
Prefijo de la tabla MySQL ufp_	*No uses 'old_' ya que este prefijo se usa para las copias de seguridad*

Y tras estas operaciones pulsaremos el botón “*Siguiente*”.

Figura 115: Paso 1 - JOOMLA

✓ PASO DOS DE INSTALACIÓN:

En el paso dos, establecemos cómo queremos que se llame la Web. Esto será el título HTML que tendrá la ventana del navegador.

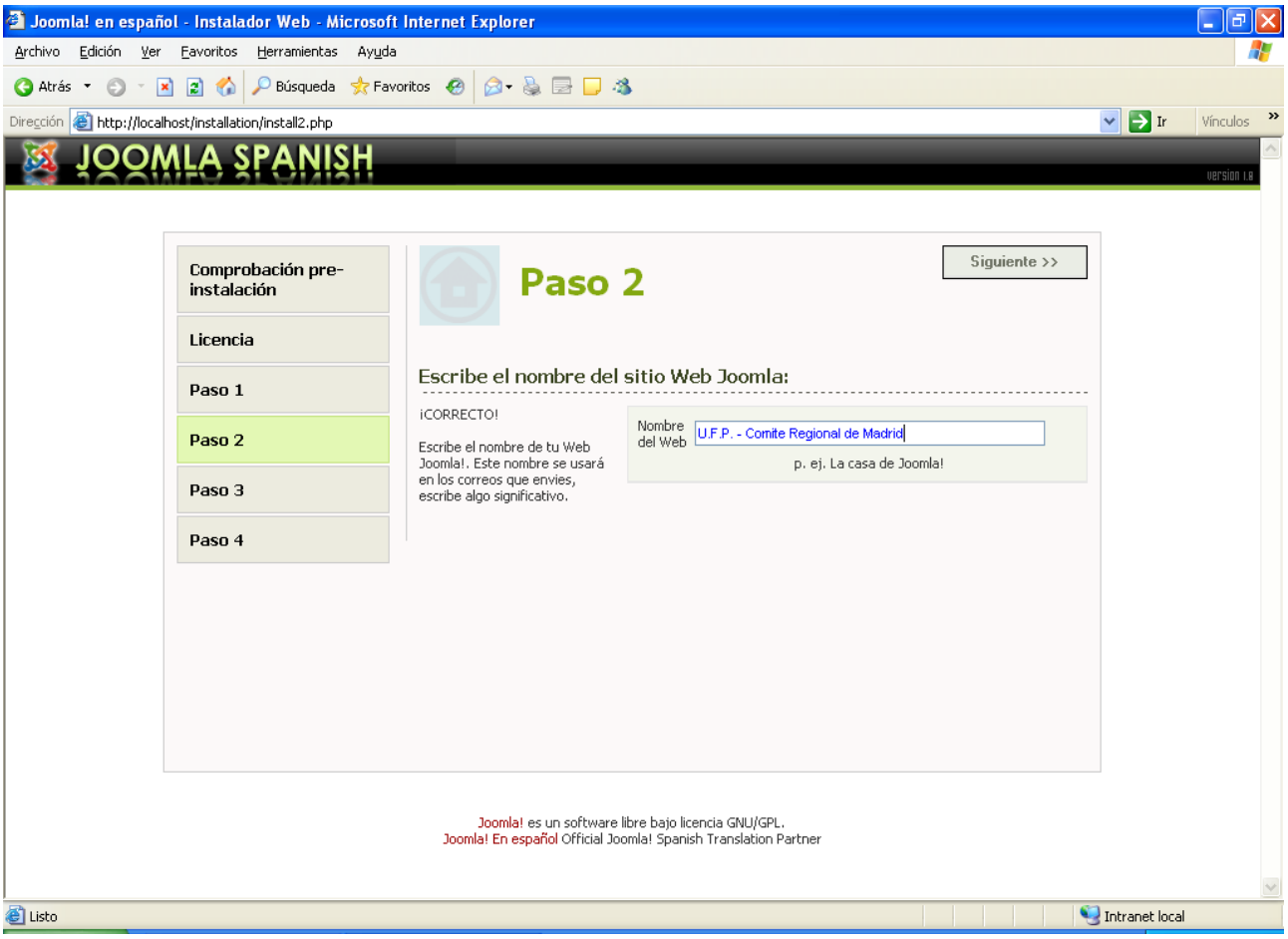


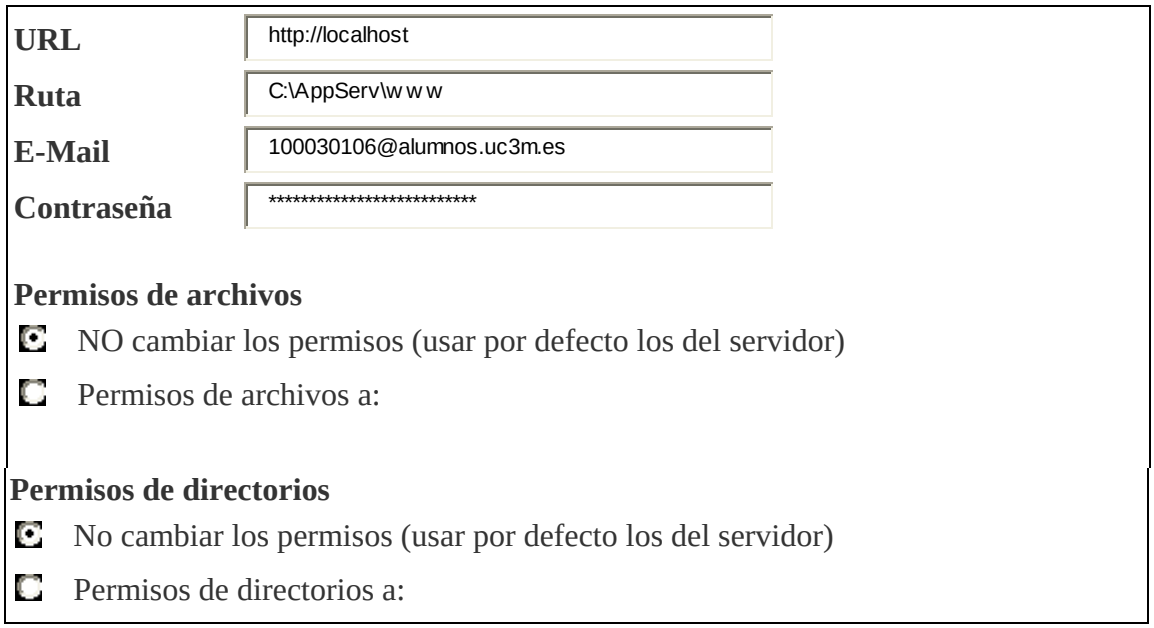
Figura 116: Paso 2 - JOOMLA

Pulsamos el botón "*Siguiente*".

- ✓ **PASO TRES DE INSTALACIÓN:**

En el paso tres de la instalación, expresaremos: los parámetros de acceso a nuestro servidor (que puede ser mediante la IP o mediante un nombre de dominio), también referenciaremos la ruta física dentro de nuestro sistema del espacio web, y por último los datos del administrador, como son el email o correo electrónico, y finalmente, la contraseña o "password".

A continuación exponemos los datos de estos parámetros para el caso concreto de nuestro proyecto:



Una vez rellenados los datos pulsamos en "*Siguiente*".

Figura 117: Paso 3 - JOOMLA

✓ PASO CUATRO DE INSTALACIÓN:

Para el último paso, la página nos dará el aviso de que debemos eliminar el directorio “C:\AppServ\www\Installation\”.

Tras esta operación quedará totalmente instalado JOOMLA.

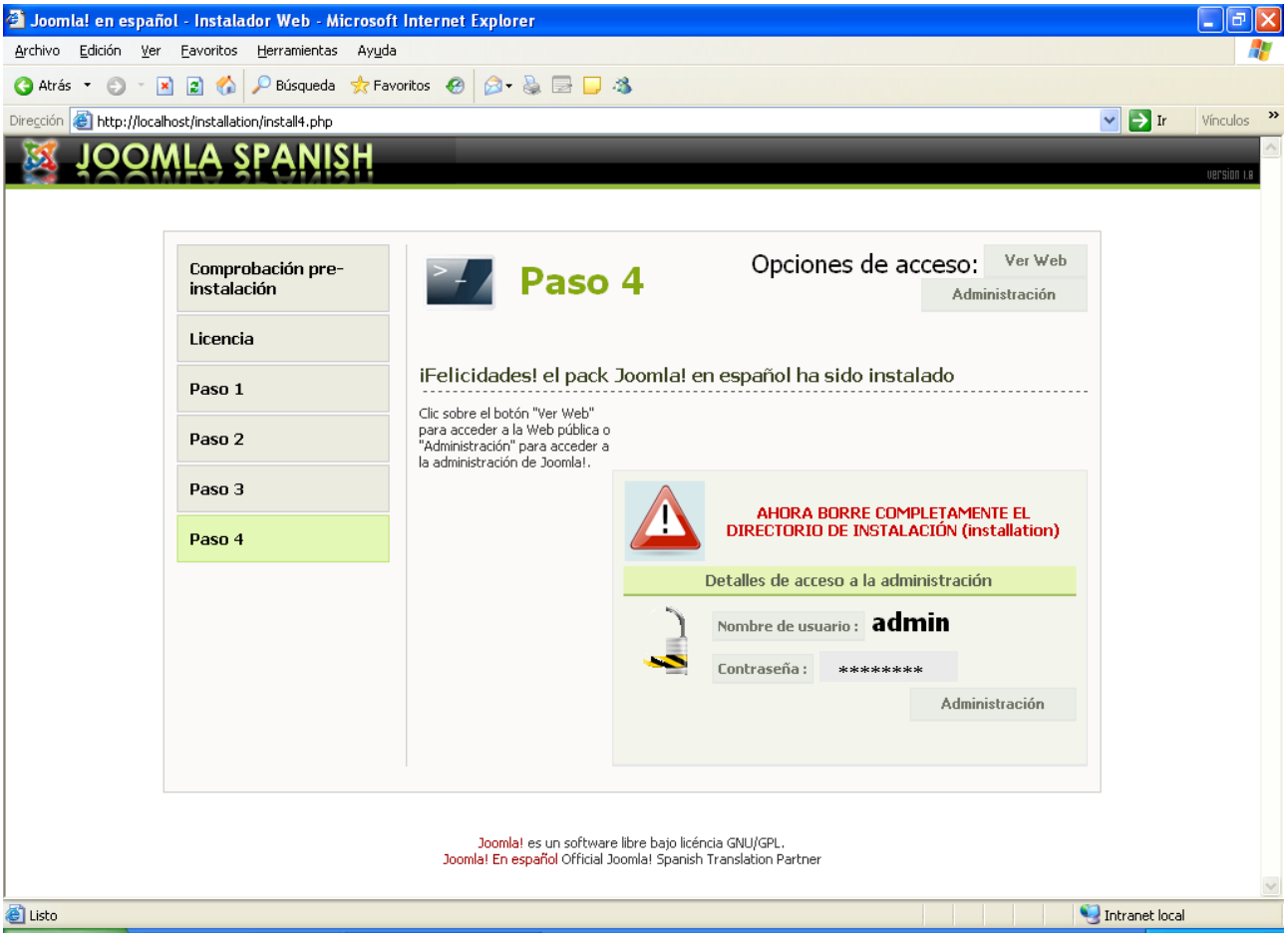


Figura 118: Paso 4 - JOOMLA

Pulsamos el botón “Ver Web”.

Se ha completado asi la instalación, al intentar acceder al Portal de JOOMLA veremos como nos muestra un mensaje de seguridad, solicitandonos que eliminemos la carpeta correspondiente a los archivos de instalación.

http://localhost/

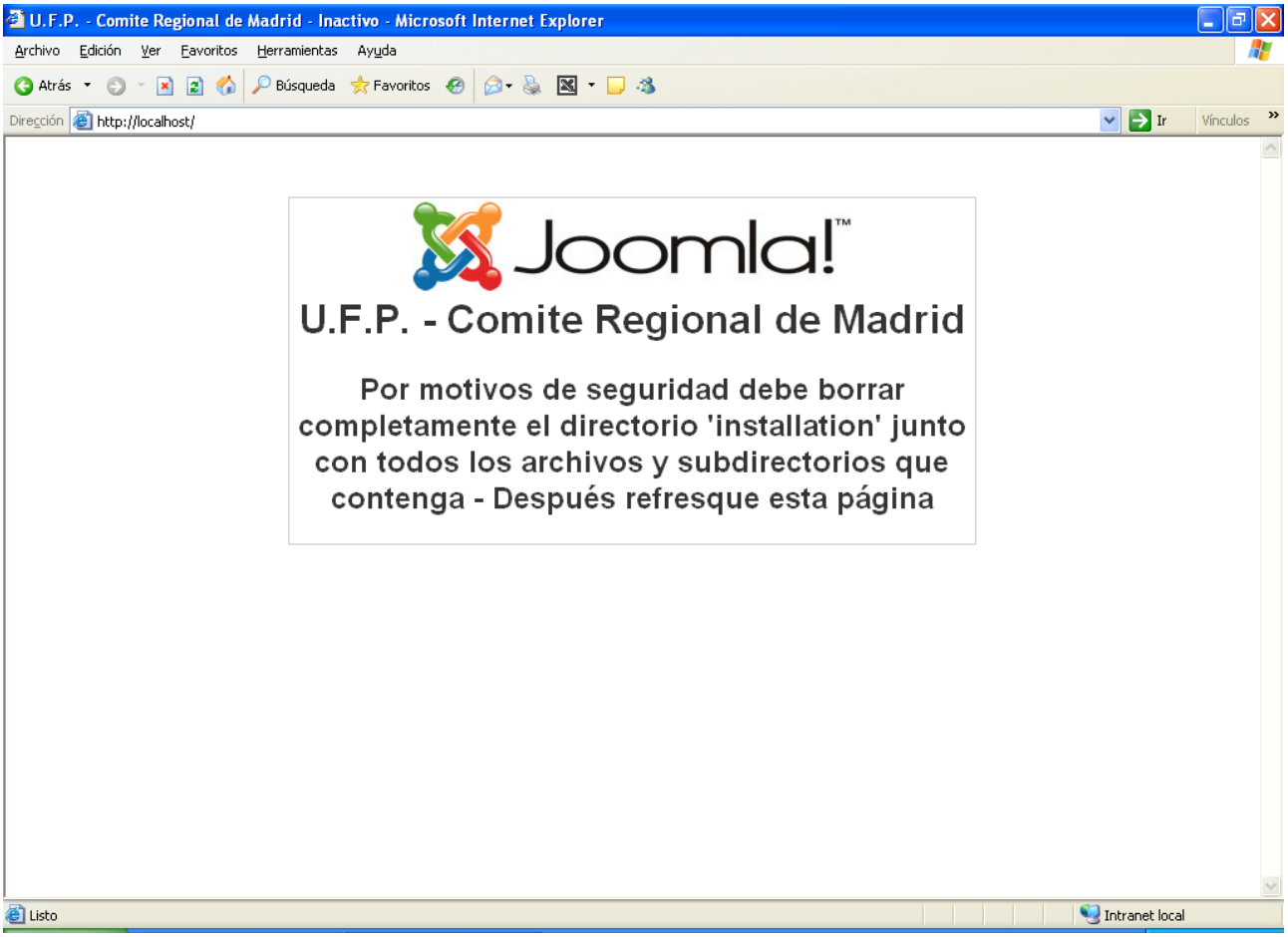


Figura 119: “http://localhost/” tras instalación

Para solucionar esto accederemos a la carpeta: “C:\AppServ\www”, cuyo contenido es el que se muestra en la siguiente figura.

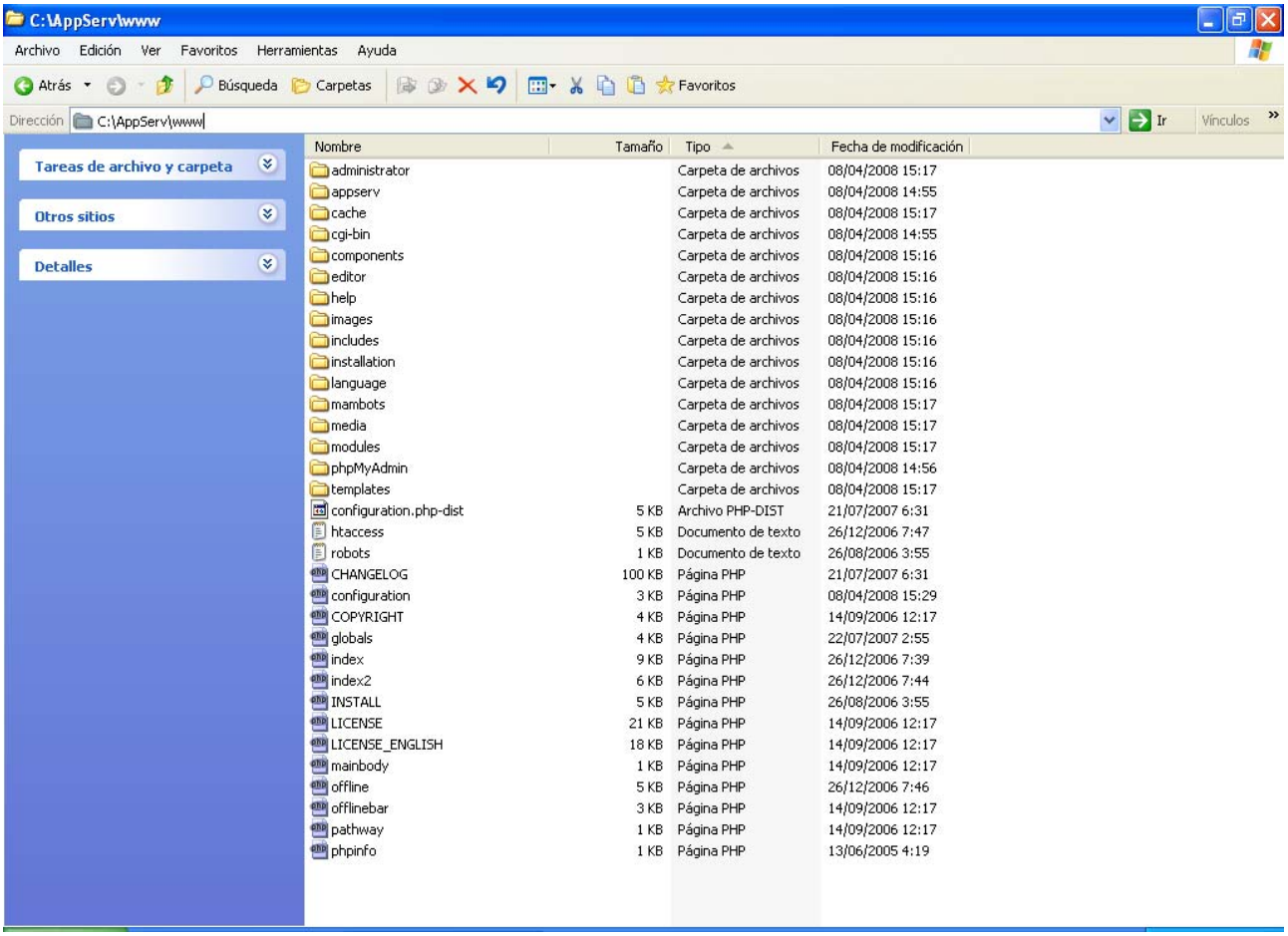


Figura 120: Contenido de “C:\AppServ\www\”

Y eliminamos la carpeta: “*installation*”

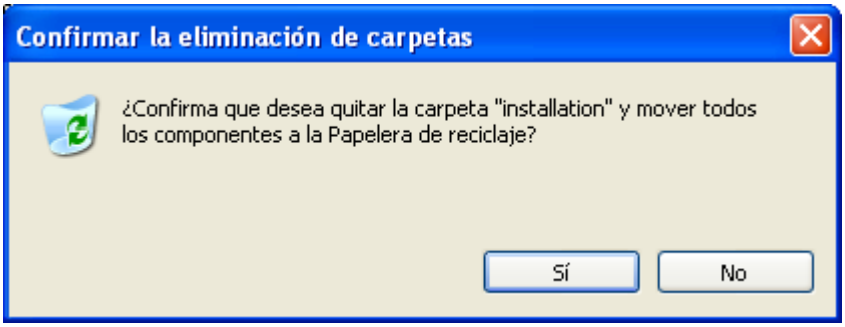


Figura 121: Eliminando carpeta “installation”

Ahora al acceder al Portal JOOMLA nos mostrará la plantilla por defecto desde la cual podremos ir desarrollando nuestro portal de manera personalizada.

http://localhost/

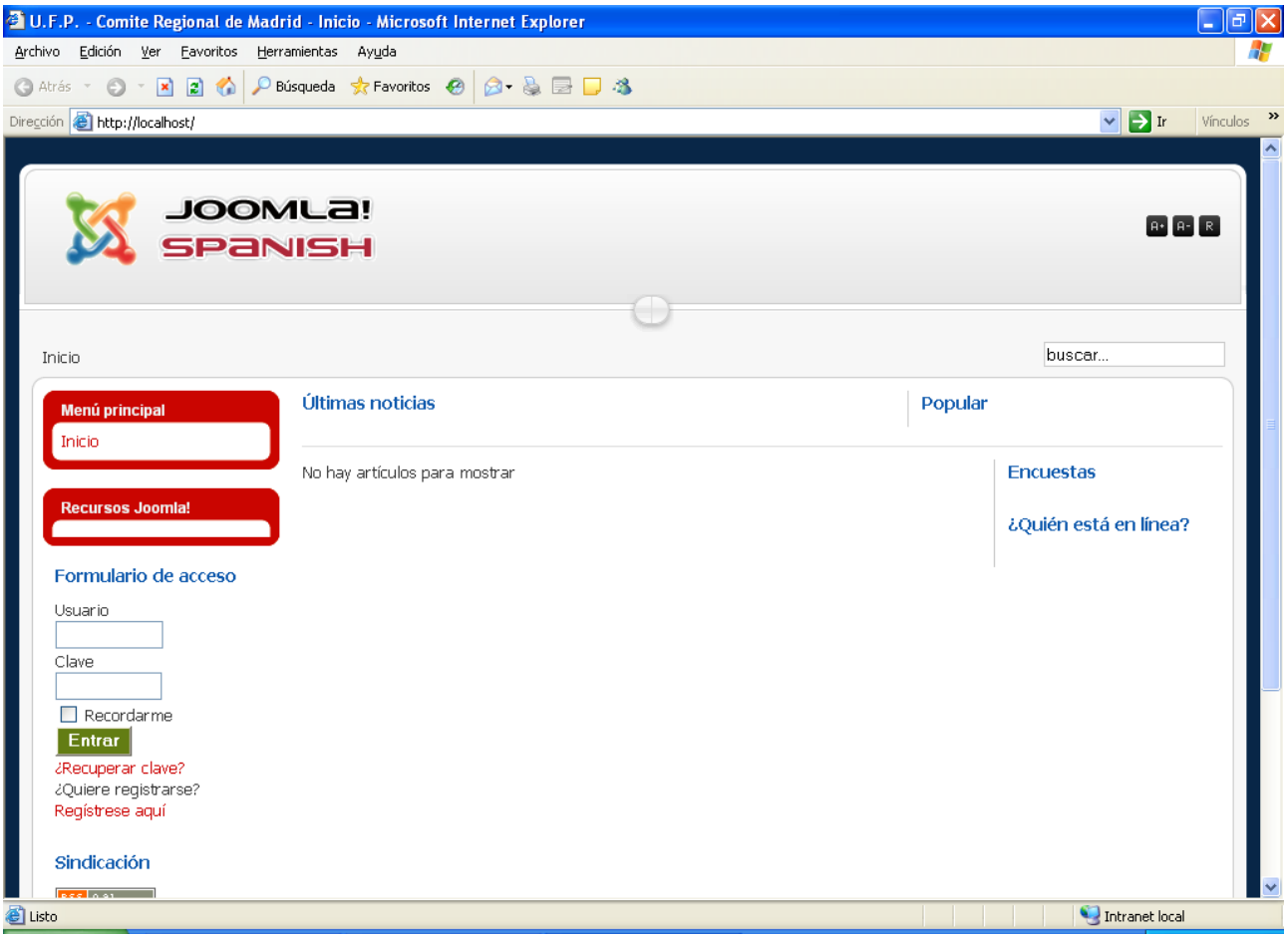


Figura 122: "http://localhost/" por defecto

APÉNDICE "D" - INSTALACIÓN DE SSL/TLS EN APACHE 2.2 SOBRE WINDOWS

D.1 - INTRODUCCION A SSL Y TLS

SSL significa "*Secure Socket Layer*" y es un marco de cifrado que puede ser usado en las conexiones de red individuales. Además de asegurar los datos contra las escuchas extrajudiciales, también permite autenticar a una conexión de red en una o ambas partes utilizando una infraestructura de clave pública basado en el estardar "*OSI X.509 standard2*".

X.509 utiliza una jerarquía centralizada, con la mayoría de entidades de confianza en su núcleo. Estas entidades en cuestión de confianza, son archivos que se utilizan para la distribución de claves públicas y certifica que el portador del archivo es quien dice ser. Los certificados son firmados digitalmente por la entidad certificadora (llamada "certificado de autoridad" o CA) para prevenir la falsificación o alteración, y el cliente puede validar la firma digital contra la clave pública, manteniendo el archivo por la autoridad certificadora y decidir si debe confiar en el certificado de servicio. Las Autoridades de Certificación, por lo tanto, tienen como función ser una especie de notario público, validando que las partes en una transacción son realmente quien ellos dicen que son.

En este anexo de la memoria del proyecto, cubriremos la generación de un certificado autofirmado. Dicho certificado proporciona protección contra las escuchas extrajudiciales, pero no proporcionan el mismo nivel de confianza al cliente final que proporcionaria la Autoridad de Certificación de una empresa dedicada a tal cometido, sobre todo si el sitio es accesible al público. En esencia, un certificado autofirmado le dice al usuario que nadie más da fe de su identidad, mientras que con un certificado comprado mediante "CA", alguien más da fe para su identidad (como un notario).

Transport Layer Security (TLS) es simplemente la última versión de SSL, y está normalizado por la IETF.

D.2 - OBTENIENDO APACHE CON SSL

Los paquetes binarios de Apache con SSL para Windows se puede obtener de:

http://www.apachelounge.com/download/

Pero, dichos paquetes no vienen con un paquete de instalación de Windows. En lugar de ello, simplemente uno tiene un archivo comprimido en ".zip" que contiene los archivos e instrucciones para su instalación. A pesar de que el proceso de instalación está cubierto en este documento, vale la pena leer el "Aviso" y "Léame primero" los archivos descargados en el archivo zip antes de continuar, sobre todo si la instalación de una versión anterior a 2.2.4.

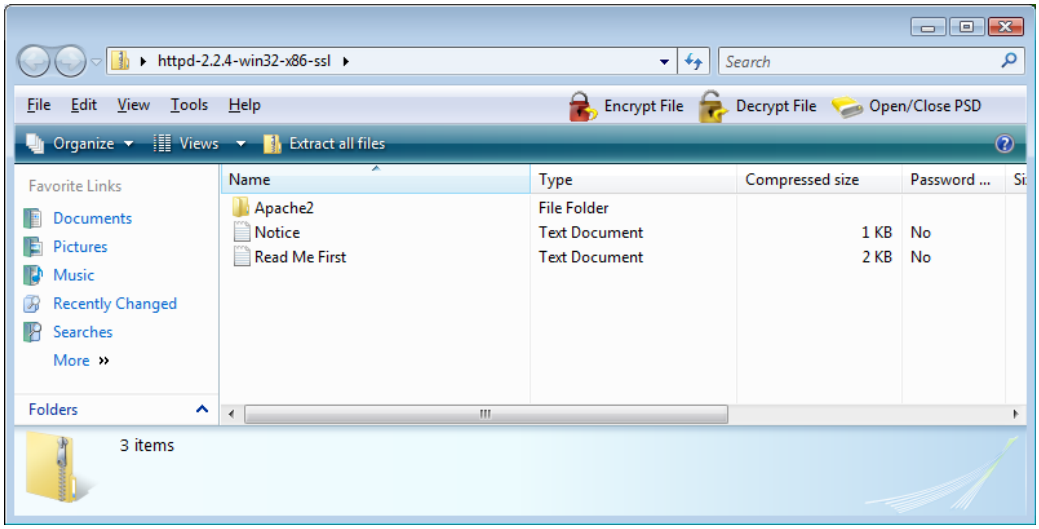


Figura 123: Paquete Zip de "httpd-2.2.4-win32-x86-ssl"

D.3 - DESCARGANDO E INSTALANDO LOS PRERREQUISITOS

El paquete requiere pero no contiene el "*Visual C++ 2005 redistribuible*". Antes de instalar el software, descargar y ejecutar el programa desde la siguiente ubicación:

http://www.microsoft.com/downloads/details.aspx?FamilyID=200b2fd9-ae1a-4a14-984d-389c36f85647&DisplayLang=en

D.4 - INSTALACIÓN SOBRE UNA INSTALACIÓN EXISTENTE DE APACHE

Para instalar manualmente a través de una instalación de Apache de la misma versión, usted debe seguir los siguientes pasos:

- ✓ **Haga una copia de seguridad de su archivo "httpd.conf":**

  Usted necesitará el archivo "httpd.conf" más tarde. Este archivo se encuentra en "C:\Archivos de programa\Apache Software Foundation\Apache2.2\conf" si ha instalado utilizando el instalador de paquetes de "http://httpd.apache.org".

- ✓ **Copie todos los archivos de la carpeta Apache2 en el archivo zip a su wwwroot:**

  Por defecto, el wwwroot se encuentra en "C:\Archivos de programa\Apache Software Foundation\Apache2.2\" si está instalado desde el paquete oficial. Tenga en cuenta que el servicio de Apache debe detenerse para que esto tenga éxito.

- ✓ **Copie de nuevo el archivo “httpd.conf”**

  Una vez terminado el proceso anterior, copiar de nuevo el archivo “httpd.conf”, Apache debe ser capaz de correr como lo hizo antes. Tendrá que editar este archivo, pero el proceso está documentado a continuación.

D.5 - MANUAL DE INSTALACIÓN DESDE CERO

Si va a instalar este software en un sistema que no ha tenido Apache instalado previamente, el enfoque es más fácil de instalar manualmente. Para ello, basta con copiar los archivos desde el comprimido “.zip” a la ruta C:\Apache2 o la carpeta de su instalación de Apache y ejecutar el siguiente comando para instalar el software como un servicio de red:

c:\apache2\bin\httpd -k install

Los usuarios de Windows que deseen utilizar el Apache Monitor pueden copiar esa solicitud o crear un enlace al mismo en la carpeta de inicio. Es en la misma carpeta que httpd.

APÉNDICE "E" - MANUAL PARA GENERAR CERTIFICADOS

Los certificados se pueden generar utilizando "*Microsoft Certificate Server*" (parte del paquete de Windows Server), o con una herramienta de software libre como es "*OpenSSL*". Este anexo se centrará en OpenSSL ya que este se ve enriquecido con la versión de Apache que hemos instalado.

La primera parte de esta sección se mostrará cómo crear una solicitud de firma de certificado, o la RSE, lo que podría ser enviado a una autoridad certificadora de confianza, a fin de obtener un certificado SSL pleno. Si esta instalación va a ser accesibles al público, este es el método preferido para la generación de certificado. Para las pruebas y los fines del desarrollo, puede que desee firmar con una Autoridad Certificadora creada por usted mismo.

La primera cosa que usted debe hacer es copiar el archivo "*openssl.cnf*" de la "*wwwroot/conf*" en el directorio "*C:\openssl\ssl*" (puede que necesite crear este directorio primero). Esto es necesario porque este es el único lugar donde openssl buscará el archivo de configuración.

E.1 - GENERACIÓN DE LA SOLICITUD DE FIRMA DE CERTIFICADO

La primera etapa en la generación de un certificado es crear un servidor de claves. Esto se hace con la utilidad openssl. Tenga en cuenta que por debajo de la ruta puede tener que ser modificado dependiendo de donde esté instalado Apache en su sistema:

"C\Archivos de programa\Apache Software Foundation\Apache2.2\bin\openssl.exe" genrsa-des3-out server.key 1024

Una vez introducido dicho comando, se le pedirá una contraseña. Escriba la misma contraseña (entre 4 y 511 caracteres) en las dos indicaciones. No pierda esta contraseña, ya que hará que el certificado sea inútil.

La siguiente etapa es crear una clave sin cifrar. Esta clave debe ser protegida cuidadosamente, ya que se utiliza en la fase de intercambio de claves. Si la clave está en peligro, el sistema se vuelve vulnerable al famoso ataque del "hombre en el medio".

La clave es descifrada usando el siguiente comando (de nuevo, se adapta el camino, según sea necesario y todo en una línea):

"C\Archivos de programa\Apache Software Foundation\Apache2.2\bin\openssl.exe" rsa-in server.key -- server.pem

Ahora, podemos generar un certificado “DES” firmado, llamado RSE o solicitud de firma de certificado. El comando es la siguiente:

"C\Archivos de programa\Apache Software Foundation\Apache2.2\bin\openssl.exe" req-new-key server.key-out server.csr

Siga las instrucciones para generar el certificado SSL. Tenga en cuenta que el nombre canónico (CN) debe ser el nombre de dominio completamente calificado para el servidor que está creando.

A continuación se muestra una captura de pantalla de todo el proceso anterior con unos datos de ejemplo:

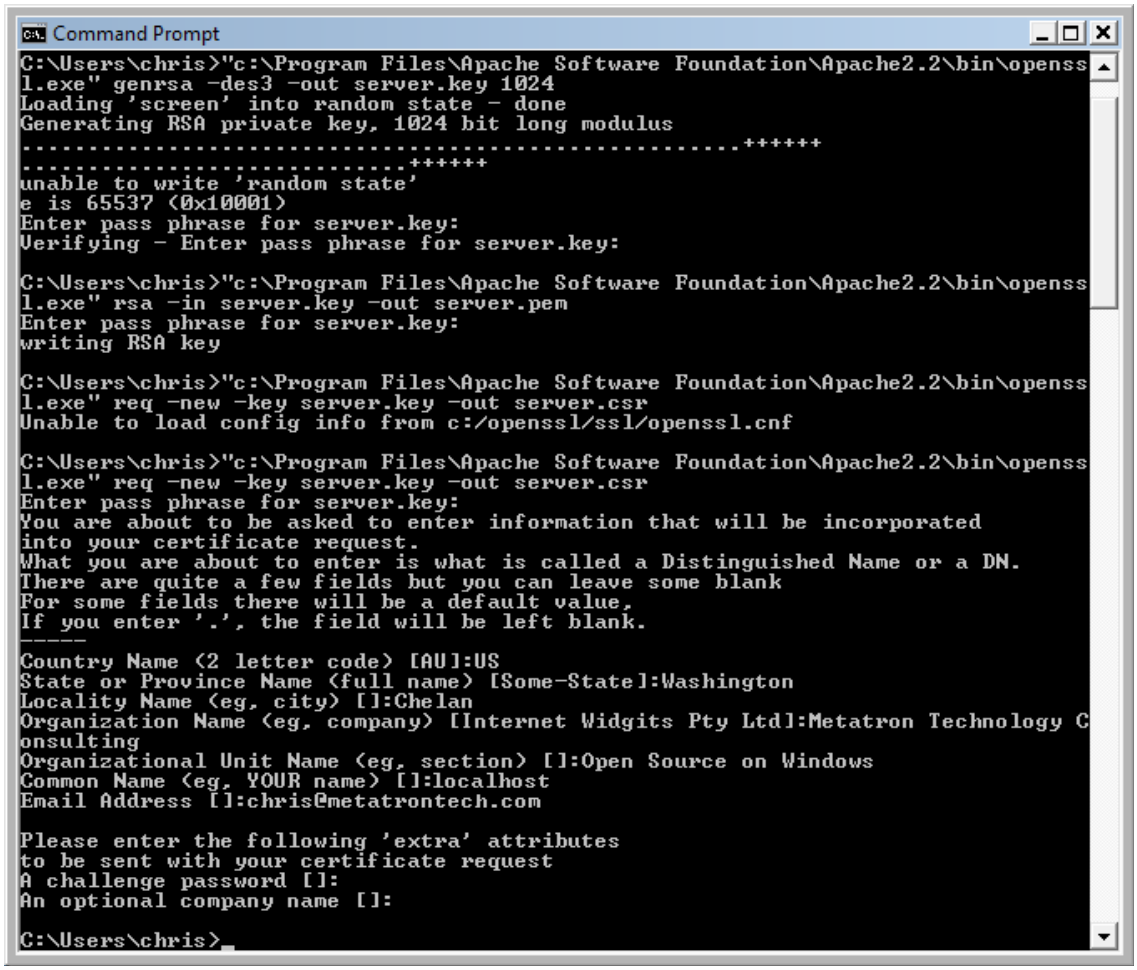


Figura 124: Command Prompt “creación de certificados”

E.2 - AUTOFIRMAR EL CERTIFICADO

Una vez que haya generado la responsabilidad social de la empresa, puede que desee enviar a una autoridad certificadora de confianza para su firma. Si el sistema es sólo para el desarrollo y pruebas de uso, usted podría proceder con sólo un certificado autofirmado. Tenga en cuenta que la mayoría de los navegadores va a informar al usuario de que la fiabilidad del certificado está en duda, por lo que esta no está recomendado para público y frente a las solicitudes.

Para generar un certificado válido por 30 días, puede usar el siguiente comando:

"C\Program Files\Apache Software Foundation\Apache2.2\bin\openssl.exe" x509 -req -days 30 -in server.csr -signkey server.key -out server.crt

Una vez más, el mando es todo en una línea. Introduzca la clave de la contraseña cuando se le solicite.

E.3 - INSTALACIÓN DEL CERTIFICADO

Copie el "*server.crt*" y "*server.pem*" en el *"wwwroot\conf\"* (la ruta por defecto será, probablemente, "*C:\Archivos de programa\Apache Software Foundation\Apache2.2\conf*").

E.4 - EDICIÓN DEL ARCHIVO "HTTPD.CONF" Y ARCHIVOS RELACIONADOS

Con el fin de ejecutar Apache con SSL/TLS, debe modificar los ficheros de configuración y reiniciar el software.

En el archivo "*httpd.conf*", cambiar las siguientes líneas. Tenga en cuenta que la forma más sencilla de hacerlo es a través de Buscar o Buscar interfaz de su editor de texto. En cada uno de estos casos, todo lo que necesita hacer es quitar el signo # (inicial) con el fin de descomentar la línea:

 # LoadModule ssl_module modules / mod_ssl.so
y
 # include conf / extra / httpd-default.conf

En el *"wwwroot\conf\extras\"* (por defecto "C:\Archivos de programa\Apache Software Foundation\Apache2.2\conf\extras"), modificar las siguientes líneas:

SSLCertificateKeyFile C:/Archivos de programa/Apache Software Foundation/Apache2.2/conf/server.key

SSLCertificateKeyFile "C:/Archivos de Programa/Apache Software Foundation/Apache2.2/conf/server.pem"

Por supuesto, si desea almacenar la clave en algún otro lugar, usted debe modificar el camino. Si hay espacios en la ruta, añadir las comillas alrededor de todo el argumento.

APÉNDICE "F" - INSTALACIÓN HERRAMIENTA CRYPTOKIT (F.N.M.T.)

F.1 - INSTALACIÓN DEL LECTOR DE CRIPTOTARJETAS

F.1.1 - Antes de conectar el lector…

Siga estas instrucciones:

1. Descargue la última versión de los drivers del *Cryptokit* desde uno de estos dos sitios:

   http://www.c3po.es/downloads/cdcrv47.zip
   http://ca.lefis.org/ (Software > CERES)

2. Descomprimir en C:\
3. Ejecutar Setup.exe
4. Aceptar la petición de reiniciar…

F.1.2 - Conectamos el lector… (sin criptotarjeta)

Una vez arrancado de nuevo Windows, conecte por primera vez el lector USB (sin tarjeta) y siga el asistente de instalación con las opciones:

- ✓ No conectar a Windows Update.
- ✓ Instalar software desde ubicación seleccionada:
  "C:\cdcrv47\Drivers\LTC3X USB"

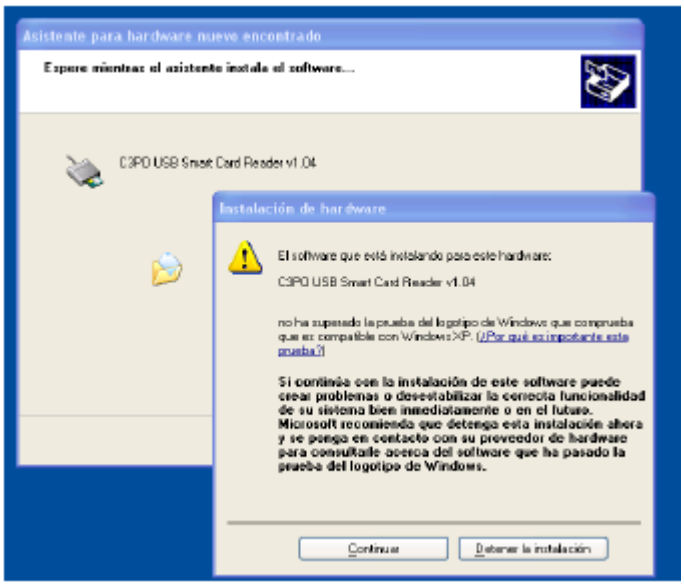


Figura 125: Instalación del driver del lector de tarjetas

- ✓ Pulsar en Continuar.

- ✓ Al terminar nos aparecerá un mensaje como este:

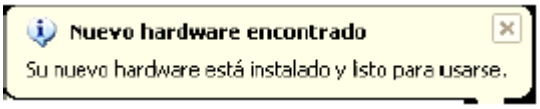


Figura 126: Nuevo hardware encontrado

F.1.3 - Insertamos la Criptotarjeta en el lector…

Al insertar la criptotarjeta en el lector, observaremos que:

- ✓ No aparecerán ningún mensaje de Windows por insertar la tarjeta. Esto es normal.
- ✓ Se encenderá la luz verde (leyendo) durante unos dos segundos y se apagará.
- ✓ Se encenderá la luz roja (esperando) y permanecerá encendida mientras no se use el lector.

En este punto usted debería tener el lector correctamente instalado.

F.2 - CÓDIGOS PIN Y DESBLOQUEO DE LA CRIPTOTARJETA

En el sobre donde ha encontrado la criptotarjeta encontrará una carta de la FNMT-RCM. En el podrá ver unos códigos impresos que ahora explicaremos.

Debe conservar este sobre en lugar seguro. Nunca destruya esta información o será imposible recuperar la tarjeta ante un bloqueo u olvido del PIN.

Las Criptotarjetas vienen de fábrica con dos códigos:

F.2.1 - CÓDIGO PIN (Personal Identification Number):

- ✓ Es el código que vd. tendrá que introducir para usar la tarjeta. Es como el PIN de la tarjeta de Crédito o el PIN del teléfono móvil.

- ✓ El PIN se solicita por sesión. Se le pide la primera vez que se accede a la criptotarjeta y el sistema lo ‘recuerda’ mientras no se reinicie o se desconecte o pase un cierto tiempo.

- ✓ El PIN es modificable y a continuación veremos cómo hacerlo.

- ✓ Cuando se introduce el PIN incorrectamente más de 3 veces, la tarjeta se bloquea. Esto es una medida de seguridad.

F.2.2 - CÓDIGO DE DESBLOQUEO

- ✓ Este código no es modificable.
- ✓ Este código sirve para desbloquear la tarjeta o para cambiar el PIN.

F.3 - CAMBIO DEL PIN DE LA CRIPTOTARJETA

Para cambiar el PIN de su tarjeta deberá seguir los siguientes pasos con el ratón desde el menú de inicio:

Inicio
→ *Todos los Programas*
→ *FNMT-RCM*
→ *Tarjeta*
→ *Desbloqueo de tarjeta y cambio de Pin*

Aparecerá el siguiente programa, en el que tendrá que introducir su *NUEVO PIN*.

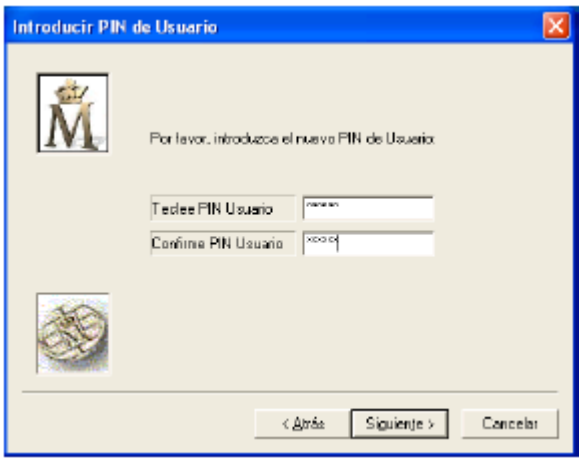


Figura 127: Introduciendo el nuevo PIN

A continuación deberá introducir el *CÓDIGO DE DESBLOQUEO* (impreso en la carta donde recibió la criptotarjeta) para realizar el cambio de PIN:

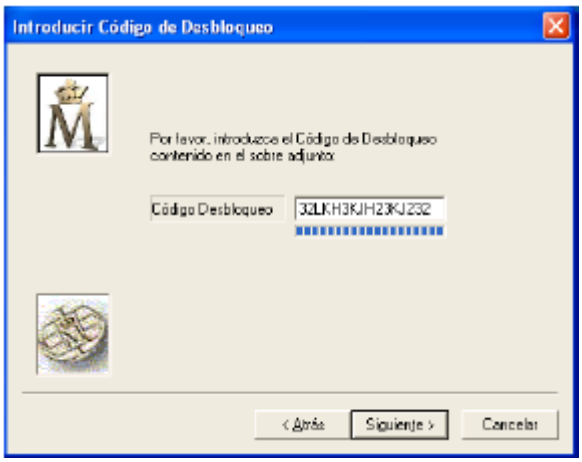


Figura 128: Introduciendo el código de desbloqueo

El procedimiento es el mismo para:

- ✓ Cambio del PIN de fábrica por un PIN personal.
- ✓ Cambio del PIN por olvido.
- ✓ Cambio del PIN porque la tarjeta está bloqueada.

F.4 - IMPORTAR UN CERTIFICADO EN LA CRIPTOTARJETA

Es posible tener hasta 4 certificados con sus claves en una tarjeta.

Las tarjetas sólo soportan claves RSA con longitud de 1024 bits.

Si dispone de un certificado LEFIS o ajeno a LEFIS y desea instalarlo en la tarjeta deberá realizar dos operaciones:

1) Exportar el certificado que ya existe en el sistema.
2) Importar el certificado en la tarjeta.

F.4.1 - Exportar un certificado del sistema

Para exportar un certificado ya instalado en el sistema, abra *Internet Explorer* y vaya a:

Herramientas
→ *Opciones de Internet*
→ *Contenido*
→ *Certificados…*

Aparecerá una lista de sus certificados, seleccione el que desea exportar y haga 'click' en el botón de Exportar:

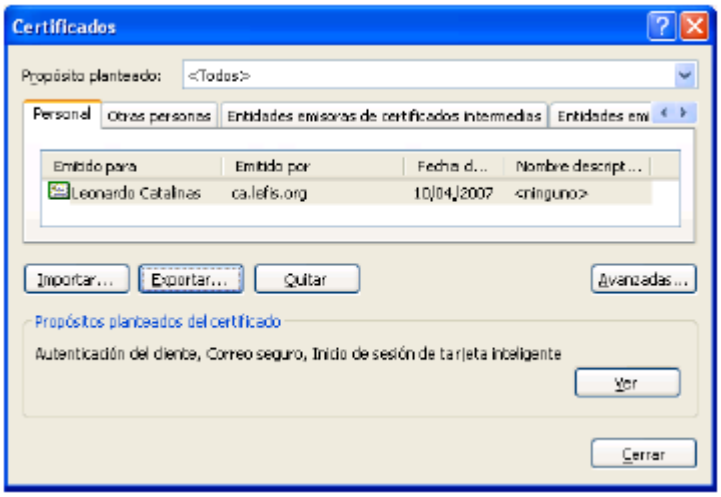


Figura 129: Exportando Certificados

Aparecerá un asistente de exportación que le permitirá exportar su certificado si es posible.

A continuación se le mostrará un asistente. Debe completarlo con las siguientes opciones:

- CLAVE PRIVADA:
  - ✓ Exportar la Clave Privada.

- FORMATO:
  - ✓ Intercambio de Información Personal PKCS#12.
  - ✓ Incluir certificados de la ruta de certificación.
  - ✓ Permitir protección segura.

- CONTRASEÑA:
  - ✓ Esta contraseña sirve para proteger el certificado y las claves.
  - ✓ La necesitará para instalarlos en la tarjeta.

- NOMBRE DE ARCHIVO:
  - ✓ Guarde su certificado en el Escritorio, con un nombre descriptivo.
  - ✓ "*ORGANIZACIÓN_Nombre_Apellidos.pfx*"

Ahora tendrá un fichero que:

1. Está protegido (cifrado) con la contraseña que ha especificado.
2. Contiene su Clave Privada y su Clave Pública con su Certificado.
3. Puede instalar en otro ordenador o importarlo en la tarjeta criptográfica.

Figura 130: Formato de archivo ".pfx"

Este formato de fichero también recibe la extensión de fichero: ".p12" en lugar de ".pfx", pero su utilización es la misma.

F.4.2 - Importar un ".PFX" o ".P12" en la criptotarjeta

Para importar un certificado en uno formato compatible en la criptotarjeta, debe navegar por la siguiente ruta:

Inicio
→ *Todos los Programas*
→ *FNMT-RCM*
→ *Tarjeta*
→ *Importador de Certificados*

Completar el asistente:

1) Seleccionar el certificado a importar.
2) Introducir la contraseña para la clave privada2.
3) Seleccionar automáticamente el almacén de certificados.
4) Se le solicitará el PIN de la tarjeta criptográfica.
5) Posiblemente el asistente le informe de que se va a instalar el certificado de la CA que firma su certificado. Acepte.
6) Detectará que tenemos una tarjeta CERES y preguntará si queremos importar el certificado en ella3. Recuerde que puede tener hasta 4 certificados en la tarjeta. Acepte.

Tenga en cuenta que según el tipo de certificado que desee importar es posible que surjan problemas o incompatibilidades. El sistema de criptotarjetas de FNMT es compatible con la PKI de LEFIS y certificados X.509 con longitud de clave RSA de 1024 bits.

Puede encontrar más documentación acerca de cómo importar certificados en:

Inicio
→ *Todos los Programas*
→ *FNMT-RCM*
→ *Ayuda*
→ *Manual Software Criptográfico*

F.4.3 - Comprobación de la importación

Desde Internet Explorer diríjase a:

Herramientas
→ *Opciones de Internet*
→ *Contenido*
→ *Certificados…*

1) Aparecerá una lista de sus certificados, compruebe que aparece el certificado que acaba de importar.
2) A continuación cierre totalmente Internet Explorer y extraiga la tarjeta.
3) Abra de nuevo Internet Explorer y compruebe que no aparece el certificado importado.
4) Vuelva a conectar la criptotarjeta.
5) Vuelva a comprobar los certificados disponibles: compruebe que ahora sí aparece el certificado.

Estos pasos verifican que el certificado está instalado en la tarjeta y sólo se puede usar cuando ésta está insertada y funciona correctamente.

APÉNDICE “G” - FICHERO “httpd.conf”

Aquí se expone el fichero de configuración básico “httpd.conf” con varios comentarios dentro del mismo código para hacer más comprensible dichas directivas.

El directorio que contendrá este fichero de configuración básica en nuestro proyecto lo podremos encontrar en la siguiente ruta: *“C:\AppServ\Apache2.2\conf\”*.

Las principales modificaciones sobre el código que trae por defecto el servidor Apache se verán reflejadas en letra negrita para facilitar la lectura.

CÓDIGO DEL FICHERO “httpd.conf”

```
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.2/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.2/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "C:/AppServ/Apache2.2" will be interpreted by the
# server as "C:/AppServ/Apache2.2/logs/foo.log".
#
# NOTE: Where filenames are specified, you must use forward slashes
# instead of backslashes (e.g., "c:/apache" instead of "c:\apache").
# If a drive letter is omitted, the drive on which Apache.exe is located
# will be used by default.  It is recommended that you always supply
# an explicit drive letter in absolute paths, however, to avoid
# confusion.
#

# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
```

```
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum  number of requests a server process serves
#ThreadsPerChild 250
MaxRequestsPerChild  0

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to point the LockFile directive
# at a local disk.  If you wish to share the same ServerRoot for multiple
# httpd daemons, you will need to change at least LockFile and PidFile.
#

###############################################################################
ServerRoot "C:/AppServ/Apache2.2"
###############################################################################

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80


###############################################################################
###############################################################################
###                                                                         ###
###                  - SEGURIDAD PROYECTO UC3M -                            ###
###        Configuramos el Servidor Apache de manera segura                 ###
###                Archivo "httpd.conf" de Apache                           ###
###                                                                         ###
###############################################################################
###############################################################################
##                                                                           ##

###############################################################################
## PARA QUE ESCUCHE EN MODO NO SEGURO PUERTO 80.                             ##
## PARA QUE ESCUCHE EN MODO SEGURO (SSL) PUERTO 443.                         ##
###############################################################################
##                                                                           ##

Listen 80
Listen 443

##                                                                           ##
###############################################################################

#
# This configuration file reflects default settings for Apache HTTP Server.
# You may change these, but chances are that you may not need to.
```

```
#
# Timeout: The number of seconds before receives and sends time out.
#

Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5

#
# UseCanonicalName: Determines how Apache constructs self-referencing
# URLs and the SERVER_NAME and SERVER_PORT variables.
# When set "Off", Apache will use the Hostname and Port supplied
# by the client.  When set "On", Apache will use the value of the
# ServerName directive.
#
UseCanonicalName Off

#
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives.  See also the AllowOverride
# directive.
#


##########################################################################
## NOMBRAMOS EL ARCHIVO QUE EN JOOMLA CONTENDRA LAS REGLAS DE        ##
## SEGURIDAD PARA EL PORTAL.                                         ##
##########################################################################
##                                                                      ##

AccessFileName .htaccess.txt

##                                                                      ##
##########################################################################


#
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of:  Full | OS | Minor | Minimal | Major | Prod
```

```
# where Full conveys the most information, and Prod the least.
#
ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#
ServerSignature On

#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_default_module modules/mod_authz_default.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule deflate_module modules/mod_deflate.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
#LoadModule expires_module modules/mod_expires.so
```

```
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule headers_module modules/mod_headers.so
LoadModule imagemap_module modules/mod_imagemap.so
LoadModule include_module modules/mod_include.so
#LoadModule info_module modules/mod_info.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic_module modules/mod_mime_magic.so

#####################################################################
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule negotiation_module modules/mod_negotiation.so
#####################################################################


#####################################################################
## ACTIVAMOS EL MODULO mod_rewrite PARA QUE EL ARCHIVO .htaccess     ##
## CONTENIDO DENTRO DEL PORTAL JOOMLA SEA TENIDO EN CUENTA Y SE      ##
## INTERPRETEN SUS REGLAS INTERNAS DE SEGURIDAD.                     ##
#####################################################################
##                                                                 ##

LoadModule rewrite_module modules/mod_rewrite.so

##                                                                 ##
#####################################################################


LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule speling_module modules/mod_speling.so
#LoadModule status_module modules/mod_status.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so

#####################################################################
LoadModule vhost_alias_module modules/mod_vhost_alias.so
#####################################################################


#####################################################################
## ACTIVAMOS EL MODULO mod_ssl PARA QUE SEA POSIBLE USAR             ##
## CERTIFICADOS SSL EN EL SERVIDOR APACHE.                           ##
#####################################################################
##                                                                 ##

LoadModule ssl_module modules/mod_ssl.so

##                                                                 ##
#####################################################################
```

```
#######################################################################
LoadModule php5_module C:\AppServ\php5\php5apache2_2.dll
#######################################################################

# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#


#######################################################################
## EL CORREO DEL ADMINISTRADOR DEL SERVIDOR APACHE.                  ##
#######################################################################
##                                                                   ##

ServerAdmin 100030106@alumnos.uc3m.es

##                                                                   ##
#######################################################################


#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#


#######################################################################
## EL NOMBRE DEL SERVIDOR, EN ESTE CASO SERA localhost PERO EN UN    ##
## FUTURO SE CORRESPONDERA CON LA DIRECCION http://www.ufpmadrid.com ##
#######################################################################
##                                                                   ##

ServerName localhost

##                                                                   ##
#######################################################################


#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
```

```
###########################################################################
DocumentRoot "C:/AppServ/www"
###########################################################################

#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.
#

# <Directory />
#    Options FollowSymLinks ExecCGI Indexes
#    AllowOverride None
#    Order deny,allow
#    Deny from all
#    Satisfy all
# </Directory>


###########################################################################
## MODIFICAMOS LOS PERMISOS SOBRE LOS DIRECTORIOS                        ##
###########################################################################
##                                                                       ##

<Directory />
    Options FollowSymLinks
    AllowOverride All
    Order deny,allow
    Allow from all
    Satisfy all
</Directory>

##                                                                       ##
###########################################################################


#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

#
# This should be changed to whatever you set DocumentRoot to.
#

###########################################################################
<Directory "C:/AppServ/www">
###########################################################################

    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI
MultiViews
```

```
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.2/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks MultiViews ExecCGI

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   Options FileInfo AuthConfig Limit
    #
    AllowOverride All

    #
    # Controls who can get stuff from this server.
    #
    Order allow,deny
    Allow from all

</Directory>

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.php index.html index.htm
</IfModule>

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<FilesMatch "^\.ht">
    Order allow,deny
    Deny from all
</FilesMatch>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog logs/error.log

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn
```

```
<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"
%I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    CustomLog logs/access.log common

    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #
    #CustomLog logs/access.log combined
</IfModule>

<IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar

    #
    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path
    #
    # If you include a trailing / on /webpath then the server will
    # require it to be present in the URL.  You will also likely
    # need to provide a <Directory> section to allow access to
    # the filesystem path.

    #
    # ScriptAlias: This controls which directories contain server scripts.
    # ScriptAliases are essentially the same as Aliases, except that
    # documents in the target directory are treated as applications and
    # run by the server when requested rather than as documents sent to the
    # client.  The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    #
    ScriptAlias /cgi-bin/ "C:/AppServ/www/cgi-bin/"
```

```
</IfModule>

#
# "C:/AppServ/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "C:/AppServ/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

#
# Apache parses all CGI scripts for the shebang line by default.
# This comment line, the first line of the script, consists of the symbols
# pound (#) and exclamation (!) followed by the path of the program that
# can execute this specific script.  For a perl script, with perl.exe in
# the C:\Program Files\Perl directory, the shebang line should be:

   #!c:/program files/perl/perl

# Note you _must_not_ indent the actual shebang line, and it must be the
# first line of the file.  Of course, CGI processing must be enabled by
# the appropriate ScriptAlias or Options ExecCGI directives for the files
# or directory in question.
#
# However, Apache on Windows allows either the Unix behavior above, or can
# use the Registry to match files by extention.  The command to execute
# a file of this type is retrieved from the registry by the same method as
# the Windows Explorer would use to handle double-clicking on a file.
# These script actions can be configured from the Windows Explorer View menu,
# 'Folder Options', and reviewing the 'File Types' tab.  Clicking the Edit
# button allows you to modify the Actions, of which Apache 1.3 attempts to
# perform the 'Open' Action, and failing that it will try the shebang line.
# This behavior is subject to change in Apache release 2.0.
#
# Each mechanism has it's own specific security weaknesses, from the means
# to run a program you didn't intend the website owner to invoke, and the
# best method is a matter of great debate.
#
# To enable the this Windows specific behavior (and therefore -disable- the
# equivilant Unix behavior), uncomment the following directive:
#
#ScriptInterpreterSource registry
#
# The directive above can be placed in individual <Directory> blocks or the
# .htaccess file, with either the 'registry' (Windows behavior) or 'script'
# (Unix behavior) option, and will override this server default option.
#

#
# DefaultType: the default MIME type the server will use for a document
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value.  If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
#
```

```
DefaultType text/plain

<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    #
    TypesConfig conf/mime.types

    #
    # AddType allows you to add to or override the MIME configuration
    # file specified in TypesConfig for specific file types.
    #
    #AddType application/x-gzip .tgz
    #
    # AddEncoding allows you to have certain browsers uncompress
    # information on the fly. Note: Not all browsers support this.
    #
    #AddEncoding x-compress .Z
    #AddEncoding x-gzip .gz .tgz
    #
    # If the AddEncoding directives above are commented-out, then you
    # probably should define those extensions to indicate media types:
    #
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz

    #
    # AddHandler allows you to map certain file extensions to "handlers":
    # actions unrelated to filetype. These can be either built into the server
    # or added with the Action directive (see below)
    #
    # To use CGI scripts outside of ScriptAliased directories:
    # (You will also need to add "ExecCGI" to the "Options" directive.)
    #
    #AddHandler cgi-script .cgi

    # For type maps (negotiated resources):
    #AddHandler type-map var

    #
    # Filters allow you to process content before it is sent to the client.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    #AddType text/html .shtml
    #AddOutputFilter INCLUDES .shtml
</IfModule>

#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type.  The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
#MIMEMagicFile conf/magic

#
# Customizable error responses come in three flavors:
```

```
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
<IfModule mod_php5.c>
  AddType application/x-httpd-php .php
  AddType application/x-httpd-php .php3
  AddType application/x-httpd-php-source .phps
</IfModule>


#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall is used to deliver
# files.  This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
#
#EnableMMAP off
#EnableSendfile off

# Supplemental configuration
#
# The configuration files in the conf/extra/ directory can be
# included to add extra features or to modify the default configuration of
# the server, or you may simply copy their contents here and change as
# necessary.

# Server-pool management (MPM specific)
#Include conf/extra/httpd-mpm.conf

# Multi-language error messages
#Include conf/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
#Include conf/extra/httpd-autoindex.conf

# Language settings
#Include conf/extra/httpd-languages.conf

# User home directories
#Include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#Include conf/extra/httpd-info.conf

################################################################################
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
################################################################################

# Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf
```

```
# Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf


##############################################################################
## PARAMETROS DEL SERVIDOR PARA CERTIFICADOS SSL.                           ##
##############################################################################
##                                                                          ##

# Configuracion por defecto
Include conf/extra/httpd-default.conf

# Configuracion de seguridad (SSL/TLS)
Include conf/extra/httpd-ssl.conf

#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#

# <IfModule ssl_module>
# SSLRandomSeed startup builtin
# SSLRandomSeed connect builtin
# </IfModule>

##                                                                          ##
##############################################################################


#
# Directives controlling the display of server-generated directory listings.
#
# Required modules: mod_autoindex, mod_alias
#
# To see the listing of a directory, the Options directive for the
# directory must include "Indexes", and the directory must not contain
# a file matching those listed in the DirectoryIndex directive.
#

#
# IndexOptions: Controls the appearance of server-generated directory
# listings.
#
IndexOptions FancyIndexing HTMLTable VersionSort

# We include the /icons/ alias for FancyIndexed directory listings.  If
# you do not use FancyIndexing, you may comment this out.
#
Alias /icons/ "C:/AppServ/Apache2.2/icons/"

<Directory "C:/AppServ/Apache2.2/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

```
#
# AddIcon* directives tell the server which icon to show for different
# files or filename extensions.  These are only displayed for
# FancyIndexed directories.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

#
# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
#
DefaultIcon /icons/unknown.gif

#
# AddDescription allows you to place a short description after a file in
# server-generated indexes.  These are only displayed for FancyIndexed
# directories.
# Format: AddDescription "description" filename
#
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz

#
# ReadmeName is the name of the README file the server will look for by
# default, and append to directory listings.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.
ReadmeName README.html
HeaderName HEADER.html
```

```
#
# IndexIgnore is a set of filenames which directory indexing should ignore
# and not include in the listing.  Shell-style wildcarding is permitted.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

APÉNDICE "H" - FICHERO "httpd-ssl.conf"

Aquí se expone el fichero de configuración del protocolo de seguridad SSL "httpd-ssl.conf" con varios comentarios dentro del mismo código para hacer más comprensible dichas directivas.

El directorio que contendrá este fichero de configuración de seguridad en nuestro proyecto lo podremos encontrar en la siguiente ruta: *"C:\AppServ\Apache2.2\conf\extra\"*.

Este fichero a diferencia del anterior ha sido implementado por completo para adaptarlo a este proyecto por lo que no se encontraran directivas por defecto.

CÓDIGO DEL FICHERO "httpd-ssl.conf"

```
##########################################################################
## PARAMETROS DEL SERVIDOR PARA CERTIFICADOS SSL.                       ##
##########################################################################
##                                                                      ##

## CONFIGURACION: Configuramos parametros basicos del Modulo SSL.

SSLMutex default
SSLRandomSeed startup
"c:/AppServ/Apache2.2/conf/SERVIDOR/CLAVEPRIV/servidor2.key"
SSLSessionCache "dbm:c:/AppServ/Apache2.2/logs/ssl_scache"
ErrorLog "c:/AppServ/Apache2.2/logs/SSL.log"


## SERVIDOR VIRTUAL: El canal SSL se establecera en el puerto seguro, 443.

<VirtualHost localhost:443>

    ## PROTOCOLO SSL: Activamos el Protocolo SSL.

    SSLEngine On


    ## SERVIDOR: Habilitamos el Certificado y la Clave Privada del Servidor.

    SSLCertificateFile conf/SERVIDOR/servidor.crt
    SSLCertificateKeyFile conf/SERVIDOR/CLAVEPRIV/servidor2.key


    ## CA: Habilitamos a la Autoridad de Certificacion.

    SSLCACertificatePath conf/demoCA
    SSLCACertificateFile conf/demoCA/cacert.pem


    ## CLIENTE: Forzamos a que el Cliente tenga Certificado para establecer el
    ## canal SSL.
```

```
    SSLVerifyClient require

</VirtualHost>

##                                                                              ##
################################################################################



################################################################################
##  Control de Accesos correspondiente a ZONA PORTAL.                         ##
################################################################################
##                                                                              ##

    <Directory "C:/AppServ/www">


       ############################################################################
       ## CONEXION SEGURA CON SSL.                                               ##
       ## Redireccionar a Modo Seguro:  desde http a https.                      ##
       ############################################################################


        ## Activamos el modulo rewrite.

        RewriteEngine on


        ## CONDICION: Si no es un acceso por el puerto 443 ...

        RewriteCond %{SERVER_PORT} !443$


        ## CONDICION: Y cuando la URL contenga cualquiera de las siguientes
        ## cadenas: ...


        # FORMULARIO DE ACCESO: Boton entrar.

        RewriteCond %{QUERY_STRING} (option=login)+ [OR]


        # FORMULARIO DE ACCESO: Opcion Recuperar Clave.

        RewriteCond %{QUERY_STRING} (option=com_registration)+ [OR]


        # MENU USUARIO.

        RewriteCond %{QUERY_STRING} (task=blogcategory)+ [OR]


        # FORO USUARIO.

        RewriteCond %{QUERY_STRING} (com_fireboard)+
```

```
## ... REGLA: Automaticamente cambiamos de http a https esa misma URL.
##            Si la URL, no contiene a ninguna de dichas cadenas, o
##            el puerto por el que atendiamos ya era el 443, no se
##            ejecutara dicha linea.

RewriteRule ^(.*)$ https://localhost/$1 [L,R]



#########################################################################
## CONEXION SEGURA CON SSL.                                            ##
## Redireccionar a Modo Abierto:  desde https a http.                  ##
#########################################################################


## CONDICION: Si no es un acceso por el puerto 80 ...

RewriteCond %{SERVER_PORT} !80$


## CONDICION: Y cuando la URL contenga cualquiera de las siguientes
## cadenas: ...


# FORMULARIO DE ACCESO: Al pulsar el boton salir.

RewriteCond %{QUERY_STRING} (option=logout)+ [OR]


# MENU PRINCIPAL: Inicio.

RewriteCond %{QUERY_STRING} (option=com_frontpage)+ [OR]


# MENU PRINCIPAL: Quienes Somos, Documentos, Enlaces de Interes.
# y
# DEPENDENCIAS.

RewriteCond %{QUERY_STRING} (task=view)+ [OR]


# MENU PRINCIPAL: Noticias.

RewriteCond %{QUERY_STRING} (task=blogsection)+ [OR]


# MENU PRINCIPAL: Galeria de Imagenes.

RewriteCond %{QUERY_STRING} (com_zoom)+ [OR]


# MENU PRINCIPAL: Calendario.

RewriteCond %{QUERY_STRING} (com_events)+ [OR]


# ENCUESTAS (Componente para realizar encuestas).

RewriteCond %{QUERY_STRING} (com_poll)+ [OR]
```

```
        # BURCAR (Componente motor de busqueda).

        RewriteCond %{QUERY_STRING} (com_search)+


        ## ... REGLA: Automaticamente cambiamos de https a http y salimos de
        ##            la administracion.
        ##            Si la URL, no contiene a ninguna de dichas cadenas, o
        ##            el puerto por el que atendiamos ya era el 80, no se
        ##            ejecutara dicha linea.

        RewriteRule ^(.*)$ http://localhost/$1 [L,R]


    </Directory>

##                                                                            ##
################################################################################



################################################################################
##  Control de Accesos correspondiente a ZONA ADMINISTRADOR.                  ##
################################################################################
##                                                                            ##

    <Directory "C:/AppServ/www/administrator">


       ############################################################################
       ## CONEXION SEGURA CON SSL.                                               ##
       ## Redireccionar a Modo Seguro:  desde http a https.                      ##
       ############################################################################


        ## Activamos el modulo rewrite.

        RewriteEngine on


        ## CONDICION: Si no es un acceso por el puerto 443 ...

        RewriteCond %{SERVER_PORT} !443$


        ## ... REGLA: Automaticamente cambiamos de http a https esa misma URL.
        ##            Si ya era una URL https, no se ejecutara dicha linea.

        RewriteRule ^(.*)$ https://localhost/administrator/$1 [L,R]


       ############################################################################
       ## CONEXION SEGURA CON SSL.                                               ##
       ## Redireccionar a Modo Abierto: desde https a http.                      ##
       ############################################################################


        ## CONDICION: Si no es un acceso por el puerto 80 ...

        RewriteCond %{SERVER_PORT} !80$
```

```
        ## CONDICION: Y cuando la URL contenga la cadena "option=logout" ...

        RewriteCond %{QUERY_STRING} (option=logout)+


        ## ... REGLA: Automaticamente cambiamos de https a http y salimos de
        ##            la administracion.
        ##            Si la URL, no contiene la cadena "option=logout", o el
        ##            puerto por el que atendiamos ya era el 80, no se
        ##            ejecutara dicha linea.

        RewriteRule ^(.*)$ http://localhost/$1 [L,R]


    </Directory>

##                                                                              ##
################################################################################
```

APÉNDICE “I” - FICHERO “.htaccess”

Aquí se expone el fichero de configuración “.htaccess” con varios comentarios dentro del mismo código para hacer más comprensible dichas directivas.

El directorio que contendrá este fichero de configuración en nuestro proyecto lo podremos encontrar en la ruta raiz del portal: *“C:\AppServ\www\”*.

Las principales modificaciones sobre el código que trae por defecto el servidor Apache se verán reflejadas en letra negrita para facilitar la lectura.

CÓDIGO DEL FICHERO “.htaccess”

```
###############################################################################
###############################################################################
###                                                                         ###
###                  - SEGURIDAD PROYECTO UC3M -                            ###
###            Configuramos el portal de manera segura                      ###
###         Archivo ".htaccess" de la Zona Raiz del Portal                  ###
###                                                                         ###
###############################################################################
###############################################################################
##                                                                           ##


###############################################################################
## Protegemos el archivo ".htaccess" de posibles ataques.                   ##
###############################################################################

## Denegamos el acceso al archivo ".htaccess".

<Files .htaccess>
    order allow,deny
    deny from all
</Files>


###############################################################################
## Desactivamos register_globals para prevenir posibles ataques.            ##
###############################################################################

php_value register_globals 0


##                                                                           ##
###############################################################################
###############################################################################
```

```
################################################################################
################################################################################
###                                                                          ###
###              - SEGURIDAD JOOMLA POR DEFECTO -                            ###
###                                                                          ###
################################################################################
################################################################################
##                                                                          ##

##
# @version $Id: htaccess.txt 5973 2006-12-11 01:26:33Z robs $
# @package JOOMLA
# @copyright Copyright (C) 2005 Open Source Matters. All rights reserved.
# @license http://www.gnu.org/copyleft/gpl.html GNU/GPL
# JOOMLA! is Free Software
##

#####################################################
#  READ THIS COMPLETELY IF YOU CHOOSE TO USE THIS FILE
#
# The line just below this section: 'Options +FollowSymLinks' may cause
problems
# with some server configurations.  It is required for use of mod_rewrite, but
may already
# be set by your server administrator in a way that dissallows changing it in
# your .htaccess file.  If using it causes your server to error out, comment
it out (add # to
# beginning of line), reload your site in your browser and test your sef
url's.  If they work,
# it has been set by your server administrator and you do not need it set
here.
#
# Only use one of the two SEF sections that follow.  Lines that can be
uncommented
# (and thus used) have only one #.  Lines with two #'s should not be
uncommented
# In the section that you don't use, all lines should start with #
#
# For Standard SEF, use the standard SEF section.  You can comment out
# all of the RewriteCond lines and reduce your server's load if you
# don't have directories in your root named 'component' or 'content'
#
# If you are using a 3rd Party SEF or the Core SEF solution
# uncomment all of the lines in the '3rd Party or Core SEF' section
#
#####################################################

#####  SOLVING PROBLEMS WITH COMPONENT URL's that don't work #####
# SPECIAL NOTE FOR SMF USERS WHEN SMF IS INTEGRATED AND BRIDGED
# OR ANY SITUATION WHERE A COMPONENT's URL's AREN't WORKING
#
# In both the 'Standard SEF', and '3rd Party or Core SEF' sections the line:
# RewriteCond %{REQUEST_URI} ^(/component/option,com) [NC,OR] ##optional - see
notes##
# May need to be uncommented.  If you are running your JOOMLA!/Mambo from
# a subdirectory the name of the subdirectory will need to be inserted into
this
```

```
# line.  For example, if your JOOMLA!/Mambo is in a subdirectory called
'/test/',
# change this:
# RewriteCond %{REQUEST_URI} ^(/component/option,com) [NC,OR] ##optional - see
notes##
# to this:
# RewriteCond %{REQUEST_URI} ^(/test/component/option,com) [NC,OR] ##optional
- see notes##
#
######################################################

##  Can be commented out if causes errors, see notes above.
Options +FollowSymLinks

#  Uncomment following line if your webserver's URL
#  is not directly related to physical file paths.
#  Update Your JOOMLA!/MamboDirectory (just / for root)

# RewriteBase /

########## Begin - JOOMLA! core SEF Section
############## Use this section if using ONLY JOOMLA! core SEF
## ALL (RewriteCond) lines in this section are only required if you actually
## have directories named 'content' or 'component' on your server
## If you do not have directories with these names, comment them out.
#
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
#RewriteCond %{REQUEST_URI} ^(/component/option,com) [NC,OR]
      ##optional - see notes##
RewriteCond %{REQUEST_URI} (/|\.htm|\.php|\.html|/[^.]*)$  [NC]
RewriteRule ^(content/|component/) index.php
#
########## End - JOOMLA! core SEF Section

########## Begin - 3rd Party SEF Section
############## Use this section if you are using a 3rd party (Non JOOMLA! core)
SEF extension - e.g. OpenSEF, 404_SEF, 404SEFx, SEF Advance, etc
#
#RewriteCond %{REQUEST_URI} ^(/component/option,com) [NC,OR]
      ##optional - see notes##
#RewriteCond %{REQUEST_URI} (/|\.htm|\.php|\.html|/[^.]*)$  [NC]
#RewriteCond %{REQUEST_FILENAME} !-f
#RewriteCond %{REQUEST_FILENAME} !-d
#RewriteRule (.*) index.php
#
########## End - 3rd Party SEF Section

########## Begin - Rewrite rules to block out some common exploits
## If you experience problems on your site block out the operations listed
below
## This attempts to block the most common type of exploit `attempts` to
JOOMLA!
#
# Block out any script trying to set a mosConfig value through the URL
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z_]{1,21}(=|\%3D) [OR]
# Block out any script trying to base64_encode crap to send via URL
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [OR]
```

```
# Block out any script that includes a <script> tag in URL
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
# Block out any script trying to set a PHP GLOBALS variable via URL
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
# Block out any script trying to modify a _REQUEST variable via URL
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2})
# Send all blocked request to homepage with 403 Forbidden error!
RewriteRule ^(.*)$ index.php [F,L]
#
########## End - Rewrite rules to block out some common exploits
```