



UNIVERSIDAD
POLITECNICA
DE VALENCIA



UNIVERSIDAD POLITÉCNICA DE VALENCIA

Escuela técnica superior de informática aplicada

MEMORIA DEL PROYECTO FINAL DE CARRERA



APORTAL WEB DE
YUNTAMIENTO

Autor:
Daniel Alcalá Alcocer

Director:
Félix Buendía García



A mi madre,

que ostenta el máximo tiempo (varios años a decir verdad)
persistiendo en su afán por que que concluyera este proyecto.

A mi esposa,

más persistente que la anterior si cabe,
consiguiendo lo mismo en menor tiempo.

A mi padre,

por defenderme de las 2 primeras,
aunque tuvo que recurrir a su gran elocuencia
cuando comprobó que el proyecto se demoraba *in extremis*.

A mi hermana,

por resultar la más democrática y darme la calma,
al saber hacerme ver a su manera, que por muchos años que pasaran,
este día al fin llegaría.

A la Universidad Politécnica de Valencia,
y en especial a la Escuela Técnica Superior de Ingeniería Informática,
por la trepidante juventud que allí pasé
y los grandes recuerdos que me han dejado.



Índice

1. INTRODUCCIÓN.....	5
1.1. Propósito y objetivos.....	5
1.2. Motivaciones.....	6
1.3. Estructura.....	7
2. ESPECIFICACIÓN DE REQUISITOS.....	9
2.1. Introducción.....	9
2.1.1. Propósito.....	9
2.1.2. Ámbito.....	10
2.1.3. Definiciones, acrónimos y abreviaturas.....	10
2.1.4. Referencias.....	14
2.1.5. Visión global.....	15
2.2. Descripción general.....	15
2.2.1. Perspectiva del producto.....	15
2.2.2. Funciones del producto.....	16
2.2.3. Características del usuario.....	18
2.2.4. Restricciones generales.....	20
2.2.5. Supuestos y dependencias.....	20
2.3. Requisitos específicos.....	21
2.3.1. Requisitos de interfaces externos.....	21
2.3.2. Requisitos funcionales.....	23
2.3.3. Atributos.....	49
3. ANÁLISIS.....	51
3.1. Introducción.....	51
3.2. Casos de uso.....	52
3.3. Diagramas de clases.....	59
3.4. Diagramas de secuencia.....	62
3.4.1. Escenario 1: Autenticarse.....	63
3.4.2. Escenario 2: Buscar trámites.....	64
3.4.3. Escenario 3: Crear un nuevo usuario.....	65
3.4.4. Escenario 4: Eliminar un trámite.....	65
3.4.5. Escenario 5: Recuperar contraseña olvidada.....	68
4. DISEÑO.....	69
4.1. Introducción.....	69
4.2. Nivel de interfaz.....	71



4.3. Nivel de negocio.....	73
4.4. Nivel de almacenamiento.....	76
4.4.1. Diagrama Entidad-Relación.....	76
4.4.2. Diseño lógico.....	79
5. IMPLEMENTACIÓN.....	83
5.1. Introducción.....	83
5.2. Tecnologías.....	85
5.2.1. HTML.....	85
5.2.2. CSS.....	86
5.2.3. PHP.....	91
5.2.4. JavaScript.....	92
5.2.5. MySQL.....	94
5.3. Herramientas.....	95
5.3.1. Apache.....	95
5.3.2. Komodo Edit.....	96
5.3.3. Adobe Photoshop.....	97
5.3.4. FileZilla.....	99
5.4. Detalles de implementación.....	101
5.4.1. Implementación del interfaz.....	101
5.4.2. Implementación de la funcionalidad.....	115
5.4.3. Implementación de la gestión de datos.....	165
6. EVALUACIÓN Y PRUEBAS.....	171
7. CONCLUSIONES.....	179
8. BIBLIOGRAFÍA.....	181



1 .Introducción .-

En los tiempos que corren, es posible afirmar sin temor a equivocación que en Internet, quien no está, no existe. Y es que cada vez son menos los reticentes a cuestionar tal pensamiento. Si bien constituye una afirmación que puede escucharse con gran rotundidad en un entorno empresarial o de negocio, hoy día resulta también factible aplicarla en una amplia variedad de contextos, como el que ahora nos ocupa, el de un Ayuntamiento.

Sería especialmente de lamentar, no aprovechar todas las posibilidades que Internet brinda para aproximar Ayuntamiento y ciudadanos: mirar, participar, gestionar, resolver, expresar o compartir, son sólo algunas de las principales motivaciones que miles de usuarios encuentran a diario.

En esta primera década del siglo XXI, la creación o renovación de un portal web para un ayuntamiento propicia un escenario privilegiado de comunicación, un transitado lugar de encuentro para ciudadanos, donde el ayuntamiento sale reforzado como principal protagonista, al servicio de los mismos. Lo contrario sería rechazar la evolución, significaría en resumidas cuentas, lo que esbozaba la frase que abría esta breve introducción: dejar de tener relevancia, caer en el olvido, dejar de existir.

1.1. Propósito y objetivos .-

El presente documento pretende poner de manifiesto las diferentes etapas que comportan el desarrollo y creación de un portal web para un ayuntamiento, sirviendo el mismo como memoria de un Proyecto de Final de Carrera.



A groso modo, se pretenden cubrir los siguientes objetivos principales:

- El portal deberá adaptarse a los hábitos de utilización de cada usuario, por lo que la programación deberá ser eminentemente dinámica, posibilidad que brinda la tecnología PHP. En este sentido, deben mostrarse de forma privilegiada aquellos elementos más utilizados por cada usuario, en detrimento de otros por los que el mismo muestre menor interés.
- El portal web deberá resultar intuitivo y potente, ofreciendo al usuario la información que busca de forma coherente y en el menor tiempo posible.
- Deberá brindar servicios diferenciados en virtud del rol del usuario. Distinguirá por tanto varios niveles de perfiles de usuario.
- Tratará de forma permanente de "cautivar" al usuario no registrado (Anónimo), ofreciendo los elementos persuasivos necesarios para que el mismo tome la iniciativa de registrarse y pasar a formar parte de la comunidad de usuarios. En este sentido, se limitará al máximo el acceso a la diversas secciones del portal, si bien se le restringirá su participación (una frase que sintetiza bien la estrategia a seguir sería "podrá ver pero no tocar", para eso tendría que registrarse)
- El portal deberá presentar un carácter marcadamente interactivo. De este modo, deben aparecer apartados de esta índole, por ejemplo: foros de discusión o bien gestión de trámites.

1.2 Motivaciones -

Entre los diversos factores que han podido influenciarme en la decisión de afrontar un proyecto de la envergadura del que nos ocupa, los englobaría en 2 bloques principales: aquellos orientados a colaborar en mi desarrollo personal por un lado, y los que propician una formación óptima para el desarrollo profesional por el otro.



De este modo, conseguiría cubrir una parte de mis inquietudes personales, esto es, hacer servir Internet para dar a conocer y compartir mi afición por la composición musical, alcanzando el nivel de dominio necesario sobre las tecnologías actuales en programación y diseño web. Tal dominio llegaría casi de forma natural, fruto del esfuerzo y rigurosidad que impone el desarrollo de un proyecto final de carrera como el que nos ocupa, es decir, la creación de un portal web de ayuntamiento, el cual supone un marco incomparable para poder involucrar una amplia variedad de recursos tecnológicos tanto de programación como de diseño web.

Finalmente, lograría asimilar por otro lado, los conocimientos técnicos necesarios que me permitirían desarrollar una profesión cuya presencia en casi todos los sectores, durante esta primera década de siglo, ha ido *"in crescendo"* y que cada día que transcurre, continua consolidándose y convergiendo hacia una profesión de gran interés y demanda.

1.3. Estructura

El presente documento está constituido por una serie de capítulos, siguiendo un enfoque basado en el modelo de "cadena de valor", esto es, una serie consecutiva de actividades o etapas que progresivamente van adicionando un valor al proyecto.

De esta forma, la primera etapa, ya abordada, ha sido la propuesta del proyecto, donde se han definido las motivaciones y los objetivos.

La siguiente etapa consiste en la Especificación de requisitos, que no es mas que una descripción de las necesidades del proyecto, atendiendo a las opiniones conseguidas por parte de los usuarios finales.

Posteriormente aparecerá la etapa de Análisis y Diseño donde serán descritos nuevamente los requisitos anteriores pero desde una perspectiva más cercana al desarrollador de la aplicación web, estableciéndose aquí una primera toma de contacto con los modelos formales para su representación.



La etapa de Implementación recoge las diversas tecnologías recurridas con el objetivo de obtener una primera fase de productos o prototipos funcionales. Es posible reutilizar en este apartado, aquellas aplicaciones de terceros a las que se hayan recurrido.

Posteriormente es posible encontrar la etapa de Evaluación. Es en esta etapa donde la aplicación web será sometida a una batería de pruebas para testear a fondo que la aplicación discurra según lo previsto, en cualquiera de sus vertientes posibles.

Finalmente el documento concluye con la etapa de Conclusiones, donde se documenta el trabajo realizado durante todo el desarrollo del proyecto, los resultados alcanzados así como una breve argumentación de las experiencias contraídas. Por último, esta etapa contempla la inclusión de posibles ampliaciones realizables en un futuro.

Es preciso asimismo documentar las principales fuentes de información en un apartado de Bibliografía. Deben referenciarse todas las fuentes o materiales consultados durante el desarrollo del proyecto, ya sean libros, apuntes o páginas Web.



2. Especificación de requisitos .-

La Especificación de requisitos debe recoger todas aquellas características a cumplir por nuestra aplicación para satisfacer las expectativas o exigencias del Ayuntamiento, en el caso que nos ocupa, así como de los usuarios finales.

2.1. Introducción .-

La descripción de los requisitos debe realizarse tratando de conservar una perspectiva cercana al usuario final, evitando términos o tecnicismos que habitualmente emplearía un desarrollador. Con este fin, se emplea habitualmente un documento que reúne de forma ordenada estos requisitos y que se conoce como Especificación de Requisitos Software (ERS), el cual puede tener diferentes formatos.

2.1.1. Propósito .-

En el proyecto que nos ocupa, se ha optado por organizar los requisitos capturados empleando una ERS en base al estándar IEEE 830-1998, el cual puede ser utilizado como referencia en la especificación de proyectos Web, dadas las analogías existentes con la metodología de proyectos software.

Esta decisión está orientada en favor no sólo de estructurar los requisitos definidos previamente, sino también en pro de obtener una especificación consistente y no ambigua.

Esta especificación de requisitos representa una parte vital dentro de la memoria, pues el desarrollo de apartados posteriores deberán cumplir y satisfacer las exigencias de cada uno de estos requerimientos de partida.



2.1.2. **Ámbito .-**

El producto objeto del proyecto, debe ser una aplicación web fácilmente adaptable a cualquier ayuntamiento. En el caso que nos ocupa, los contenidos versarán en torno al Ayuntamiento de Paterna.

El portal debe denotar un marcado carácter interactivo y amigable, apto para abarcar la mayor tipología de público objetivo posible. Deberá prevalecer durante todo el desarrollo del mismo, la importancia de que sus elementos principales deben "estudiar" y adaptarse al comportamiento de cada usuario, invitando de este modo a permanecer el mayor tiempo posible en él.

De obligado cumplimiento será la inclusión de todo tipo de informaciones de carácter más estático, como pudieran ser la estructura jerárquica del ayuntamiento o una guía de la ciudad, como no menos obligado debe ser la integración de apartados de gran dinamismo, muy demandados por otro lado en la actualidad, como aquellos que permiten al ciudadano gestionar trámites cómodamente desde su hogar, expresarse libremente en foros de discusión o bien disponer de las noticias o eventos siempre actualizados.

Por último, la aplicación deberá contemplar un apartado para la administración de sus contenidos, esto es, aquel personal del ayuntamiento o externo, responsables de mantener actualizada la información de gran parte de los apartados del website. En este sentido, el portal presentará una interface intuitiva y segura, que confine la coherencia de la información almacenada en la base de datos.

2.1.3. **Definiciones, acrónimos y abreviaturas .-**

Apache: servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.



Autenticarse: acción de introducir el identificador de usuario (Username) y la contraseña (Password) asociada al mismo, de forma que el sistema pueda verificar si los valores almacenados en la Base de Datos son los mismos, en cuyo caso, los dará por válidos e iniciará una serie de acciones, programadas previamente por el diseñador web.

Base de Datos o BD: conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En el contexto en que nos encontramos, su almacenamiento se produce de forma digital (electrónica). En lo sucesivo en este documento, nos referimos a la Base de Datos como BD.

CSS (Cascade Style Sheets): lenguaje que describe cómo deben ser presentadas por pantalla u otros dispositivos de salida, las diferentes páginas de un website. Permite la separación forma-contenido de los documentos escritos en HTML.

E-mails: servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente (también denominados mensajes electrónicos o emails) mediante sistemas de comunicación electrónicos.

Foro: también conocido como foro de mensajes, foro de opinión o foro de discusión, es una aplicación web que da soporte a discusiones u opiniones en línea.

FTP (File Transfer Protocol): protocolo para intercambiar archivos en Internet. Se utiliza principalmente para descargar un archivo de un servidor o para subir un archivo a un servidor a través de Internet.

Hipertexto: cualquier texto disponible en el WWW que contenga enlaces con otros documentos.

HTML (HyperText Markup Language): lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP (Hypertext Transfer Protocol): protocolo usado en cada transacción de la World Wide Web. Está orientado a transacciones y sigue el esquema petición-respuesta



entre un cliente y un servidor.

HTTPS (Hypertext Transfer Protocol Secure): Protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto. Garantiza un nivel de seguridad mayor que el de HTTP.

IEEE (The Institute of Electrical and Electronics Engineers): asociación técnico-profesional mundial dedicada a la redacción y aprobación de normas que se establecen para garantizar el acoplamiento de elementos construidos independientemente.

Interface de usuario: representa la forma y disposición visual de los elementos en pantalla (texto, imágenes, objetos gráficos) que posibilitan la interacción de un usuario en un programa o aplicación.

Javascript: lenguaje interpretado (no requiere compilación) utilizado en páginas Web, con una sintaxis parecida a Java. Permite el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

MySQL: sistema de gestión de base de datos, desarrollado bajo la filosofía del software libre.

Navegador web: programa que permite visualizar la información que contiene una página web.

Newsletter: boletín informativo electrónico, distribuido de forma regular y típicamente mediante envío de e-mails, generalmente centrado en un tema principal que es del interés de sus suscriptores.

Oficina Virtual: espacio del portal en el que pueden desarrollarse actividades similares a las de una oficina. Los Usuarios Residenciales podrán iniciar tramitaciones y los Usuarios Interinos y/o Administradores podrán aprobarlas o denegarlas.

Página web: documento creado en formato HTML que es parte de un grupo de



documentos hipertexto o recursos disponibles en el World Wide Web. Una serie de páginas web componen lo que se llama un sitio web.

Password o contraseña: serie secreta de caracteres que permite a un usuario, junto a su Username, identificarse en el portal, "conociendo" éste los servicios e información personalizada que debe mostrar al mismo.

PHP: lenguaje de programación usado normalmente para la creación de contenido para sitios Web. Genera contenido dinámico, esto es, generado en el momento de visualizarla o solicitarla a un servidor.

Post o mensaje: cada uno de los mensajes publicados en los foros de discusión.

Servidor: ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras. También se entiende como el software que configura un PC como servidor para facilitar el acceso a la red y sus recursos.

Sesión: secuencia de páginas web que un usuario visita en un sitio web, desde que entra en nuestro sitio, hasta que lo abandona.

Tramitaciones o gestiones: están asociadas a alguno de los posibles trámites y deben haber sido previamente iniciadas por un Usuario Residencial. Sus posibles estados son: Iniciado, En Revisión, Aprobado o Denegado.

Trámites: cada una de las gestiones que un Usuario Residencial podría iniciar. Algunos trámites llevan asociado un impreso descargable. También existen trámites que requieren una tramitación presencial y por tanto no son tramitables de forma electrónica.

Topic o tema: cada uno de los temas principales de un foro de discusión. Normalmente, cuando se crea un nuevo tema, sistemáticamente es creado también un nuevo y primer post o mensaje para dicho topic o tema.

Username o nombre de usuario: palabra con la que un usuario es reconocido en el portal.



Usuario Administrador (ADM): usuario que además de estar registrado, tiene el máximo nivel de control sobre la administración de los contenidos del portal.

Usuario Interino (INT): usuario que además de estar registrado, tiene control sobre la administración de parte de los contenidos del portal.

Usuario Anónimo: cualquier usuario que acceda a nuestro portal Web y no se haya autenticado.

Usuario Registrado (USU): usuario que tras introducir su Username y Password, ha sido validado por la base de datos y autenticado por tanto correctamente.

Usuario Residencial (RES): usuario que además de estar registrado, tiene acceso a los servicios de tramitación online (Oficina Virtual).

Web, WWW o World Wide Web: sistema de documentos de hipertexto enlazados y accesibles a través de Internet.

Website: serie de Páginas web que generalmente conforman un concepto único o temática.

2.1.4. Referencias .-

A continuación se relacionan las fuentes de información consultadas para la redacción de la Especificación de Requisitos:

- Álvarez, M.A.
Desarrollo Web, Tu mejor ayuda para aprender a hacer webs.
(<http://www.desarrolloweb.com>).
Accedido en sept.2009.
- Buendía, F.
Una guía para la realización y supervisión de proyectos final de carrera (PFC) en el ámbito de la web.



Ed. UPV, 2008.

- Sanger L., Wales J.
Wikipedia, La enciclopedia libre.
(<http://es.wikipedia.org>).
Accedido en sept.2009.

2.1.5. Visión global .-

A continuación, la Especificación de Requisitos tratará de ofrecer una visión genérica sobre los diferentes funcionalidades o subapartados de la aplicación.

Posteriormente, una descripción de los requisitos específicos será necesaria.

2.2. Descripción general .-

El objetivo del presente apartado pretende ofrecer una visión lo más generalizada posible de la aplicación, evitando incurrir en descripciones pormenorizadas o mas específicas, evitando así innecesarias modificaciones en futuras versiones del documento.

2.2.1. Perspectiva del producto .-

El producto a desarrollar no se integrará en un sistema más amplio. De este modo, funcionará de forma autónoma sobre cualquier plataforma, valiéndose para ello de cualquier navegador web de actualidad y obviamente, de una conexión a Internet.



Las tecnologías principales empleadas en su implementación han sido:

- Lenguaje PHP, en gran medida, dado el carácter eminentemente dinámico.
- Lenguaje HTML, en menor medida.
- Plantillas CSS para optimizar la legibilidad de la información mostrada, así como para propiciar un diseño actual, acorde a las últimas tendencias.
- JavaScript de forma puntual, para moldear la forma en que mostraremos algunas páginas web que emergerán sobre otras (efecto pop-up) o bien, visualizar el tiempo, servicio ofrecido por 'www.tutiempo.net'.
- MYSQL como sistema de gestión de base de datos.

Será necesario por tanto un servidor donde alojar el website, que soporte las tecnologías citadas anteriormente, siendo recomendable asimismo una descripción de dominio natural y fácil de recordar por la mayoría de los potenciales usuarios.

2.2.2. Funciones del producto .-

Las funciones ofrecidas por el portal en función de la tipología de usuario y que pueden ser distinguidas son las siguientes :

- Acceso a la información referente al municipio y al ayuntamiento (parte más estática del website).
- Operaciones sobre noticias:
 - Visualización de noticias.
 - Creación, edición y eliminación de noticias.
- Operaciones sobre eventos de la agenda:
 - Visualización de eventos.
 - Creación, edición y eliminación de eventos.



- Operaciones sobre oposiciones:
 - Visualización y descarga de bases.
 - Creación, edición y eliminación de oposiciones.

- Operaciones sobre trámites y gestiones (tramitaciones) :
 - Visualización de trámites y descarga de impresos.
 - Creación, edición y eliminación de trámites.
 - Filtros de búsqueda de trámites.
 - Visualización del estado de las gestiones (tramitaciones).
 - Gestión de tramitaciones (iniciación, modificación de su estado y eliminación).
 - Filtro de búsqueda de tramitaciones en base al usuario.

- Operaciones sobre foros de discusión:
 - Visualización de los posts de un determinado topic.
 - Creación y eliminación de posts.
 - Creación y eliminación de topics.

- Operaciones sobre usuarios:
 - Visualización de usuarios y toda la información asociada en su registro.
 - Registro y autenticación de usuarios.
 - Modificación de la información del registro, así como de sus preferencias.
 - Modificación del password y envío de uno nuevo en caso de olvido.
 - Creación, edición y eliminación de usuarios.
 - Cierre de sesión de usuario.

- Gestión y seguimiento de los apartados más utilizados por cada usuario:
 - Seguimiento permanente del sistema de los hábitos de uso de cada usuario.
 - Ubicación dinámica de los diversos apartados del portal atendiendo a los hábitos de uso de cada usuario.

- Gestión y envío automático de boletines:
 - Usuarios registrados podrán subscribirse al servicio de envío de boletines informativos.
 - El sistema enviará automáticamente por e-mail un boletín a todos los suscritos



cada vez que sean añadidos un total de 10 nuevos elementos (noticias y/o eventos).

- Implementación de un motor de búsqueda interno:
 - Definición de uno y hasta dos términos simultáneos para la búsqueda.
 - Opción de búsqueda en todo el website o sólo en la página/sección actual.

2.2.3. Características del usuario .-

Atendiendo a la tipología de los usuarios que van a utilizar la aplicación, los mismos pueden ser clasificados como sigue :

- **Usuario Anónimo**
 - Cualquier visitante esporádico o no, el cual no se haya autenticado.
 - La estrategia a seguir para este tipo de usuario será la de tratar permanentemente de persuadirlo para que termine registrándose en el portal. En este sentido, se le permitirá visualizar los diversos apartados del website, pero sin posibilidad de interactuar (podrá ver pero no participar). De forma más detallada :
 - Tendrá acceso total al apartado de Noticias y Agenda, pero no podrá recibir boletines informativos (newsletters) sobre las mismas.
 - En el apartado de Trámites, podrá visualizarlos y descargar impresos, pero no podrá iniciar tramitaciones.
 - En el apartado de Foros, podrá leer mensajes pero no podrá responder a los mismos, ni crear nuevos temas.
 - Tendrá acceso a toda la información de la parte más estática del portal (Ayuntamiento, Municipio, Servicios), además de a los apartados de El Tiempo y Callejero.
- **Usuario Registrado (USU)**
 - Cualquier usuario que previamente haya completado el proceso de registro y se haya autenticado en el portal.



- Es importante destacar que en el apartado de Trámites, dispondrá de las mismas opciones que un Usuario Anónimo. Esto es así porque para poder gestionar tramitaciones, es preciso introducir un DNI o CIF en el momento del registro, en cuyo caso, el apartado de "Trámites" pasaría a ser mostrado como "Oficina Virtual".
- Dispondrá de las mismas posibilidades de actuación que un Usuario Anónimo, además de las siguientes:
 - Los elementos de la página principal cambiarán su ubicación en función de las preferencias que tenga por los mismos. Para esto, el sistema "estudiará" desde el primer momento, todos sus "movimientos".
 - Dispondrá de la opción de suscribirse al servicio de Newsletter.
 - En el apartado de Foros, podrá leer mensajes así como responderlos e iniciar nuevos temas de discusión.
- **Usuario Residencial (RES)**
 - Cualquier usuario que previamente haya completado el proceso de registro introduciendo DNI en el momento del mismo, expresando así su deseo de disponer del servicio de Oficina Virtual, habiéndose autenticado asimismo en el portal.
 - Dispondrá de las mismas opciones que un Usuario Registrado, además de poder gestionar tramitaciones (Oficina Virtual).
- **Usuario Interino (INT)**
 - Este tipo de usuario sólo puede ser creado por un Usuario Administrador.
 - El rol que cumple es la propia de un funcionario del ayuntamiento, es decir, dispondrá de un interface dedicado con el objetivo de gestionar los contenidos del portal, como sigue:
 - Visualizar, dar de alta y editar: noticias, eventos de la agenda, trámites y oposiciones.
 - Gestionar tramitaciones previamente iniciadas por un Usuario Residencial (podrá visualizarlas, añadir notificaciones y modificar su estado).
 - Visualizar el Website como si de un Usuario Anónimo se tratase, con el fin de verificar que los cambios introducidos son mostrados de la forma esperada.



- **Usuario Administrado (ADM)**

- Este tipo de usuario sólo puede ser creado por otro Usuario Administrador.
- Dispondrá de las mismas opciones que un Usuario Interino, además de las siguientes :
 - Visualizar, dar de alta y eliminar usuarios.
 - Eliminar noticias, eventos de la agenda, trámites y/o oposiciones.

2.2.4. Restricciones generales .-

Como ya se ha comentado anteriormente, no existirá dependencia de un sistema mayor y bastará una computadora con un navegador web y una conexión Internet para tener acceso al portal.

Obviamente serán adoptadas consideraciones orientadas a garantizar la seguridad e integridad de la información, como son la encriptación de las contraseñas de los usuarios, medidas contra ataques de inyección SQL o también la edición conveniente del archivo de configuración para PHP.

Por otro lado, se hace altamente recomendable alojar el website en un servidor con HTTPS.

2.2.5. Supuestos y dependencias .-

El portal será visualizado de forma correcta a través de un navegador web compatible con el protocolo HTTP 1.0 o 1.1, JavaScript y con plantillas CSS, esto es, la amplia mayoría de los navegadores de actualidad.

Como también se indicara anteriormente, el servidor donde esté alojado deberá cumplir unos requisitos mínimos como son: soporte para el lenguaje PHP y para administrar bases de datos MYSQL.



2.3. Requisitos específicos

En este apartado serán descritos los requisitos, atendiendo al nivel de detalle preciso que permita abordar con éxito las siguientes etapas del proyecto, esto es:

- descripción de las entradas ("inputs") al sistema.
- descripción del comportamiento o de las funciones adoptados por el sistema, en reacción a las entradas ("inputs") o bien para cubrir las exigencias de una posible respuesta.
- descripción de las respuestas ("outputs") producidas por el sistema.

2.3.1. Requisitos de Interfaces externos

Interfaces de usuario

Atendiendo a la tipología de usuario, pueden diferenciarse 2 tipos de interface de usuario, como sigue :

Usuarios no administradores (Anónimos, Registrados e Residenciales):

Dispondrán de una página web principal, cuyo espacio se distribuye del siguiente modo:

- Sendas barras de navegación situadas tanto en la parte superior como en el pie de página, incluyendo repetición de enlaces directos a algunas de los apartados principales: Trámites y Gestiones, Sugerencias, Agenda, Aviso Legal, etc.
- También en la parte superior, barra de navegación principal con acceso a los apartados principales (parte más dinámica) del portal: Noticias, Agenda, Trámites, Foros.
- Debajo de la barra de navegación principal anterior, se encuentra la caja de búsquedas, la cual ofrecerá la opción de buscar en todo el portal (única



opción disponible si nos encontramos en la página principal) o buscar sólo en el apartado actual (Noticias, Agenda, Trámites o Foros).

- En la zona lateral izquierda, barra de navegación vertical con acceso a los apartados principales (parte más estática) sobre información del Ayuntamiento, Municipio y Servicios.
- La zona central (tronco de la página principal), estará dedicado a mostrar un resumen de los últimos elementos incorporados a cada uno de los apartados principales: Noticias, Agenda o Foros.
- La zona lateral derecha estará dedicada a mini-secciones de funciones de visualización rápida como son: calendario del mes con días marcados si existen eventos, callejero, el tiempo con 3 días de previsión.
- La zona lateral derecha superior siempre estará dedicada al área de autenticación de usuarios, modificación del perfil de usuario y su contraseña, así como cierre de la sesión.
- Todo el ancho de la zona inferior albergará una subsección "Secciones al detalle", mostrando las últimas oposiciones publicadas dando opción a descargar sus bases, así como descarga también de los últimos impresos publicados o información sobre los últimos usuarios registrados (Comunidad).

Usuarios administradores (Administradores e Interinos):

Dispondrán de una página web principal, cuyo espacio quedará distribuido del siguiente modo:

- En la zona más alta de cabecera, se dispondrá de sendos enlaces: uno poder visualizar el website como si de un Usuario Anónimo se tratase y otro para acceder a la página principal de administración.
- En la zona superior, barra de navegación principal con acceso a los apartados principales de administración: Oficina Virtual, Noticias, Eventos, Oposiciones y Usuarios (ésta última no disponible para Interinos).
- El área lateral izquierda superior estará reservada para las diversas opciones de gestión y que irán variando según el apartado principal seleccionado en la barra de navegación principal superior.
- El área central principal (tronco de la página principal) estará dedicado a



mostrar las informaciones propias de cada gestión en curso.

- Sendas barras de navegación tanto en la parte superior como a pie de página, con repetición de enlaces directos a algunas de los apartados principales: Trámites y Gestiones, Sugerencias, Agenda, Aviso Legal, etc.

Interfaces de hardware

Como ha sido mencionado anteriormente, los requisitos de hardware necesarios para el portal serán un servicio de alojamiento web con soporte para las tecnologías PHP y MYSQL, así como una computadora con conexión a Internet, capaz de ejecutar un navegador web, para los usuarios que accederán al portal.

Interfaces de software

El producto es desarrollado bajo el sistema operativo Windows XP SP3, sobre el cual se ha instalado y configurado un servidor Apache.

Asimismo los lenguajes de programación utilizados son PHP y HTML, junto con plantillas CSS, y JavaScript en menor medida.

El sistema de gestión de base de datos correrá a cargo del sistema archiconocido MySQL.

2.3.2. Requisitos funcionales .-

Siguiendo una clasificación basada en la tipología de usuario, a continuación se muestra una descripción detallada de los distintos requisitos funcionales:

Usuario Anónimo

Conviene recordar que este tipo de usuario, es aquel que accede a nuestro portal de forma esporádica o no, el cual probablemente representa un potencial "cliente" para el portal del Ayuntamiento. La estrategia de "comportamiento" del portal en este caso,



debe focalizarse en tratar de captar la atención del mismo, hasta el grado de involucrarlo en el proceso del registro, pasando así a formar parte finalmente de la Comunidad de Usuarios.

2.3.2.1. Registrarse en el portal	
Introducción	El Usuario decide formar parte de la Comunidad de Usuarios registrados, probablemente atraído por alguno de los servicios exclusivos para los mismos (información mostrada de forma llamativa mediante iconos animados)
Entradas	Cada uno de los campos del formulario de registro.
Proceso	El sistema muestra el formulario de alta de nuevo usuario. Si el usuario confirma su envío, el sistema verifica que hayan sido cumplimentados todos los campos marcados como obligatorios, así como la corrección del formato en algunos de ellos. Si las condiciones anteriores se cumplen, el nuevo usuario es insertado en la Base de Datos (BD), cerciorándose previamente el sistema de que no exista duplicidad en los campos Username y Email. Asimismo es añadida una tupla a la tabla 'preferencias_usuarios' inicializando las puntuaciones. Previamente se realiza un control anti-inyección SQL.
Salidas	Texto informando de que el proceso fue exitoso. Se ofrece un enlace redirigiendo al usuario a la página principal, donde el área de Usuarios exhibirá su nombre y el resto de opciones.

2.3.2.2. Autenticarse en el portal.	
Introducción	Para identificarse en el portal e iniciar así un sesión, será suficiente con que el usuario ingrese su Username y Password en el área de Usuarios.
Entradas	Los datos Username y Password ingresados por el usuario.
Proceso	El sistema contrasta la información recibida contra la Base de Datos. En caso de encontrar una tupla que coincida, valida la operación, inicia una sesión y registra algunas variables de sesión con algunos datos de frecuente consulta sobre el usuario. Finalmente el usuario es redirigido a la página de la sección desde donde se inició la autenticación (si el Usuario tuviera la categoría de Administrador o Interino es redirigido a la página principal de admón).
Salidas	Mensaje de bienvenida, redirigiendo al usuario a la página correspondiente, en virtud del criterio expresado en el apartado anterior.
	En caso de fallo del proceso de autenticación, se sugiere al usuario volver a intentarlo.



2.3.2.3.	Recuperar la contraseña en caso de olvido.
Introducción	El usuario pretende autenticarse sin éxito, debido a que no recuerda la contraseña. Bastará con que ingrese su Username y Password para recibir una nueva contraseña aleatoria por email, que posteriormente podrá modificar a conveniencia.
Entradas	Los datos Username e E-mail ingresados por el usuario.
Proceso	existan en una misma tupla, en cuyo caso, generará una contraseña aleatoria basada en un diccionario con miles de palabras, almacenado en un archivo. A continuación, la nueva contraseña es almacenada en la BD y enviada por email al Usuario para su conocimiento. Si los datos ingresados por el usuario no son encontrados en ninguna tupla, se informa al usuario y se le invita a volver a intentarlo.
Salidas	Se informa al usuario que revise su correo electrónico donde habrá recibido una nueva contraseña.
	Se invita al usuario a volver intentarlo con motivo de que algunos de los datos ingresados no es correcto.

2.3.2.4.	Consultar eventos de un día del calendario.
Introducción	El usuario puede conocer todos los eventos programados para un día en concreto, simplemente haciendo click en un día del calendario, el cual deberá estar marcado de forma destacada como no vacío. Los días sin eventos no dispondrán de enlace alguno. Asimismo, el usuario puede cambiar de mes y año.
Entradas	El día seleccionado en el calendario, así como el año y mes seleccionados en los campos junto al calendario.
Proceso	Se recupera de la BD toda la información referente a eventos existentes en el día seleccionado.
Salidas	Ventada emergente mostrado el detalle de los eventos programados para el día seleccionado.



2.3.2.5. Visualizar mapa/callejero.	
Introducción	El usuario puede consultar el mapa integrado en la página principal del portal, interactuando directamente sobre el mismo (zoom, desplazamientos, etc). Puede también acceder a una versión ampliada del mismo, ubicada fuera del portal. El servicio es ofrecido por Google Maps.
Entradas	Eventos de ratón enviados por el usuario.
Proceso	Interpretación de los eventos de ratón enviados por el usuario y respuesta a los mismos.
Salidas	Mapa modificado en tiempo real (zoom, desplazamientos) en relación a los eventos de ratón.
	Se muestra en pantalla una versión del mapa ampliada, con el ayuntamiento de la población preseleccionada.

2.3.2.6. Consultar el tiempo.	
Introducción	El usuario puede visualizar el tiempo con una predicción a 3 días. También puede ampliar la información (predicción de 7 días e información detallada) haciendo click en la imagen del tiempo integrada en la página principal del portal. El servicio es ofrecido por www.tutiempo.net .
Entradas	Click en la imagen del tiempo.
Proceso	El usuario es redirigido al website de www.tutiempo.net .
Salidas	Se muestra el website de www.tutiempo.net , con la información ampliada referida a la población de Paterna.

2.3.2.7. Contactar con el Ayuntamiento.	
Introducción	El usuario dispone de un aparatado de Sugerencias para hacer llegar al Ayuntamiento cualquier tipo de comentario.
Entradas	Datos del formulario enviados por el usuario.
Proceso	El sistema verifica que hayan sido cumplimentados todos los campos marcados como obligatorios, así como la corrección del formato en algunos de ellos. A continuación envía un correo al servicio de atención al cliente del Ayuntamiento, incluyendo los datos de contacto del usuario así como el texto de la sugerencia.
Salidas	Mensaje informando que el envío fue exitoso y que en breve será atendido. En el caso de que se produzca algún error en el envío del e-mail, se invita al usuario a volver a intentarlo pasados unos minutos.



2.3.2.8.	Iniciar búsquedas selectivas dentro del portal.
Introducción	El usuario dispone de un buscador de cualquier información perteneciente a cualquiera de las secciones del portal. Puede buscar empleando una sólo palabra o incluso 2 palabras. También puede restringir la búsqueda a la sección en la que se encuentre o bien en todo el portal, en cuyo caso, un resumen de resultados por sección será mostrado (única opción disponible si el usuario se encuentra en la página principal).
Entradas	La palabra o palabras introducidas por el usuario, así como la opción de búsqueda en en todo el portal o sólo en la sección actual.
Proceso	El sistema creará la consulta a la BD en función del número de palabras introducidas por el usuario, así como del tipo de búsqueda (total o selectiva). Si la búsqueda es compleja (2 palabras), la consulta a la BD estará basadas en índices FULLTEXT en modo booleano. Asimismo si se trata de una búsqueda selectiva, se consultará el valor de la variable de sesión <code>\$_SERVER["PHP_SELF"]</code> para conocer la única sección en la que debe buscarse.
Salidas	Página con los resultados hallados. Si se tratase de una búsqueda total, sería mostrado un resumen por secciones. Todos los resultados contendrán enlaces al mismo o a la sección a la que pertenece, con el fin de ampliar la información.

2.3.2.9.	Visualizar las últimas noticias.
Introducción	El usuario puede visualizar un resumen con las últimas noticias destacadas. Para ello, bastará con que seleccione la sección Noticias y el enlace Ver todas las noticias en la página principal.
Entradas	Enlace seleccionado por el usuario.
Proceso	El sistema busca las últimas noticias introducidas en la BD, extrayendo toda la información de las mismas y ordenando el resultado desde la más moderna a la más antigua.
Salidas	Página con las últimas noticias, mostrando todos los detalles de las mismas.

2.3.2.10.	Visualizar en detalle una noticia.
Introducción	El usuario puede visualizar una sólo noticia con todos sus detalles (texto completo, foto, etc) pulsando sobre el titular de la misma, ya se encuentre en la página principal o en el apartado de Noticias o en el de resultado de una búsqueda.
Entradas	Identificador de la noticia seleccionada por el usuario.
Proceso	El sistema busca la noticia en la BD y extrae toda la información necesario de la tupla encontrada.
Salidas	Ventada emergente mostrando todas las informaciones disponibles sobre la noticia.



2.3.2.11.	Visualizar todos los eventos del mes.
Introducción	El usuario puede visualizar un resumen con todos los eventos programados en la agenda para el mes actual o bien para otro mes que el mismo seleccione.
Entradas	Enlace de la sección 'Agenda' seleccionado por el usuario, o bien formulario con los valores de un mes y un año enviados por el mismo.
Proceso	Si la entrada es el enlace seleccionado por el usuario, el sistema busca en la BD, todos los detalles referidos a los eventos programados para el mes actual. Si la entrada es el formulario, el sistema ceñiría la búsqueda a los eventos programados para los valores del formulario.
Salidas	Página con todos los detalles de cada evento del mes actual o bien del mes seleccionado mediante el formulario. Se muestra también un nuevo formulario para seleccionar otra fecha.

2.3.2.12.	Visualizar en detalle un evento.
Introducción	El usuario puede visualizar un sólo evento con todos sus detalles pulsando sobre el título del mismo, en el resumen de eventos mostrados en la página principal.
Entradas	Identificador del evento seleccionado por el usuario.
Proceso	El sistema busca el evento en la BD y extrae toda la información necesaria de la tupla encontrada.
Salidas	Ventana emergente mostrando toda la información disponible sobre el evento.

2.3.2.13.	Visualizar trámites disponibles.
Introducción	El usuario puede buscar un determinado trámite con tan sólo aplicar unos filtros de búsqueda. Es posible buscar por tipología de trámite y también es posible indicar que sean listados sólo aquellos con opción de tramitación electrónica (Oficina Virtual).
Entradas	Formulario de búsqueda enviado por el usuario, junto con los filtros seleccionados por el mismo.
Proceso	El sistema muestra el formulario de búsqueda. Si el usuario confirma su envío, el sistema construye una consulta a la BD en función de los filtros seleccionados, obteniendo aquellos trámites que cumplen con los mismos.
Salidas	El sistema muestra una página con un listado de todos los trámites que reúnen las condiciones de búsqueda seleccionadas por el usuario, así como un enlace a cada uno de ellos con el fin de mostrar información más detallada.



2.3.2.14.	Visualizar en detalle un trámite.
Introducción	El usuario puede visualizar todos los detalles de un trámite pulsando sobre el nombre del mismo, en el resumen de trámites mostrados en la sesión de Trámites. También muestra un enlace para descarga el impreso asociado al mismo.
Entradas	Identificador del trámite seleccionado por el usuario.
Proceso	El sistema busca el trámite en la BD y extrae toda la información necesaria de la tupla encontrada.
Salidas	Ventada emergente mostrando toda la información disponible sobre el trámite, así como un enlace al impreso para su descarga, en caso de que el trámite lleve asociado el mismo.

2.3.2.15.	Visualizar temas (topics) del foro.
Introducción	EL usuario puede consultar el título de todos los temas que existen en el foro, así como el Username de su propietario, la fecha en que fue creado y la cantidad de mensajes que contiene.
Entradas	Enlace "Foros" o "Ver Temas" seleccionado por el usuario.
Proceso	El sistema obtiene de la BD la información sobre cada tema.
Salidas	Listado de todos los títulos de los temas (topics) del foro, así como el resto de informaciones descritas en la Introducción. Cada título es un enlace que permite visualizar los mensajes contenidos en el tema.

2.3.2.16.	Visualizar todos los mensajes (posts) de un tema.
Introducción	El usuario puede consultar los mensajes de cada tema del foro y sus fechas de publicación, así como el Username y antigüedad del propietario de cada uno de ellos.
Entradas	Identificador del tema seleccionado por el usuario.
Proceso	El sistema obtiene de la BD la información sobre todos los mensajes del tema seleccionado por el usuario.
Salidas	Listado con todos los mensajes del tema seleccionado, así como el resto de informaciones descritas en el apartado Introducción.



2.3.2.17.	Mostrar alguna de las secciones sobre el Ayuntamiento (Ayuntamiento, Municipio o Servicios)-parte más estática-
Introducción	El usuario puede consultar cualquiera de las informaciones referentes al Ayuntamiento o Municipio (Saludo del alcalde, órganos de gobierno, Guía de Paterna, Tfnos de interés,etc)
Entradas	Enlace seleccionado por el usuario, en menú lateral dedicado al Ayuntamiento y al Municipio
Proceso	El sistema obtiene de la BD la información sobre el apartado seleccionado por el usuario.
Salidas	Página con la información sobre el apartado del Ayuntamiento o del Municipio seleccionado por el usuario

2.3.2.18.	Mostrar fecha y día de la semana actuales.
Introducción	El usuario puede visualizar permanentemente el nombre del día de la semana, así como la fecha completa actuales, en la parte superior derecha del portal.
Entradas	Ninguna
Proceso	El sistema actualiza la información sobre la fecha y día de la semana cada vez que es cargada una página.
Salidas	Permanentemente es visualizado el nombre del día de la semana, así como la fecha completa actuales, en la parte superior derecha del portal.

2.3.2.19.	Visualizar listado de oposiciones.
Introducción	El usuario puede visualizar todas las últimas oposiciones publicadas así como descargar sus bases.
Entradas	Enlace seleccionado en el apartado "Secciones al detalle" de la página principal.
Proceso	El sistema obtiene de la BD la información sobre cada una de las oposiciones vigentes.
Salidas	Ventana emergente mostrando el detalle de todas las oposiciones publicadas, así como un enlace que permite descargar las bases de cada una de ellas.



2.3.2.20.	Visualizar últimos impresos publicados.
Introducción	En el apartado "Secciones al detalle" de la página principal, de forma permanente se muestran los últimos impresos publicados, así como un enlace que permite su descarga.
Entradas	Enlace seleccionado para iniciar la descarga del impreso.
Proceso	El sistema obtiene de la BD la información sobre los últimos impresos añadidos.
Salidas	Listado con los últimos impresos añadidos junto con un enlace que permite su descarga.

2.3.2.21.	Visualizar últimos usuarios registrados.
Introducción	En el apartado "Secciones al detalle" de la página principal, de forma permanente se muestran el Username de los últimos usuarios que se unieron a la comunidad de usuarios registrados (no usuarios administradores ni interinos), así como la fecha con que lo hicieron. (Forma parte de la estrategia captación de nuevos usuarios).
Entradas	Ninguna
Proceso	El sistema obtiene de la BD el Username y la fecha de antigüedad de los últimos usuarios que se registraron cuya categoría no sea la de Administrador ni tampoco la de Interino.
Salidas	Listado con los últimos usuarios registrados (Username y fecha de registro).

2.3.2.22.	Consultar el Aviso Legal.
Introducción	A pie de página en el menú de navegación, existe un enlace permanente al texto de Aviso Legal.
Entradas	Enlace 'Aviso Legal' seleccionado por el usuario.
Proceso	El sistema abre una ventana emergente, imprimiendo en sus interior el texto del Aviso Legal que se encuentra en un fichero con extensión PHP.
Salidas	Ventana emergente mostrando el texto del Aviso Legal.



Usuario Registrado

Conviene recordar que este usuario dispone de las posibilidades de un Usuario Anónimo, además de las descritas a continuación.

2.3.2.23.	Cambiar contraseña.
Introducción	El usuario puede en cualquier momento cambiar su contraseña de autenticación por otra, en el área de Usuarios.
Entradas	Formulario cumplimentado por el usuario con la contraseña antigua, la nueva contraseña y la confirmación de la nueva contraseña.
Proceso	El sistema muestra el formulario de cambio de contraseña. Si el usuario confirma su envío, el sistema consulta a la BD para verificar que la contraseña antigua sea correcta. Comprueba asimismo que la nueva contraseña coincida con su confirmación. Finalmente, actualiza la nueva contraseña en la Base de Datos e informa al usuario que el proceso concluyó con éxito.
Salidas	Mensaje indicando que la contraseña fue cambiada con éxito o que por el contrario, existe algún error en los datos que introdujo.

2.3.2.24.	Modificar perfil y preferencias de usuario.
Introducción	El usuario puede editar los datos de registro así como alterar sus preferencias para el servicio de Newsletter (en caso q estuviera suscrito).
Entradas	Formulario cumplimentado con los campos modificados a conveniencia del usuario.
Proceso	El sistema muestra el formulario de modificación al usuario. Si éste confirma su envío, el sistema verifica que los campos modificados presenten el formato adecuado. Asimismo comprueba que en caso de haber modificado el Username y/o Email, éstos no existiesen ya. A continuación, actualiza la tupla en la Base de Datos con los nuevos valores y registra nuevamente las variables de sesión de frecuente consulta sobre el usuario. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que la modificación concluyó con éxito o bien, que el proceso fue cancelado.



2.3.2.25.	Cerrar sesión.
Introducción	El usuario puede cerrar la sesión activa en cualquier instante desde el área de Usuarios.
Entradas	Enlace "Cerrar Sesión" seleccionado por el usuario, el cual envía una variable logout empleando el método \$_GET.
Proceso	Es destruido el identificador de sesión así como las variables de sesión.
Salidas	Se muestra la página principal con las opciones propias de un Usuario Anónimo.

2.3.2.26.	Reubicar automáticamente las secciones de la pág. principal
Introducción	Cada vez que un usuario consigue autenticarse, se obtienen de la BD su histórico de preferencias de uso de cada sección principal del portal. Atendiendo a este histórico, cada vez que es mostrada la página principal, ésta reubicará sus subsecciones, dando preferencia de visibilidad a aquellas con mayor puntuación.
Entradas	Los datos de autenticación enviados desde la página principal o bien el enlace 'Inicio' seleccionado por el usuario.
Proceso	El sistema obtiene de la BD las puntuaciones para el usuario autenticado, referentes al acumulado de ocasiones que el mismo ha visitado cada una de las secciones principales del portal. En función de estas puntuaciones, en la página principal serán mostrados unos apartados antes que otros.
Salidas	La página principal mostrará los elementos de cada sección, mostrando de forma preferente aquellos que el usuario visitó más veces, atendiendo al criterio expresado en el punto anterior.

2.3.2.27.	Aumentar la puntuación de preferencia de una sección.
Introducción	Cada usuario dispone de una puntuación para cada sección, en función de la cantidad de veces que las visita o utiliza. De esta forma, cuando un usuario visita un sección, es aumentado en 1 unidad su contador de puntuación asociado.
Entradas	Ninguna.
Proceso	El sistema obtiene de la BD la puntuación asociada a la sección que el usuario acaba de visitar, le adiciona 1 unidad y actualiza su valor en la BD.
Salidas	Ninguna



2.3.2.28.	Subscribirse al servicio de Newsletter.
Introducción	El usuario puede subscribir al servicio de envío de boletines. Dispone de 3 modalidades: recibir sólo noticias, sólo eventos o ambos. Para esto, el usuario debe pulsar sobre un icono animado situado en la página principal (sólo es mostrado cuando el usuario no se ha suscrito aún al servicio) el cual enlaza con un formulario que permite subscribirse seleccionando antes el tipo de Newsletter.
Entradas	Formulario con las opciones de subscripción seleccionadas por el usuario.
Proceso	El sistema muestra el formulario de subscripción. Si el usuario confirma su envío, verifica que se haya seleccionado al menos una de las opciones, en cuyo caso actualiza en la BD la tupla de la tabla 'preferencias_usuarios' con este valor.
Salidas	Se muestra un mensaje informando que la subscripción concluyó con éxito o bien que el proceso fue cancelado.

2.3.2.29.	Abrir un nuevo topic (tema) en los foros.
Introducción	El usuario puede abrir un nuevo tema de discusión, adicionando además un primer mensaje o no.
Entradas	Formulario con el título del tema y (opcionalmente) texto del primer mensaje (post).
Proceso	El sistema muestra un formulario con el título del tema y texto del mensaje por cumplimentar. Si el usuario confía su envío, se comprueba que el título del tema no esté vacío, en cuyo caso analiza si existe además un primer mensaje. Seguidamente inserta en la BD el nuevo topic, así como el mensaje, si lo hubiera. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario de que el tema fue creado correctamente, listándose a continuación todos los temas del foro. Si el usuario no introdujo algún título para el tema, se le informa de que esta circunstancia es imprescindible para continuar.

2.3.2.30.	Crear un nuevo post (mensaje) en un tema del foro.
Introducción	El usuario puede crear un mensaje asociado a algún tema del foro, respondiendo al mensaje de otro usuario o bien simplemente aportando una opinión.
Entradas	Formulario con el texto del mensaje introducido por el usuario.
Proceso	El sistema comprueba que el texto del mensaje no esté vacío, en cuyo caso inserta una nueva tupla en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario de que el mensaje fue creado correctamente, listándose a continuación todos los mensajes del tema asociado. Si el usuario envió el formulario el blanco, se le informa del error.



2.3.2.31.	Eliminar un mensaje del foro.
Introducción	El usuario puede eliminar un mensaje del foro, siempre y cuando sea propietario del mismo. El sistema no dará la opción de borrado si no se cumple esta condición.
Entradas	Enlace 'Eliminar mensaje' seleccionado por el usuario, el cual lleva asociado el identificador del mensaje a eliminar.
Proceso	El sistema elimina de la DB la tupla asociada al identificador del mensaje.
Salidas	Se informa al usuario de que el mensaje fue eliminado correctamente, listándose a continuación todos los mensajes del tema asociado.

2.3.2.32.	Eliminar un tema del foro.
Introducción	El usuario puede eliminar un tema del foro, siempre y cuando sea propietario del mismo y no existan mensajes de otros usuarios en el mismo. El sistema no dará la opción de borrado si no se cumplen las condiciones anteriores.
Entradas	Enlace 'Eliminar tema' seleccionado por el usuario, el cual lleva asociado el identificador del tema a eliminar.
Proceso	Se elimina de la BD la tupla correspondiente al tema seleccionado. Si existen mensajes del propietario del tema, también son eliminados de la BD.
Salidas	Se informa al usuario de que el tema fue eliminado correctamente, listándose a continuación los temas del foro.

Usuario Residencial

Conviene recordar que este usuario dispone de las posibilidades de un Usuario Registrado, además de la funcionalidad de Oficina Virtual.



2.3.2.33.	Visualizar tramitaciones o gestiones.
Introducción	El usuario puede visualizar todas las tramitaciones iniciadas previamente, así como su estado actual (Iniciado, Revisión, Aprobado o Denegado). El sistema dará la opción de eliminar aquéllas que aún no hayan sido revisadas así como de visualizar en detalle cada una de ellas.
Entradas	Enlace 'Gest. Tramitaciones' o 'Oficina Virtual' seleccionado por el usuario.
Proceso	Se obtienen de la BD todas las tramitaciones cuyo propietario sea el usuario autenticado.
Salidas	Se muestra un listado con todas las tramitaciones que haya iniciado el usuario autenticado, así como el estado de cada una de ellas (Iniciado, Revisión, Aprobado o Denegado), su fecha de iniciación, la opción de eliminar la tramitación si ésta se encuentra en estado de Iniciado así como también la posibilidad de ver todos los detalles de la misma.

2.3.2.34.	Visualizar en detalle una tramitación o gestión.
Introducción	El usuario puede visualizar todos los detalles de una tramitación, esto es, el histórico de notificaciones usuario-funcionario, el estado de la misma, así como toda la información del trámite asociado.
Entradas	Enlace 'Ver estado' seleccionado por el usuario, el cual lleva asociado el identificador de la tramitación a visualizar.
Proceso	Se obtienen de la BD la información de la tramitación seleccionada por el usuario, así como la información del trámite a que corresponde la tramitación.
Salidas	Se muestra en una ventana emergente, todos los detalles de la tramitación seleccionada, es decir, el histórico de notificaciones usuario-funcionario, el estado de la misma y la información del trámite asociado.



2.3.2.35.	Eliminar una tramitación o gestión.
Introducción	El usuario puede eliminar una tramitación que hubiera iniciado previamente, siempre y cuando no estuviera en revisión, es decir, su estado fuera de Iniciado, Aprobado o Denegado.
Entradas	Formulario de confirmación de borrado de tramitación, enviado por el usuario.
Proceso	El sistema muestra un formulario de confirmación de borrado. Si éste es confirmado, consulta a la BD el estado de la tramitación a eliminar para comprobar que no sea el de "Revisión" (control del acceso concurrente ya que un Administrador podría haber cambiado su estado de Iniciado a Revisión), en cuyo caso, elimina de la BD la tupla asociada a la tramitación.
Salidas	Se muestra un mensaje informando que la eliminación concluyó con éxito o bien que el proceso fue cancelado.

2.3.2.36.	Editar una tramitación o gestión.
Introducción	El usuario puede añadir nuevas notificaciones a una tramitación, siempre y cuando no haya sido aprobada o denegada por un funcionario del Ayuntamiento. Estas notificaciones son el recurso disponible para comunicarnos con el funcionario durante el curso de la gestión de la tramitación.
Entradas	Formulario con el texto de la notificación a adjuntar a la tramitación.
Proceso	El sistema muestra en una ventana emergente el histórico actual de notificaciones así como un formulario para añadir una nueva notificación. A continuación actualiza la tramitación en BD, habiendo añadido previamente el nuevo texto de notificación. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario de que la tramitación fue modificada con éxito o bien, que el proceso fue cancelado.



2.3.2.37. Iniciar una tramitación o gestión.	
Introducción	El usuario puede iniciar una nueva tramitación. Para ello, deberá realizar previamente una búsqueda de trámites empleando filtros. El resultado de la búsqueda mostrará un listado de trámites disponibles, dando el sistema las siguientes opciones sobre cada uno de ellos: <ul style="list-style-type: none">- Visualizar todos sus detalles.- Iniciar su tramitación (siempre y cuando disponga de opción de tramitación electrónica).
Entradas	En primer lugar, formulario con los filtros de búsqueda seleccionados por el usuario. A continuación, formulario de confirmación de iniciación de tramitación, con el texto (opcional) de notificación a adjuntar.
Proceso	El sistema muestra formulario de búsqueda de tramitaciones. Posteriormente, inserta en la BD una nueva tramitación, así como el texto de notificación si lo hubiera. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario de que la tramitación se inició correctamente.

Usuario Interino

Conviene recordar que este tipo de usuario es el responsable de manipular la información sobre los contenidos del portal. El único matiz que lo diferencia con el Usuario Administrador es que no dispone de opciones para eliminar ninguna información, así como de gestionar usuarios.

2.3.2.38. Visualizar tramitaciones pendientes de resolución.	
Introducción	El usuario podrá visualizar todas las tramitaciones pendientes de resolución de un usuario determinado, mostrando diferentes opciones de actuación para cada una de ellas. Previamente habrá realizado una búsqueda filtrando por el nombre de usuario (Username).
Entradas	Formulario de búsqueda con el Username seleccionado por el usuario.
Proceso	El sistema obtiene de la BD las tramitaciones cuyo estado sea 'Iniciado' o 'Revision', del usuario seleccionado en el formulario de búsqueda
Salidas	Listado con las tramitaciones (nombre, estado y fecha de iniciación) pendientes de resolución del usuario seleccionado, ofreciendo sobre cada una de ellas, la siguiente opción: <ul style="list-style-type: none">- Visualizar sus detalles.- Modificar su estado, añadiendo opcionalmente una notificación.



2.3.2.39.	Modificar una tramitación.
Introducción	El usuario podrá cambiar el estado de la tramitación a alguno de los estados posible (Revision, Aprobado, Denegado), así como opcionalmente anexar una notificación. Asimismo este usuario podrá conservar el estado de Revisión y añadir opcionalmente una notificación requiriendo alguna información del usuario que originó la tramitación.
Entradas	Formulario con el nuevo valor del estado de la tramitación, así como el texto a notificar.
Proceso	El sistema obtiene de la BD el histórico de notificaciones existente sobre la tramitación seleccionada por el usuario y lo muestra en una ventana emergente, acompañándolo de un formulario donde el usuario puede añadir una nueva notificación así como indicar el nuevo estado para la tramitación. En caso de que éste anexara una nueva notificación, el sistema lo adiciona al histórico obtenido antes. A continuación, se actualiza en la BD la tupla correspondiente a la tramitación, con el nuevo valor de estado y del histórico de notificaciones.
Salidas	Se informa al usuario que la modificación concluyó con éxito o bien, que el proceso fue cancelado.

2.3.2.40.	Visualizar trámites disponibles.
Introducción	El usuario puede obtener un listado de trámites disponibles, utilizando para ello un formulario de búsqueda. El listado resultado mostrará para cada trámite, opciones de edición o visualización en detalle.
Entradas	Formulario de búsqueda enviado por el usuario, junto con los filtros seleccionados por el mismo.
Proceso	El sistema muestra el formulario de búsqueda. Si el usuario confirma su envío, el sistema construye una consulta a la BD en función de los filtros seleccionados, obteniendo aquellos trámites que cumplen con los mismos.
Salidas	Listado con todos los trámites que reúnen las condiciones de búsqueda seleccionadas por el usuario, junto con algunas opciones de actuación según se explica en el apartado Introducción.



2.3.2.41.	Visualizar en detalle un trámite.
Introducción	El usuario puede visualizar todos los detalles de un trámite determinado, así como descargar el impreso asociado al mismo.
Entradas	Identificador del trámite seleccionado por el usuario.
Proceso	El sistema busca el trámite en la BD y extrae toda la información necesaria de la tupla encontrada.
Salidas	Ventada emergente mostrando toda la información disponible sobre el trámite, así como un enlace al impreso para su descarga, en caso de que exista un impreso asociado.

2.3.2.42.	Crear un nuevo trámite.
Introducción	El usuario puede dar de alta nuevos trámites, cumplimentando cada uno de los campos del formulario. Asimismo, es posible anexar un impreso que permita ser descargado posteriormente.
Entradas	Formulario cumplimentado y enviado por el usuario.
Proceso	El sistema muestra el formulario de apertura de nuevo trámite. Si el usuario confirma su envío, el sistema comprobará que estén cumplimentados los campos obligatorios. Si además anexó algún impreso, comprobará que el mismo haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema insertará el trámite en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que el trámite fue introducido con éxito o bien, que el proceso fue cancelado.

2.3.2.43.	Editar un trámite.
Introducción	El usuario puede modificar un trámite existente, así como cambiar el impreso asociado al mismo.
Entradas	Formulario de modificación enviado por el usuario.
Proceso	El sistema muestra el formulario de edición. Si el usuario confirma su envío, el sistema comprobará que estén cumplimentados los campos obligatorios. Si además anexó algún impreso, comprobará que el mismo haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema actualizará el trámite en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que la modificación concluyó con éxito o bien, que el proceso fue cancelado.



2.3.2.44. Visualizar noticias.	
Introducción	El usuario puede obtener un listado de todas las noticias disponibles, utilizando para ello un formulario de búsqueda. El listado resultado mostrará para cada noticia, opciones de edición y visualización en detalle.
Entradas	Formulario de búsqueda enviado por el usuario, junto con los filtros seleccionados por el mismo.
Proceso	El sistema muestra el formulario de búsqueda. Si el usuario confirma su envío, el sistema construye una consulta a la BD en función de los filtros seleccionados, obteniendo aquellas noticias que cumplen con los mismos.
Salidas	Listado con todas las noticias que reúnen las condiciones de búsqueda seleccionadas por el usuario, junto con algunas opciones de actuación según se explica en el apartado Introducción.

2.3.2.45. Visualizar en detalle una noticia.	
Introducción	El usuario puede visualizar todos los detalles de una noticia determinada.
Entradas	Identificador de la noticia seleccionada por el usuario.
Proceso	El sistema busca la noticia en la BD y extrae toda la información necesaria de la tupla encontrada.
Salidas	Ventana emergente mostrando toda la información disponible sobre la noticia, así como su foto en caso de que exista.

2.3.2.46. Crear una nueva noticia.	
Introducción	El usuario puede dar de alta nuevas noticias, cumplimentando cada uno de los campos del formulario. Asimismo, es posible anexar una foto a la misma.
Entradas	Formulario cumplimentado y enviado por el usuario.
Proceso	El sistema muestra el formulario de apertura de nueva noticia. Si el usuario confirma su envío, el sistema comprobará que estén cumplimentados los campos obligatorios. Si además anexó alguna foto, comprobará que la misma haya sido recibida correctamente. Si se cumplen las condiciones anteriores, el sistema insertará el trámite en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que la noticia fue introducida con éxito o bien, que el proceso fue cancelado.



2.3.2.47.	Editar una noticia.
Introducción	El usuario puede modificar una noticia existente, así como cambiar la foto asociada a la misma.
Entradas	Formulario de modificación enviado por el usuario.
Proceso	El sistema muestra el formulario de edición. Si el usuario confirma su envío, el sistema comprobará que estén cumplimentados los campos obligatorios. Si además anexó alguna foto, comprobará que la misma haya sido recibida correctamente. Si se cumplen las condiciones anteriores, el sistema actualizará la noticia en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que la modificación concluyó con éxito o bien, que el proceso fue cancelado.

2.3.2.48.	Visualizar eventos de la agenda.
Introducción	El usuario puede obtener un listado de todos los eventos disponibles, utilizando para ello un formulario de búsqueda. El listado resultado mostrará para cada evento, opciones de edición y visualización en detalle.
Entradas	Formulario de búsqueda enviado por el usuario, junto con los filtros seleccionados por el mismo.
Proceso	El sistema muestra el formulario de búsqueda. Si el usuario confirma su envío, el sistema construye una consulta a la BD en función de los filtros seleccionados, obteniendo aquellos eventos que cumplen con los mismos.
Salidas	Listado con todos los eventos que reúnen las condiciones de búsqueda seleccionadas por el usuario, junto con algunas opciones de actuación según se explica en el apartado Introducción.

2.3.2.49.	Visualizar en detalle un evento.
Introducción	El usuario puede visualizar todos los detalles de un evento determinado.
Entradas	Identificador del evento seleccionado por el usuario.
Proceso	El sistema busca el evento en la BD y extrae toda la información necesaria de la tupla encontrada.
Salidas	Ventana emergente mostrando toda la información disponible sobre el evento.



2.3.2.50.	Crear un nuevo evento.
Introducción	El usuario puede dar de alta nuevos eventos, cumplimentando cada uno de los campos del formulario.
Entradas	Formulario cumplimentado y enviado por el usuario.
Proceso	El sistema muestra el formulario de apertura de nuevo evento. Si el usuario confirma su envío, el sistema comprobará que estén cumplimentados los campos obligatorios, en cuyo caso, se insertará el evento en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que el evento fue introducido con éxito o bien, que el proceso fue cancelado.

2.3.2.51.	Editar un evento
Introducción	El usuario puede modificar un evento existente.
Entradas	Formulario de modificación enviado por el usuario.
Proceso	envío, el sistema comprobará que estén cumplimentados los campos obligatorios, en cuyo caso, se actualizará el evento en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que la modificación concluyó con éxito o bien, que el proceso fue cancelado.

2.3.2.52.	Visualizar oposiciones.
Introducción	El usuario puede obtener un listado de todas las oposiciones disponibles. El listado resultado mostrará para cada oposición, opciones de edición y visualización en detalle.
Entradas	Enlace "Gest. Oposiciones" o sección "Oposiciones", seleccionados por el usuario.
Proceso	El sistema obtiene de la BD toda la información sobre las oposiciones.
Salidas	Listado con todas las oposiciones, junto con algunas opciones de actuación según se explica en el apartado Introducción.



2.3.2.53.	Visualizar en detalle una oposición.
Introducción	El usuario puede visualizar todos los detalles de una oposición determinada, así como descargar sus bases si éstas existiesen.
Entradas	Identificador de la oposición seleccionada por el usuario.
Proceso	El sistema busca la oposición en la BD y extrae toda la información necesaria de la tupla encontrada.
Salidas	Ventada emergente mostrando toda la información disponible sobre la oposición, así como un enlace que permite la descarga de sus bases, en caso de que éstas existiesen.

2.3.2.54.	Crear un nueva oposición.
Introducción	El usuario puede dar de alta nuevas oposiciones, cumplimentando cada uno de los campos del formulario.
Entradas	Formulario cumplimentado y enviado por el usuario.
Proceso	El sistema muestra el formulario de apertura de nueva oposición. Si el usuario confirma su envío, el sistema comprobará que estén cumplimentados los campos obligatorios. Si además anexó el documento con las bases, comprobará que el mismo haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema insertará la oposición en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que la oposición fue introducida con éxito o bien, que el proceso fue cancelado.

2.3.2.55.	Editar una oposición.
Introducción	El usuario puede modificar una oposición existente, así como cambiar el documento con las bases asociado a la misma.
Entradas	Formulario de modificación enviado por el usuario.
Proceso	El sistema muestra el formulario de edición. Si el usuario confirma su envío, el sistema comprobará que estén cumplimentados los campos obligatorios. Si además anexó algún documento con las bases, comprobará que el mismo haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema actualizará la oposición en la BD. Previamente se produce un control anti-inyección SQL.
Salidas	Se informa al usuario que la modificación concluyó con éxito o bien, que el proceso fue cancelado.



2.3.2.56.	Enviar una newsletter a los subscriptores.
Introducción	De forma automática y transparente, es enviada una newsletter cada vez que son insertados 10 nuevos elementos (entre noticias y eventos). Especial atención requiere el hecho de que un usuario puede estar suscrito sólo en el apartado noticias, sólo en el de eventos o en ambos, por lo que debe recibir únicamente las últimas 10 incorporaciones según su condición.
Entradas	Ninguna.
Proceso	Cada vez que un usuario administrador introduce un nuevo elemento (noticia o evento) en la BD, el sistema incrementa los contadores asociados a cada tipo de newsletter (tabla 'control_newsletter'). A continuación, analiza si se cumple la condición de que con esta nueva incorporación, hayan sido introducidos ya un total de 10 nuevos elementos (bien sólo de noticias, bien sólo de eventos o entre ambos), en cuyo caso son enviados esos 10 elementos únicamente a aquellos usuarios suscritos a ese tipo de newsletter. Para esto, se obtendrán de la BD la información referida a esos 10 elementos para construir el boletín. Se obtendrán también los emails de los usuarios suscritos al tipo de newsletter en cuestión y finalmente será enviado el boletín.
Salidas	Aparece un mensaje en pantalla informando de que fue enviada una Newsletter.

Asimismo, el Usuario Interino dispone de las siguientes requisitos funcionales ya descritos anteriormente:

2.3.2.23.	Cambiar contraseña.
2.3.2.24.	Modificar perfil y preferencias de usuario.
2.3.2.25.	Cerrar sesión.



Usuario Administrador

Por último, conviene recordar que este usuario dispone de las posibilidades de un Usuario Interino, con el añadido añadidos de poder eliminar informaciones así como gestionar usuarios.

2.3.2.57.	Visualizar usuarios registrados en el portal.
Introducción	El usuario puede obtener un listado de todos los usuarios registrados en el portal. El listado resultado estará ordenado en orden alfabético por primer apellido y mostrará para cada usuario los siguientes datos: primer apellido, nombre, antigüedad, Username y categoría, así como opciones de eliminación y visualización en detalle.
Entradas	Enlace 'Gest. Usuarios' o sección 'Usuarios', seleccionados por el usuario.
Proceso	El sistema obtiene de la BD la información necesaria de cada usuario registrado en la misma.
Salidas	Listado ordenado alfabéticamente por apellido de todos los usuarios junto con algunas opciones de actuación según se explica en el apartado Introducción.

2.3.2.58.	Visualizar en detalle un usuario.
Introducción	El usuario puede visualizar todos los detalles de un usuario determinado a excepción de su contraseña.
Entradas	Username del usuario, seleccionado por el usuario.
Proceso	El sistema busca el usuario en la BD y extrae toda la información necesaria de la tupla encontrada.
Salidas	Ventada emergente mostrando toda la información disponible sobre el usuario a excepción de su contraseña.



2.3.2.59. Crear un nuevo usuario.	
Introducción	El usuario puede dar de alta nuevos usuarios, bien con privilegios de administración de contenidos (Administrador o Interino), bien con privilegios de Registrado o Residencial.
Entradas	Formulario cumplimentado y enviado por el usuario.
Proceso	El sistema muestra el formulario de nuevo usuario. Si el usuario confirma su envío, el sistema verifica que hayan sido cumplimentados todos los campos marcados como obligatorios, así como la corrección del formato en algunos de ellos. Si las condiciones anteriores se cumplen, el nuevo usuario es insertado en la Base de Datos (BD), cerciorándose previamente el sistema de que no exista duplicidad en los campos Username y Email. Asimismo es añadida una tupla a la tabla 'preferencias_usuarios' inicializando las puntuaciones. Previamente se realiza un control anti-inyección SQL.
Salidas	Se informa al usuario que el evento fue introducido con éxito o bien, que el proceso fue cancelado.

2.3.2.60. Eliminar un usuario.	
Introducción	El usuario puede eliminar usuarios de la BD a conveniencia.
Entradas	Formulario de confirmación de borrado de usuario, enviado por el usuario.
Proceso	El sistema muestra un formulario de confirmación de borrado. Si éste es confirmado, se elimina de la BD la tupla asociada al usuario.
Salidas	Se muestra un mensaje informando que la eliminación concluyó con éxito o bien que el proceso fue cancelado.

2.3.2.61. Eliminar un trámite.	
Introducción	El usuario puede eliminar un trámite (no así una tramitación o gestión) de la BD.
Entradas	Formulario de confirmación de borrado de usuario, enviado por el usuario.
Proceso	El sistema muestra un formulario de confirmación de borrado. Si éste es confirmado, se elimina de la BD la tupla asociada al trámite.
Salidas	Se muestra un mensaje informando que la eliminación concluyó con éxito o bien que el proceso fue cancelado.



2.3.2.62.	Eliminar una noticia.
Introducción	El usuario puede eliminar noticias de la BD.
Entradas	Formulario de confirmación de borrado de noticia, enviado por el usuario.
Proceso	El sistema muestra un formulario de confirmación de borrado. Si éste es confirmado, se elimina de la BD la tupla asociada a la noticia.
Salidas	Se muestra un mensaje informando que la eliminación concluyó con éxito o bien que el proceso fue cancelado.

2.3.2.63.	Eliminar un evento.
Introducción	El usuario puede eliminar eventos de la BD.
Entradas	Formulario de confirmación de borrado de evento, enviado por el usuario.
Proceso	El sistema muestra un formulario de confirmación de borrado. Si éste es confirmado, se elimina de la BD la tupla asociada al evento
Salidas	Se muestra un mensaje informando que la eliminación concluyó con éxito o bien que el proceso fue cancelado.

2.3.2.64.	Eliminar una oposición.
Introducción	El usuario puede eliminar oposiciones de la BD.
Entradas	Formulario de confirmación de borrado de oposición, enviado por el usuario.
Proceso	El sistema muestra un formulario de confirmación de borrado. Si éste es confirmado, se elimina de la BD la tupla asociada a la oposición.
Salidas	Se muestra un mensaje informando que la eliminación concluyó con éxito o bien que el proceso fue cancelado.

2.3.2.65.	Eliminar un mensaje del foro.
Introducción	El usuario puede eliminar cualquier mensaje del foro a conveniencia, con el fin de retirar comentarios ofensivos, o simplemente si se diera la circunstancia que el mensaje no guardase relación con el tema donde fue publicado.
Entradas	Enlace 'Eliminar mensaje' seleccionado por el usuario, el cual lleva asociado el identificador del mensaje a eliminar.
Proceso	Se elimina de la BD la tupla correspondiente al tema seleccionado. Si existen mensajes del propietario del tema, también son eliminados de la BD.
Salidas	Se informa al usuario de que el tema fue eliminado correctamente, listándose a continuación los temas del foro.



2.3.3. Atributos.-

Seguridad

La seguridad y consistencia de la información almacenada en la Base de Datos es un aspecto fundamental y de especial consideración dentro del proyecto. Por este motivo, se han adoptado diversas metodologías modernas en pro de evitar o, en el peor de los casos minimizar, las consecuencias que podría originar un percance en este apartado. A continuación se describen algunas de las buenas prácticas adoptadas :

1. Garantizaremos el acceso ordenado y confiable de la información a través de un sistema de autenticación de usuarios, donde cada usuario requerirá estar registrado en la base de datos con un nombre de usuario y una contraseña. Esta información estará almacenada en una tabla de la base de datos.
2. Las contraseñas serán almacenadas de forma encriptada en un sentido (no reversible) mediante la función de encriptación PASSWORD de MYSQL.
3. Las funciones propias de un tipo de usuario únicamente podrán ser ejecutadas por usuarios de su tipo. Esto se consigue consultando, previo a la ejecución de una función, la variable de sesión que hace referencia a la categoría (y por tanto a los privilegios) del usuario autenticado. Por ejemplo, un Usuario Registrado nunca podrá iniciar una tramitación (función sí disponible para un Usuario Residencial), puesto que previo a la ejecución de la función, se comprueba la categoría (variable de sesión) del usuario autenticado.
4. Activaremos la directiva magic_quotes_gpc del archivo de configuración PHP.INI, con el fin de evitar vulnerabilidades que podrían originarse a raíz de datos provenientes de los métodos GET y POST. Evitamos de este modo los escapes indeseados de información.
5. No se usarán variables de sesión por ser fácilmente modificables por el usuario. En su lugar, utilizaremos funciones y variables de sesión. Las funciones de sesión proporcionan un identificador único para el usuario.
6. Se eliminarán etiquetas HTML y PHP de las cadenas introducidas en formularios por el usuario, empleando la función *strip_tags* de PHP.
7. Para minimizar riesgos de ataques, nunca se enviará información de contenido o



de control importantes, a través de url (método GET). En su lugar, utilizaremos métodos POST.

8. La carga de librerías de funciones se hará de forma discriminada. De este forma, un usuario no administrador, no podrá "ver" funciones propias del mismo.
9. El archivo de conexión con la Base de Datos que almacena la contraseña y demás información que permite el enlace con la misma, estará almacenado fuera de la ruta del Website.

Mantenimiento

El mantenimiento de la información de contenido del portal (aquel que requiere de una actualización constante como por ejemplo el apartado de noticias o eventos, será responsabilidad de los usuarios con categoría de Administrativo o Interino, por tanto se tratará típicamente de personal del propio ayuntamiento.

Los cambios en el rediseño de la estructura de la base de datos o del grafismo del portal, así como la actualización de la información de la parte más estática del portal (Ayuntamiento, Municipio, etc) y resolución de incidencias eventuales, serán solicitados por personal del Ayuntamiento al ingeniero del website.



3. Análisis .-

Una vez han sido descritos los requisitos capturados, en el capítulo anterior, es preciso profundizar en alguna metodología que propicie la detección de aquellos apartados generales que guardan una relación directa con esos requisitos. Esta metodología debe tener la virtud de ser lo suficientemente abstracta como para conseguir mantener la independencia con los diversos recursos tecnológicos disponibles para su implementación, los cuales serán abordados en capítulos posteriores.

En la actualidad, existe una metodología ya consolidada y denominada RUP (*Rational Unified Process*), que cubre ampliamente las expectativas expresadas en el párrafo anterior. Esta metodología se apoya en modelos UML (*Unified Modeling Language*) que ayudan a describir la función, la arquitectura o el diseño del software.

3.1. Introducción .-

UML es el lenguaje de modelado de sistemas software más utilizado actualmente, respaldado por el OMG (Object Management Group), que sirve eficazmente para describir procesos haciéndose valor para ello de elementos gráficos. Permite visualizar, especificar, construir y documentar un sistema. Ofrece por tanto de un estándar que podemos hacer servir óptimamente en nuestra aplicación web.

UML dispone de diversos diagramas para identificar las diversas entidades del sistema a modelar. De entre todos ellos, nos basaremos en el Diagrama de Casos de Uso así como en el Diagrama de Clases para continuar con el desarrollo del proyecto que nos ocupa.

3.2. Casos de uso .-

Un Diagrama de Casos de Uso muestra cómo debería reaccionar el sistema con el usuario, o bien con otro sistema, para conseguir un objetivo. Se trata por tanto de una representación de la secuencia de acciones entre un sistema y sus actores, respondiendo a un evento que inicia uno de los actores principales.

El uso de este tipo de diagramas está ampliamente extendido debido a la notación relajada de la que hace uso, resultando por tanto muy cercano para el usuario final.

En la Fig. 3.2.1 se muestran los diversos actores identificados cuyas peculiaridades ya fueron descritas en apartados anteriores. El diagrama pone de manifiesto la especialización de algunos de los usuarios.

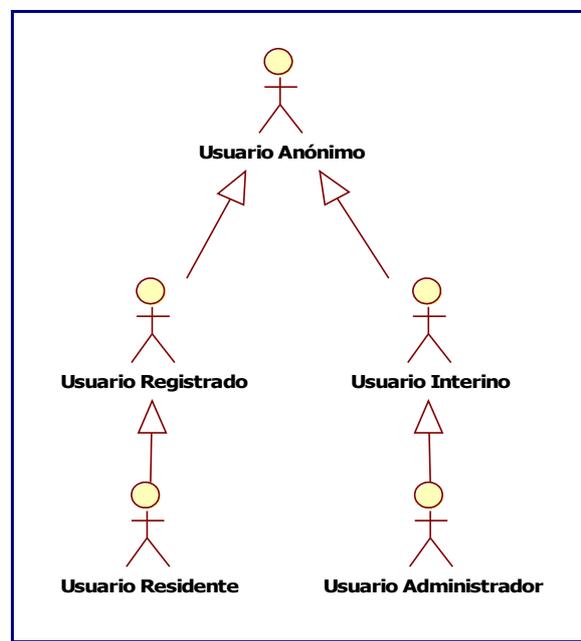


Fig. 3.2.1 - Actores principales del sistema

Expondremos a continuación los diferentes casos de uso para cada uno de los actores identificados.

Los casos de uso del Diagrama para el Usuario Registrado (Fig. 3.2.3) deberían incluir al Caso de Uso "Autenticarse". No obstante, se han obviado en pro de mejorar la legibilidad del mismo. En lo sucesivo para el resto de Diagramas que serán mostrados, se obviará representar este Caso de Uso con este fin, el de facilitar una interpretación ágil, salvo que su no inclusión sea contraproducente por propiciar resultados ambiguos.

Puede apreciarse en el Diagrama de Casos de Uso para el Usuario Registrado (Fig. 3.2.3), cómo algunos escenarios involucran sistemáticamente a otros (aquellos identificados con la etiqueta <<include>> o inclusión). Por ejemplo, para responder o eliminar un mensaje de un tema del foro, inevitablemente será preciso visualizar previamente los mensajes del tema en cuestión.

Del mismo modo, puede apreciarse en el mismo Diagrama, el uso de la etiqueta <<extend>> o extensión, para representar que diversos escenarios pueden desencadenarse en función de la condición que se cumpla (especificada entre corchetes []). Por ejemplo, el Caso de Uso "Modificar preferencias" involucraría actualizar la Base de Datos con la información enviada por usuario, siempre y cuando los datos del formulario estén correctos, o bien, implicaría informar al usuario de que existe un error en los datos introducidos, invitándole a volver a intentarlo.

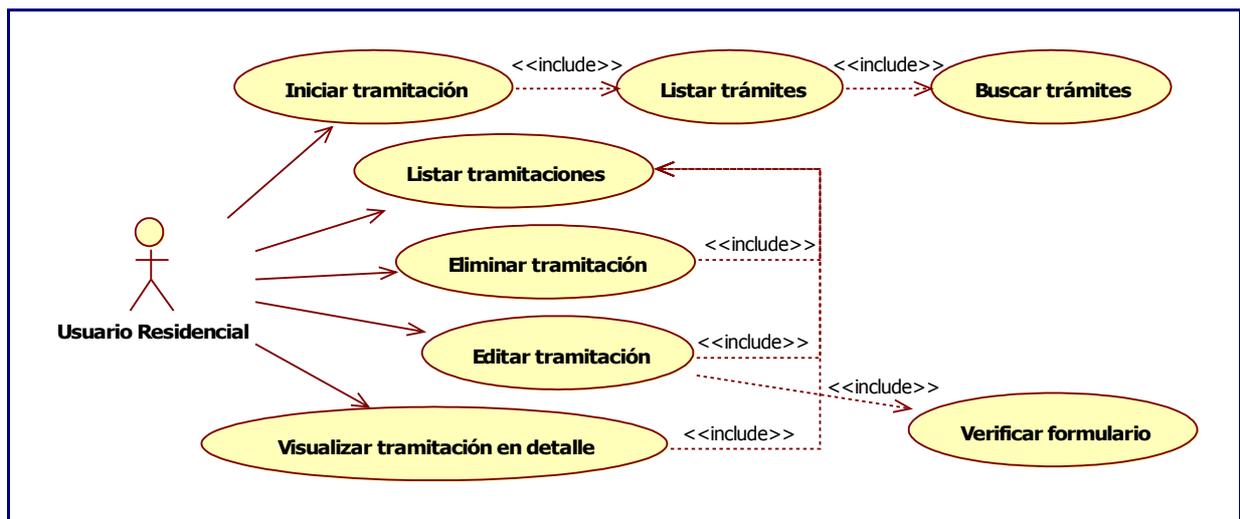


Fig. 3.2.4 – Casos de uso para el Usuario Residencial

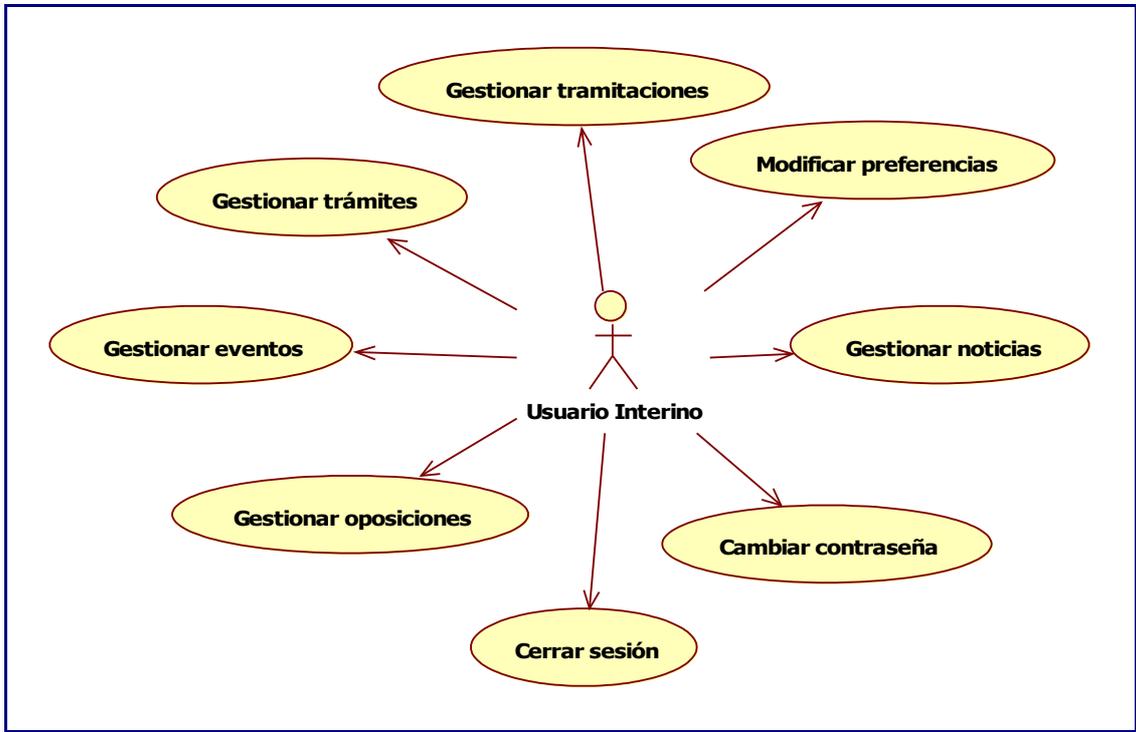


Fig. 3.2.5 – Visión general Casos de Uso para el Usuario Interino

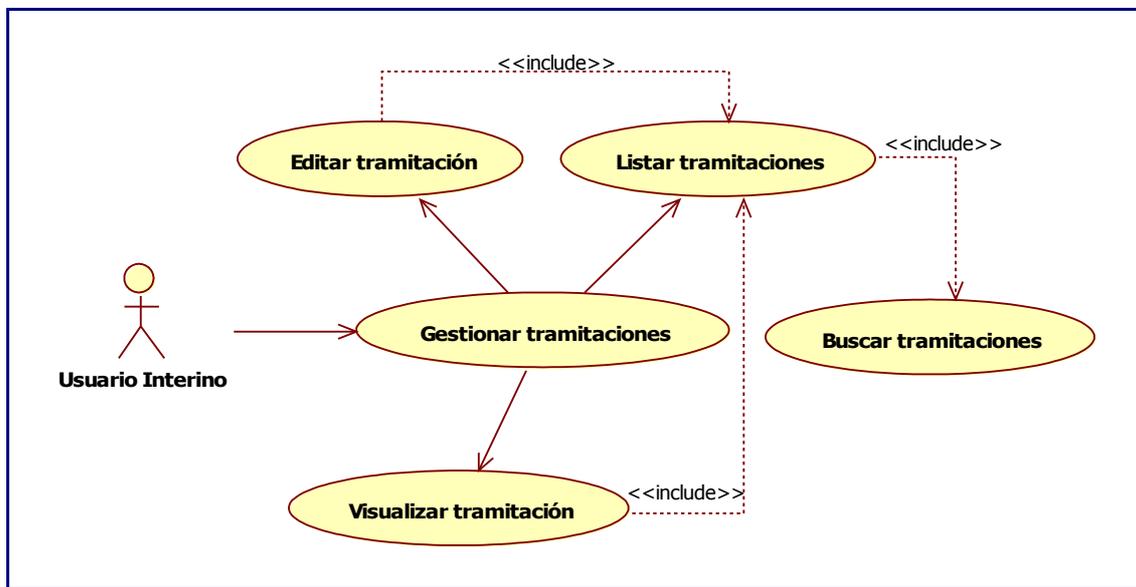


Fig. 3.2.6 – Visión en detalle del Caso de Uso "Gestionar tramitaciones"

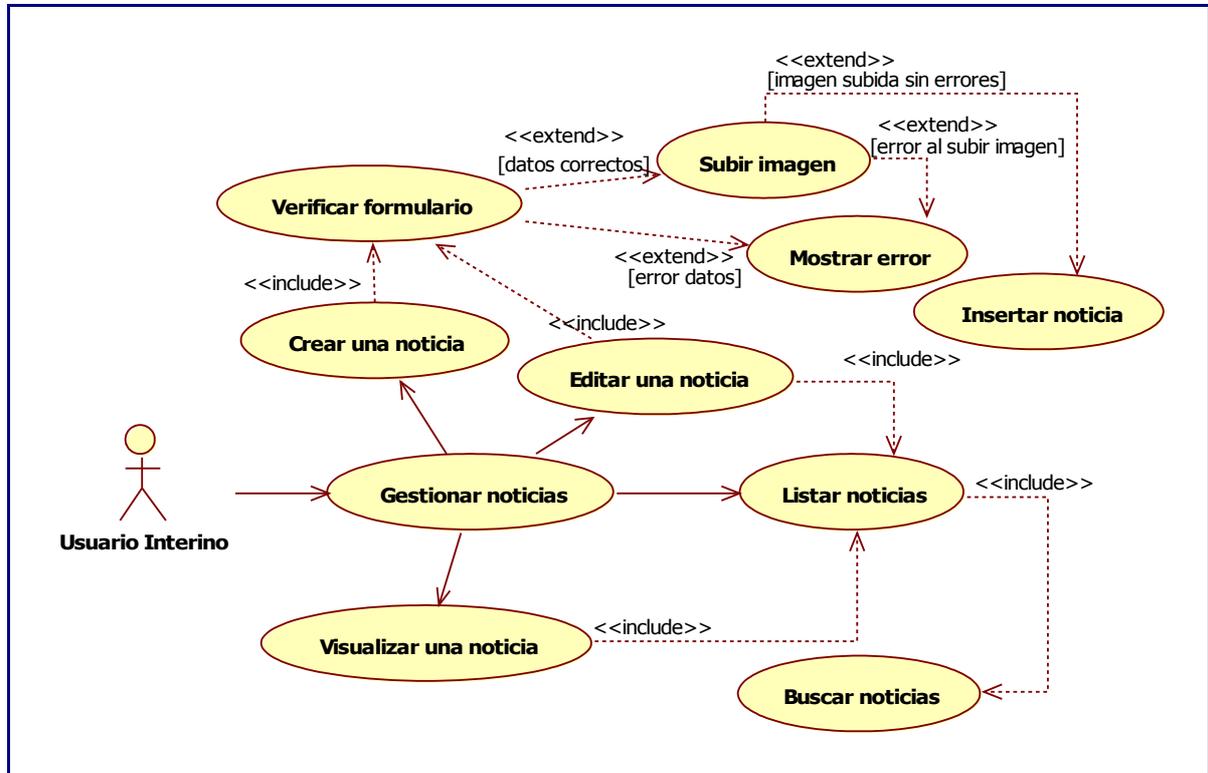


Fig. 3.2.7 – Visión en detalle del Caso de Uso "Gestionar noticias"

En la fig. 3.2.7, se muestra con mayor nivel de detalle, el Caso de Uso "Gestionar noticias" correspondiente al Usuario Interino. Obviaremos representar los casos de uso "Gestionar eventos", "Gestionar trámites" y "Gestionar oposiciones" por disponer de prácticamente idénticos Diagramas (siguen el mismo patrón).

Cabe destacar para los Casos de Uso "Crear una noticia" y "Editar una noticia", la importancia de verificar que el fichero que contiene la imagen de la noticia haya sido subido al servidor sin errores, previo a la inserción de la tupla correspondiente en la Base de Datos. Si ocurriese alguna incidencia durante su envío, se informaría al usuario y éste debería volver a intentarlo.

A continuación se muestran en la fig. 3.2.8, los diferentes Casos de Uso para el Usuario Administrador, que lo diferencian del Usuario Interino, esto es, la posibilidad de eliminar información de la Base de Datos, así como la gestión de usuarios.

En los Casos de Uso "Eliminar usuario" o "Crear usuario" (fig. 3.2.9), se hace necesario eliminar o insertar respectivamente, una tupla en la tabla 'preferencias_usuarios', sólo si el tipo del usuario en cuestión es Registrado o Residencial. Esto es así debido a que esta tabla es consultada cada vez que debe mostrarse la página principal (Inicio) con el fin de reubicar sus elementos en función de las preferencias de hábitos del usuario, funcionalidad que no está disponible para los otros tipos de usuarios (Administrador, Interino o Anónimo).

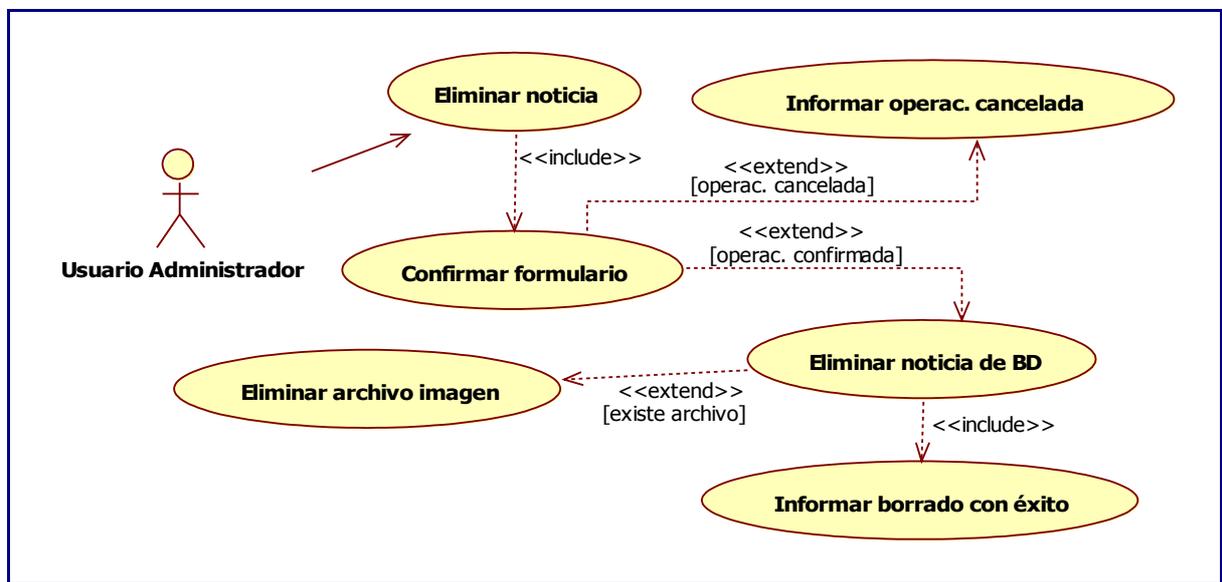


Fig. 3.2.10 – Visión en detalle del Caso de Uso "Eliminar noticia"

En la fig. 3.2.10 se muestra un Caso de Uso para la eliminación de un elemento, concretamente una noticia. No se expondrán los Casos de Uso para la eliminación del resto de elementos por seguir el mismo patrón.

Como puede apreciarse, siempre se solicitará confirmación de borrado al usuario y se comprobará si existe algún archivo en el servidor asociado al elemento que se solicita eliminar, para de esta forma proceder también a su eliminación.



3.3. Diagramas de clases -

Los Diagramas de Clases muestran las diversas entidades que componen un sistema y cómo se relacionan unas con otras. Podríamos decir que son diagrama estáticos en tanto en cuanto describen la estructura del sistema.

Una clase define los atributos y los métodos de una serie de objetos, de forma que todos los objetos de una clase determinada tienen el mismo comportamiento y el mismo conjunto de atributos. Para el caso que nos ocupa, un objeto podría ser un usuario, una noticia o un mensaje del foro.

Este tipo de diagramas también se sustentan de forma relevante sobre el concepto de herencia, esto es, la capacidad de crear sub-clases o clases-hijas más especializadas pues pueden incorporar nuevos atributos y métodos, además de conservar los que heredan de la clase-padre. Por poner un ejemplo que facilite la comprensión: en el proyecto que nos ocupa, un Usuario Residencial dispone de la mismas propiedades que un Usuario Registrado, además de ampliarlas con otras nuevas. Un Usuario Residencial debe contener indefectiblemente algún valor para el atributo DNI, mientras que para un Usuario Registrado esto no es requisito. Asimismo, un Usuario Residencial puede iniciar tramitaciones (acceso al servicio Oficina Virtual) mientras que un Usuario Registrado no dispone de tales privilegios. Por tanto, podríamos decir que un objeto de la clase Usuario Registrado podría especializarse en Usuario Interino, heredando sus atributos y métodos además de tener el potencial de adicionar otros nuevos.

A continuación se muestra el Diagrama de Clases (Fig. 3.2.11) que representa el sistema objeto de estudio, indicando atributos y algunos de los métodos de especial relevancia. Asimismo se incluye la cardinalidad y nombre de algunas de las relaciones, con el objetivo de facilitar la comprensión de aquellos escenarios que podrían suscitar cierto grado de ambigüedad.

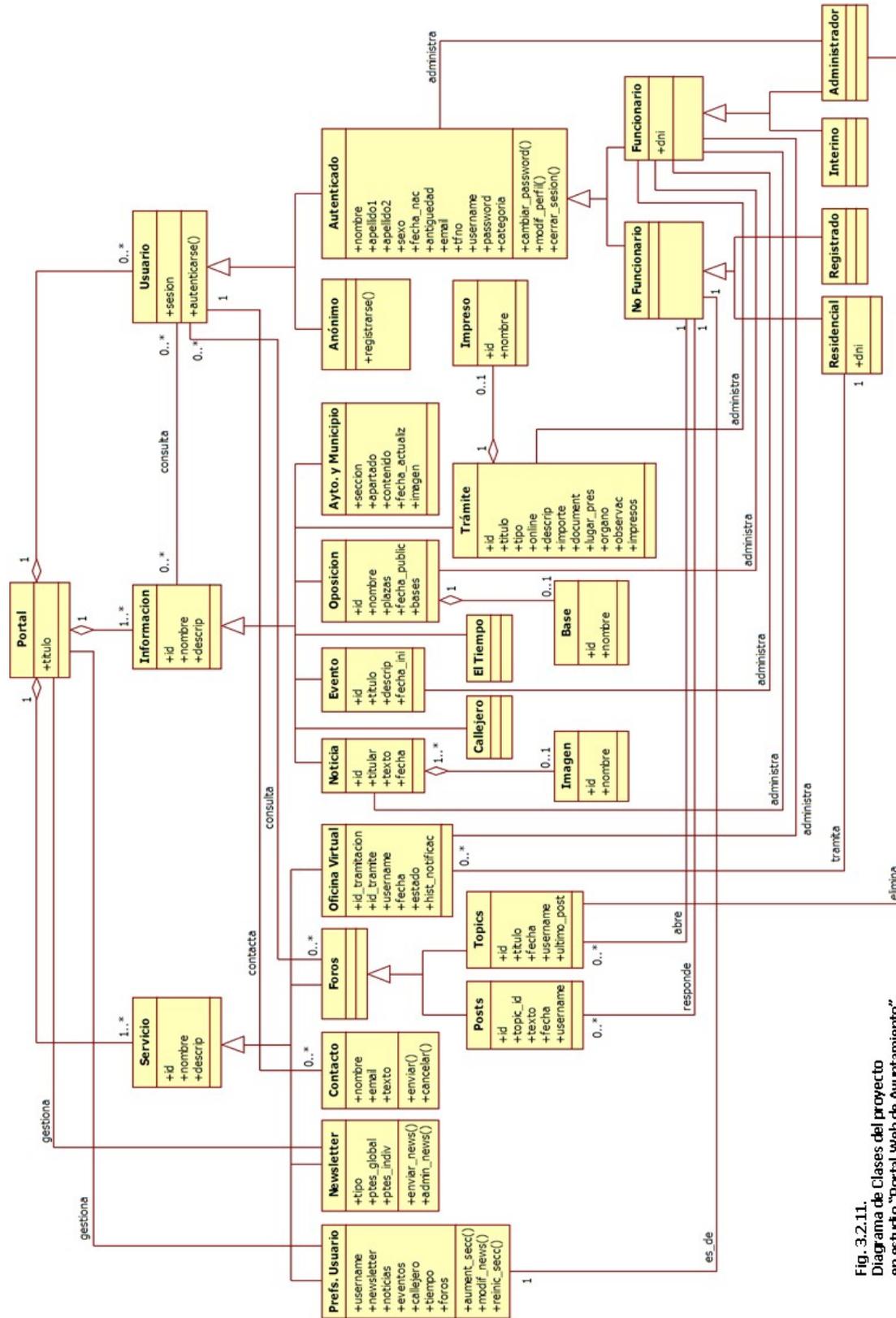


Fig. 3.2.11. Diagrama de Clases del proyecto en estudio "Portal Web de Ayuntamiento"



A continuación se exponen las diferentes estrategias que argumentan el por qué del modelado de algunos de los posibles escenarios del Diagrama de Clases:

- El portal está compuesto por 3 clases principales, como sigue:
 - Usuarios: se ha decidido modelar su jerarquía agrupándolos según sus funcionalidades o privilegios. En este sentido, en primera instancia se establece un primer nivel de jerarquía atendiendo al hecho de que el usuario se haya autenticado en el sistema o no, y en segunda instancia, de entre el grupo de usuarios autenticados se distingue de entre aquellos usuarios con privilegios de administración de contenidos (Funcionarios del Ayuntamiento) y aquellos que no los tienen.
 - Servicios: aquellos apartados del portal que posibilitan una destacada interactividad al usuario (obviamente nos referimos al Usuario No Funcionario). Por tanto, son apartados del portal que típicamente requieren la participación del usuario en modo alguno.
 - Informaciones: aquellos apartados con un nivel de interactividad bajo con el usuario, básicamente se dedican a informar al usuario. La inmensa mayoría de estos apartados son mantenidos por un Usuario de tipo Funcionario.
- En nuestro sistema y como ya se definiera en un capítulo anterior, un Usuario Registrado pasaría automáticamente a ser un Usuario Residencial, si éste introdujera un DNI válido en el formulario de registro o de modificación de su perfil, teniendo de este modo inmediatamente activo el servicio de Oficina Virtual. Este es el motivo por el que el atributo DNI se ha decidido introducir en la clase Usuario Residencial y no en ninguna de las clases-padre, con el objetivo de especificar que un Usuario Registrado no dispondrá nunca de DNI.
- Se observa cómo cualquier usuario del tipo que sea, dispone de la opción de consultar los foros, si bien únicamente los usuarios de la subclase "No Funcionario" podrán crear nuevos mensajes y abrir temas.



- Por otro lado, si bien un Usuario Administrador dispondrá de las mismas relaciones que un Usuario Interino (a través de la clase-padre Funcionario), sólo éste dispondrá de la relación 'elimina' con la entidad "Posts", así como la relación 'administra' con la entidad que hace referencia a los usuarios Autenticados.
- También conviene destacar la relación 'es_de' que viene a especificar que todo usuario No Funcionario debe tener fortuitamente una y sólo una entrada en la clase Prefs. Usuario y viceversa. Esta relación garantizará el buen funcionamiento de la funcionalidad asociada a la ubicación automática de los elementos de la pág. principal, cada vez que ésta es solicitada por el usuario. De importante relevancia es también destacar que la clase anterior es mantenida automáticamente por el sistema, sin requerir la intervención expresa de usuario alguno, hecho que está modelado con la relación 'gestiona'.
- De forma análoga a como se explicaba en el punto anterior, el portal será el encargado de gestionar la entidad "Newsletter", sin requerir la intervención expresa de usuario alguno. Recordamos que los boletines son enviados automáticamente cada vez que son insertados en la Base de Datos un total de 10 nuevos elementos (entre noticias y eventos).

3.4, Diagramas de secuencia -

Los Diagramas de Secuencia están ampliamente extendidos. Se usan frecuentemente sobre otros debido a que modelan a la perfección cómo interaccionan entre sí los diferentes objetos del sistema.

Este tipo de diagramas muestra la secuencia de mensajes entre objetos que tiene lugar durante alguno de los posibles escenarios, entendiendo éstos como algunas de las posibles ejecuciones concretas de nuestro sistema.

Cabe destacar que el diagrama muestra los mensajes en orden cronológico según van surgiendo, desde arriba hacia abajo, a lo largo de unas líneas discontinuas verticales, las cuales están asociadas a algún objeto posible.

Mostraremos algunos de los escenarios posibles, aquéllos de mayor relevancia:

3.4.1. Escenario 1: Autenticarse

Cualquier usuario que visita el portal, de forma esporádica o no, dispone de la opción de identificarse en el mismo, autenticándose mediante el envío de un sencillo formulario.

1. El usuario introduce un Username y un Password y envía el formulario.
2. El portal contrasta la información con la almacenada en la BD.
3. Si la información no existe en la BD en una misma tupla, informamos del error al usuario.
4. Si la información sí existe en una misma tupla, validamos el usuario y mostramos la página de inicio conforme a sus preferencias.

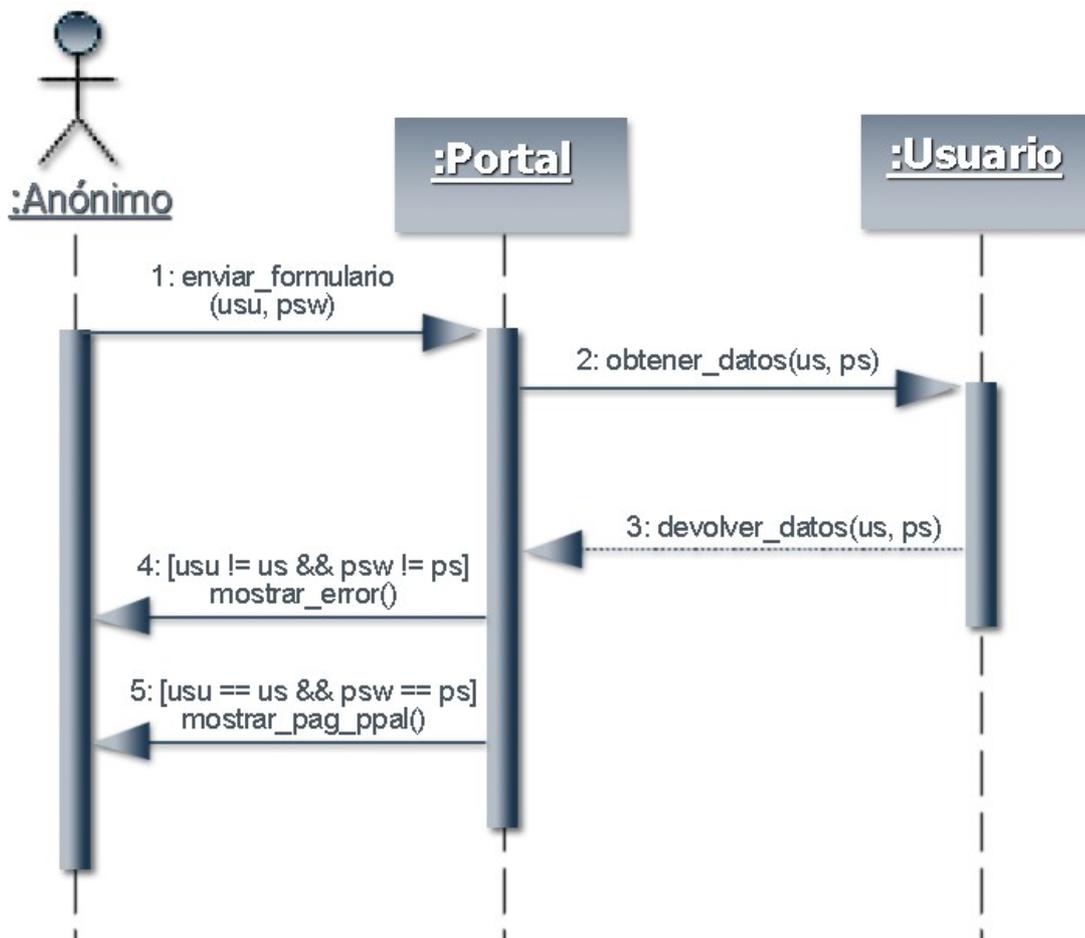


Fig. 3.4.1.1. – Diagrama de Secuencia: Autenticarse

3.4.2. Escenario 2: Buscar trámites. (Fig. 3.4.2.1)

Cualquier usuario puede obtener un listado de trámites disponibles, que cumplan una serie de requisitos de búsqueda (formulario con filtros).

1. Se muestra al usuario un formulario con varios filtros.
2. El usuario lo cumplimenta y envía.
3. El sistema obtiene la información que coincide con los filtros y la prepara previo a mostrarla.
4. Finalmente, el sistema muestra en base a un formato determinado, la información anterior. Si ninguna información fuera encontrada, informa igualmente al usuario.

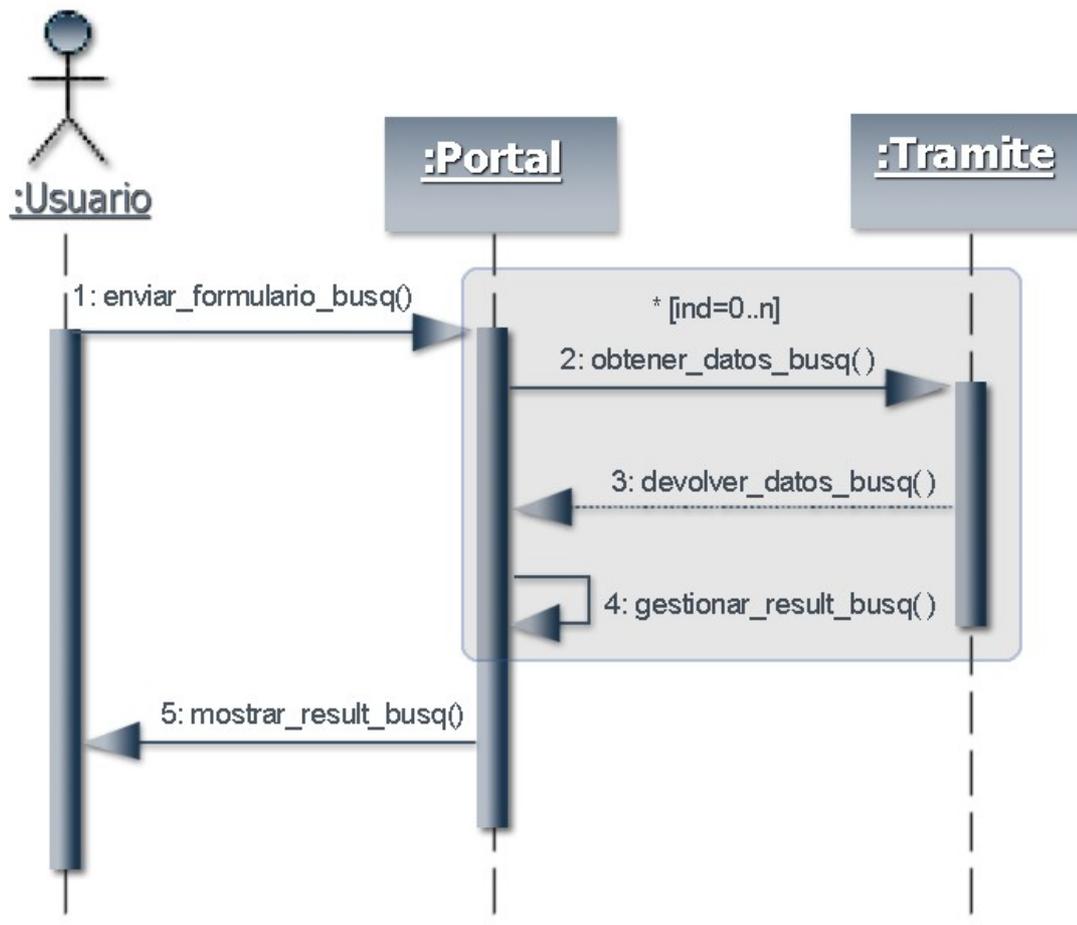


Fig. 3.4.2.1. – Diagrama de Secuencia: Buscar trámites



3.4.3. Escenario 3: Crear un nuevo usuario (Fig. 3.4.3.1)

El usuario Administrador es el único que puede dar de alta nuevos usuarios.

1. El usuario envía el formulario con los datos del nuevo socio.
2. El sistema comprueba que el Username y el Password no existiesen ya para un mismo socio.
3. Se verifican que el resto valores de campos del formulario estén en el formato correcto y hayan sido cumplimentados los obligatorios.
4. El usuario es insertado en la BD.
5. El usuario Administrador es informado del éxito. Si ocurriese cualquier error durante el escenario, también será informado del mismo.

3.4.4. Escenario 4: Eliminar un trámite. (Fig. 3.4.4.1)

El usuario Administrador puede eliminar un trámite, para lo que previamente deberá listar los trámites disponibles aplicando una búsqueda por filtrado. Si existe algún impreso asociado, si fichero deberá ser eliminado igualmente.

1. Se muestra al usuario un formulario con varios filtros. Éste lo cumplimenta y envía.
2. El sistema obtiene la información que coincide con los filtros y la prepara previo a mostrarla.
3. El sistema muestra un listado con los trámites, dando sobre cada uno la opción de eliminarlo, entre otras.
4. El usuario selecciona la opción de eliminar y el sistema solicita el borrado de la BD.
5. Asimismo, si comprueba si existe algún impreso asociado, en cuyo caso el sistema solicita también el borrado del mismo.
6. Finalmente el sistema informa al usuario del resultado de la operación.

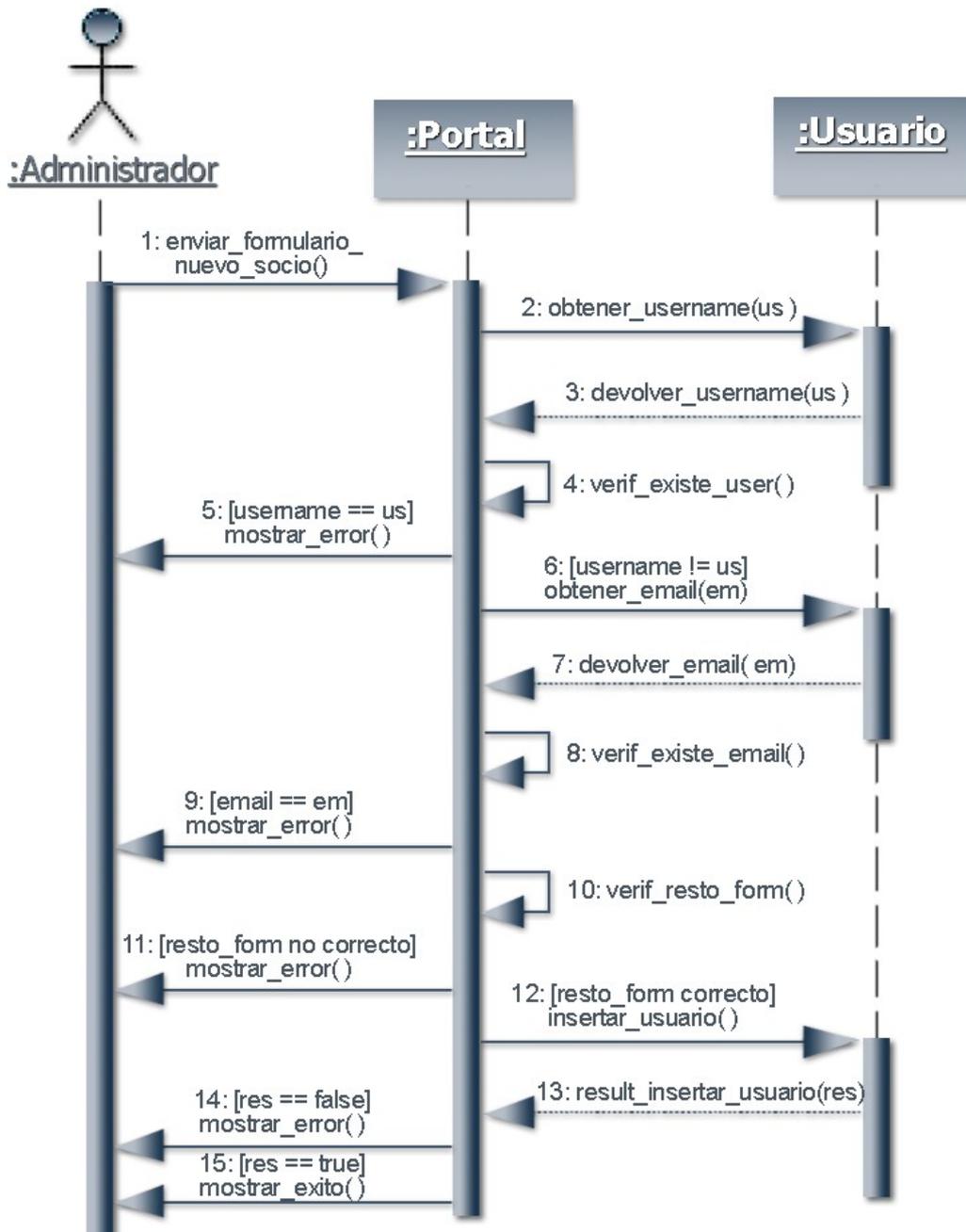


Fig. 3.4.3.1. – Diagrama de Secuencia: Crear un nuevo usuario

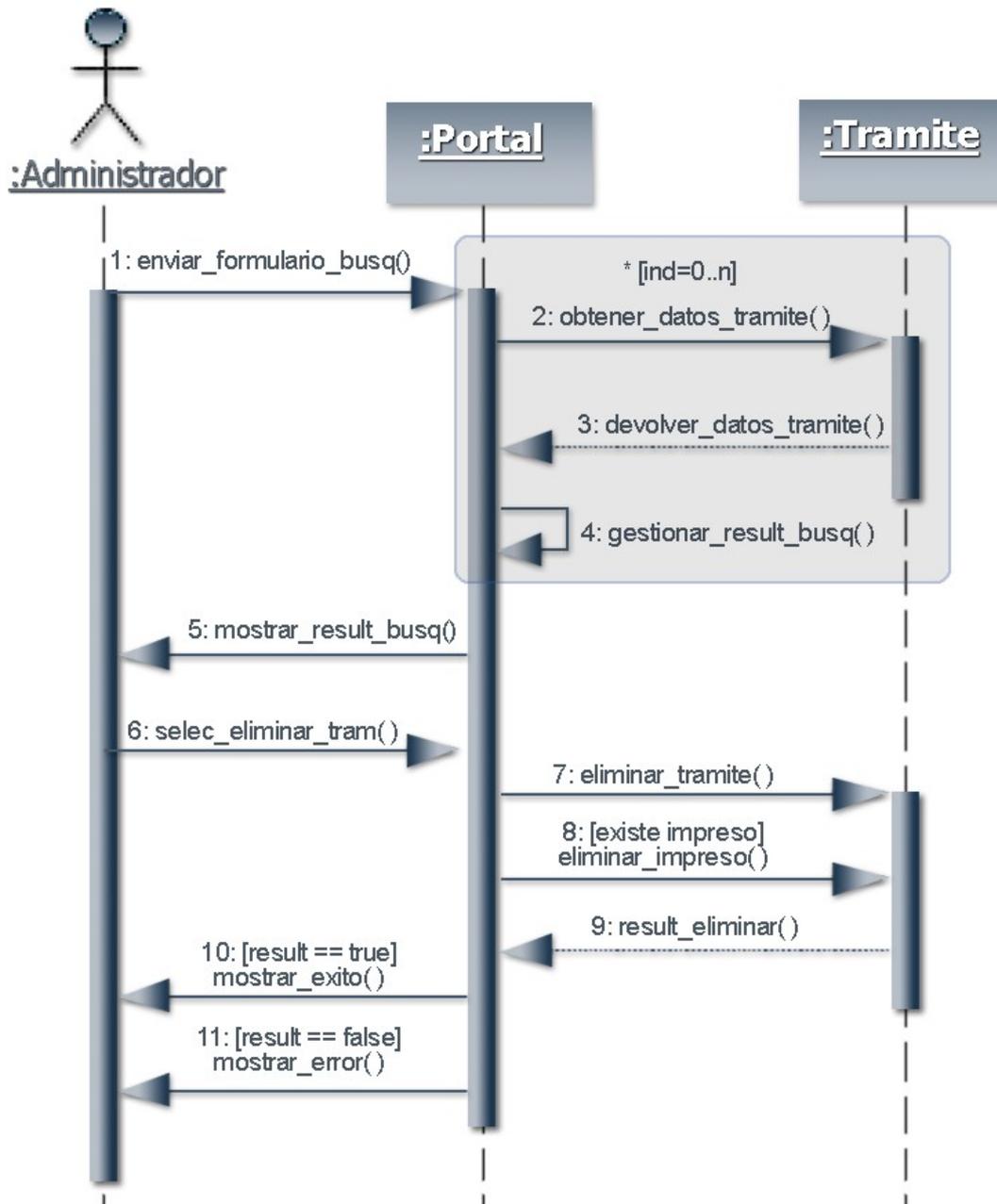


Fig. 3.4.4.1. – Diagrama de Secuencia: Eliminar un trámite

3.4.5. Escenario 5: Recuperar contraseña olvidada. (Fig. 3.4.5.1)

Un usuario puede solicitar una nueva contraseña en caso de olvido de la misma. Para ello, deberá enviar un formulario indicando su Username y su Email. En sistema comprueba que la información es correcta, previo consulta a la BD. Si la información enviada es fidedigna, el sistema generará una nueva contraseña e informará al usuario de la misma, enviándole un email, informando al mismo de que debe revisar su email.

1. El usuario envía vía formulario, su Username y su Password.
2. El portal contrasta la información con la BD y en caso de éxito, genera un nuevo password.
3. Se envía la contraseña por email al usuario y se le informa de que debe revisar el correo para conocer el nuevo password.

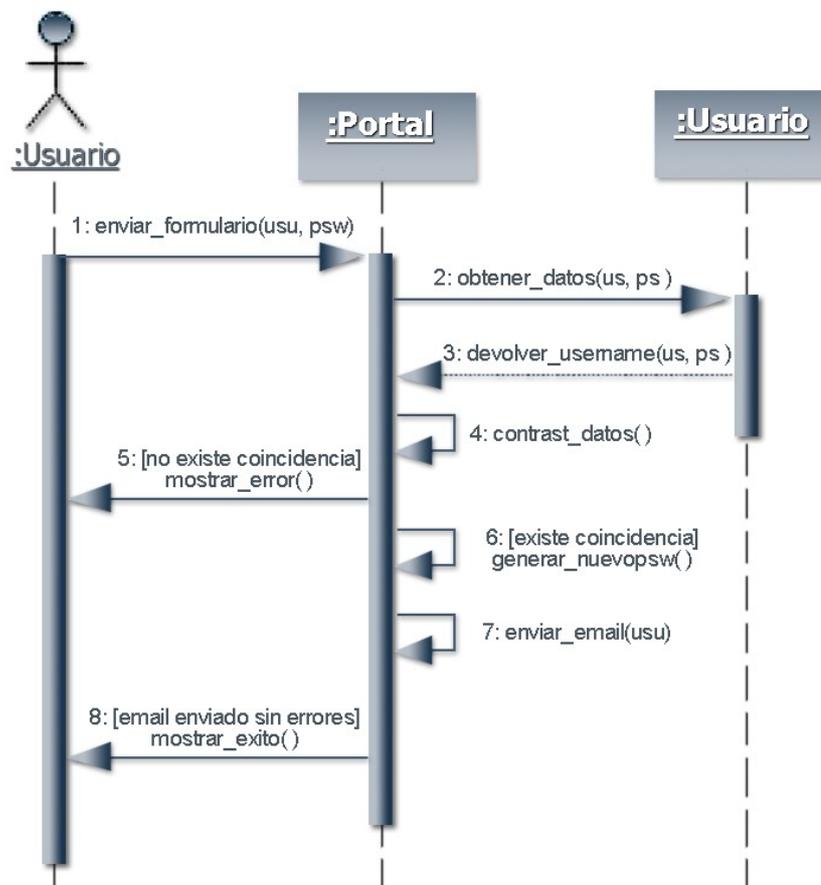


Fig. 3.4.5.1. – Diagrama de Secuencia: Recuperar contraseña en caso de olvido



4 .Diseño .-

En este capítulo, la aplicación será descrita con un nivel de detalle todavía más elevado si cabe que el conseguido en el capítulo de Análisis.

Para conseguir el objetivo anterior, nos haremos servir de una arquitectura Web basada en el modelo de tres niveles.

4.1. Introducción .-

Algunas de las virtudes de utilizar un modelo de 3 niveles o capas, es que permite continuar con el desarrollo en cualquiera de los niveles, siendo necesario revisar únicamente uno de ellos, en caso de precisar de alguna modificación. Otra virtud es que permite distribuir el trabajo por equipos, de forma que cada equipo consigue focalizar sus esfuerzos en el nivel que tenga asociado; en este caso, bastaría con conocer la API que existe entre el resto de niveles (se consigue así abstraer a los equipos, enfatizándose el “qué hace” más que el “cómo lo hace”).

A continuación describiremos en detalle cada uno de los niveles. Asimismo bastará con observar la Fig. 4.1.1. para asimilar rápidamente lo que se pretende al emplear este tipo de arquitectura, en un entorno Web como el que nos ocupa:

- **Nivel de Interfaz o Presentación:** formado por aquellos elementos que permiten al usuario interactuar con el sistema (formularios, informes, confirmación de eventos, ...). También utilizada por el sistema para informar al usuario (mensajes de error, información demandada, etc).



Este nivel representa el enlace que comunica los otros 2 niveles.

- **Nivel de Almacenamiento o Persistencia:** integra la Base de Datos y el Sistema de Gestión de la Base de Datos (MySQL y phpMyAdmin respectivamente, en el caso que nos ocupa). No sólo es el encargado de almacenar la información, sino también el responsable de la seguridad e integridad de la misma.

A continuación se detalla cómo han sido abordados cada uno de los niveles para el caso en estudio.

4.2. Nivel de interfaz .-

En cualquier interfaz de usuario que se precie, debe prevalecer el carácter intuitivo y amigable del mismo, esto es, que cualquier usuario que se aventure a utilizarlo, sin importar la experiencia del mismo, consiga identificar los elementos principales en el menor tiempo posible. También el término amigable expresa la necesidad de “gustar” al usuario y esto se consigue con diferentes estrategias: tipos de letras legibles y adecuadas a cada contexto, así como la combinación de grafismos y simbologías empleada, colores suaves y poco estridentes, uniformidad en los diseños, etc.

En el caso que nos ocupa, para representar cada una de las páginas que conforman el portal, se emplearán 2 formatos o estructura de página diferentes:

- La Figura 4.2.1 muestra la estrategia utilizada para la página principal (Inicio), con cabida para albergar una mayor cantidad de elementos y secciones. Es posible detectar una disposición del área central en base a 3 columnas donde la columna central será dedicada a exhibir los contenidos principales, mientras que las columnas laterales, de menor extensión, albergarán elementos como menús de navegación, subsecciones u otros servicios menores.
- Por otro lado, la Figura 4.2.2 representa el formato que será empleado para el resto de páginas, tanto para usuarios Funcionarios como No Funcionarios. Es



fácil comprobar que su aspecto, aunque similar al formato anterior, presenta una concentración menor de elementos dispares. De esta forma, la disposición del área central es ahora en base a 2 columnas y se ha eliminado el área inferior dedicado a las Secciones al Detalle. Obviamente, las opciones de usuario disponibles en las distintas barras de navegación o menús, cambiarán automáticamente en función del tipo de usuario autenticado.



Fig. 4.2.1. – Formato de página empleado para la Página Principal (Inicio)



Fig. 4.2.2. – Formato de página empleado para el resto de páginas.

4.3. Nivel de negocio.-

El nivel de negocio está representado por el conjunto de componentes software que implementa de forma completa el comportamiento de las clases del dominio.



Asimismo este nivel desarrolla un papel vital de enlace o comunicación entre los otros 2 niveles: por un lado, se encarga de obtener los datos de la Base de Datos y formatearlos según las necesidades especificadas en el nivel de interfaz, y por otro lado, también debe obtener la información procedente del nivel de interfaz y readecuarla según las necesidades y restricciones impuestas por el nivel de almacenamiento.

Es precisamente el papel de enlace descrito anteriormente, el que le confiere su carácter de independencia y que por tanto permite efectuar modificaciones en los otros 2 niveles, sin que ello suponga una revisión en profundidad en el resto de niveles.

Es posible utilizar la información de los casos de uso representados durante la fase de Análisis para expresar en este caso la división que existe entre los procesos externos (los que dependen del interfaz) y los internos (aquéllos cuyo cometido no es otro que el de procesar la información de entrada para reaccionar en consecuencia, devolviendo los resultados alcanzados).

En la Fig. 4.3.1. puede observarse una muestra de diagrama de procesos o tareas donde son representados los procesos externos como puntos de entrada de la información a la aplicación del portal objeto de estudio, así como también quedan representados los conjuntos de procesos internos que se desencadenarán respondiendo al "estímulo" de los primeros. Estos procesos internos pueden descomponerse en otros a su vez más específicos, en coherencia con las definiciones dadas en los casos de uso.

Según puede extraerse de la figura anterior, el diagrama representa los procesos externos como aquéllos que son accedidos por elementos "actor", así como los conjuntos de tareas relacionadas con otras "subtareas", relación manifiesta mediante el uso de flechas indicando relaciones de dependencia.

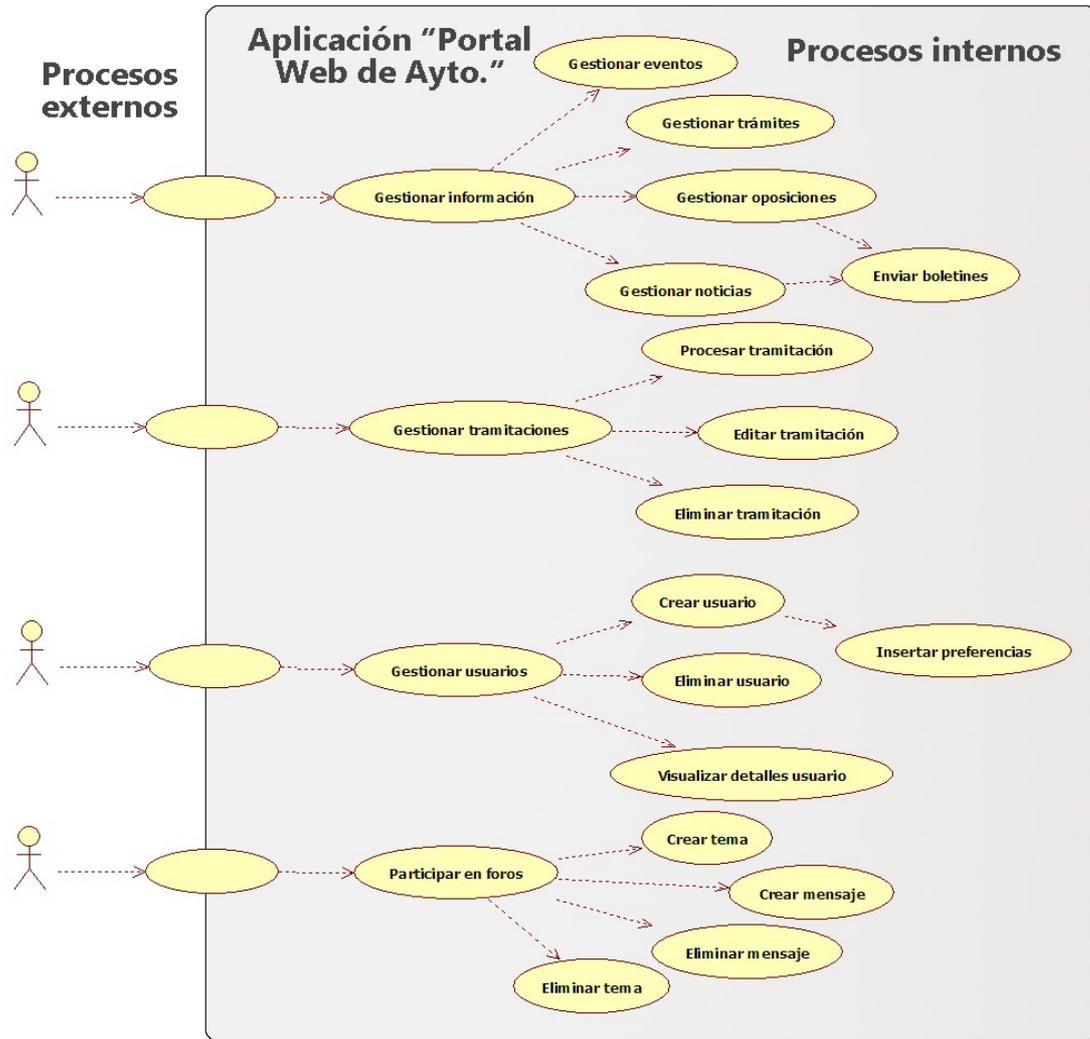


Fig. 4.3.1. – Diagrama de tareas del Portal Web de Ayuntamiento



4.4. Nivel de almacenamiento

En este nivel, se enfatizará en aquellos recursos orientados a garantizar la persistencia de los objetos del nivel de negocio, habitualmente recurriendo a un sistema gestor de base de datos relacional.

4.4.1. Diagrama Entidad-Relación

A continuación se muestra el Diagrama Entidad-Relación (Fig. 4.4.1.1) como diseño conceptual de nuestra Base de Datos. Estos tipos de diagramas son idóneos para mostrar la estructura de una Base de Datos, basándose en 3 elementos gráficos principales:

- las entidades: representadas mediante un rectángulo, hacen referencia a un objeto o cosa del mundo real con existencia independiente. En nuestro proyecto podría ser un usuario o una noticia.
- las relaciones: representadas mediante rombos, hacen referencia a la asociación de 2 o más entidades. En nuestro proyecto podrían ser las tramitaciones o gestiones, las cuales hacen corresponder un trámite a un usuario.
- los atributos: representados mediante elipses, hacen referencia a las características relevantes de un entidad o relación. En nuestro proyecto serían el DNI o Username de un usuario o el titular de una noticia.

Por motivos de legibilidad, no incluiremos en el diagrama todos los atributos sino sólo aquellos de mayor relevancia o que ayudan a explicar mejor el comportamiento de algunas entidades.

Si observamos con detenimiento el Diagrama E-R del proyecto en estudio, especial mención requiere los siguientes posibles escenarios:



- La entidad “TRAMITES_USUARIOS” desempeña un papel fundamental en el apartado de 'Oficina Virtual' de nuestra aplicación. Esta entidad viene a representar el conjunto de tramitaciones que fueron iniciadas en algún instante por algún Usuario Residencial. A continuación, detallamos los posibles escenarios que pueden darse, relacionados con esta entidad:
 - la relación “tramita” representa el posible escenario de que un usuario puede iniciar una nueva tramitación o bien puede editar una ya existente, si bien una tramitación dada sólo puede pertenecer a un usuario determinado.
 - la relación “tramita_con” representa el hecho de que toda tramitación debe estar asociada a uno y tan sólo un trámite. Asimismo pone de manifiesto la posibilidad de que un trámite determinado puede estar presente en varias tramitaciones. Por otro lado, un Usuario Funcionario puede gestionar (aprobar, denegar, notificar, etc) cualquiera de las tramitaciones, así como éstas pueden ser gestionadas por cualesquiera de los funcionarios.

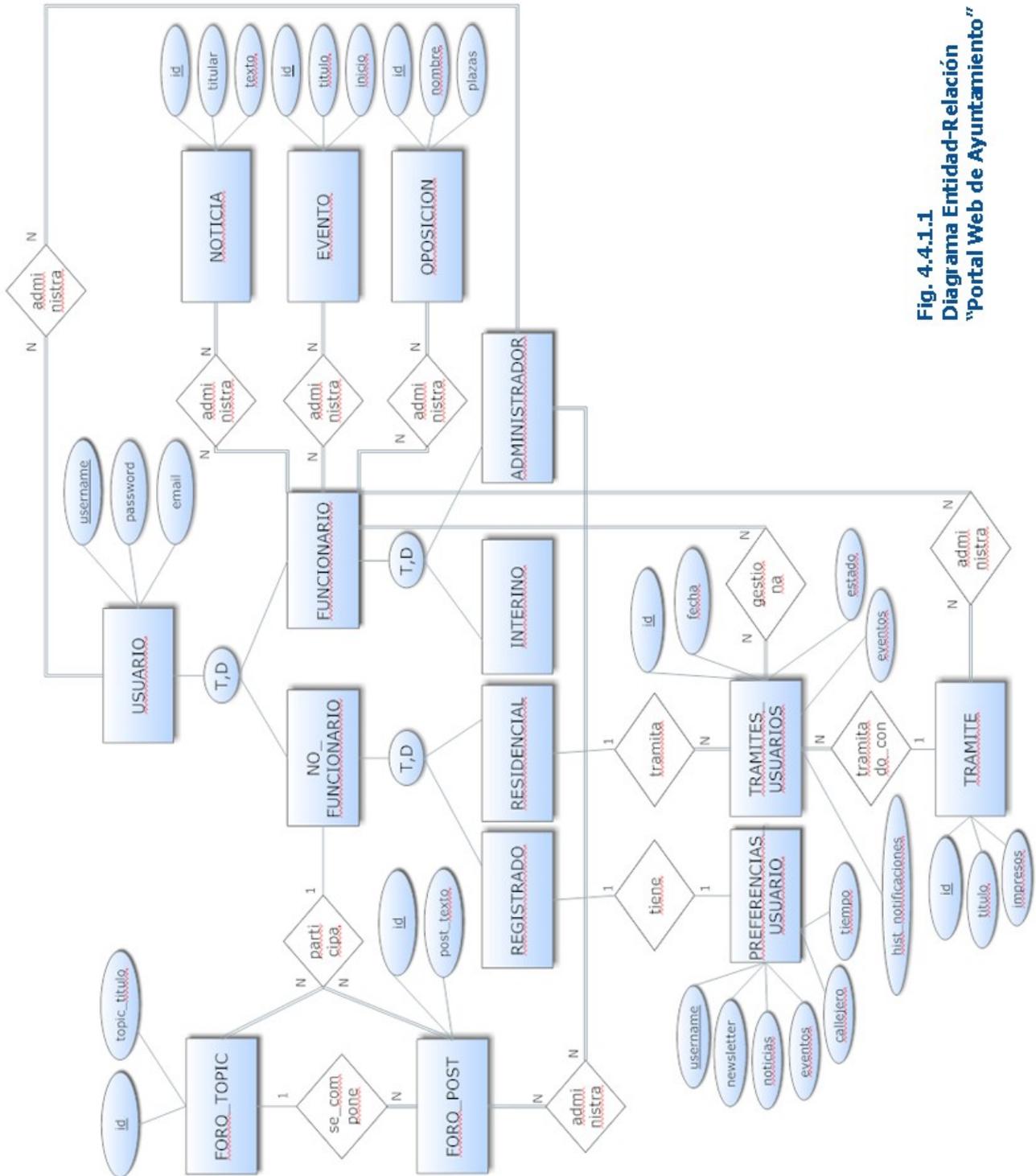


Fig. 4.4.1.1 Diagrama Entidad-Relación "Portal Web de Ayuntamiento"



- Se ha optado por crear varios niveles de especialización para la entidad "USUARIO", con el fin de delimitar de un forma más legible en el Diagrama, los privilegios de actuación de cada uno de los diferentes perfiles de usuarios posibles:
 - si bien, cualquier Usuario No Funcionario podrá participar en los foros, sólo aquéllos que sean Residenciales, podrán hacer uso del servicio de Oficina Virtual (relacionarse con la entidad "TRAMITES_USUARIOS").
 - si bien, cualquier Usuario Funcionario podrá administrar los contenidos de diversas secciones del portal, sólo aquel especializado como Usuario Administrador podrá administrar (crear y eliminar) Usuarios o bien eliminar mensajes del Foro.

4.4.2. Diseño lógico -

A continuación se complementa el Diagrama analizado en el apartado anterior con el diseño lógico, esto es, la estructura de la base de datos representada en un lenguaje comprensible por cualquier sistema de gestión de bases de datos relacionales.

USUARIO (dni: varchar(9), nombre: varchar(20), apellido1: varchar(25), apellido2: varchar(25), sexo: varchar(1), fecha_nac: date, antiguedad: date, email: varchar(50), tfno: varchar(15), username: varchar(20), password: varchar(50), categoria: varchar(3))

CP: {dni}

VNN: {nombre, antiguedad, email, username, password, categoria}

TRAMITES_USUARIOS (id: entero_largo, id_tram: entero_largo, username: varchar(20), fecha: date, estado: varchar(20), hist_notificaciones: varchar(255))

CP: {id}

VNN: {id_tram, username, fecha, estado}

CAj: {username} → USUARIOS

CAj: {id_tram} → TRAMITES



TRAMITES (id: entero_largo, titulo: varchar(100), tipo: varchar(50), online: varchar(2), desc: varchar(255), importe: entero, documentacion: varchar(255), lugar_present: varchar(50), organo_gestor: varchar(50), observaciones: varchar(255), impresos: varchar(50))

CP: {id}

VNN: {tipo, online, importe}

CAj: {id} → TRAMITES_USUARIOS

CALENDARIO_EVENTOS (id: entero_largo, titulo_evento: varchar(100), desc_evento: text, inicio_evento: date)

CP: {id}

VNN: {titulo_evento, inicio_evento}

CONTROL_NEWSLETTER (tipo_newsletter: varchar(18), pendientes_global: entero(2), pendientes_individual: entero(2))

CP: {tipo_newsletter}

VNN: {pendientes_global, pendientes_individual}

FORO_POSTS (post_id: entero_largo, topic_id: entero_largo, post_texto: text, post_fecha: date, post_prop: varchar(100))

CP: {post_id}

VNN: {topic_id, post_texto, post_prop}

CAj: {topic_id} → FORO_TOPICS

FORO_TOPICS (topic_id: entero_largo, topic_titulo: varchar(100), topic_fecha: date, topic_prop: varchar(20), topic_ultimo: date)

CP: {topic_id}

VNN: {topic_titulo, topic_prop, topic_fecha}

NOTICIAS (id: entero_largo, titular:varchar(200), texto: text, imagen: varchar(50), fecha: date)

CP: {id}

VNN: {titular, fecha}



OPOSICIONES (id: entero_largo, nombre: varchar(50), plazas: entero(3), fecha_publicacion: date, bases: varchar(100))

CP: {id}

VNN: {nombre, plazas, fecha_publicacion}

PREFERENCIAS_USUARIOS (username: varchar(40), newsletter: varchar(8), noticias: entero(4), eventos: entero(4), callejero: entero(4), tiempo: entero(4), foros: entero(4),)

CP: {username}

VNN: {newsletter, noticias, eventos, callejero, tiempo, foros}

CAj: {username} → USUARIOS

SECCION_ESTATICA (seccion: varchar(4), contenido: text, imagen: varchar(30), fecha_actualizacion: date, apartado: varchar(15))

CP: {seccion}

VNN: {contenido, apartado}





5. Implementación .-

En el capítulo anterior, la aplicación objeto de este estudio fue estructurada en diversos niveles y fueron descritos todos ellos de la forma más abstracta posible, evitando entrar en detalles sobre tecnologías o recursos de implementación.

En el capítulo que nos ocupa, serán revisadas las tecnologías Web recurridas para implementar la aplicación, progresando de forma ordenada a través de la estructura de niveles que fue especificada.

5.1. Introducción .-

En síntesis, aquellos recursos o tecnologías webs a los que se ha recurrido son:

- Entorno de desarrollo web **WampServer** versión 2.0. Se trata de un paquete que incluye, instala y configura casi de manera automática: Apache, PHP, MySQL y PHPMyAdmin.
- **Lenguaje PHP** para todo el apartado de programación y lógica de control, debido al carácter dinámico e interactivo perseguido, y habitualmente será entremezclado o combinado con el lenguaje de marcado HTML. La versión utilizada en el momento de editar este documento es la 5.3.0.
- **Plantillas CSS** con el objetivo de ofrecer una interfaz atractiva y un diseño del website moderno, acorde a las últimas tendencias. Por otro lado, esta decisión mejorará notablemente el tiempo requerido para futuras modificaciones en el diseño. Para la maquetación del website se ha recurrido a uno de los Frameworks CSS más populares: **960 Grid System**.



- Adobe **Photoshop** CS2 para para el diseño de algunas partes de las páginas como las diversas cajas que las componen, efectos de degradados y sombras en gran multitud de elementos, composición de imágenes empleadas en la cabecera o pie de página, efectos dinámicos para los barras de navegación, diseño de tipos de letra complejas así como iconos animados repartidos a lo largo de la página.
- **Javascript** para gobernar el comportamiento de algunas ventanas del portal, reaccionando a la acción del usuario (típicamente un evento de ratón). También se ha utilizado para integrar el código necesario para mostrar el tiempo, servicio ofrecido por 'www.tutiempo.net'.
- **MySQL** como Base de Datos, siendo ésta de las más extendidas y eficientes en la actualidad. Aunque se ha diseñado una interfaz al uso en el mismo portal para usuarios administradores de contenidos, en ocasiones puntuales podría precisarse gestionar la Base de Datos con una herramienta externa. En este caso, se ha optado por la utilidad PHPMyAdmin, también muy extendida y con un alto grado de amigabilidad para con el usuario. La versión empleada en el momento de edición de este documento es la 5.1.36.
- Servidor Web **Apache**, concretamente la versión 2.2.11 en el momento de edición del presente documento. Se trata sin duda de uno de los servidores más populares debido a su robustez y calidad de servicios. Es compatible sobre multitud de plataformas y se integra a la perfección con el lenguaje PHP.
- Otras utilidades menores como editores de código (Komodo Edit 5) o diferentes navegadores Web (Mozilla Firefox, Opera, Google Chrome, etc) para validar las diferentes páginas.



5.2. Tecnologías .-

A continuación serán descritas con mayor nivel de profundidad, algunas de las tecnologías citadas anteriormente.

5.2.1. HTML .-

HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje que predomina en el desarrollo de páginas web. Gracias a él, es posible no sólo describir el contenido de texto de una página web sino también su forma o estructura, permitiendo incrustar imágenes y otros objetos en la misma.

La historia del estándar se remonta a principio de 1990, cuando Tim Berners-Lee define el HTML como un subconjunto de un conocido lenguaje de etiquetas denominado SGML y además crea algo más valioso aún, el World Wide Web.

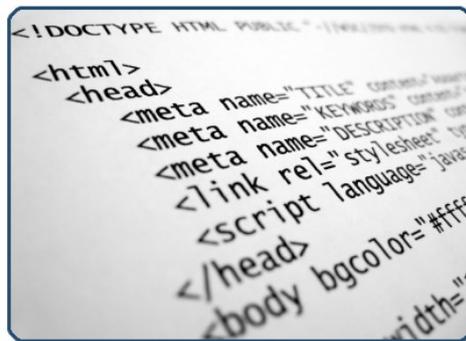


Fig. 5.2.1.1. – Etiquetas empleadas por el lenguaje PHP.

Este lenguaje se escribe a base de etiquetas delimitadas por corchetes angulares (< , >). Cuando se utiliza una etiqueta, habitualmente debe utilizarse otra de cierre, la cual antepondrá el símbolo de barra (/). Sirva un ejemplo:

<p>Esto está escrito con una fuente normal mientras que ****ésto está escrito con una fuente más gruesa o negrita**</p>**



El párrafo anterior será mostrado como :

Esto está escrito con una fuente normal mientras que **ésto está escrito con una fuente más gruesa o negrita**

El lenguaje HTML puede ser escrito y editado con cualquier editor de textos sencillo, aunque habitualmente se emplean editores potentes que permiten visualizar el resultado de lo que se está haciendo así como identificar errores de programación.

Una página web HTML debe contener una estructura mínima de etiquetas como sigue:

```
<html>
  <head>
    <title>Aquí iría el título de la página</title>
  </head>
  <body>
    Aquí debe ir el contenido de la página
  </body>
</html>
```

El diseño en HTML debe respetar unos criterios de accesibilidad web, siguiendo unas pautas o normativas. Se encuentra disponible y desarrollado por el W3C a través de las Pautas de Accesibilidad al Contenido Web 1.0 WCAG (actualizadas recientemente con la especificación 2.0).

5.2.2. CSS .

El lenguaje HTML está algo limitado si queremos emplearlo para definir la forma de un documento, ya que no fue concebido precisamente para este fin. Aún así, durante mucho tiempo los diseñadores han debido recurrir a "trucos" para salvar esta dificultad, causando a menudo problemas en las páginas a la hora de su visualización en distintas plataformas.



CSS u hojas de estilo en cascada (Cascading Style Sheets), es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. Lo que se pretende con esta tecnología es separar la estructura de una página web de su presentación, esto es, el contenido en sí a mostrar de la forma con que se desea que aparezca.

CSS puede incorporarse en el mismo documento HTML o bien adjuntarse como un documento por separado (obviamente el documento HTML deberá entonces referenciarle).

Enumeraremos algunas de las ventajas de CSS:

- Podemos definir la forma de presentación de un website entero, de sólo una página o una porción de la misma. Se agilizan por tanto enormemente, las tareas de modificación: por ejemplo, bastaría con editar una hoja de estilo CSS para que tuviera una repercusión inmediata en todas las páginas del website.
- El documento HTML es más fácil de entender, pues conseguimos con CSS separar la forma del contenido (el código que hace referencia a la forma, ya no estará entremezclado con el código restante).
- Una misma página Web puede ser mostrada de forma correcta en diversos dispositivos de salida: un móvil, una impresora, una PDA. Para ello bastará con sustituir la hoja de estilo empleada.
- Aumento considerable de la accesibilidad, ya que un navegador web permite a un usuario especificar su propia hoja de estilo local. Idóneo por tanto para usuarios con deficiencias visuales (hojas que aumentan el tamaño de letra o la combinación de colores).

Otro de los recursos ampliamente buscados por diseñadores web y que están basados en CSS son los denominados Frameworks CSS. Éstos no son más que declaraciones de estilos escritas en CSS que preparan y agilizan la maquetación de una página web, principalmente porque resetean y normalizan los estilos, con el fin de homogeneizar los



resultados (conseguimos que nuestras páginas sean mostradas del mismo modo en cualquier navegador web).

Para nuestro caso en estudio, se ha empleado el Framework 960 Grid System, el cual ha despertado el interés de muchos diseñadores web, partiéndose de un diseño inicial en el editor Photoshop.

Framework 960 Grid System

960 Grid System define una serie de clases que formatean nuestra página web como si de una rejilla se tratara, es decir, habilita para uso del diseñador, una plantilla o rejilla con un ancho de 960 píxeles y dividida en un número fijo de columnas (12 columnas en nuestro caso) del mismo tamaño.

Una de las ventajas principales de esta división en columnas de nuestro espacio de trabajo es que permite anclar o fijar, a conveniencia del diseñador, los distintos elementos que comportarán la página web, encajando éstos de forma precisa sobre la retícula, además de conseguir un alineamiento uniforme entre los elementos. Se trata por tanto de un recurso orientado principalmente al posicionamiento.

960 Grid System permite agrupar fácilmente estas columnas para conseguir columnas mayores y así estructurar la página web en base a 2, 3 o más columnas o áreas principales, de igual o diferente tamaño cada una de ellas. De este modo, es posible formatear fácilmente una página web con un diseño por ejemplo a 3 columnas. A continuación se muestra una breve explicación de cómo sería abordado este ejemplo:

- Primera columna: compuesto por 2 columnas según el Framework. Este área lo reservaremos para ubicar por ejemplo un menú vertical.
- Segunda columna: compuesto por 7 columnas según el Framework. Este área lo reservaremos al cuerpo principal de la página.
- Tercera columna: compuesto por 3 columnas según el Framework. Esta última columna principal o área lo reservaremos para ubicar algunas cajas pequeñas

mostrando información de distintas secciones.

En total, según el ejemplo anterior habríamos ocupado las 12 columnas de la rejilla a nuestra conveniencia.



Fig. 5.2.2.1. - Ejemplo de página web diseñada a partir de la rejilla que habilita el Framework CSS 960 Grid System (versión 12 columnas).

En el caso de nuestro portal objeto de estudio, la página principal (Inicio) ha sido diseñada en base a una estructura de 3 columnas o áreas principales, mientras que el resto de páginas respetan una estructura de 2 columnas principales.

Modelo de Caja

CSS pone a disposición del diseñador otro recurso muy extendido que es el Modelo de Caja. Este modelo permite estructurar los diferentes contenidos de la página como si de cajas rectangulares se tratara. Cada caja puede pertenecer a un servicio (por ejemplo, el buscador), una sección (por ejemplo el apartado de noticias) o a su vez una parte de una sección (por ejemplo la fotografía de la noticia).

Las cajas a su vez están compuestas por partes o capas, presentando en conjunto una estructura tridimensional. De esta forma, toda caja tiene una parte que es el borde,

otra que es el relleno, un margen, un color y una imagen de fondo, un relleno y finalmente un contenido. Cada una de estas partes podríamos representarla como una estructura de capas superpuestas, como puede verse en la Fig. 5.2.2.2.

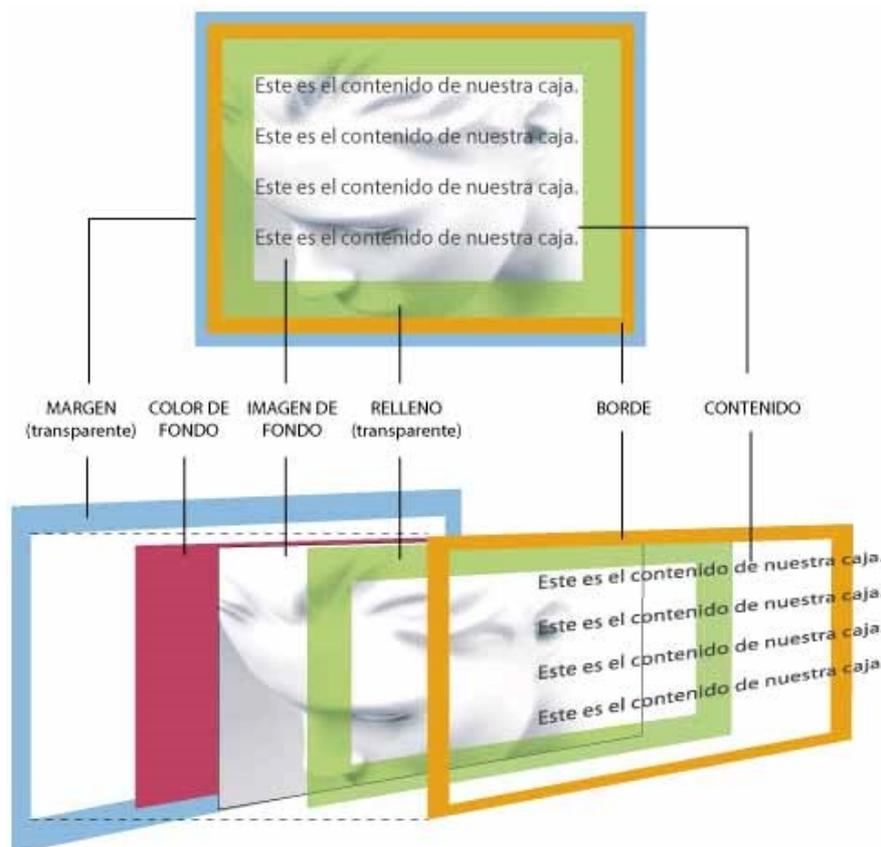


Fig. 5.2.2.2. - Modelo de Caja en CSS

Las cajas se implementan en HTML, delimitándolas con un par de etiquetas `<div>` `</div>`. Posteriormente se asigna a cada caja una serie de estilos CSS a través de un identificador o una clase. Sirva como ejemplo el siguiente extracto de código HTML que describe una de las diversas cajas empleadas en nuestro portal objeto de estudio:

5.2.3. PHP

PHP (PHP Hypertext Pre-processor) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Fue creado en 1994 por el programador danés-canadiense Rasmus Lerdorf (Personal Home Page Tools por aquel entonces) para mostrar su currículum vitae y conocer el tráfico que su web recibía.

PHP es un lenguaje poderoso y está ampliamente extendido, se utiliza en más de 20 millones de sitios web además de en un millón de servidores. Uno de los motivos es un gran parecido con otros populares lenguajes de programación estructurada, lo que facilita su dominio en el menor tiempo. Otra de las razones es que es libre (licencia GNU) y por tanto, fácilmente accesible a todos. Y por añadir alguna razón más, indicaremos que PHP permite ser ejecutado en la mayoría de los sistemas operativos e interactuar con los servidores web y motores de base de datos más populares.

El código fuente escrito en PHP es invisible al navegador y al cliente. El servidor es el encargado de ejecutar el código y enviar su resultado HTML al navegador (véase Fig. 5.2.3.1). Esto hace que la programación en PHP sea segura y confiable.



Fig. 5.2.3.1. - Esquema funcionamiento PHP



Nuestro caso de estudio está fundamentado principalmente en PHP, integrado entre el código HTML. El esquema de funcionamiento para un potencial usuario que visualizará alguna de sus páginas web es muy similar al descrito en la Fig. 5.2.3.1., y ayuda a entender 2 conceptos con los que abrimos la introducción de este apartado: PHP como "lenguaje interpretado" y orientado al desarrollo de "páginas web dinámicas". A continuación describiremos brevemente un posible escenario de nuestro portal web que se ajusta a la figura anterior.

Un usuario pretende visualizar alguna de las páginas de nuestro, posiblemente enviando alguna información adicional a través de un formulario. El navegador web realizará esta solicitud. La petición será recibida por el servidor web donde estará alojada la página web, analizará la misma y allí donde encuentre instrucciones PHP, las interpretará y las "cambiará" por un resultado ya escrito en HTML. Si fuera necesario, podría consultar a la Base de Datos para incorporar alguna información sustraída de la misma. Finalmente devolverá el resultado final al cliente (navegador web) que inició la solicitud o petición.

Como es posible comprobar en la descripción del escenario anterior, la página web será **interpretada** por el servidor previo a ser devuelta al cliente y además, su contenido se generará a partir de algún dato introducido por el usuario, probablemente a través de un formulario, es decir, este contenido no estaba inicialmente en el archivo HTML original, lo que responde al concepto de **página web dinámica** (el resultado no siempre es el mismo como sucedería en una página web estática).

5.2.4. JavaScript :-

JavaScript es un lenguaje de scripting, esto es, un archivo de órdenes, generalmente almacenado en formato de texto plano, que representa a un programa habitualmente de complejidad sencilla que puede efectuar diversas tareas como combinar componentes o bien interactuar con el sistema operativo o con el usuario, motivo por el que es frecuentemente recurrido para diseñar interfaces de usuario.



JavaScript es interpretado por la amplia totalidad de navegadores web modernos. Puede incluirse en cualquier documento y es compatible con HTML en el navegador del cliente. La W3C define que el método óptimo para utilizar JavaScript es incluirlo en un archivo externo, por razones de accesibilidad y rendimiento.

Fue desarrollado por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript coincidiendo con el anuncio de que Netscape soportaría oficialmente Java en su, por aquel tiempo, popular navegador web, estrategia que no estuvo exenta de algunas polémicas. Para el portal objeto de estudio, JavaScript será utilizado principalmente para hacer emerger una subventana o ventana-hija, por ejemplo, para mostrar los detalles de alguno de los elementos de un listado o resultado de una búsqueda (véase Fig. 5.2.4.1).

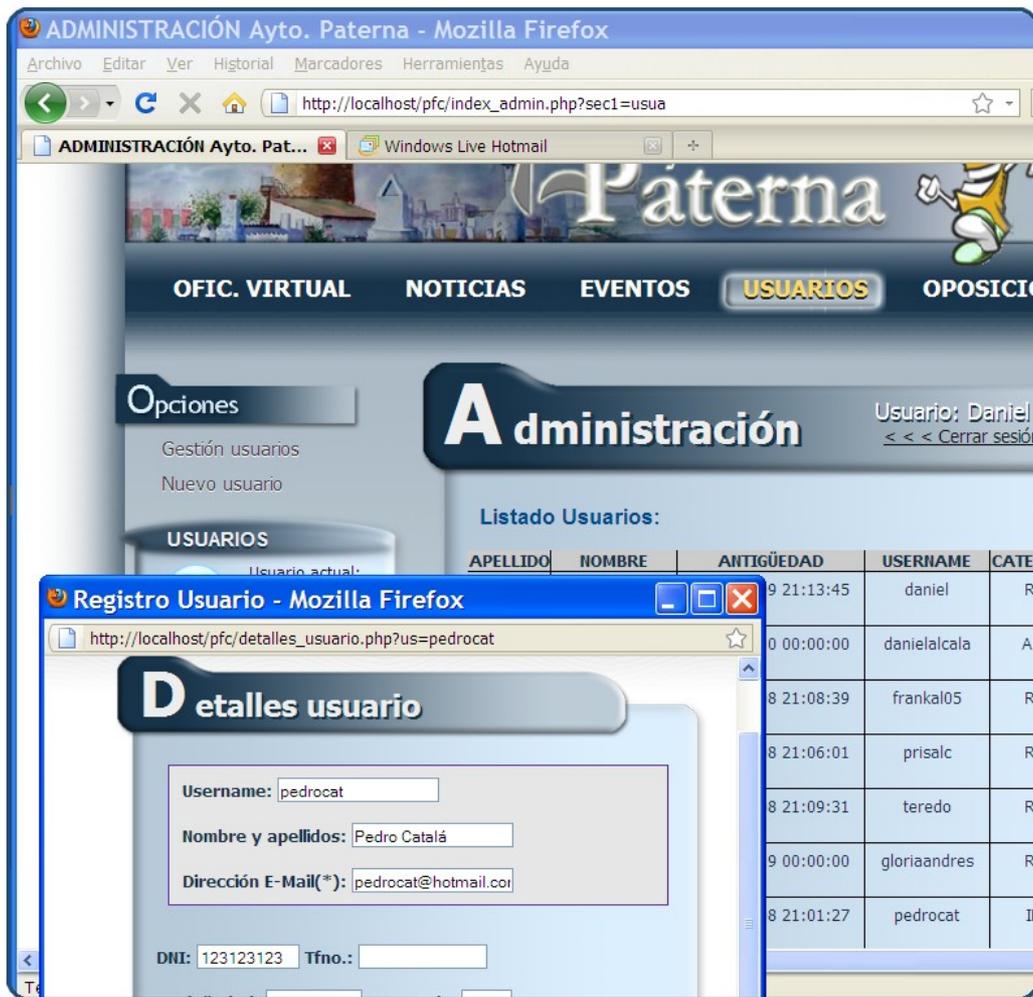


Fig. 5.2.4.1. - Ventana emergente mostrando los detalles de uno de los usuarios del listado

5.2.5. MySQL.

MySQL es un sistema de gestión de base de datos, relacional, multihilo y multiusuario. Relacional porque se sustenta en un modelo de datos (modelo relacional) basado en la lógica de predicados y en la teoría de conjuntos. Multihilo porque permite el acceso concurrente a la información, de forma ordenada y segura. Finalmente es multiusuario porque permite proveer servicio a múltiples usuarios simultáneamente.

MySQL se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, lo que implica comprar una versión comercial licenciada si el usuario no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales.

El uso de MySQL está muy extendido en aplicaciones web, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. Su origen se remonta al año 1995 cuando la empresa MySQL AB busca adaptar el lenguaje SQL (que ya existía desde 1981) de forma que continúe cumpliendo su estándar pero sin sacrificar velocidad, fiabilidad o usabilidad.

The screenshot shows the PHPMYAdmin web interface. The browser address bar displays 'http://localhost/phpmyadmin/'. The page title is 'Servidor: localhost > Base de datos: pfc > Tabla: usuarios'. The interface includes a sidebar with a tree view of databases and tables. The main content area shows a table with columns: dni, nombre, apellido1, apellido2, sexo, fecha_nac, antiguedad, and email. The table contains 5 rows of user data. A SQL query editor is visible above the table, showing a SELECT statement: 'SELECT * FROM `usuarios` LIMIT 0, 30'. The interface also includes navigation buttons like 'Examinar', 'Estructura', 'SQL', 'Buscar', 'Insertar', 'Exportar', 'Importar', 'Operaciones', and 'Vaciar'.

	dni	nombre	apellido1	apellido2	sexo	fecha_nac	antiguedad	email
<input type="checkbox"/>	48123411X	Gloria	Andrés	Gómez	M	1985-12-17	2009-10-19 00:00:00	gloria_despacho@hotmail.com
<input type="checkbox"/>	48384111B	Daniel	Alcalá	Alcocer	H	1977-07-11	2009-02-10 00:00:00	daniel.alcala33@gmail.com
<input type="checkbox"/>	48287722W	Rafa	Nadal		H	1987-12-11	2010-04-09 21:25:06	rafaelnadal@clubtenismadrid.com
<input type="checkbox"/>	22887766W	Maria Teresa	Alcocer		M	1961-09-11	2010-04-08 21:09:31	alcocer_terdom@gva.es
<input type="checkbox"/>	41223311B	Daniel			H	1977-11-11	2010-04-09	trackkman@hotmail.com

Fig. 5.2.5.1. - Utilidad PHPMYAdmin para la gestión de MySQL



Para el caso que nos ocupa, se optó por MySQL por su extremada sencillez de manejo, gracias en parte a la interface PHPMyAdmin. Esta interface organiza la información en base a tablas y permite manipular a información, tanto de contenido como de estructura, de forma rápida y confiable.

5.3. Herramientas .-

A continuación se examinan algunas de las utilidades o aplicaciones que han sido de gran ayuda para conformar el resultado final.

5.3.1. Apache .-

Apache es un servidor web HTTP de código abierto para plataformas Unix, Windows y Macintosh, entre otras. Su uso está muy extendido: desde el año 1996 ocupa la mayor cuota del mercado, detectándose un altísimo valor de cuota del 70% en el año 2005.

Apache se usa originariamente para enviar páginas web en internet. Inclusive existe una gran mayoría de aplicaciones web que están diseñadas asumiendo que serán utilizadas algunas de sus características.

También es utilizado como entorno de desarrollo web, como es nuestro caso, con el fin de previsualizar y probar el código que va siendo implementado. También es usado en aquellas ocasiones donde la información debe estar dispuesta de forma segura y confiable (por ejemplo, al compartir ficheros desde un ordenador personal con Internet).

Otras de las características destacadas que sustentan su gran éxito son: es altamente configurable, modular (es posible extender su funcionalidad) y provisiona autenticación en base de datos y negociado de contenido.

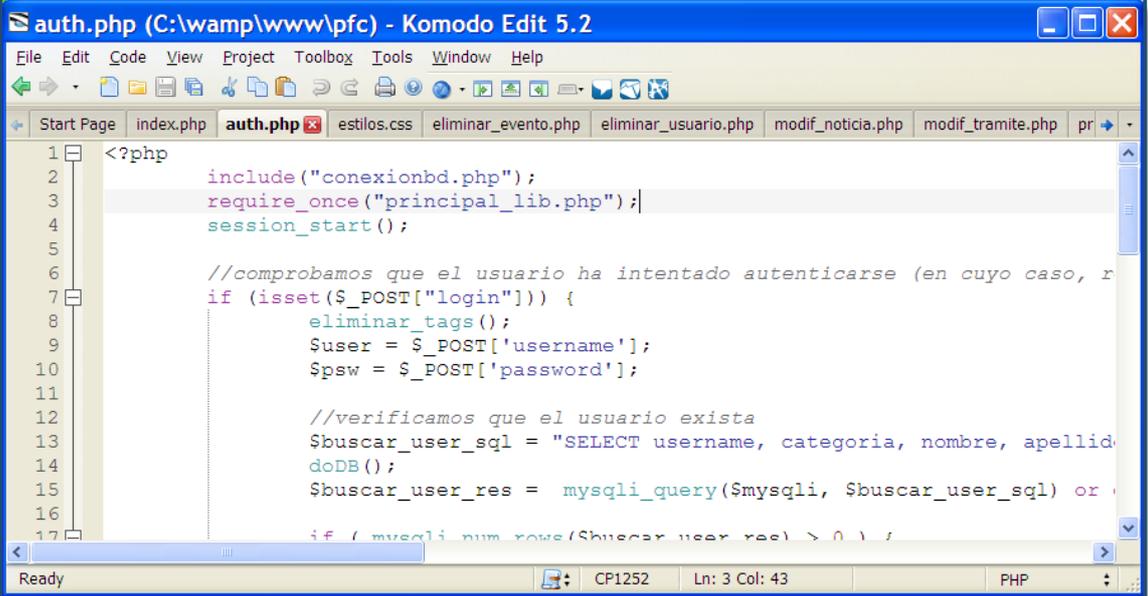


5.3.2. Komodo Edit .-

Komodo Edit es un entorno integrado de desarrollo (IDE) multiplataforma y multilenguaje para la elaboración de aplicaciones web dinámicas. Incluye entre otras, funciones muy demandadas en este tipo de aplicaciones como: autocompletado, comprobación de sintaxis y resaltado del lenguaje.

Komodo Edit ha sido preparado para los lenguajes dinámicos PHP, Ruby, Perl, Python y TCL; también permite trabajar con XML, HTML, Javascript y CSS. La aplicación hace uso de los denominados 'calltips', menús desplegables que permite seleccionar los atributos de las diferentes etiquetas durante la edición del código.

Para el proyecto que nos ocupa, Komodo Edit ha resultado ser de gran utilidad, aumentando de forma constante la legibilidad del código, gracias a sus funciones de autotabulación, coloreado y ayuda en la sintaxis. Por otro lado, se integra a la perfección con programas de FTP, por lo que podíamos subir a un proveedor de alojamiento web el archivo que estaba siendo editado y verificar los resultados de forma instantánea.



```
1 <?php
2 include("conexionbd.php");
3 require_once("principal_lib.php");
4 session_start();
5
6 //comprobamos que el usuario ha intentado autenticarse (en cuyo caso, r
7 if (isset($_POST["login"])) {
8     eliminar_tags();
9     $user = $_POST['username'];
10    $psw = $_POST['password'];
11
12    //verificamos que el usuario exista
13    $buscar_user_sql = "SELECT username, categoria, nombre, apellido
14    doDB();
15    $buscar_user_res = mysqli_query($mysqli, $buscar_user_sql) or
16
17    if ( mysqli_num_rows($buscar_user_res) > 0 ) {
```

Fig. 5.3.2.1. - Captura de la aplicación Komodo Edit

5.3.3. Adobe Photoshop -

Posiblemente Photoshop sea una de esas herramientas que no requieren presentación alguna.

Su historia se remonta a finales de los 80, cuando su creador, el recién licenciado Thomas Knoll, comenzó a trabajar en su doctorado sobre "el procesamiento de imágenes digitales". Fue entonces cuando Thomas adquirió un ordenador Mac (por aquel entonces, ni tan siquiera mostraban imágenes en escala de grises) y escribió un programa llamado Display que permitiese mostrar imágenes en escala de grises empleando tramas de 1 bit (blanco y negro). Éste fue el comienzo de Photoshop; Thomas iría ampliando sus las funcionalidades de Display al mismo tiempo que iría dispersándose de su tesis doctoral.

En la actualidad, Photoshop es el resultado de muchos años de investigación, reinventándose una y otra vez. Su versión más actualizada es CS5 y algunas de sus funciones más recientes especialmente interesantes en un entorno web son:

- Creación de logotipos, botones web e ilustraciones 3D desde cualquier capa de texto, selección, trazado o máscara de capa y, a continuación, es posible torcer, rotar, extrujar, biselar e inflar los diseños.



- Selección extremadamente fácil de elementos intrincados de la imagen, como el pelo, para perfeccionarlos, componerlos o colocarlos en una composición.
- Deformación o estiramiento de gráficos, texto o elementos de la imagen con precisión con el fin de crear apariencias únicas para los diseños web.
- Servicio online que añaden comentarios a las imágenes desde el navegador web de un usuario y estos aparecerán inmediatamente en su pantalla.
- Creación de objetos y diseños 3D con una sencilla creación de sombras, luces basadas en la imagen y materiales como cristal y cromo.
- Multitud de efectos novedosos, entre ellos, efectos pictóricos realistas y naturales con el pincel mezclador, que ofrece la posibilidad de fusionar colores en lienzos.

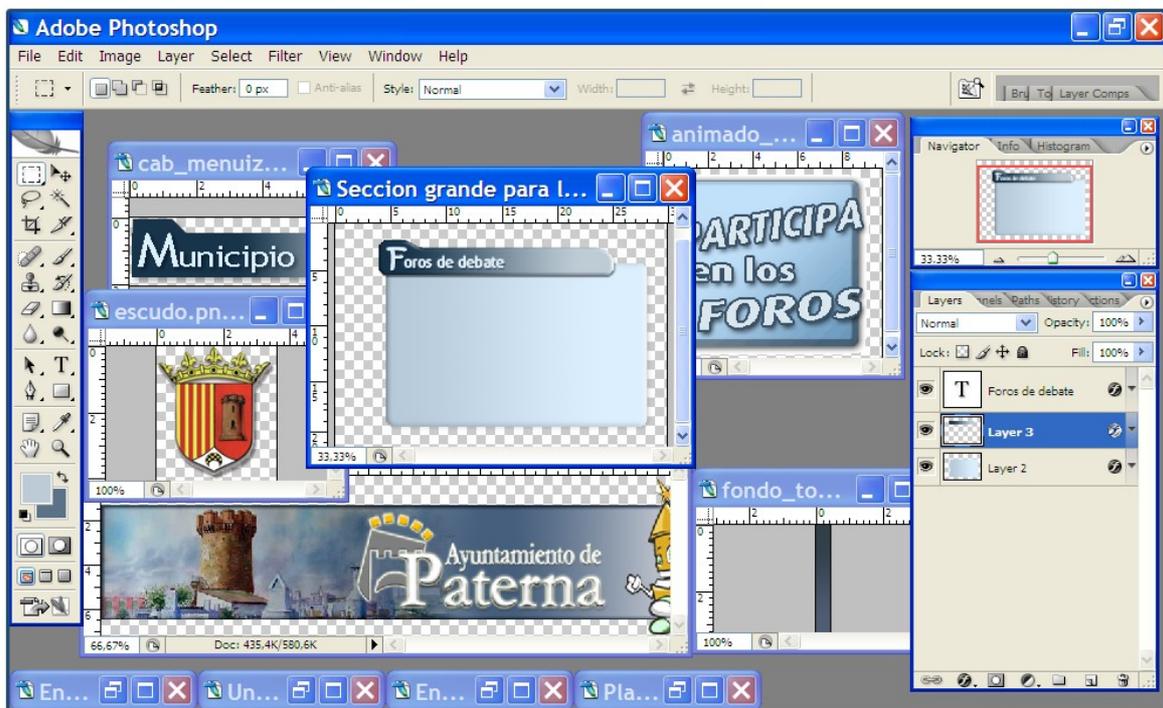


Fig. 5.3.3.1. - Captura de la aplicación Adobe Photoshop



En nuestro caso de estudio, fue utilizado Photoshop con los siguientes objetivos principales:

- diseño de cabeceras y pies de página, realizando fotomontajes y fusión de varias fotografías, añadiendo textos y aplicando diversos efectos.
- diseño de las diferentes cajas que muestran la información de una sección o subsección en particular, así como la caja de búsquedas.
- diseño de algunos tipos de letra complejos que no era posible obtener mediante HTML+CSS.
- diseño de iconos animados (GIF).
- optimización de la calidad en relación al espacio en bytes ocupado por cada uno de los elementos gráficos incluidos en una página web (optimización tiempos de carga sin merma visual aparente), en función del formato gráfico más adecuado a cada contexto.
- diseño de los efectos dinámicos utilizados en las barras de navegación (elemento activo, elemento no seleccionado, etc).

5.3.4. FileZilla.-

FileZilla es un cliente FTP, gratuito, libre (GNU) y de código abierto. Se trata por tanto de una aplicación que permite transferir archivos, conectándose para ello a un servidor FTP.

La vista de archivo y carpeta, situada debajo del registro de mensajes, proporciona una interfaz gráfica amigable. Los usuarios pueden navegar por las carpetas, ver y alterar sus contenidos tanto en la máquina local como en la remota, utilizando una interfaz de tipo árbol de exploración. Los usuarios pueden arrastrar y soltar archivos entre los ordenadores local y remoto.



Empleamos este cliente FTP en nuestro caso de estudio, fundamentalmente con el objetivo de ir subiendo a un servidor que ofreciese alojamiento web, las diferentes partes del portal conforme éstas eran programadas, pudiendo verificar de esta forma si todas ellas cumplían y se comportaban según lo esperado. Por otro lado, FileZilla se integró a la perfección con la aplicación de edición de código Komodo Edit, lo que aún facilitó más esta labor.

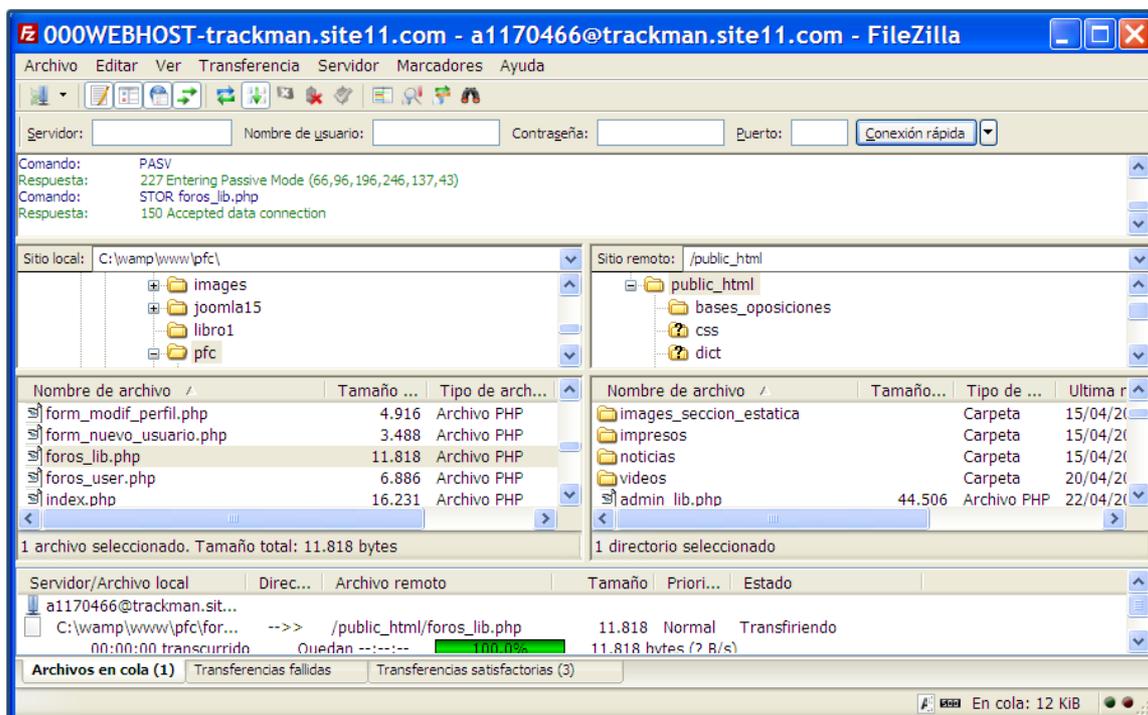


Fig. 5.3.4.1. - Captura de la aplicación FileZilla

5.4. Detalles de implementación .-

Atendiendo a la estructura de 3 niveles que fue definida en un capítulo anterior y una vez revisados de forma general aquellos recursos y tecnologías web a los que se ha recurrido, se describirá de forma concisa cómo han sido implementadas estas tecnologías para el caso que nos ocupa.

5.4.1. Implementación del Interfaz .-

La elaboración del interfaz puede iniciarse mediante aspectos básicos de presentación como ahora un “blueprint” (Rosenfeld, 2002) o retícula (véase Figura 5.4.1.1) que pone de manifiesto las diferentes áreas en que puede dividirse la página principal del portal.

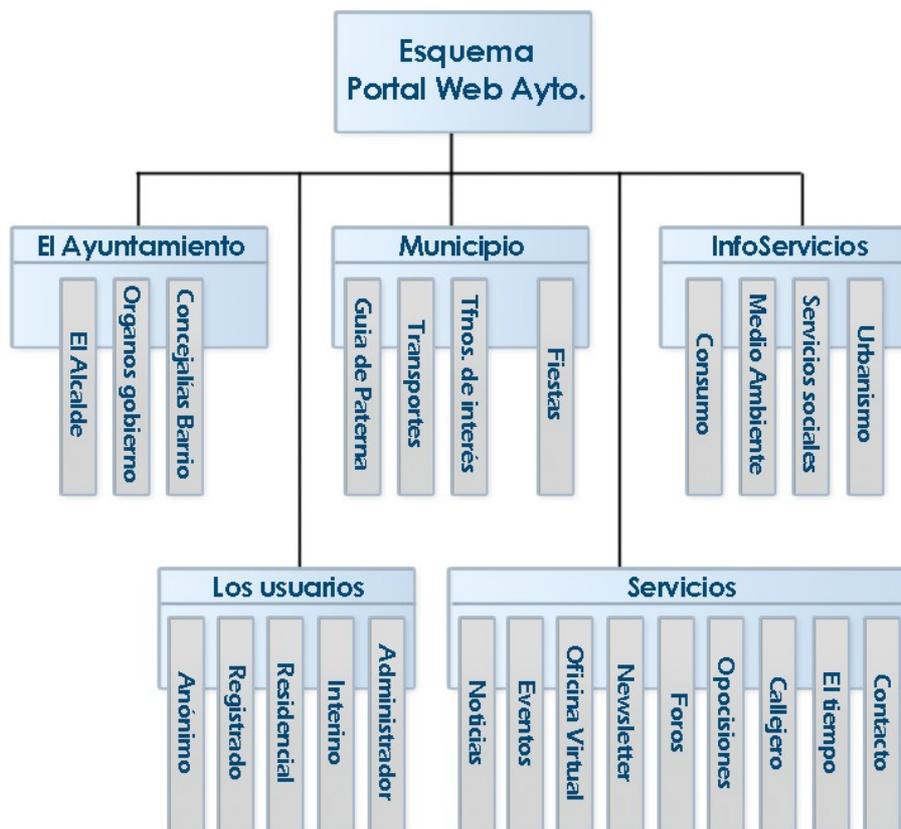


Fig. 5.4.1.1. – Esquema de contenidos de la aplicación Portal Web de Ayuntamiento.



La retícula ilustra las principales secciones del portal y sus contenidos, basándose en una estructura jerárquica aunque no se ha utilizado una notación concreta para representar sus componentes.

Tomando la retícula anterior como punto de partida, se representa el diseño de la página Web mediante un formato de documento HTML (véase Figura 5.4.1.2.) basado en elementos de tipo "capa" (<div>). El diseño original se efectuó en el editor gráfico Photoshop, siendo trasladado posteriormente a lenguaje HTML junto con CSS. Se describen a continuación algunas de las partes relevantes en que se divide el diseño, junto con la estrategia buscada, incluyendo la página en sí de forma global:

Página principal (Inicio)

La página principal muestra un compendio sintetizado de las secciones principales que pueden encontrarse a lo largo del portal. Presenta un aspecto visual dinámico, conteniendo diversos iconos animados que invitan a participar en alguna de las secciones, para lo que previamente habrá que autenticarse en el portal. El dinamismo se consigue también integrando una pequeña área dedicada a una galería fotográfica en la misma cabecera donde es exhibido el título del portal.

Se ha buscado una combinación de colores uniforme, basándose principalmente en variaciones y degradados del color azul, prevaleciendo tonalidades suaves.

El diseño de las diferentes cajas que muestran la información propia de cada sección, responden a un patrón de diseño similar, pudiéndose detectar sutiles variaciones.

El escudo del ayuntamiento dispondrá de una ubicación fija ubicada en la columna izquierda.

Menú Superior

En la parte superior izquierda, sobre la cabecera, se muestra se muestra un menú con enlaces a apartados del portal de relevancia secundaria: Inicio, Mapa Web y Contacto o Sugerencias. Cuando el usuario autenticado sea Funcionario, este menú le permitirá además visualizar el portal como si de un usuario No Funcionario se tratase.



Fig. 5.4.1.2. – Ejemplo de diseño de la página principal de la aplicación Portal Web de Ayuntamiento.



Fecha

En la parte superior derecha es mostrada la fecha actualizada, junto con el nombre del día de la semana.

Cabecera (Fig. 5.4.1.3)

Muestra una panorámica de uno de los monumentos más emblemáticos de Paterna, la Torre, junto con una dibujo caricaturizando al motivo de la fiesta más popular del pueblo, la Cordá. Finalmente, existe una parte dinámica con fotografías del pueblo que van siendo mostradas sucesivamente.



Fig. 5.4.1.3 – Cabecera del portal

Cuerpo o área de contenidos

Área principal dedicada a mostrar de forma sintetizada una porción de las diversas secciones principales del portal, esto es, noticias, eventos y foros. En el caso de cualquier otra página distinta de la principal, este área se dedicará exclusivamente a mostrar la información propia de esa sección.

Subsecciones (Secciones al detalle) (Fig. 5.4.1.4)

Área disponible únicamente en la página principal, en la parte inferior justo precediendo al pie de página. Su cometido es mostrar un resumen de los últimos elementos introducidos en el portal referente a impresos, oposiciones o usuarios registrados en la Comunidad.

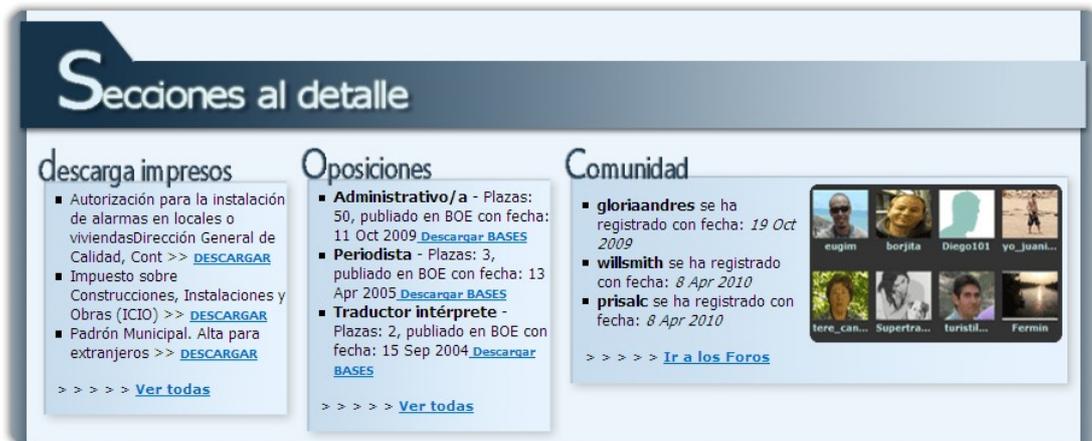


Fig. 5.4.1.4 – Secciones al detalle de la página principal

En la Figura 5.4.1.5. se muestra una relación de elementos <div> empleados en el diseño de la página principal, comenzando con la capa principal que “anida” al resto de capas. Conviene destacar que el tamaño de éstas, está especificado como “automático”, representando de este modo el hecho de que están anidadas o incluidas en la capa principal.



Clase	Tamaño	Contenidos
Principal	100%	
Submenú Superior	Automático	Lista de opciones
Título	Automático	Nombre del Ayuntamiento
Galería fotos animada	Automático	Icono animado con fotografías de la población
Menú navegación principal	Automático	Lista de secciones (servicios) principales
Escudo	Automático	Escudo del Ayuntamiento
Menú navegación lateral	Automático	Lista de secciones InfoAyuntamiento (sección estática)
General	Automático	Área de contenidos
Noticias	Automático	Últimas noticias publicadas
Eventos	Automático	Últimos eventos publicados
Foros	Automático	Últimos mensajes publicados
Subsecciones lateral derecho	Automático	Área dedicado a secciones menores
Usuarios	Automático	Área de autenticación y perfil de usuario
Agenda	Automático	Vista mensual de la agenda con eventos marcados
Callejero	Automático	Vista reducida del mapa de la población
El tiempo	Automático	Vista reducida predicción a 3 días del tiempo
Secciones al detalle	Automático	Área dedicado a secciones menores
Impresos	Automático	Lista de últimos impresos publicados
Oposiciones	Automático	Lista de últimas oposiciones publicados
Comunidad	Automático	Lista de últimos usuarios registrados
Submenú pie de página	Automático	Lista con repetición de opciones y enlace a secciones

Fig. 5.4.1.5 – Estructura de elementos "div".

Asimismo se muestra en la Figura 5.4.1.6. una tabla que representa una estructura de aquellos elementos CSS de mayor relevancia que intervienen en la apariencia de la página principal.

Clase	Características
container_12	margin-left:auto;margin-right:auto;width:960px
grid_X	display:inline;float:left;margin-left:10px;margin-right:10px
clear	clear:both;display:block;overflow:hidden;visibility:hidden;Width:0;height:0
BODY	background-color: #eee; color: #FFF; font-family: verdana, Arial, helvetica;
contenedor_para_sombras	width:1222px; height:auto; position: absolute; top: 80px; left: -40px; background: url(..images/sombras_laterales.png) Repeat-y; overflow:hidden;
cabecera_sup	background: #acbbc6 url(..images/fondo-cabecera2.gif) Repeat-x;
menu_cabecera_sup	margin-bottom: 35px; padding:5px 0 0 5px;
cab_gral_izda	margin-top: -30px;
cab_gral_dcha	padding:0 0 0 3px; margin-top: -30px;
cabecera_inf	background: url(..images/fondo-cabecera2-inv.gif) repeat-x; Padding-bottom: 60px; margin-top: -35px;
buscador	float: right; margin: -78px -10px -20px 0;
cuerpo	background: #ecf5fb url(..images/fondo-cuerpo.gif) repeat-x;
escudo	margin: 10px 0 20px 40px;
cabecera_pie	margin: 0 0 10px -8px;
piepagina	background: transparent url(..images/piepagina3.png) no-Repeat; width: 1000px; height: 205px; margin: 30px 0 -8px -7px;
piepagina p	margin: 80px 0 0 -30px; font-size: 120%; text-align: right;
texto_pie	text-align: center; margin: 60px 70px 0 0; font-size: 110%;

Fig. 5.4.1.6. – Relación de elementos CSS.

Mención especial aparte requiere la estrategia de diseño perseguida para mostrar las diversas cajas que contienen la información en sí de las distintas secciones o apartados (Noticias, Impresos, Usuarios, etc). Estas cajas están programadas en HTML y CSS, haciendo uso del modelo de caja que fue presentado en el apartado de Tecnologías. Veamos a continuación el código HTML asociado a una de ellas, por ejemplo de la caja de la sección de “El Tiempo”:

```
<div class="cajapeq">

    <div class="cajapeq_arriba">
        <h3>EL TIEMPO</h3>
    </div>

    <div class="cajapeq_centro" id="el_tiempo">
        Aquí debe ir el código que muestra la información del tiempo
    </div>

    <div class="cajapeq_abajo">
    </div>

</div> <!-- end of cajapeq-->
```

Fig. 5.4.1.7 – Implementación en HTML de la caja "El Tiempo"

Como se observa en la Figura 5.4.1.7, la caja está definida por una serie de capas que están contenidas en otra principal. A esta capa principal, se le asigna una clase en CSS de nombre 'cajapeq'. Posteriormente se definen otras 3 capas con sus respectivas clases, de las cuales solamente las 2 primeras tienen algún contenido: la primera muestra el título "El Tiempo" y la segunda la información propiamente sobre el tiempo (cuyo código se ha obviado para aumentar la legibilidad).

Lo más destacable es que además del contenido definido en el código HTML, se recurre a CSS para definir una imagen de fondo a mostrar con cada subcapa de la principal, independientemente de si existe o no contenido en HTML. De esta forma, cada subcapa contiene una parte de la imagen final de la caja "El Tiempo". Asimismo se utiliza CSS para corregir la posición de cada parte, buscando la unión de todas ellas para mostrar una imagen final contigua. Por este motivo, a pesar de que la subcapa cuya clase CSS es 'cajapeq_abajo' está vacía de contenido en HTML, es preciso definirla para posteriormente aplicarle una imagen de fondo en CSS (la parte de la imagen inferior que cierre la imagen global de la capa sobre "El Tiempo").

A continuación se muestra en la Figura 5.4.1.8., la implementación en CSS de las clases asociadas a la caja anterior:

```
.cajapeq { width: 249px; margin: 10px 0 0 -20px; color: #111; }
.cajapeq a { color: #024b8b; text-decoration: none; }
.cajapeq a: hover {
    color: #fff1a6; font-weight: bold;
    text-shadow: 1px 1px #18344b; text-decoration: underline;
}
.cajapeq_arriba {
    width: 249px; height: 44px;
    background-image: url("../images/cajapeq_arriba.png");
    background-position: top left; background-repeat: no-repeat; color: #eee;
}
.cajapeq_arriba h3 {
    padding: 13px 0 0 45px; font-size: 120%; text-shadow: 0 2px #18344b;
}
.cajapeq_centro {
    width: 249px; padding: 0 28px 0 35px;
    background-image: url("../images/cajapeq_centro.png");
    background-repeat: repeat-y; overflow: hidden;
}
.cajapeq_centro p { margin: 0; padding: 0 20px 0 0; font-size: 90%; }
.cajapeq_abajo {
    width: 249px; height: 19px;
    background-image: url("../images/cajapeq_abajo.png");
    background-position: bottom left; background-repeat: no-repeat;
}
```

Fig. 5.4.1.8 – Implementación en CSS de las clases asociadas a la caja "El Tiempo"

Se observa en el código CSS anterior cómo son definidas las imágenes de fondo que conforman la imagen de la caja final, así como las instrucciones necesarias para hacer que éstas se vean unidas y de forma contigua, entre otras. Obsérvese también como la clase 'cajapeq' asociada a la capa principal, no contiene ninguna imagen de fondo puesto que su cometido no es otro que el de contener a las subcapas, definiendo así una estructura final que pueda ser manipulada como un todo.

Finalmente se muestra en la Figura 5.4.1.9 una captura de la caja final que

acabamos de describir en base a la implementación vista en CSS y HTML, acompañándose del esquema de clases definido.



Fig. 5.4.1.9 - Estructura de capas de la caja que contiene la información sobre 'El Tiempo'

Una vez han sido tratados los aspectos de presentación, se describen a continuación los elementos que permiten interactuar con los mismos:

Menu de navegación principal (MENU PPAL.) (Fig. 5.4.1.10)

Bajo la cabecera que exhibe el título, se habilita una barra de navegación horizontal con enlaces a las diferentes secciones principales del portal, con un comportamiento dinámico (un efecto tipo 'pestaña activa' realza la sección actual). Cabe destacar que los efectos dinámicos han sido implementados únicamente en CSS, sin precisar de añadir parte alguna de código en JavaScript.



Fig. 5.4.1.10 – Menú de navegación principal y buscador en modo 'sencillo'

Mención especial aparte requiere la estrategia empleada, utilizando HTML y CSS,

para implementar la barra de navegación anterior.

```
<div id="top-nav">

<ul id="navi">

$navegador = array("INICIO" => "index_user.php?", "NOTICIAS" => "noticias_public.php?",
"AGENDA" => "agenda_public.php?", "TRAMITES" => "tramites_public.php?
list_tramites&", "FOROS" => "foros_user.php?list_topics&");

foreach ($navegador as $clave => $valor) {
    if ($_GET["pag"] == $clave)
        echo "
        <li class=\"current_page_item\">
            <a href=\"\$navegador[$clave]pag=$clave\">$clave</a>
        </li>";
    else
        echo "
        <li>
            <a href=\"\$navegador[$clave]pag=$clave\">$clave</a>
        </li>";
}

</ul>

</div> <!-- end of top-nav-->
```

Fig. 5.4.1.11 – Implementación HTML para la barra de navegación principal

Se observa en la Figura 5.4.1.1 cómo son generadas las distintas opciones de la barra de navegación, empleando para ello una estructura de tipo vector. Cada elemento del mismo está representado por un par (clave → valor) donde la clave contiene una opción del menú (el título de una sección) mientras que su valor es el nombre del archivo que contiene la página web de la sección. Es decir, se asocia la página de un sección al título de la misma.

Posteriormente, se recorre cada elemento del vector, imprimiéndose en pantalla el título de cada opción del menú (el nombre de la clave), al mismo tiempo que se le

asocia un enlace a la página web que contiene dicha sección (el valor "apuntado" por la clave).

Para resolver el apartado dinámico (se resalta la sección actual con un efecto tipo "pestaña"), se recurre a variables enviadas mediante el método GET. De este modo, cuando se imprime cada opción del menú asociándosele un enlace a la página web de la sección a que corresponde, se añade además a la parte final del enlace, una variable de nombre 'pag' cuyo valor será precisamente la clave del elemento en cuestión (esto es, el título de la opción del menú). De esta forma, antes de imprimir cada opción del menú se revisa el valor de 'pag' enviado a través del método GET y se le asigna la clase CSS 'current_page_item' sólo en el de coincidir con la sección actual. Esta clase CSS contiene precisamente la descripción de cómo debe mostrarse un elemento activo del menú, esto es, el efecto de resaltado referido antes.

```
#top-nav a:hover { text-decoration: underline; font-size: 150%; }
#top-nav ul { padding: 0; }
#top-nav li { padding-left: 10px; }
#navi { height: 60px; }
#navi, #navi ul { list-style: none; line-height: 1; z-index: 100; padding: 0; }
#navi a {
    display: block; line-height: 25px; z-index: 100; padding: 0 10px; color:#000;
    font-size: 14px; font-weight: 700;
}
#navi li { float: left; line-height: 30px; margin: 0; }
#navi li a {
    color:#fff; text-decoration: none; padding: 0 0 0 16px;
    text-shadow: 0 1px #18344b; font-size: 140%;
}
#navi li a span { display: inline-block; height:30px; padding: 5px 15px 5px 5px; }
#navi li a: hover, #navi li.active a, #navi li.current_page_item a {
    background: url(../images/top-nav-left.png) no-repeat left; color: #fdd967;
    cursor: pointer; text-decoration: none;
}
#navi li.active a span, #navi li.current_page_item a span, #navi li a: hover span {
    background: url(../images/top-nav-largo.png) no-repeat right;
    text-shadow: 0 2px #222;
}
}
```

Fig. 5.4.1.12 – Implementación CSS para la barra de navegación principal



Algunos aspectos a destacar del código CSS mostrado en la Figura 5.4.1.12 son:

- la definición de la imagen de fondo a utilizar para el resaltado de la opción del menú. Es preciso dividir la imagen en 2 partes (izquierda y derecha). De esta forma, la misma puede crecer para adaptarse a la longitud de cualquier palabra.
- la disposición en horizontal de la lista que contiene los elementos de la barra de navegación.
- la definición del efecto de sombra para los títulos de las diferentes opciones del menú.

Menú Información Ayuntamiento, Municipio y Servicios (INFO. AYO) (Fig. 5.4.1.13)

Sobre la columna izquierda se dedica un área a la parte más estática del portal, esto es, enlaces a la información sobre el Ayuntamiento, el Municipio y los Servicios. Se utiliza para ello un menú vertical dinámico implementado siguiendo la misma estrategia que fue mostrada para el menú horizontal principal.

Menú opciones usuarios (Fig. 5.4.1.14)

En aquellos apartados que requieren interacción con el Usuario (tramitaciones, participar en foros, etc), serán mostrados enlaces a las diversas opciones posibles de actuación. Estas opciones cambiarán en función de la sección en que se encuentre el usuario así como de los privilegios de que disponga el mismo. Se comprueba con este fin el valor de la variable de sesión 'categoria'.

Menú pie de página (Fig. 5.4.1.15)

El pie de página contiene una repetición de enlaces a los apartados o servicios de mayor popularidad además de una imagen panorámica de fondo que exhibe una de las zonas más turísticas del pueblo: la Torre de Paterna.



Fig. 5.4.1.13 – Menu Info. Ayto (parte estática)



Fig. 5.4.1.14 – Menú de opciones de usuario



Fig. 5.4.1.15 – Enlaces a pie de página

Buscador (Fig. 5.4.1.16)

El buscador ha sido integrado en una caja como muestra la imagen. El mismo dispondrá de 2 versiones: una simple que será mostrada para la página principal y otra más avanzada, que será mostrada en el resto de páginas. La versión avanzada permitirá al usuario limitar la búsqueda a la sección actual o bien iniciarla en todo el portal.

Fig. 5.4.1.6 – Caja del buscador en su versión avanzada.

Usuarios (Fig. 5.4.1.17)

Caja que permite autenticarse, así como también mostrar la información referente al usuario autenticado y otras opciones (modificar su perfil, cerrar la sesión). Se trata de un elemento destacado que siempre dispondrá de una ubicación fija, habitualmente preferente en la columna derecha (página principal) o bien izquierda (resto de páginas).

Fig. 5.4.1.17 – Caja dedicada al apartado Usuarios

5.4.2. Implementación de la funcionalidad .-

En este apartado, se expondrán algunas secciones de código que implementan las funciones o procesos que forman parte del nivel de Negocio. Por cuestiones obvias de extensión, únicamente se plasman aquéllas cuya relevancia sea destacada.



Visualización de la página principal. Reubicación dinámica de las cajas.

La página principal contendrá 5 cajas dedicadas a diversas secciones, cuya ubicación no será fija sino dinámica. La posición que ocupen dentro de la página principal estará relacionada con la frecuencia con que el usuario ha visitado antes cada una de esas secciones (ubicación más arriba para cajas con mayor puntuación).

El área principal de contenidos contendrá 3 de esas cajas dinámicas, concretamente las dedicadas Noticias, Eventos y Foros, mientras que la columna lateral derecha contendrá las 2 cajas restantes, en este caso, las dedicadas a las secciones El Tiempo y Callejero. Estas cajas aparecerán por orden de preferencia (arriba las de las secciones más visitadas por el usuario).

Cada vez que un usuario visita una sección de las 5 citadas anteriormente, es aumentado el contador asociado a la sección en cuestión. Los contadores se encuentran almacenados en la Base de Datos, en la tabla 'preferencias_usuarios'. El código de la función que aumenta el contador de una sección, se muestra a continuación, así como el código de la función que limita el valor máximo que puede alcanzar un contador (para evitar que los contadores alcance valores innecesariamente elevados):

```
function resetear_preferencias() { //divide entre 10 el contador de cada sección
    $obtener_prefer_sql = "SELECT noticias, eventos, foros, tiempo, callejero FROM
    preferencias_usuarios WHERE username = ".$_SESSION["username"]."";
    $mysqli = doDB();
    $obtener_prefer_res = mysqli_query($mysqli, $obtener_prefer_sql) or
    die(mysqli_error($mysqli));

    $prefer_info = mysqli_fetch_array($obtener_prefer_res);
    foreach( $prefer_info as $variable => $valor ) {
        $prefer_info[$variable] = ceil($prefer_info[$variable] / 10); // ceil calcula la
        div entera, redondeando por arriba
    }
    $actualizar_prefer_sql = "UPDATE preferencias_usuarios SET noticias =
    $prefer_info[0], eventos = $prefer_info[1], foros = $prefer_info[2], tiempo =
    $prefer_info[3], callejero = $prefer_info[4] WHERE username =
```



```
    ".$SESSION["username"]."";
    $actualizar_prefer_res = mysqli_query($mysqli, $actualizar_prefer_sql) or
    die(mysqli_error($mysqli));

    mysqli_free_result($obtener_prefer_res);
    mysqli_free_result($actualizar_prefer_res);
    mysqli_close($mysqli);
} // end of function resetear_preferencias()

function aumentar_preferencias($seccion) {
    $obtener_preferencias_sql = "SELECT $seccion FROM preferencias_usuarios
    WHERE username = ".$SESSION["username"]."";
    $mysqli = doDB();
    $obtener_preferencias_res = mysqli_query($mysqli, $obtener_preferencias_sql)
    or die(mysqli_error($mysqli));

    $preferencias_info = mysqli_fetch_array($obtener_preferencias_res);
    if ($preferencias_info[0] == 9999) { // limitamos el valor máximo de un contador a
    10.000. Cuando alguno alcance este valor, se dividen todos entre 10.
        $nuevo_ranking = 1000;
        resetear_preferencias();
    } else
        $nuevo_ranking = $preferencias_info[0] + 1;

    $actualizar_preferencias_sql = "UPDATE preferencias_usuarios SET $seccion =
    $nuevo_ranking WHERE username = ".$SESSION["username"]."";
    $actualizar_preferencias_res = mysqli_query($mysqli, $actualizar_preferencias_sql)
    or die(mysqli_error($mysqli));

    mysqli_free_result($obtener_preferencias_res);
    mysqli_free_result($actualizar_preferencias_res);
    mysqli_close($mysqli);
} // end of function aumentar_preferencias()
```

Se muestra a continuación el código de la función encargada de obtener los contadores asociados a las cajas del área principal de contenidos (se ha obviado representar el código que realiza la misma función sobre las cajas del área lateral



derecho, por ser análogo). La función devuelve un vector ordenado de mayor a menor preferencia de la sección en cuestión.

```
function obtener_prefs_cajagran() { //obtiene un vector con las cajas grandes de la zona
central del website, ordenadas de mayor a menor preferencias según el usuario consulte
más unas cajas u otras. Las primeras del vector serán las más visitadas por el usuario.
    $obtener_prefsgran_sql = "SELECT noticias, eventos, foros FROM
preferencias_usuarios WHERE username = ".$_SESSION["username"]."";
    $mysqli = doDB();
    $obtener_prefsgran_res = mysqli_query($mysqli, $obtener_prefsgran_sql) or
die(mysqli_error($mysqli));

    $info_prefsgran = mysqli_fetch_assoc($obtener_prefsgran_res);
    arsort($info_prefsgran);
    return $info_prefsgran;

    mysqli_free_result($obtener_prefsgran_res);
    mysqli_close();
} // end of fuction obtener_prefs_cajagran()
```

Una vez obtenido el vector, desde la página principal se recorre cada uno de sus elementos y para cada uno de ellos, se invoca a la función que imprime por pantalla la caja con sus contenidos de la sección indicada por el elemento en cuestión.

```
function imprimir_prefs_cajasgran($seccion) {

    $salida_php = "
<div class=\"cajagrande\">
    <div class=\"cajagrande_arriba\">

switch($seccion) {

case 'noticias':

    $obtener_noticias_sql = "SELECT id, titular, texto, imagen, fecha FROM
noticias ORDER BY fecha DESC LIMIT 0, 3"; // obtenemos las 3 noticias más
recientes para mostrarlas en la página principal del website
```



```
$mysqli = doDB();
$obtener_noticias_res = mysqli_query($mysqli, $obtener_noticias_sql) or
die(mysqli_error($mysqli));

$salida_php .= "
    <h3>Últimas Noticias</h3>
</div>
<div class=\"cajagrande_centro\">;
$noticia_info = mysqli_fetch_array($obtener_noticias_res);
$noticia_imagen = "noticias/".$noticia_info['imagen'];
$noticia_titular = $noticia_info['titular'];
$fecha_temp = explode(" ", $noticia_info['fecha']);
$fecha_temp = explode("-", $fecha_temp[0]);
$noticia_fecha = $fecha_temp[2].".".$fecha_temp[1].".".$fecha_temp[0];
$noticia_texto = $noticia_info['texto'];
$salida_php .= "
<p><strong>$noticia_titular</strong><br>
<span style=\"font-size:90%;font-style:italic;\">$noticia_fecha</span>
<img src=\"".$noticia_imagen.\" width=\"200\" height=\"205\"
alt=\"Foto_noticia\" class=\"imagen_noticia\" />
<div class=\"texto_noticia\">
    $noticia_texto
</div></p>;
while ($noticia_info = mysqli_fetch_array($obtener_noticias_res)) {
    $noticia_titular = $noticia_info['titular'];
    $noticia_id = $noticia_info['id'];
    $fecha_temp = explode(" ", $noticia_info['fecha']);
    $fecha_temp = explode("-", $fecha_temp[0]);
    $noticia_fecha = $fecha_temp[2].".".$fecha_temp[1].".".$
    $fecha_temp[0];
    $salida_php .= "
    <hr><p> --> <strong><a href=\"javascript:eventWindow
('detalles_noticia.php?id=$noticia_id')\">\".$noticia_fecha.\" - \".
    $noticia_titular.\"</strong></a></p>;
}
mysqli_free_result($obtener_noticias_res);
$salida_php .= "
<hr><p> > > > > > <span style=\"font-size:120%;\"><strong><a
href=\"noticias_public.php?pag=NOTICIAS\"> VER TODAS LAS NOTICIAS
```




```
case 'foros':

    $obtener_posts_sql = "SELECT post_texto, post_fecha, post_prop FROM
    foro_posts ORDER BY post_fecha LIMIT 0, 3";
    $mysqli = doDB();
    $obtener_posts_res = mysqli_query($mysqli, $obtener_posts_sql) or
    die(mysqli_error($mysqli));
    $salida_php .= "
        <h3>Mensajes Foros</h3>
    </div>
    <div class=\"cajagrande_centro\">
    <p>";
    while ($post_info = mysqli_fetch_array($obtener_posts_res)) {
        $post_texto = $post_info['post_texto'];
        $fecha_temp = explode(" ", $post_info['post_fecha']);
        $fecha_temp = explode("-", $fecha_temp[0]);
        $post_fecha =
        $fecha_temp[2].".".$fecha_temp[1].".".$fecha_temp[0];
        $post_prop = $post_info['post_prop'];
        $salida_php .= "
            <p><strong><span style=\"color:#951316; font-size:90%;\">
            $post_texto</span></strong><br>
            <span style=\"font-style:italic;font-size:80%;\">
            <br>Posteado por: $post_prop con fecha: $post_fecha</p><hr>";
        }
        $salida_php .= "
        <p>>>>>>> <span style=\"font-size:120%;\"><strong>
        <a href=\"foros_user.php?list_topics?pag=FOROS\"> IR a los FOROS
        </strong></span></a> <<<<<<< </p></p>";
        mysqli_free_result($obtener_posts_res);

        break;
    };
    $salida_php .= "
        </div> <!-- end of cajacentro -->
        <div class=\"cajagrande_abajo\"></div>
    </div> <!-- end of cajagrande -->";
    echo $salida_php;
    mysqli_close();
} //end of function imprimir_prefs_cajasgran($seccion)
```



Se muestra a continuación el breve código que se precisa en la página principal para coordinar las funciones vistas anteriormente.

```
<?php
$prefs_cajagran = obtener_prefs_cajagran();
foreach ($prefs_cajagran as $clave => $valor) {
    imprimir_prefs_cajasgran($clave);
}
?>
```

Control y envío de Newsletter.

El sistema envía automáticamente un boletín (newsletter) cada que vez que se incorporan 10 nuevos elementos a la base de datos, ya sean de tipo noticias o eventos. No obstante, un usuario puede estar suscrito en una de las 3 variaciones de newsletter posibles: sólo noticias, sólo eventos o ambos (global).

El sistema sólo debe enviar boletines con los últimos 10 elementos incorporados, no debiendo incluir ninguno de estos elementos en futuros boletines. Por este motivo, el sistema controlará de forma permanente la cantidad de nuevo elementos pendientes de envío, para cada una de los 3 tipos de newsletter. Veamos un sencillo ejemplo que ayude a comprender esta estrategia:

Si en el caso particular de una newsletter de tipo global, existiesen 3 noticias y 6 eventos pendientes de envío, cuando se produzca la siguiente incorporación (del tipo que sea), el sistema deberá construir un boletín con esos elementos y enviarlo sólo a aquellos usuarios suscritos en el tipo de newsletter global.

La acción que desencadena el envío de una newsletter, lógicamente tendrá lugar cuando un usuario de tipo funcionario (Administrador o Interino), inserta un nuevo



elemento en la base de datos. En ese preciso instante, el sistema verifica si se cumplen las condiciones para construir y enviar un boletín, para los 3 tipos existentes del mismo.

Cabe destacar del algoritmo diseñado para cumplir con los requisitos anteriores, que serán necesarios 4 contadores o índices distintos que se almacenarán en una tabla de la base de datos y cuyo cometido es el siguiente :

- un contador para controlar el envío de una newsletter parcial de tipo noticias. Cuando este contador llegue a 10, se resetea y a continuación se envía una newsletter con las últimas 10 noticias existentes en la Base de Datos.
- un contador para controlar el envío de una newsletter parcial de tipo eventos, siguiendo el mismo *modus operandi* que en el caso anterior.
- Dos contadores para controlar el envío de una newsletter global. Uno para contabilizar cuántos elementos deben ser incluidos del tipo noticias y otro para contabilizar cuántos del tipo eventos. Obviamente, deberá ser enviada la newsletter cuando ambos contadores sumen un valor de 10.

Se muestra a continuación el código encargado de gestionar cuándo debe ser enviada una newsletter, siguiendo la estrategia descrita:

```
function administrar_newsletter($tipo) {
    /*se obtienen de la BD los contadores asociados a la cantidad de elementos
    pendientes de envío, atendiendo al tipo de newsletter*/
    $obtener_ptes_envio_sql = "SELECT * FROM control_newsletter";
    $mysqli = doDB();
    $obtener_ptes_envio_res = mysqli_query($mysqli, $obtener_ptes_envio_sql) or
    die(mysqli_error($mysqli));

    while ( $ptes_envio_info = mysqli_fetch_array($obtener_ptes_envio_res) ) {
        if ($ptes_envio_info['tipo_newsletter'] == "noticias") {
            $ptes_global_noticias = $ptes_envio_info['pendientes_global'];
            $ptes_individual_noticias =
            $ptes_envio_info['pendientes_individual'];
        } else {
            $ptes_global_eventos = $ptes_envio_info['pendientes_global'];
```



```
        $ptes_individual_eventos =
        $ptes_envio_info['pendientes_individual'];
    }
}
//se incrementan los contadores y si procede, se envia newsletter
if ($tipo == "noticias") { // si se ha insertado una noticia
    $ptes_global_noticias++;
    if ($ptes_individual_noticias < 9)
        $ptes_individual_noticias++; /*se incrementa el contador para
        el tipo de newsletter parcial-noticias*/
    else { /*se han alcanzado 10 nuevas noticias insertadas, se resetea
    contador y se envia newsletter a aquellos subscriptores sólo de noticias*/
        $ptes_individual_noticias = 0;
        enviar_newsletter("noticias");
    }
    //se actualiza contador en la BD
    $actualizar_ptes_individual_sql = "UPDATE control_newsletter SET
    pendientes_individual = '$ptes_individual_noticias' WHERE
    tipo_newsletter = 'noticias'";
    $actualizar_ptes_individual_res = mysqli_query($mysqli, $actualizar_ptes_
    individual_sql) or die(mysqli_error($mysqli));
    mysqli_free_result($actualizar_ptes_individual_res);
} else { //se ha insertado un elemento de tipo evento
    $ptes_global_eventos++;
    if ($ptes_individual_eventos < 9)
        $ptes_individual_eventos++;
    else {
        $ptes_individual_eventos = 0;
        enviar_newsletter("eventos");
    }
    $actualizar_ptes_individual_sql = "UPDATE control_newsletter SET
    pendientes_individual = '$ptes_individual_eventos' WHERE
    tipo_newsletter = 'eventos'";
    $actualizar_ptes_individual_res = mysqli_query($mysqli, $actualizar_ptes_
    individual_sql) or die(mysqli_error($mysqli));
    mysqli_free_result($actualizar_ptes_individual_res);
}
/*si entre noticias y eventos, se alcanza la cantidad de 10 nuevos elementos
insertados, se envía una newsletter de tipo global*/
if ( (($ptes_global_noticias + $ptes_global_eventos) % 10) == 0 ) {
```



```
$actualizar_ptes_global_sql = "UPDATE control_newsletter SET
pendientes_global = '0'";
/*se envía newsletter, indicándose además la cantidad de elementos
a incluir de cada tipo*/
$actualizar_ptes_global_res = mysqli_query($mysqli, $actualizar_ptes_
global_sql) or die(mysqli_error($mysqli));
mysqli_free_result($actualizar_ptes_global_res);
enviar_newsletter("global", $ptes_global_noticias, $ptes_global_eventos);
} else {
    if ($tipo == "noticias")
        $actualizar_ptes_global_sql = "UPDATE control_newsletter SET
pendientes_global = '$ptes_global_noticias' WHERE
tipo_newsletter = 'noticias'";
    else
        $actualizar_ptes_global_sql = "UPDATE control_newsletter SET
pendientes_global = '$ptes_global_eventos' WHERE
tipo_newsletter = 'eventos'";
    $actualizar_ptes_global_res = mysqli_query($mysqli, $actualizar_ptes_
global_sql) or die(mysqli_error($mysqli));
}

mysqli_free_result($obtener_ptes_envio_res);
mysqli_close($mysqli);
} //FIN function administrar_newsletter()
```

A continuación se muestra la implementación de la función encargada de construir el boletín y enviarlo a los usuarios subscriptos, atendiendo al tipo de newsletter:

```
function enviar_newsletter($tipo, $ptes_global_noticias = 0, $ptes_global_eventos = 0) {
    echo "<p><span style=\"font-size:150%; font-style:italic; color:#d64100;\">
Newsletter enviada</span></p><br><br>";
    $para = "novedades@boletin.ayto-paterna.com";
    $asunto = "Boletín novedades AYTO. PATERNA";
    $salida_newsletter = "<hr>ATENCIÓN: No respondas este mensaje, si quieres
comunicarte con nosotros entra en:
http://www.ayto-paterna.com/contacta/";
}
```



PARA DARSE DE BAJA:

Si deseas dejar de recibir nuestros boletines debes darte de baja entrando en <http://www.ayto-paterna.com/boletines>

En esa página encontrarás un formulario para introducir tu correo y darte de baja automáticamente.

Gracias por tu atención y hasta pronto.

```
<hr>;
```

```
$headers = "MIME-Version: 1.0\r\n";
```

```
$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";
```

```
$headers .= "From: newsletter@ayto-paterna.com\r\n";
```

```
$headers .= "Bcc:";
```

```
if ($tipo == "noticias") {
```

```
    $obtener_noticias_sql = "SELECT titular, texto, imagen, fecha FROM  
    noticias ORDER BY fecha DESC LIMIT 0, 10";
```

```
    $obtener_subscriptores_sql = "SELECT email FROM usuarios,  
    preferencias_usuarios WHERE usuarios.username =  
    preferencias_usuarios.username AND newsletter='noticias';"
```

```
    $mysqli = doDB();
```

```
    $obtener_noticias_res = mysqli_query($mysqli, $obtener_noticias_sql) or  
    die(mysqli_error($mysqli));
```

```
    $obtener_subscriptores_res = mysqli_query($mysqli,  
    $obtener_subscriptores_sql) or die(mysqli_error($mysqli));
```

```
    $salida_newsletter .= "<h1>Newsletter Noticias</h1>";
```

```
    while ($noticia_info = mysqli_fetch_array($obtener_noticias_res)) {
```

```
        $titular = $noticia_info["titular"];
```

```
        $texto = $noticia_info["texto"];
```

```
        $imagen = "/pfc/noticias/".$noticia_info["imagen"];
```

```
        $fecha = $noticia_info["fecha"];
```

```
        $salida_newsletter .= "
```

```
        <h3>Noticia: $titular<br>
```

```
        <strong style=\ "font-size:60%\">$fecha</strong></h3>
```

```
        <img src=\ "$imagen\" width=\ "256\" alt=\ "Foto_noticia\">
```

```
        <p>$texto</p><hr>";
```

```
    }
```

```
    while ($suscriptor_info = mysqli_fetch_array($obtener_subscriptores_res)) {
```

```
        $headers .= $suscriptor_info['email'].",";
```

```
    }
```

```
}
```



```
if ($tipo == "eventos") {
    $obtener_eventos_sql = "SELECT * FROM calendario_eventos
ORDER BY id DESC LIMIT 0, 10";
    $obtener_subscriptores_sql = "SELECT email FROM usuarios,
preferencias_usuarios WHERE usuarios.username =
preferencias_usuarios.username AND newsletter='eventos'";
    $mysqli = doDB();
    $obtener_eventos_res = mysqli_query($mysqli, $obtener_eventos_sql) or
die(mysqli_error($mysqli));
    $obtener_subscriptores_res = mysqli_query($mysqli,
$obtener_subscriptores_sql) or die(mysqli_error($mysqli));
    $salida_newsletter = "<h1>Newsletter Eventos</h1>";
    while ($evento_info = mysqli_fetch_array($obtener_eventos_res)) {
        $titulo_evento = $evento_info["titulo_evento"];
        $desc_evento = $evento_info["desc_evento"];
        $inicio_evento = $evento_info["inicio_evento"];
        $salida_newsletter .= "
<h3>Evento: $titulo_evento<br>
<strong style=\"font-size:60%\">$inicio_evento</strong></h3>
<p>$desc_evento</p><hr>";
    }
    while ($suscriptor_info = mysqli_fetch_array($obtener_subscriptores_res)) {
        $headers .= $suscriptor_info['email'].",";
    }
}

if ($tipo == "global") {
    /*se verifica si al menos existe un elemento pendiente de envío del tipo
noticias, para no ejecutar la consulta a la BD de forma innecesaria*/
    if ($ptes_global_noticias > 0) {
        $obtener_noticias_sql = "SELECT titular, texto, imagen, fecha FROM
noticias ORDER BY fecha DESC LIMIT 0, $ptes_global_noticias";
        $mysqli = doDB();
        $obtener_noticias_res = mysqli_query
($mysqli, $obtener_noticias_sql) or die(mysqli_error($mysqli));
    }
    if ($ptes_global_eventos > 0) {
        $obtener_eventos_sql = "SELECT * FROM calendario_eventos
ORDER BY id DESC LIMIT 0, $ptes_global_eventos";
```



```
        $mysqli = doDB();
        $obtener_eventos_res = mysqli_query
        ($mysqli, $obtener_eventos_sql) or die(mysqli_error($mysqli));
    }
    $obtener_subscriptores_sql = "SELECT email FROM usuarios,
    preferencias_usuarios WHERE usuarios.username =
    preferencias_usuarios.username AND newsletter='total'";
    $salida_newsletter = "<h1>Newsletter Noticias y Eventos</h1>";
    if ($ptes_global_noticias > 0)
        while ($noticia_info = mysqli_fetch_array($obtener_noticias_res)) {
            $titular = $noticia_info["titular"];
            $texto = $noticia_info["texto"];
            $imagen = "/pfc/noticias/" . $noticia_info["imagen"];
            $fecha = $noticia_info["fecha"];
            $salida_newsletter .= "
            <h3>Noticia: $titular<br>
            <strong style='font-size:60%'>$fecha</strong></h3>
            <img src=\"$imagen\" width=\"256\" alt=\"Foto_noticia\">
            <p>$texto</p><hr>";
        }
    if ($ptes_global_eventos > 0)
        while ($evento_info = mysqli_fetch_array($obtener_eventos_res)) {
            $titulo_evento = $evento_info["titulo_evento"];
            $desc_evento = $evento_info["desc_evento"];
            $inicio_evento = $evento_info["inicio_evento"];
            $salida_newsletter .= "
            <h3>Evento: $titulo_evento<br>
            <strong style='font-size:60%'>$inicio_evento</strong></h3>
            <p>$desc_evento</p><hr>";
        }

    while ($suscriptor_info = mysqli_fetch_array($obtener_subscriptores_res)) {
        $headers .= $suscriptor_info["email"].",";
    }
}

$headers .= "\r\n";
mail($para, $asunto, $salida_newsletter, $headers);
```



```
mysqli_free_result($obtener_noticias_res);  
mysqli_free_result($obtener_eventos_res);  
mysqli_free_result($obtener_subscriptores_res);  
mysqli_close($mysqli);  
} //FIN function enviar_newsletter()
```

Autenticación de usuarios y cierre de sesión.

Un usuario anónimo puede autenticarse empleando un sencillo formulario de dos campos (Username y Password) desde cualquier página del portal. En todas ellas, estará disponible el área de Usuarios.

La primera comprobación que el sistema efectúa al recibir el formulario, es verificar que exista una variable de tipo oculta (hidden). En caso de existir, contrasta si existe en la Base de Datos, alguna coincidencia con los datos enviados por el usuario. Una vez hallada coincidencia, se inicializan diversas variables de sesión con los datos de consulta más frecuentes sobre el usuario (a destacar aquella que indica que existe algún usuario autenticado o bien, aquella que informa sobre los privilegios del mismo).

Una vez ha sido validado el usuario, si éste es del tipo funcionario, se le reenvía a la página de administración de contenidos. Para el resto de usuario, serán remitidos a la página desde la cual efectuaron la autenticación, mostrando además las opciones a que tenga derecho según su categoría (por ejemplo, para un usuario de tipo Residencial, se habilitará la opción del menú principal 'Oficina Virtual').

En caso de que no se hallara coincidencia en la Base de Datos, se remite al usuario a la página principal, informándole además de los motivos.

Por otro lado, el sistema también se encarga de desconectar o cerrar la sesión de un usuario previamente autenticado, si así lo hubiera solicitado. En este caso, el sistema recibirá una variable de nombre 'logout' mediante el método GET, desasignando a



continuación todas las variables de sesión existentes, así como también eliminando el fichero que almacena el identificados de sesión. Por último, el usuario es remitido a la página principal.

```
/*comprobamos que el usuario ha intentado autenticarse de forma no malintencionada
(en cuyo caso, se recibirá la variable login de tipo hidden)*/
if (isset($_POST["login"])) {
    eliminar_tags();
    $user = $_POST['username'];
    $psw = $_POST['password'];

    //se comprueba que exista coincidencia en la BD
    $buscar_user_sql = "SELECT username, categoria, nombre, apellido1, email FROM
    usuarios WHERE username='".$user.'" and password=PASSWORD('".$psw."');
    doDB();
    $buscar_user_res = mysqli_query($mysqli, $buscar_user_sql) or
    die(mysqli_error($mysqli));

    if ( mysqli_num_rows($buscar_user_res) > 0 ) {
        /*inicializamos una variable de sesión, que indicará al sistema que
        existe un usuario autenticado*/
        $_SESSION["autenticado"] = "1";
        //se almacenan los datos de frecuente consulta en variables de sesión
        while ($user_info = mysqli_fetch_array($buscar_user_res)) {
            $_SESSION["username"] = $user_info['username'];
            $_SESSION["categoria"] = $user_info['categoria'];
            $_SESSION["nombre_comp"] = $user_info['nombre']." ";
            $user_info['apellido1'];
            $_SESSION["email"] = $user_info['email'];
            mysqli_free_result($buscar_user_res);
            mysqli_close($mysqli);
        }
        //se remite al usuario a la página convenida, atendiendo a sus privilegios
        if ( $_SESSION["categoria"] == "ADM" || $_SESSION["categoria"] == "INT" ) {
            header("Location: index_admin.php");
        } else
            switch($_POST["login"]) {
```



```
        case "noticias":
            header("Location: noticias_public.php?pag=
NOTICIAS"); break;
        case "agenda":
            header("Location: agenda_public.php?pag=
EVENTOS"); break;
        case "tramites":
            if ($_SESSION["categoria"] == "RES")
                header("Location: tramites_user.php?pag=
OFIC. VIRTUAL");
            else
                header("Location: tramites_public.php?pag=
TRAMITES");
            break;
        case "foros":
            header("Location: foros_user.php?pag=
FOROS"); break;
        default:
            header("Location: saludo.php?pag=INICIO"); break;
    };
} else { //el usuario no ha sido encontrado en la base de datos
    $_SESSION["autenticado"] = "0";
    header("Location: index.php?pag=INICIO");
}

session_write_close();
/*termina la sesión, forzando el almacenamiento de los datos de sesión. Útil para
accesos concurrentes.*}

if (isset($_GET["logout"])) { //un usuario ha solicitado desconectarse
    session_unset(); // desasigna las variables de sesión
    session_destroy(); //termina sesión y borra su archivo
    header("Location: index.php?logout&pag=INICIO");
}
```



Gestión de tramitaciones (Oficina Virtual).

Los usuarios de tipo Residencial pueden iniciar y gestionar tramitaciones de aquellos trámites que estén disponibles para gestión online. El modus operandi es sencillo: el usuario inicia una tramitación, adjuntando si lo desea una notificación u observación, tras lo cual, un usuario funcionario recogerá esa tramitación, aceptándola o rechazándola (dándose por concluida) o bien requiriendo alguna información del usuario residencial (en cuyo caso, adjuntará una notificación). Por tanto, una tramitación dispondrá de un historial de notificaciones, en algunos casos, como medio de “conversación” entre usuario residencial y funcionario.

Se muestra a continuación un extracto de código, el cual muestra la primera fase del proceso, donde se exhibe al usuario residencial un formulario informando de la tramitación que está a punto de iniciar, solicitando su confirmación. En caso de que el usuario confirme, el sistema insertará la nueva tramitación en la Base de Datos, adjuntando asimismo la notificación, en caso la hubiere.

```
if ( $_SESSION["categoria"] == "RES" ) {  
  
if ( !isset($_POST["confirmar"]) && !isset($_POST["cancelar"]) ) {  
    $id = $_GET["id"];  
    $nombre = $_GET["nombre"];  
    $importe = $_GET["importe"];  
    $salida_php = "  
    <h2>Trámite ID: $id <br>$nombre</h2><p><strong>Importe: $importe €</strong>  
    <p><form action=\"\".$_SERVER["PHP_SELF"].\"\" method=\"POST\">  
    <strong>A continuación indique las observaciones que desee adjuntar como anexo  
    a la tramitación:</strong><br><br>  
    <textarea cols=\"53\" rows=\"8\" name=\"notificaciones\"></textarea>  
    <input type=\"hidden\" name=\"id\" value=$id><br><br>  
    <input type=\"submit\" name=\"confirmar\" value=\"Confirmar\">  
    <input type=\"submit\" name=\"cancelar\" value=\"Cancelar\"></form>";  
} else {  
    if (isset($_POST["confirmar"])) {  
        eliminar_tags();  
    }  
}
```



```
$notif = "<span style=\"color:#024b8b; font-weight:bold;\">>".
$_POST["notificaciones"]."</span>";
$id = $_POST["id"];
$fecha = getdate();
$fecha = $fecha[mday]."-".$fecha[mon]."-".$fecha[year];
$notificaciones = "<ul><li>Usuario \"".$_SESSION["username"]."\" con
Fecha ".$fecha.", informa que:<ul><li>\"".$_notif."\"</li></ul></li></ul>";
$insertar_tramitacion_sql = "INSERT INTO tramites_usuarios
(id_tram, username, fecha, estado, hist_notificaciones) VALUES
('$id', '".$_SESSION["username"]."'.", now(), 'Iniciado', '$notificaciones)";
doDB();
$insertar_tramitacion_res = mysqli_query($mysqli, $insertar_tramitacion_sql)
or die(mysqli_error($mysqli));
$salida_php .= "
<h2>Tramitación enviada correctamente. En breve, será revisada y
actualizado su estado</h2>
<p><a href=# onClick=\"window.close()\">Cerrar</a></p>";
}
if (isset($_POST["cancelar"]))
    $salida_php .= "
<h2>Tramitación cancelada</h2>
<p><a href=# onClick=\"window.close()\">Cerrar</a></p>";
}

mysqli_free_result($insertar_tramitacion_sql);
mysqli_close($mysqli);

} else
    echo "<h2>Debes identificarte para poder acceder a Oficina Virtual</h2>";
```

Se describe a continuación el código relevante que forma parte de la segunda fase de una tramitación. El sistema recurre a esta implementación, tanto en el caso de un usuario funcionario que haya solicitado modificar el estado de la tramitación iniciada en la primera fase o bien, en el caso de un usuario residencial que haya solicitado incorporar alguna notificación a la misma.



```
if ( $_SESSION["categoria"] == "ADM" || $_SESSION["categoria"] == "INT" ||
$_SESSION["categoria"] == "RES" ) {
    if (isset($_GET["id"]))
        $id = $_GET["id"];
    if (isset($_POST["id"]))
        $id = $_POST["id"];
    //se obtiene el historial de notificaciones asociado a la tramitación
    $obtener_notificaciones_sql = "SELECT hist_notificaciones FROM tramites_usuarios
WHERE id = '$id'";
    doDB();
    $obtener_notificaciones_res = mysqli_query($mysqli, $obtener_notificaciones_sql)
or die(mysqli_error($mysqli));

    if (mysqli_num_rows($obtener_notificaciones_res) > 0) {
        $notif_info = mysqli_fetch_array($obtener_notificaciones_res);
        $hist_notificaciones = $notif_info["hist_notificaciones"];
    } else
        $hist_notificaciones = "No existen notificaciones";

    //si el usuario aún no ha enviado el formulario, se le muestra
    if ( !isset($_POST["confirmar"]) && !isset($_POST["cancelar"]) ) {
        $id = $_GET["id"];
        $username = $_GET["us"];
        $estado = $_GET["es"];

        $salida_php = "
<h2>Usuario: $username<br>Estado actual: $estado</h2>
<form action=\"".$_SERVER["PHP_SELF"]."\" method=\"POST\">
<h3>Histórico notificaciones:</h3>
<div id=\"notific_modif_tramitacion\">
$hist_notificaciones
</div><br><br>";
        if ($_SESSION["categoria"] != "RES") //para usuarios funcionarios
            $salida_php .= "
<h3>Indique un nuevo estado para la<br>tramitación:</h3>
<input type=\"radio\" name=\"nuevo_estado\" value=\"Revision\"
checked> Revisión
<input type=\"radio\" name=\"nuevo_estado\"
value=\"Aprobado\"> Aprobado
```



```
        <input type="radio" name="nuevo_estado"
            value="Denegado"> Denegado";
$salida_php .= "
<br><br><strong>Información a notificar :</strong><br>
<textarea cols=53 rows=8 name="nueva_notificacion"></textarea>
<input type="hidden" name="id" value=$id><br><br>
<input type="submit" name="confirmar" value="Confirmar">
<input type="submit" name="cancelar" value="Cancelar">";
} else { //el usuario ha enviado el formulario
    if (isset($_POST["confirmar"])) {
        eliminar_tags();
        /*las notificaciones de un usuario funcionario tendrán un texto de
        diferente color que las de un usuario residencial*/
        $nueva_notif = "<span style="color:#";
        if ($_SESSION["categoria"] != "RES")
            $nueva_notif .= "024b8b";
        else
            $nueva_notif .= "46754f";
        $nueva_notif .=
        "; font-weight:bold;\>".$_POST["nueva_notificacion"]."</span>";
        $id = $_POST["id"];
        $fecha = getdate();
        $fecha = $fecha[mday]."-".$fecha[mon]."-".$fecha[year];
        if ($_SESSION["categoria"] != "RES")
            $hist_notificaciones .=
            "<ul><li>Funcionario \"".$_SESSION["username"]."\" con
            Fecha ".$fecha.", notifica que:<ul><li>\"".$nueva_notif.
            "\"</li></ul></li></ul>";
        else
            $hist_notificaciones .=
            "<ul><li>Usuario \"".$_SESSION["username"]."\" con Fecha ".
            $fecha.", notifica que:<ul><li>\"".$nueva_notif.
            "\"</li></ul></li></ul>";
        $actualizar_tramitacion_sql = "UPDATE tramites_usuarios SET
        hist_notificaciones = '$hist_notificaciones";
        if ($_SESSION["categoria"] != "RES")
            $actualizar_tramitacion_sql .=
            ", estado = '".$_POST["nuevo_estado"]."\"";
        $actualizar_tramitacion_sql .= " WHERE id = '$id";
```



```
doDB();
$actualizar_tramitacion_res = mysqli_query($mysqli,
$actualizar_tramitacion_sql) or die(mysqli_error($mysqli));
$salida_php .= "
<h2>Tramitación modificada</h2>
<p><a href=# onClick=\"window.close()\">Cerrar</a></p>";
}
if (isset($_POST["cancelar"]))
$salida_php .= "
<h2>Modificación cancelada</h2>
<p><a href=# onClick=\"window.close()\">Cerrar</a></p>";
}
} else
$salida_php = "<h2>No dispones de permisos para acceder a este recurso.</h2>";

mysqli_free_result($actualizar_tramitacion_res);
mysqli_free_result($obtener_notificaciones_res);
mysqli_close($mysqli);
```

Como se observa en la implementación anterior, las notificaciones van siendo incrementadas y posteriormente actualizadas, haciéndose valor para ello de un único campo de una tupla (tabla 'tramites_usuario') en la Base de Datos. Se recurre a una tipología de letra de diferente color, entre otros, con el fin de aumentar la legibilidad del histórico de conversaciones (notificaciones) entre usuario residencial y funcionario.

Cabe destacar que el código con las acciones a adoptar por el sistema, tanto para el caso de un usuario funcionario como residencial, se encuentra entremezclado, decidiéndose sobre la marcha qué "camino" seguir, en base a sencillas instrucciones condicionales. De este modo, el formulario mostrado tanto al usuario residencial como al funcionario es análogo, con la diferencia de que el primero únicamente tendrá una opción disponible, la de adjuntar una notificación, mientras que el usuario funcionario podrá además alterar el estado de la tramitación. Cualquier modificación efectuada tanto por uno como por otro, podrá ser inmediatamente visualizada tras confirmar la misma.



Contraseña olvidada. Recuperación.

Aquellos usuarios que no recuerden la contraseña para autenticación, podrán en cualquier instante solicitar una nueva. Bastará para ello ingresar el Username y Email. El modus operandi implementado es el siguiente :

El sistema verifica que existe una coincidencia en la Base de Datos para los datos ingresados por el usuario en el formulario. Si no existe la incidencia, se informa al usuario de los motivos (el Email introducido no se corresponde con la contraseña, o bien el Username no existe, etc) y se le invita a volver a intentarlo.

Si después de la verificación anterior, se encuentra una coincidencia, se genera una nueva contraseña a partir de una palabra seccionada al azar de un diccionario (almacenado en un fichero) con miles de palabras en inglés. Además, para aumentar la seguridad de la contraseña, se le adicionará al final de la misma, un número aleatorio de 3 dígitos. Finalmente, la contraseña es enviada al usuario por correo y actualizada en la Base de Datos, previo a su encriptado.

```
if (isset($_POST["form_resetpsw_enviado"])) { //si el usuario ha enviado el formulario
    try {
        if (empty($_POST["username"]))
            throw new Exception('Debe introducir un username.');
```

```
        if (empty($_POST["email"]))
            throw new Exception('Debe introducir un email.');
```

```
    }
    catch (Exception $e) {
        echo "<p><span style='font-size:150%; color:#a5271a;'>ERROR: ".
            $e->getMessage()."</span><p>";
    }
    if (empty($e)) {
        /*si no hay errores, se verifica que el usuario exista, aprovechando
        además para obtener su email, que será reutilizado más tarde*/
        $obtener_email_sql = "SELECT email FROM usuarios WHERE
        username = '".$_POST["username"]."'";
        doDB();
        $obtener_email_res = mysqli_query($mysqli, $obtener_email_sql) or
```



```
die(mysqli_error($mysqli));
$email_info = mysqli_fetch_array($obtener_email_res);
$email_user = $email_info['email'];

try {
    if (mysqli_num_rows($obtener_email_res) == 0) {
        throw new Exception('El usuario introducido no existe.
        Por favor, ingrese de nuevo un username válido.');
```

```
    }
```

```
    if ( (mysqli_num_rows($obtener_email_res) > 0) &&
```

```
        ($_POST["email"] != $email_user) ) {
```

```
        throw new Exception('El email no se corresponde con el
        usuario introducido.');
```

```
    }
```

```
    } catch(Exception $e) {
```

```
        echo "<p><span style=\"font-size:150%; color:#a5271a;\">
```

```
        ERROR: ".$e->getMessage()."</span><p>";
```

```
    }
```

```
if (empty($e)) {
```

```
    $palabra = ""; //se inicializa la contraseña con un cadena vacía
```

```
    //indicamos la ruta al archivo con el diccionario de palabras
```

```
    $diccionario = "../usr/dict/words/english.0";
```

```
    $fp = fopen($diccionario, 'r');
```

```
    if (!$fp) exit;
```

```
    $tamaño = filesize($diccionario);
```

```
    /*nos situamos en un punto aleatorio del diccionario, el cual
    podría caer en una posición intermedia dentro de una palabra*/
```

```
    srand ((double) microtime()*1000000);
```

```
    $posicion_aleat = rand(0, $tamaño);
```

```
    fseek($fp, $posicion_aleat);
```

```
    /*obtenemos la siguiente palabra cuyo tamaño sea el correcto
    (en este caso, nro. de caracteres permitido para la contraseña). El
    máximo indicado es 17 porque luego añadiremos hasta 3 dígitos
    más para añadir más seguridad a la contraseña*/
```

```
    while ( strlen($palabra) < 6 || strlen($palabra) > 17 ||
```

```
    strstr($palabra, "'") ) { // strstr descarta palabras con apóstrofes
```



```
        // si se alcanza el final del diccionario, se vuelve al principio
        if (feof($fp)) fseek($fp, 0);

        //se lee una línea, pero se descarta esta primera palabra,
        ya que puede no estar completa
        $palabra = fgets($fp, 80);
        $palabra = fgets($fp, 80); //esta segunda es correcta
    }
    $palabra = trim($palabra); // para recortar el \n final de fgets

    //se aumenta la seguridad, añadiéndosele un nro. entre 0 y 999
    srand ((double) microtime()*1000000);
    $nro_aleat = rand(0, 999);
    $new_password = $palabra.$nro_aleat;

    //se actualiza nueva contraseña en la BD
    $cambiar_psw_sql = "
    UPDATE usuarios SET password = PASSWORD('$new_password')
    WHERE username = '$_POST[\"username\"]';
    $cambiar_psw_res = mysqli_query($mysqli, $cambiar_psw_sql) or
    die(mysqli_error($mysqli));

    //se informa por email al usuario, de la nueva contraseña
    $mens = "Tu contraseña para el portal Web del Ayto. de Paterna
    ha sido cambiada por la siguiente: $new_password \r\n";
    $from = "From: support@ayto-paterna \r\n";
    mail($email_user, 'Contraseña modificada - Portal Ayto. De
    Paterna', $mens, $from);

    mysqli_free_result($cambiar_psw_res);
    mysqli_free_result($obtener_email_res);
    mysqli_close($mysqli);

    $salida_php = "
    <h2>Contraseña cambiada exitosamente.<br><br>Revise su email
    para conocerla.</h2>
    <p><a href=# onClick=\"window.close()\">Cerrar</a></p>";
    }
}
```



```
} else {
    $salida_php = "
    <h2>Introduzca su Username y su E-mail :</h2>
    <form method=\"post\">
    <strong>Username:</strong>
    <input type=\"text\" name=\"username\"><br><br>
    <strong>E-Mail:</strong>
    <input type=\"text\" name=\"email\"><br><br>
    <input type=\"submit\" name=\"form_resetpsw_enviado\" value=\"Cambiar
    password\">
    </form>";
}
```

Buscador multifuncional.

La caja de búsquedas permite introducir uno o dos términos de búsqueda, mostrándose los resultados ordenados por relevancia. Además, si el usuario se encuentra en una página que no sea la principal, podrá seleccionar además que la búsqueda sea efectuada únicamente en la página en la que se encuentre o bien en todo el portal, en cuyo caso, se mostrará un resumen de resultados por cada sección del portal.

Con el fin de mejorar la eficiencia del buscador, algunos campos de texto de las tablas de la Base de Datos serán indexados mediante índices 'fulltext'. Esto además permite ordenar los resultados por orden de relevancia atendiendo a la cantidad de ocurrencias del patrón de búsqueda.

En primer lugar, cabe destacar la implementación ideada para que el sistema pueda reconocer en todo momento, la página desde la cual se inicia una búsqueda limitada a los contenidos de la misma (búsqueda parcial), así como para no ofrecer la opción de búsqueda parcial si el usuario se encuentra en la página principal. Para conseguirlo, se consulta la variable de sesión `$_SERVER["PHP_SELF"]` y se pasa la misma como parámetro de la función de búsqueda. Por último, la función de búsqueda discriminará (utilizando una instrucción switch) la búsqueda sobre una/s tabla/s determinada/s en función del valor de esta variable.



```
function mostrar_caja_búsqueda() {
    $salida_php = "
    <div class=\"cajabusc\" id=\"buscador\">
        <div class=\"cajabusc_arriba\"></div>
        <div class=\"cajabusc_centro\">;
    $path_pag_actual = $_SERVER["PHP_SELF"];
    $pag_actual = explode('/', $path_pag_actual);
    $pos_pag_actual = count($pag_actual)-1;
    $pag_actual = $pag_actual[$pos_pag_actual];
    $salida_php .= "
        <p><h5>Introduzca 1 ó 2 palabras :</h5>
        <form action=\"result_búsqueda.php\" method=\"post\">

        <input type=\"text\" name=\"búsqueda_enviada\" size=\"18\">
        <input value=\"Buscar\" type=\"submit\"
        name=\"form_búsqueda_enviado\"><br>;
    if ( $pag_actual != "index_user.php" && $pag_actual != "index.php" &&
    $pag_actual != "index_lateral.php" )
        $salida_php .= "
            <input type=\"radio\" name=\"tabla\" value=\"$pag_actual\"
            checked>Sólo en esta sección<br>
            <input type=\"radio\" name=\"tabla\" value=\"todas\">Todo el
            portal<br>;
    if ( $pag_actual == "index_user.php" || $pag_actual == "index.php" )
        $salida_php .= "
            <input type=\"hidden\" name=\"tabla\" value=\"todas\">;
    if ( $pag_actual == "index_lateral.php" )
        $salida_php .= "
            <input type=\"hidden\" name=\"tabla\" value=\"estática\">;

    $salida_php .= "
        </form>
        </p>
    </div>
        <div class=\"cajabusc_abajo\"></div>
    </div> <!-- end of cajabusc/buscador -->;

    echo $salida_php;
} //end of function mostrar_caja_búsqueda()
```



```
function busqueda($cadena, $tabla) {
    /*búsquedas de una sólo palabra, son más rápidas con LIKE que con FULLTEXT,
    por eso verificamos cuántas palabras contiene la búsqueda*/
    $nro_palabras = count(explode(" ", $cadena));
    $mysqli = doDB();

    switch($tabla) {

    case 'noticias_public.php':

    if ($nro_palabras == 1) { //consulta a la BD usando LIKE
        $obtener_noticias_sql = "
            SELECT id, titular, fecha FROM noticias WHERE titular LIKE '%$cadena%'
            OR texto LIKE '%$cadena%' ORDER BY fecha DESC LIMIT 0, 5";
    } elseif ($nro_palabras > 1) //consulta a la BD usando FULLTEXT
        $obtener_noticias_sql = "
            SELECT id, titular, fecha , MATCH (titular, texto) AGAINST ('$cadena') AS
            puntuacion FROM noticias WHERE MATCH (titular, texto) AGAINST
            ('$cadena' IN BOOLEAN MODE) ORDER BY puntuacion DESC LIMIT 5";
        $obtener_noticias_res = mysqli_query($mysqli, $obtener_noticias_sql) or
        die(mysqli_error($mysqli));

        if (mysqli_num_rows($obtener_noticias_res) > 0) {
            $salida_noticias = "
                <h4 align=\"center\">NOTICIAS</h4>
                <table cellpadding=\"3\" cellspacing=\"1\" border=\"1\">
                <tr style=\"background-color: #CCCCCC\">
                <th>Titular noticia</th>
                <th>Fecha</th>
                </tr>";

            while ( $noticia_info = mysqli_fetch_array($obtener_noticias_res) ) {
                $noticia_id = $noticia_info['id'];
                $noticia_titular = $noticia_info['titular'];
                $fecha_temp = explode(" ", $noticia_info['fecha']);
                $fecha_temp = explode("-", $fecha_temp[0]);
                $noticia_fecha =
                $fecha_temp[2].".".$fecha_temp[1].".".$fecha_temp[0];
                $salida_noticias .= "
```



```
        <tr><td><strong><a href=\"javascript:eventWindow
        ('detalles_noticia.php?id=$noticia_id');\">\".$noticia_titular.
        \"</a></strong></td>
        <td align=\"center\" width=\"20%\">\".$noticia_fecha.
        \"</td></tr>\";
    }
    $salida_noticias .= \"</table><br><br><br>\";
    echo $salida_noticias;
    mysqli_free_result($obtener_noticias_res);
    mysqli_close($mysqli);
} else
    echo \"<h2>NOTICIAS: No existen coincidencias</h2><hr>\";

break;

case 'agenda_public.php':

if ($nro_palabras == 1) { //consulta a la BD usando LIKE
    $obtener_eventos_sql = \"
    SELECT id, titulo_evento, inicio_evento FROM calendario_eventos WHERE
    titulo_evento LIKE '%$cadena%' OR desc_evento LIKE '%$cadena%'
    ORDER BY inicio_evento DESC LIMIT 0, 5\";
} elseif ($nro_palabras > 1) //consulta a la BD usando FULLTEXT
    $obtener_eventos_sql = \"
    SELECT id, titulo_evento, inicio_evento , MATCH
    (titulo_evento, desc_evento) AGAINST ('$cadena') AS puntuacion FROM
    calendario_eventos WHERE MATCH (titulo_evento, desc_evento) AGAINST
    ('$cadena' IN BOOLEAN MODE) ORDER BY puntuacion DESC LIMIT 5\";
    $obtener_eventos_res = mysqli_query($mysqli, $obtener_eventos_sql) or
    die(mysqli_error($mysqli));

    if (mysqli_num_rows($obtener_eventos_res) > 0) {
        $salida_eventos = \"
        <h4 align=\"center\">EVENTOS</h4>
        <table cellpadding=\"3\" cellspacing=\"1\" border=\"1\">
        <tr style=\"background-color: #CCCCCC\">
        <th>Título Evento</th>
        <th>Fecha de inicio</th>
        </tr>\";
    }
}
```



```
while ($evento_info = mysqli_fetch_array($obtener_eventos_res)) {
    $evento_id = $evento_info['id'];
    $evento_titulo = $evento_info['titulo_evento'];
    $fecha_temp = explode(" ", $evento_info['inicio_evento']);
    $fecha_temp = explode("-", $fecha_temp[0]);
    $inicio_evento =
    $fecha_temp[2].".".$fecha_temp[1].".".$fecha_temp[0];
    $salida_eventos .= "<tr><td align=\"center\"><strong>
    <a href=\"javascript:eventWindow('detalles_evento.php?
    id=$evento_id');\">\".$evento_titulo.</a></strong></td>
    <td align=\"center\" width=\"20%\">\".$inicio_evento.\"
    </td></tr>";
}
$salida_eventos .= "</table><br><br><br>";
echo $salida_eventos;
mysqli_free_result($obtener_eventos_res);
mysqli_close($mysqli);
} else
    echo "<h2>AGENDA: No existen coincidencias</h2><hr>";

break;

case 'foros_user.php':

if ($nro_palabras == 1) { //consulta a la BD usando LIKE
    $obtener_posts_sql = "SELECT * FROM foro_posts INNER JOIN foro_topics
    ON foro_topics.topic_id = foro_posts.topic_id WHERE foro_posts.post_texto
    LIKE '%$cadena%' OR foro_topics.topic_titulo LIKE '%$cadena%' LIMIT 50";
} elseif ($nro_palabras > 1 ) //consulta a la BD usando FULLTEXT
    $obtener_posts_sql = "SELECT * , MATCH (foro_posts.post_texto) AGAINST
    ('$cadena') AS puntuacion FROM foro_posts INNER JOIN foro_topics ON
    foro_topics.topic_id = foro_posts.topic_id WHERE MATCH
    (foro_posts.post_texto) AGAINST ('$cadena' IN BOOLEAN MODE) OR
    MATCH (foro_topics.topic_titulo) AGAINST ('$cadena' IN BOOLEAN MODE)
    ORDER BY puntuacion DESC LIMIT 50";

$obtener_posts_res = mysqli_query($mysqli, $obtener_posts_sql) or
die(mysqli_error($mysqli));
```



```
if (mysqli_num_rows($obtener_posts_res) > 0) {
    $salida_foros = "
    <h4 align=\"center\">FOROS</h4>
    <table cellpadding=\"3\" cellspacing=\"1\" border=\"1\">
    <tr style=\"background-color: #CCCCCC\">
    <th>Texto Mensaje</th>
    <th>TEMA</th>
    </tr>";

    while ($post_info = mysqli_fetch_array($obtener_posts_res)) {
        $topic_id = $post_info['topic_id'];
        $post_id = $post_info['post_id'];
        $post_texto = $post_info['post_texto'];
        $post_prop = $post_info['post_prop'];
        $post_fecha = $post_info['post_fecha'];

        $obtener_tema_sql = "
        SELECT topic_titulo FROM foro_topics WHERE topic_id = '$topic_id';
        $obtener_tema_res = mysqli_query($mysqli, $obtener_tema_sql) or
        die(mysqli_error($mysqli));
        $topic_info = mysqli_fetch_array($obtener_tema_res);
        $titulo = $topic_info['topic_titulo'];

        $salida_foros .= "
        <tr><td><strong>".$post_texto."</strong><br/>
        <span style=\"font-style:italic\"><br>Creado con fecha: ".
        $post_fecha."<br>por <strong>".$post_prop."</strong></td></span>
        <td align=center width=\"30%\"><a href=\"foros_user.php?pag=
        FOROS&mostrar_topic=".$topic_id.""><strong>".$titulo."</strong>
        </a></td></tr>";
    }
    $salida_foros .= "</table><br><br><br>";
    echo $salida_foros;

    mysqli_free_result($obtener_posts_res);
    mysqli_free_result($obtener_tema_res);
    mysqli_close($mysqli);
} else
    echo "<h2>FOROS: No existen coincidencias</h2><hr>";
```



```
break;

case 'oposiciones.php':

if ($nro_palabras == 1) { //consulta a la BD usando LIKE
    $obtener_oposiciones_sql = "
        SELECT id, nombre, plazas FROM oposiciones WHERE nombre
        LIKE '%$cadena%' ORDER BY fecha_publicacion DESC LIMIT 0, 5";
} elseif ($nro_palabras > 1 ) //consulta a la BD usando FULLTEXT
    $obtener_oposiciones_sql = "
        SELECT id, nombre, plazas , MATCH (nombre) AGAINST ('$cadena') AS
        puntuacion FROM oposiciones WHERE MATCH (nombre) AGAINST
        ('$cadena' IN BOOLEAN MODE) ORDER BY puntuacion DESC LIMIT 5";
$obtener_oposiciones_res = mysqli_query($mysqli, $obtener_oposiciones_sql) or
die(mysqli_error($mysqli));

if (mysqli_num_rows($obtener_oposiciones_res) > 0) {
    $salida_oposiciones = "
        <h4 align=\"center\">OPOSICIONES</h4>
        <table cellpadding=\"3\" cellspacing=\"1\" border=\"1\">
        <tr style=\"background-color: #CCCCCC\">
        <th>Nombre Oposición</th>
        <th>Plazas</th>
        </tr>";
    while ($oposicion_info = mysqli_fetch_array($obtener_oposiciones_res)) {
        $oposicion_id = $oposicion_info['id'];
        $oposicion_nombre = $oposicion_info['nombre'];
        $oposicion_plazas = $oposicion_info['plazas'];
        $salida_oposiciones .= "<tr><td align=\"center\"><strong>
        <a href=\"javascript:eventWindow('detalles_oposicion.php?
        id=$oposicion_id');\">\".$oposicion_nombre.</a></strong></td>
        <td align=\"center\" width=\"20%\">\".$oposicion_plazas.
        \"</td></tr>";
    }
    $salida_oposiciones .= "</table><br><br><br>";
    echo $salida_oposiciones;

    mysqli_free_result($obtener_oposiciones_res);
    mysqli_close($mysqli);
}
```



```
} else
    echo "<h2>OPOSICIONES: No existen coincidencias</h2><hr>";

break;

case 'tramites_user.php': case 'tramites_public.php':

if ($nro_palabras == 1) { //consulta a la BD usando LIKE
    $obtener_tramites_sql = "
        SELECT id, titulo, online FROM tramites WHERE titulo LIKE '%$cadena%' OR
        tramites.desc LIKE '%$cadena%' OR documentacion LIKE '%$cadena%'
        OR observaciones LIKE '%$cadena%' ORDER BY tipo DESC LIMIT 0, 5";
} elseif ($nro_palabras > 1) //consulta a la BD usando FULLTEXT
    $obtener_tramites_sql = "
        SELECT id, titulo, online , MATCH (titulo, documentacion, lugar_present,
        organo_gestor, observaciones, impresos, tramites.desc) AGAINST
        ('$cadena') AS puntuacion FROM tramites WHERE MATCH (titulo,
        documentacion, lugar_present, organo_gestor, observaciones, impresos,
        tramites.desc) AGAINST ('$cadena' IN BOOLEAN MODE) ORDER BY
        puntuacion DESC LIMIT 5";

$obtener_tramites_res = mysqli_query($mysqli, $obtener_tramites_sql) or
die(mysqli_error($mysqli));

if (mysqli_num_rows($obtener_tramites_res) > 0) {
    $salida_tramites = "
        <h4 align=\"center\">TRÁMITES</h4>
        <table cellpadding=\"3\" cellspacing=\"1\" border=\"1\">
        <tr style=\"background-color: #CCCCCC\">
        <th>Nombre Trámite</th>
        <th>Tramitación Online</th></tr>";

    while ($tramite_info = mysqli_fetch_array($obtener_tramites_res)) {
        $tramite_id = $tramite_info['id'];
        $tramite_titulo = $tramite_info['titulo'];
        $tramite_online = $tramite_info['online'];
        $salida_tramites .= "<tr><td align=\"center\"><strong>
        <a href=\"javascript:eventWindow
        ('detalles_tramiteytramitacion.php?id_tramite=$tramite_id');\">
        $tramite_titulo.</a></strong></td>";
    }
}
```



```
        <td align=\"center\" width=\"20%\">;
    if (!empty($tramite_online))
        $salida_tramites .= \"SI</td></tr>\";
    else
        $salida_tramites .= \"NO</td></tr>\";
    }
    $salida_tramites .= \"</table><br><br><br>\";
    echo $salida_tramites;

    mysqli_free_result($obtener_tramites_res);
    mysqli_close($mysqli);
} else
    echo \"<h2>TRÁMITES: No existen coincidencias</h2><br>\";

break;

case 'estatica':

if ($nro_palabras == 1) { //consulta a la BD usando LIKE
    $obtener_estatica_sql = \"
        SELECT apartado, seccion, contenido FROM seccion_estatica WHERE
        contenido LIKE '%$cadena%' ORDER BY apartado DESC LIMIT 0, 5\";
} elseif ($nro_palabras > 1) //consulta a la BD usando FULLTEXT
    $obtener_estatica_sql = \"
        SELECT apartado, seccion, contenido , MATCH (contenido) AGAINST
        ('$cadena') AS puntuacion FROM seccion_estatica WHERE MATCH
        (contenido) AGAINST ('$cadena' IN BOOLEAN MODE) ORDER BY
        puntuacion DESC LIMIT 5\";
$obtener_estatica_res = mysqli_query($mysqli, $obtener_estatica_sql) or
die(mysqli_error($mysqli));

if (mysqli_num_rows($obtener_estatica_res) > 0) {
    $salida_estatica = \"
        <h4 align=\"center\">INFO AYUNTAMIENTO</h4>
        <table cellpadding=\"3\" cellspacing=\"1\" border=\"1\">
        <tr style=\"background-color: #CCCCCC\">
        <th>Apartado</th>
        <th>Sección</th>
        <th>Contenido</th></tr>\";
```



```
while ($estatica_info = mysqli_fetch_array($obtener_estatica_res)) {
    $estatica_apartado = $estatica_info['apartado'];
    $secciones = array("alca" => "El Alcalde", "orga" => "Órgs. de
gobierno", "conc" => "Concejalías Barrio", "tfno" =>
"Tfnos. de interés", "fies" => "Fiestas", "guia" => "Guia de Paterna",
"tran" => "Transportes", "sani" => "Sanidad", "cons" => "Consumo",
"medi" => "Medio ambiente", "serv" => "Servicios sociales", "urba"
=> "Urbanismo");
    $estatica_seccion = $estatica_info['seccion'];
    $nombre_seccion = $secciones[$estatica_seccion];
    $estatica_contenido =
substr($estatica_info['contenido'], 0, 300)." ..... ";

    $salida_estatica .= "
<tr><td width=\"15%\"><strong>".$estatica_apartado."</strong>
</td><td width=\"25%\"><strong>".$nombre_seccion."</strong>
</td><td width=\"60%\">
<div id=\"busq_infoayto\">$estatica_contenido</div><br>
<a href=\"index_lateral.php?sec1=$estatica_apartado&sec2=
$estatica_seccion\">VER MÁS</a></td></tr>";
}
$salida_estatica .= "</table><br><br><br>";

echo $salida_estatica;

mysqli_free_result($obtener_estatica_res);
mysqli_close($mysqli);
} else
echo "<h2>INFO AYUTANMIENTO: No existen coincidencias</h2><hr>";

break;

};

} //end of function busqueda()
```



Finalmente, será preciso el siguiente código, en la página que muestra los resultados de la búsqueda, para pilotar la búsqueda en todas las tablas (búsqueda global) o en una sola (búsqueda parcial) como sigue :

```
if ( empty($_POST["busqueda_enviada"]) )
    echo "El campo de búsqueda está vacío";
else {
    if ($_POST["tabla"] == "todas") {
        busqueda($_POST["busqueda_enviada"], "noticias_public.php");
        busqueda($_POST["busqueda_enviada"], "agenda_public.php");
        busqueda($_POST["busqueda_enviada"], "foros_user.php");
        busqueda($_POST["busqueda_enviada"], "oposiciones.php");
        busqueda($_POST["busqueda_enviada"], "tramites_user.php");
        busqueda($_POST["busqueda_enviada"], "tramites_public.php");
        busqueda($_POST["busqueda_enviada"], "estatica");
    } else
        busqueda($_POST["busqueda_enviada"], $_POST["tabla"]);
}
```

Foros de discusión.

Un usuario anónimo puede consultar cualquier mensaje del foro, aunque para poder participar abriendo nuevos temas y respondiendo a otros mensajes, precisa de autenticarse.

Lo primero con lo que se encuentra un usuario al acceder a la página de foros, es una relación con todos los temas disponibles. Bastará con hacer click en cualquiera de ellos para acceder a los mensajes del mismo.

A continuación se expone la implementación de la función que muestra la relación de temas de discusión disponibles. A destacar la sección donde el sistema exhibe la opción de eliminar un tema, sólo si no contiene mensajes de cualquier otro usuario (esto es, si no existe participación aparte de la de su creador). En otro orden de cosas, el sistema debe contabilizar la cantidad de mensajes contenidos en cada tema.



```
function listar_topics() {

    $obtener_topics_sql = "
    SELECT topic_id, topic_titulo, DATE_FORMAT(topic_fecha, '%b %e %Y at %r') AS
    ft_topic_fecha, topic_prop, DATE_FORMAT(topic_ultimo, '%b %e %Y at %r') AS
    ft_topic_ultimo FROM foro_topics ORDER BY topic_fecha DESC";
    $mysqli = doDB();
    $obtener_topics_res = mysqli_query($mysqli, $obtener_topics_sql) or
    die(mysqli_error($mysqli));

    if (mysqli_num_rows($obtener_topics_res) < 1)
        $salida_php = "<p><em>No existen temas de debate</em></p>";
    else
        $salida_php .= "<strong> <span style='font-size:150%'\>
        Seleccione un tema de debate:</span></strong><br><br>";

    $salida_php .= "
    <table cellpadding='3' cellspacing='1' border='1'\>
    <tr style='background-color: #CCCCCC'\>
    <th>TITULOS DE LOS TEMAS</th>
    <th>MENSAJES</th>
    <th>ÚLTIMO MENSAJE</th>
    </tr>";

    while ($topic_info = mysqli_fetch_array($obtener_topics_res)) {
        $topic_id = $topic_info['topic_id'];
        $topic_titulo = $topic_info['topic_titulo'];
        $topic_fecha = $topic_info['ft_topic_fecha'];
        $topic_prop = $topic_info['topic_prop'];
        $topic_ultimo = $topic_info['ft_topic_ultimo'];

        //se contabiliza la cantidad de mensajes para cada tema del foro
        $obtener_posts_sql = "
        SELECT post_prop FROM foro_posts WHERE topic_id = ".$topic_id."";
        $obtener_posts_res = mysqli_query($mysqli, $obtener_posts_sql) or
        die(mysqli_error($mysqli));

        $cant_posts = mysqli_num_rows($obtener_posts_res);
```



```
/*si el usuario autenticado es quien abrió el topic y no existe participación
alguna de cualquier otro usuario, se exhibe la opción de eliminarlo*/
$participantes = "solo_creador";
while ( ($post_info = mysqli_fetch_array($obtener_posts_res)) &&
($participantes == "solo_creador") ) {
    if ($post_info["post_prop"] != $_SESSION["username"])
        $participantes = "otros";
}
$salida_php .= "
<tr><td width=55%>
<a href=\"foros_user.php?pag=FOROS&mostrar_topic=\".$topic_id.\">
<strong>\".$topic_titulo.\"</strong></a>
<br>Creado con fecha: \".$topic_fecha.\"<br>Por: \".$topic_prop.\"<br>";

if ( $topic_prop == $_SESSION["username"] && $participantes ==
"solo_creador" )
    $salida_php .= "
    <strong><a style=\"font-style:normal; color:#a0281f;\" href=\"foros_
user.php?pag=FOROS&del_topic_id=$topic_id&cant_posts=
$cant_posts\"> > Eliminar Tema < < /a></strong>";

$salida_php .= "
</td>
<td align=center>\".$cant_posts.\"</td>
<td align=center>\".$topic_ultimo.\"</td>
</tr>";
}

$salida_php .= "</table>";
echo $salida_php;

mysqli_free_result($obtener_topics_sql);
mysqli_free_result($obtener_posts_res);
mysqli_close($mysqli);
} //FIN function listar_topics()
```



Un usuario autenticado puede abrir un nuevo tema de discusión en el foro, incluyendo opcionalmente un primer mensaje para el mismo. Para ello, deberá cumplimentar un sencillo formulario de 2 campos, el título del tema y el texto (opcional) del primer mensaje.

El sistema recibe el formulario y verifica que al menos, se haya indicado algún título para el tema, en cuyo caso, inserta una nueva tupla en la tabla 'foro_topics' de la Base de Datos. Asimismo, comprueba si existe un primer mensaje y si lo hubiera, lo inserta en la tabla 'foro_posts' relacionándolo con el tema insertado anteriormente. La implementación de este apartado se expone a continuación:

```
function insertar_topic() {
    if ($_SESSION["autenticado"] == "1") {
        try {
            if (empty($_POST["topic_titulo"]))
                throw new Exception("El título no puede estar en blanco.");
        } catch (Exception $e) {
            echo "<span style='font-size:150%;>ERROR: ";
            echo $e->getMessage()."</span>";
        }
        if (empty($e)) {
            eliminar_tags();
            $insertar_topic_sql = "INSERT INTO foro_topics (topic_titulo, topic_fecha, topic_prop, topic_ultimo) VALUES ('".$_POST["topic_titulo"]."',now(), '".$_SESSION["username"]."',now())";
            $mysqli = doDB();
            $insertar_topic_res = mysqli_query($mysqli, $insertar_topic_sql) or die(mysqli_error($mysqli));

            if (!empty($_POST["post_texto"])) {
                $topic_id = mysqli_insert_id($mysqli); //retorna el id de la
                //última consulta de aquella columna que tenga el atributo
                //AUTO_INCREMENT, en este caso, ese campo es topic_id
                $insertar_post_sql = "INSERT INTO foro_posts (topic_id, post_texto, post_fecha, post_prop) VALUES ('$topic_id', '".$_POST["post_texto"]."',now(), '".$_SESSION["username"]."')";
            }
        }
    }
}
```



```
        $insertar_post_res = mysqli_query($mysqli, $insertar_post_sql)
        or die(mysqli_error($mysqli));
        mysqli_free_result($insertar_post_res);
    }
    echo "<span style=\"font-size:150%; font-style:italic;
    color:#d64100;\">Tema creado con éxito</span><br><br>";

    mysqli_free_result($insertar_topic_res);
    mysqli_close($mysqli);
}
} else
    echo "Debe identificarse para poder continuar.";
} //FIN function insertar_topic()
```

Cuando el usuario selecciona la eliminación de algún tema, el sistema elimina la tupla correspondiente de la tabla 'foro_topics' pero también verifica si el tema contiene algún mensaje, en cuyo caso, también debe eliminarlos.

```
function eliminar_topic($topic_id, $cant_posts) {
    if ($_SESSION["autenticado"] == "1") {
        $eliminar_topic_sql = "DELETE FROM foro_topics WHERE topic_id = '$topic_id'";
        $mysqli = doDB();
        $eliminar_topic_res = mysqli_query($mysqli, $eliminar_topic_sql) or
        die(mysqli_error($mysqli));
        if ($cant_posts > 0) {
            $eliminar_posts_sql = "DELETE FROM foro_posts WHERE topic_id =
            '$topic_id'";
            $eliminar_posts_res = mysqli_query($mysqli, $eliminar_posts_sql) or
            die(mysqli_error($mysqli));
        }
        echo "<span style=\"font-size:150%; font-style:italic; color:#d64100;\">
        TEMA eliminado completamente</span><br><br>";
        mysqli_free_result($eliminar_topic_res);
        mysqli_free_result($eliminar_posts_res);
        mysqli_close($mysqli);
    } else
        $salida_php = "Debes estar identificado para poder participar en los foros";
} //FIN function eliminar_topic()
```



Calendario de eventos.

Tanto la página principal como la página dedicada a la sección de Agenda, muestran una vista reducida de un calendario. En el mismo, aparecen de forma destacada aquellos días que contienen algún evento, siendo esos días en este caso, seleccionables por el usuario. Si el mismo hace click en alguno de ellos, se mostrará una ventana emergente que informará de todos los eventos disponibles para ese día.

A continuación se muestra la implementación que muestra la vista del calendario y habilita, para aquellos días con eventos, un enlace a la página que será mostrada en una ventana emergente.

```
function imprimir_agenda($mes, $año, $vista="reducida"){
    //se define una constante con la cantidad de segundos que tiene un día
    define("UNDIA", (60*60*24));
    //$primer_dia_mes contendrá un TIMESTAMP con el primer día del mes
    $primer_dia_mes = mktime(12, 0, 0, $mes, 1, $año);
    //se obtiene un vector asociativo a partir del TIMESTAMP creado antes
    $vect_primer_dia = getdate($primer_dia_mes);
    $meses = array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto",
    "Septiembre", "Octubre", "Noviembre", "Diciembre");
    $dias = Array("Lun", "Mar", "Mie", "Jue", "Vie", "Sab", "Dom");
    $mes_actual = $mes - 1;
    $salida_php = "
    <table border=\"1\" cellpadding=\"5\"><tr>\n
    <td colspan=\"7\" align=\"center\"><strong>$meses[$mes_actual] $año
    </strong></td></tr><tr>\n";
    foreach($dias as $dia) {
        $salida_php .= "
        <td style=\"background-color: #CCCCCC; text-align: center; width: 14%\">
        <strong>$dia</strong></td>\n";
    }
    for($cont=0; $cont < (6*7); $cont++) {
        /*el valor (6*7) garantiza las celdas suficientes para albergar aquellos meses que
        pueden llegar a tener 6 semanas diferenciadas*/
        $vect_dia = getdate($primer_dia_mes);
        //$primer_dia_mes se transforma en un vector de fechas
```



```
$dia_mes = $vect_dia["mday"];
if (($cont % 7) == 0) { //cuando $cont valga 0 o múltiplo de 7
    if ($vect_dia["mon"] != $mes) { //si cambia el mes,
        break; //se sale del bucle, terminando
    } else
        //el mes no ha cambiado, por lo que se cierre una
        //fila(semana) y se abre otra
        $salida_php .= "</tr><tr>\n";
}
if ($cont < $vect_primer_dia["wday"]-1 || $vect_dia["mon"] != $mes)
/*"$vect_primer_dia["wday"]-1" contiene el nro. del día de la semana
original (que no se incrementa en cada bucle, como sí lo hace $vect_
dia). Es necesario restarle 1 porque $cont comienza en 0 y no en 1 */
    $salida_php .= "<td>&nbsp;</td>\n";
    //o todavía no estamos en el día 1 del mes o bien ya terminó el
    //mes y se debe continuar imprimiendo celdas en blanco del mes
else {
    $obtener_eventos_sql = "
SELECT titulo_evento, DATE_FORMAT(inicio_evento, '%e') AS
dia_mes FROM calendario_eventos WHERE month(inicio_evento) =
'$mes' AND dayofmonth(inicio_evento) = '$dia_mes' AND
year(inicio_evento) = '$año' ORDER BY inicio_evento";
    $mysqli = doDB();
    $obtener_eventos_res = mysqli_query($mysqli, $obtener_eventos
_sql) or die(mysqli_error($mysqli));
    if ( (mysqli_num_rows($obtener_eventos_res)) > 0 )
        //se imprime el día como un enlace a una ventana
        //emergente que mostrará los eventos del día
        $salida_php .= "<td style=\"background-color:#aab7c1;
color:#cce5fc; valign:top; text-align:center;\"><a
href=\"javascript:eventWindow('eventos_del_dia.php?
m=$mes&d=$dia_mes&y=$año');\"><strong>".$dia_mes.
"</strong></a><br/></td>\n";
    else //día sin eventos
        $salida_php .= "<td valign=\"top\" align=\"center\">".
        $dia_mes."</td>\n"; //imprimimos el día sin ningún enlace
    $primer_dia_mes += UNDIA; //incrementamos un día
}
}
```



```
$salida_php .= "</tr></table>";  
return $salida_php;  
  
mysqli_free_result($obtener_eventos_res);  
mysqli_close($mysqli);  
} // end of function agenda()
```

El código relevante de la página (eventos_del_dia.php) mostrada en una ventana emergente, con la información sobre los eventos del día, se expone a continuación:

```
$m = $_GET["m"];  
$d = $_GET["d"];  
$y = $_GET["y"];  
  
$buscar_eventos_sql = "SELECT titulo_evento, desc_evento, DATE_FORMAT(inicio_evento,  
'%!:%i %p') as fce_evento FROM calendario_eventos WHERE month(inicio_evento) =  
\"$.m.\" AND dayofmonth(inicio_evento) = \"$.d.\" AND year(inicio_evento) = \"$.y.\"  
ORDER BY inicio_evento";  
doDB();  
$buscar_eventos_res =  
mysqli_query($mysqli, $buscar_eventos_sql) or die(mysqli_error($mysqli));  
  
if (mysqli_num_rows($buscar_eventos_res) > 0) {  
    $salida_php = "<ul>";  
    while ($ev = mysqli_fetch_array($buscar_eventos_res)) {  
        $titulo_evento = $ev["titulo_evento"];  
        $desc_evento = $ev["desc_evento"];  
        $fce_evento = $ev["fce_evento"];  
        $salida_php .= "  
<li><strong>\". $fce_evento .\": \". $titulo_evento .\"</strong>.<br/>\". $desc_  
evento.\"</li><br>";  
    }  
    $salida_php .= "</ul>";  
}  
  
mysqli_free_result($buscar_eventos_res);  
mysqli_close($mysqli);
```



Administración de contenidos.

Un usuario de tipo funcionario, dispondrá de opciones para visualizar, añadir y editar contenidos (noticias, eventos, oposiciones, etc). La implementación es análoga para cualesquiera de los contenidos, por lo que únicamente se mostrará para uno de ellos, en este caso, para el apartado de noticias.

En el caso de que el usuario funcionario además alcanzara la categoría de Administrador, dispondrá además de opciones adicionales, como ahora la eliminación de esos contenidos o bien la gestión de usuarios registrados en el portal portal.

A continuación se muestra la implementación de cada una de las opciones posibles para el apartado de noticias.

En primer lugar, el sistema muestra una relación de noticias disponibles, exhibiendo las acciones posibles a adoptar junto a cada una de ellas.

```
function mostrar_noticias() {
    if ( $_SESSION["categoria"] == "ADM" || $_SESSION["categoria"] == "INT" ) {
        $meses = array("Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",
            "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre");
        $obtener_noticias_sql = "
        SELECT id, titular, DATE_FORMAT(fecha, '%e %b %Y') AS fecha FROM
        noticias WHERE month(fecha) = " . $_GET["mes"] . " AND year(fecha) =
        " . $_GET["año"] . " ORDER BY id";
        $mysqli = doDB();
        $obtener_noticias_res = mysqli_query($mysqli, $obtener_noticias_sql) or
        die(mysqli_error($mysqli));

        if (mysqli_num_rows($obtener_noticias_res) < 1)
            $salida_php = "<p><em>No existen noticias</em></p>";
        else {
            $salida_php = "
            <br><h2>Listado Noticias para " . $meses[$_GET["mes"]-1] . ".
            $_GET["año"] . "</h2><br>
            <table cellpadding=\"3\" cellspacing=\"1\" border=\"1\">
            <tr style=\"background-color: #CCCCCC\">
```



```
<th>ID</th>
<th>TITULAR</th>
<th>FECHA</th>
<th>ACCION</th>
</tr>;
while ( $noticia_info = mysqli_fetch_array($obtener_noticias_res) ) {
    $sid = $noticia_info['id'];
    $titular = $noticia_info['titular'];
    $fecha = $noticia_info['fecha'];
    $salida_php .= "
    <tr>
    <td align=center>".$sid."</td>
    <td align=center>".$titular."</td>
    <td align=center width=17%>".$fecha."</td>
    <td align=center width=18%>
    <a href=\"javascript:eventWindow('detalles_noticia.php?
    id=$id')\">Ver detalles</a><br>
    <a href=\"javascript:eventWindow('modif_noticia.php?
    id=$id')\">Editar</a><br>;
    if ($_SESSION['categoria'] == "ADM")
        $salida_php .= "<a href=\"javascript:eventWindow
        ('eliminar_noticia.php?id=$id&tit=$titular')
        \">Eliminar</a>;
    $salida_php .= "</td></tr>;
    }
    $salida_php .= "</table>;
}
echo $salida_php;

mysqli_free_result($obtener_noticias_res);
mysqli_close($mysqli);
}
} //FIN function mostrar_noticias()
```

Asimismo, el usuario puede crear nuevas noticias, insertándolas así en la Base de Datos. La implementación que lo permite se muestra a continuación, exponiendo tanto la sección que muestra el formulario al usuario como aquella que describe las acciones que adopta el sistema después de recibir el mismo.



```
function mostrar_form_noticia() {
    if ( $_SESSION["categoria"] == "ADM" || $_SESSION["categoria"] == "INT" ) {
        echo "
        <h2>(*)Los campos marcados con asterisco son obligatorios</h2><br><br>
        <p><form enctype=\"multipart/form-data\" action=\"".$_SERVER
        [\"PHP_SELF\"].\"\" method=\"POST\">
        <strong>Titular(*):</strong>
        <input type=\"text\" name=\"titular\" size=\"100\" maxlength=\"200\">
        <br><br><strong>Texto: </strong>
        <textarea name=\"texto\" cols=\"75\" rows=\"10\"></textarea><br><br>
        <strong>Adjuntar imagen: </strong><br>
        <input type=\"file\" name=\"nombre_fichero\"><br><br>
        <input type=\"hidden\" name=\"MAX_FILE_SIZE\" value=\"10000000\">
        <input type=\"hidden\" name=\"form_enviado\" value=\"noticia\">
        <input type=\"submit\" name=\"form_noticia_enviado\" value=
        \"Confirmar\">
        <input type=\"submit\" name=\"form_cancelado\" value=\"Cancelar\">
        </form></p>";
    } else
        echo "Debe identificarse como Administrador para poder continuar.";
} //FIN function mostrar_form_noticia()

function insertar_noticia() {
    if ( $_SESSION["categoria"] == "ADM" || $_SESSION["categoria"] == "INT" ) {
        try {
            if (empty($_POST["titular"]))
                throw new Exception("Debe introducir un nombre para el
                titular");
        }
        catch (Exception $e) {
            echo "<p><span style=\"font-size:150%; color:#a5271a;\">
            ERROR: ".$e->getMessage()."</span><p>";
        }
        if (empty($e)) {
            eliminar_tags();
            $destino_fichero = "/wamp/www/pfc/noticias/";
            $titular = $_POST["titular"];
            $texto = $_POST["texto"];
        }
    }
}
```

```
$imagen = $_FILES["nombre_fichero"]["name"];
if (!empty($imagen)) { //si se adjuntó algún archivo
    if (subir_fichero($destino_fichero)) {
        $insertar_noticia_sql = "INSERT INTO noticias (titular,
        texto, imagen, fecha) VALUES ('$titular', '$texto',
        '$imagen', now())";
        $mysqli = doDB();
        $insertar_noticia_res = mysqli_query($mysqli,
        $insertar_noticia_sql) or die(mysqli_error($mysqli));
        mysqli_free_result($insertar_noticia_res);
    } else
        echo "La base de datos está en mantenimiento en
        estos momentos. Por favor, inténtelo de nuevo más
        tarde.<br>";
} else { // si no se ha seleccionado ningún archivo para adjuntar
    $insertar_noticia_sql = "INSERT INTO noticias (titular, texto,
    fecha) VALUES ('$titular', '$texto', now())";
    $mysqli = doDB();
    $insertar_noticia_res = mysqli_query($mysqli,
    $insertar_noticia_sql) or die(mysqli_error($mysqli));
    mysqli_free_result($insertar_noticia_res);
}
echo "
<p><span style=\"font-size:150%; font-style:italic; color:#d64100;\">
Noticia insertada con éxito</span></p><br><br>";

    administrar_newsletter("noticias");
}
} else
    echo "Debe identificarse como Administrador para poder continuar.";
} //FIN function insertar_noticia()
```

A continuación se expone la sección de código de interés del script (modif_noticia.php) que permite la edición de una noticia existente, actualizándola en la Base de Datos. En este caso, el sistema muestra al usuario un formulario cumplimentado por defecto con los datos existentes de la noticia a editar. Cuando el usuario envía el formulario una vez alterado, el sistema comprueba que los campos obligatorios no estén vacíos de contenido y verifica a continuación que se haya almacenado correctamente el archivo con la imagen de la noticia (siempre y cuando el usuario seleccionara alguna).



Finalmente, el sistema almacena los nuevos valores en la base de datos, siempre y cuando fuese transferido con éxito el archivo con la imagen, en el paso anterior.

```
if ( $_SESSION["categoria"] == "ADM" || $_SESSION["categoria"] == "INT" ) {
    if ( !isset($_POST["confirmar"]) && !isset($_POST["cancelar"]) ) {
        $id = $_GET["id"];
        $obtener_noticia_sql = "SELECT titular, texto, imagen, DATE_FORMAT(fecha,
        '%e %b %Y') AS fecha FROM noticias WHERE id = '$id'";
        doDB();
        $obtener_noticia_res = mysqli_query($mysqli, $obtener_noticia_sql) or
        die(mysqli_error($mysqli));
        if (mysqli_num_rows($obtener_noticia_res) > 0) {
            $noticia_info = mysqli_fetch_array($obtener_noticia_res);
            $titular = $noticia_info["titular"];
            $texto = $noticia_info["texto"];
            $imagen = "noticias/".$noticia_info["imagen"];
            $fecha = $noticia_info["fecha"];
            $salida_php = "
            <form action=\"".$_SERVER["PHP_SELF"]."\" method=\"POST\"
            enctype=\"multipart/form-data\">
            <h3>Noticia Id: $id</h3>
            <strong>Titular:</strong>
            <input type=\"text\" name=\"titular\" value=\"$titular\" size=\"50\"
            maxlength=\"200\"><br><br>
            <strong>Texto noticia:</strong><br>
            <textarea name=\"texto\" cols=\"53\" rows=\"8\">$texto
            </textarea><br><br>
            <img class=\"imagen_detalles_noticia\" src=\"$imagen\"
            width=\"256\" alt=\"Foto_noticia\"><br><br>
            <strong>Cambiar imagen:</strong><br>
            <input type=\"file\" name=\"nombre_fichero\"><br><br>
            <input type=\"hidden\" name=\"MAX_FILE_SIZE\"
            value=\"10000000\">
            <input type=\"hidden\" name=\"id\" value=$id>
            <input type=\"submit\" name=\"confirmar\" value=\"Confirmar\">
            <input type=\"submit\" name=\"cancelar\" value=\"Cancelar\">
            </form>";
        } else
            $salida_php =
            "<h2>No se encontró la noticia en la base de datos</h2>";
    }
}
```



```
} else {
    if (isset($_POST["confirmar"])) {
        $id = $_POST["id"];
        $titular = $_POST["titular"];
        $texto = $_POST["texto"];
        $imagen = $_FILES["nombre_fichero"]["name"];
        $destino_fichero = "/wamp/www/pfc/noticias/";
        $actualizar_noticia_sql = "UPDATE noticias SET titular = '$titular',
        texto = '$texto'";
        if (!empty($imagen)) { // si se ha adjuntado alguna imagen
            if (subir_fichero($destino_fichero))
                // si se ha conseguido subir el fichero sin incidencias
                $actualizar_noticia_sql .= ", imagen = '$imagen'";
            else {
                echo "La base de datos está en mantenimiento en
                estos momentos. Por favor, inténtelo de nuevo más
                tarde.<br>";
                exit;
            }
        }
        $actualizar_noticia_sql .= " WHERE id = '$id'";
        doDB();
        $actualizar_noticia_res = mysqli_query($mysqli, $actualizar_
        noticia_sql) or die(mysqli_error($mysqli));
        $salida_php .= "
        <h2>Noticia modificada correctamente</h2>
        <p><a href=# onClick=\"window.close()\">Cerrar</a></p>";
    }
    if (isset($_POST["cancelar"]))
        $salida_php .= "
        <h2>Modificación cancelada</h2>
        <p><a href=# onClick=\"window.close()\">Cerrar</a></p>";
}
} else
    $salida_php = "<h2>No tiene permisos para acceder a este recurso.</h2>";

mysqli_free_result($obtener_noticia_res);
mysqli_free_result($actualizar_noticia_res);
mysqli_close($mysqli);
```



Por último, se expone la sección de interés del script (fichero eliminar_noticia.php) que permite a un usuario Administrador eliminar una noticia.

En este caso, el usuario debe confirmar la operación de borrado, tras lo cual, el sistema procederá al borrado de la tupla en la Base de Datos, así como a la eliminación del fichero que contiene la imagen, si lo hubiere. Destacar que el script es mostrado en una ventana emergente.

```
if ( $_SESSION["categoria"] == "ADM" ) {
    //si el usuario no ha enviado el formulario, se le pide confirmación de borrado
    if ( !isset($_POST["confirmar"]) && !isset($_POST["cancelar"]) ) {
        $id = $_GET["id"];
        $titular = $_GET["tit"];
        $salida_php = "
        <strong>Titular: $titular</strong>
        <h2>¿ Está seguro que desea continuar ?</h2>
        <form action=\"".$_SERVER["PHP_SELF"]."\" method=\"POST\">
        <input type=\"hidden\" name=\"id\" value=$id>
        <input type=\"submit\" name=\"confirmar\" value=\"Confirmar\">
        <input type=\"submit\" name=\"cancelar\" value=\"Cancelar\"></h2>
        </form>";
    } else {
        $id = $_POST["id"];
        $buscar_noticia_sql =
        "SELECT titular, imagen FROM noticias WHERE id = '$id'";
        doDB();
        $buscar_noticia_res = mysqli_query($mysqli, $buscar_noticia_sql) or
        die(mysqli_error($mysqli));

        if ($noticia_info = mysqli_fetch_array($buscar_noticia_res)) {
            if ( isset($_POST["confirmar"]) ) {
                $eliminar_noticia_sql =
                "DELETE FROM noticias WHERE id = '$id'";
                $eliminar_noticia_res = mysqli_query($mysqli, $eliminar_
                noticia_sql) or die(mysqli_error($mysqli));
                $ruta_imagen = "noticias/".$noticia_info["imagen"];
                if (($eliminar_noticia_res) &&
                !empty($noticia_info["imagen"]) )
                    unlink($ruta_imagen);
            }
        }
    }
}
```



```
        $salida_php = "  
        <h2>Noticia eliminada con éxito</h2>  
        <p><a href=# onClick=\"window.close()\">Cerrar</a></p>  
        ";  
    }  
    if ( isset($_POST["cancelar"]) )  
        $salida_php = "  
        <h2>Borrado cancelado</h2>  
        <p><a href=# onClick=\"window.close()\">Cerrar</a></p>  
        ";  
    } else  
        $salida_php = "  
        <h2>Esta noticia ya fue eliminada. Debe refrescar el listado  
        pulsando F5 o haciendo click en Gestión Noticias.</h2>  
        <p><a href=# onClick=\"window.close()\">Cerrar</a></p>";  
  
    mysqli_free_result($buscar_noticia_res);  
    mysqli_free_result($eliminar_noticia_res);  
    mysqli_close($mysqli);  
    }  
} else  
    $salida_php = "<h2>No dispone de permisos para acceder a este recurso.</h2>";
```

5.4.3. Implementación de la gestión de datos .-

Se detallan en este apartado, el modo en que han sido almacenados los datos de la aplicación web así como los mecanismos recurridos para acceder y gestionar los mismos.

En la Figura 5.4.3.1. se relacionan algunos de los elementos que representan información multimedia de la aplicación, así como en la Figura 5.4.3.2 puede comprobarse la estructura de contenidos del proyecto mediante una organización basada en directorios (se ha obviado incluir el directorio raíz donde se almacenan

Tipo	Descripción
Logotipo Ayto.	Imagen sintética; formato PNG; 102x115; Tamaño: 18,3 KB
Título (parte izda.)	Imagen sintética con texto; formato PNG; 725x205; Tamaño: 175 KB
Título (parte drcha.)	Icono animado, imagen sintética+fotografía; formato GIF; 236x203; Tamaño: 602KB
Cabecera opción Sección estática	Imagen sintética con texto; formato PNG; 209x58; Tamaño 8,29KB
Icono Usuarios	Imagen sintética; formato PNG; 91x118; Tamaño: 8,01 KB
Icono Foros	Icono animado, imagen sintética con texto; formato GIF; 278x247; Tamaño: 102KB
Icono Newsletter	Icono animado, imagen sintética con texto; formato GIF; 278x247; Tamaño: 156KB
Icono Newsletter 2	Icono animado, imagen sintética con texto; formato GIF; 550x300; Tamaño: 348KB
Icono Oficina Virtual	Icono animado, imagen sintética con texto; formato GIF; 278x247; Tamaño: 44,8KB
Imagen Comunidad	Imagen sintética con texto; formato GIF; 233x131; Tamaño: 14,7 KB
Pie Pagina	Imagen fotográfica; formato PNG; 980x214; Tamaño: 166 KB
Efecto sombra Lateral	Imagen sintética; formato PNG; 1222x14; Tamaño: 4,88 KB

Fig. 5.4.3.1 – Elementos multimedia de la aplicación web

Directorio	Descripción
Bases_oposiciones	Documentos con la información de cada oposición
Css	Plantillas de estilo
Images	Iconos animados, fotos, cajas y otras imágenes sintéticas
Images_seccion_Estatica	Imágenes empleadas en la Sección más estática (Inyoayuntamiento)
Impresos	Documentos asociados a algún trámite
Noticias	Imágenes fotográficas asociadas a alguna noticia
Raíz	Scripts PHP, páginas web estáticas y librerías de funciones

Fig. 5.4.3.2 – Organización de contenidos de la aplicación web

En la Figura 5.4.3.3 se muestra un extracto (en el apartado de Anexos puede consultarse la relación completa) de la estructura de la Base de Datos, obtenido mediante la aplicación phpMyAdmin. En él puede apreciarse cómo se ha organizado la información en base a tablas, indicándose además sus atributos



```
-- Base de datos: `pfc`

-----

-- Estructura de tabla para la tabla `tramites_usuarios`

CREATE TABLE IF NOT EXISTS `tramites_usuarios` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_tram` int(11) NOT NULL,
  `username` varchar(20) NOT NULL,
  `fecha` datetime NOT NULL,
  `estado` varchar(20) NOT NULL,
  `hist_notificaciones` text NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=71 ;

-----

-- Estructura de tabla para la tabla `usuarios`

CREATE TABLE IF NOT EXISTS `usuarios` (
  `dni` varchar(9) NOT NULL,
  `nombre` varchar(20) NOT NULL,
  `apellido1` varchar(25) NOT NULL,
  `apellido2` varchar(25) NOT NULL,
  `sexo` varchar(1) NOT NULL,
  `fecha_nac` date NOT NULL,
  `antiguedad` datetime NOT NULL,
  `email` varchar(50) NOT NULL,
  `ffno` varchar(15) NOT NULL,
  `username` varchar(20) NOT NULL,
  `password` varchar(50) NOT NULL,
  `categoria` varchar(3) NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Fig. 5.4.3.3 – Extracto de la estructura de la Base de Datos, realizado con phpMyAdmin

En la Figura 5.4.3.4. puede contemplarse cómo se ha implementado la función encargada de conectar con la Base de Datos, habilitando de este forma cualquier operación posterior de consulta o actualización de la misma.

```
function doDB() {  
  
    global $mysqli;  
  
    $mysqli = mysqli_connect("localhost", "danitrack", "pfc2010", "pfc");  
  
    if (mysqli_connect_errno()) {  
        printf("Conexión fallida a la Base de Datos. El error reportado es el  
siguiente: '%s\n'", mysqli_connect_error());  
        exit();  
    } else  
        return $mysqli;  
  
} //FIN function doDB()
```

Fig. 5.4.3.4 – Función que permite la conexión a la Base de Datos de la aplicación

Como puede apreciarse, es preciso especificar para conseguir una conexión con éxito, tanto un nombre de usuario (danitrack) como un contraseña (pfc2010), así como un acceso local (localhost) y obviamente el nombre de la Base de Datos con la que se trabajará (pfc). Asimismo, si la conexión no llegase a establecerse correctamente, se imprime por pantalla el error genérico que ha podido causarlo. El elemento retornado por la función, será referenciado posteriormente para efectuar cualquiera de las acciones posibles sobre la Base de Datos.

En otro orden de cosas, se hace preciso dedicar dentro de este apartado algunas líneas sobre aquella implementación de código orientada a conservar la seguridad e integridad de la información almacenada en la Base de Datos. En este sentido, puede comprobarse en la Figura 5.4.3.5, el breve pero eficaz recurso de implementación utilizado para evitar la inyección SQL por parte de usuarios malintencionados. Cabe destacar que ha sido posible reducir notablemente la extensión del código dedicado a estos menesteres, en parte debido a la optimización del archivo de configuración de PHP y la activación de ciertas directivas de seguridad del mismo.

```
function eliminar_tags() {  
  
    foreach( $_POST as $variable => $valor ){  
        $_POST [ $variable ] = strip_tags($_POST [ $variable ]);  
    }  
  
} //end of function eliminar_tags() {
```

Fig. 5.4.3.5 – Función orientada a evitar la inyección SQL

Un vez vista la implementación que permite la conexión con la Base de Datos, dejándola habilitada para futuras consultas, se expone en la Figura 5.4.3.6 una muestra de código que permite recuperar la información de la misma para procesarla y presentarla finalmente. En este caso, se trata del script que recupera toda la información disponible sobre una noticia previamente seleccionada, mostrándola en una ventana emergente.

```
$id = $_GET["id"];  
  
$obtener_noticia_sql = "SELECT titular, texto, imagen, DATE_FORMAT(fecha, '%e %b %Y')  
AS fecha FROM noticias WHERE id = '$id'";  
doDB();  
$obtener_noticia_res = mysqli_query($mysqli, $obtener_noticia_sql) or  
die(mysqli_error($mysqli));  
  
if (mysqli_num_rows($obtener_noticia_res) > 0) {  
    $noticia_info = mysqli_fetch_array($obtener_noticia_res);  
    $titular = $noticia_info["titular"];  
    $texto = $noticia_info["texto"];  
    $imagen = "noticias/".$noticia_info["imagen"];  
    $fecha = $noticia_info["fecha"];  
    $salida_php = "  
    <h2>Noticia Id: $id:<br>$titular<br><strong style='font-size:70%'>$fecha  
    </strong></h2>  
    <img class='imagen_detalle_noticia' src='$imagen' width='256'  
    alt='Foto_noticia'>
```



```
<p>$texto</p>;

mysqli_free_result($obtener_noticia_res);
mysqli_close($mysqli);
} else
$salida_php = "<h2>No se encuentra la noticia en la base de datos.</h2>";
```

Fig. 5.4.3.6 – Código script de generación de la vista en detalle de una noticia

Como puede apreciarse, se recupera toda la información de la noticia de la Base de Datos cuyo identificador coincida con el que proviene a través del método GET (enlace seleccionado previamente por un usuario), formateando previamente algunos de los campos como por ejemplo ahora, la fecha de publicación de la noticia. Posteriormente se verifica que la consulta haya propiciado algún resultado, en cuyo caso, se obtiene la información requerida de cada campo y se imprime por pantalla con el formato y orden convenidos.



6. Evaluación y pruebas .-

Concluida la fase de implementación, se inicia una etapa final de análisis y comprobaciones, donde se verifica minuciosamente cada subapartado o *hilo* en que puede descomponerse el portal. Se examinan asimismo aspectos como la usabilidad, la *performance* del website a nivel de ancho de banda consumido, la accesibilidad atendiendo a criterios de resolución de pantalla, entre otros.

De este modo, se han comprobado uno a uno, todas las opciones de actuación disponibles para el usuario en cada una de las secciones del portal, sientiendo el resultado el esperado en todos los casos.

Las imágenes y contenidos multimedia han sido optimizados en pro de ocupar el mínimo espacio posible, sin perjudicar la calidad de forma perceptible. Para ello se han buscado los formatos de almacenamiento más apropiados para cada elemento, así como también se ha recurrido a cierto grado de compresión, según el caso. Por otro lado, se han estructurado al máximo las librerías de funciones disponibles, con el objetivo de cargar el menor número posible de ellas en cada página que las requiera. Por último, las consultas a la Base de Datos substraen sólo la información de los campos útiles, evitándose siempre que sea posible, el uso de asteriscos junto con la instrucción SELECT. Todas estas medidas, redundan en un mejor rendimiento (*performance*) en lo que se refiere a la experiencia de navegación por el website, requiriendo un menor ancho de banda y por ende, menos demora en los procesos.

Se ha recurrido a aplicaciones de terceros o a consorcios internacionales como ahora W3C, con el fin de ratificar la corrección del código desde las perspectivas más habituales y potencialmente más recurridas (evitar enlaces rotos, cumplimiento de estándares, compatibilidad con navegadores).

La amigabilidad para con la aplicación, se manifiesta notoriamente desde el primer momento de utilización, gracias a los recursos introducidos para adaptar la exposición de los contenidos atendiendo a las preferencias e histórico de uso de cada usuario.

La seguridad ha sido ampliamente abordada, como ya se ha referido en capítulos anteriores, siguiendo patrones de programación de los denominados como "buenos

hábitos", integrando recursos orientados a evitar la inyección SQL y empleando un sistema de autenticación de usuarios que permite la ejecución de aquellas partes de códigos "sensibles" sólo para aquellos usuarios con los privilegios suficientes.

Se ha introducido reiteradamente el concepto de reusabilidad, con el fin de que el usuario consiga "comprender" los aspectos de funcionamiento y comportamiento, en el menor tiempo posible. De esta forma, son muy aparentes los diseños guardando uniformidad en sus elementos, tanto en formularios como en la gran parte de elementos interactivos e inclusive se aprecia en los interfaces de administración de contenidos.

Validación CSS.

El World Wide Web Consortium (W3C) es un consorcio internacional que produce recomendaciones para la World Wide Web, entre otras, las conocidas validaciones de código orientadas a conseguir una web más "limpia" y "accesible". En este sentido, esta validaciones garantizarán, en otro orden de cosas, que indistintamente del navegador web empleado, una página web sea representada correctamente.

La implementación del código CSS del portal objeto de estudio ha sido validada con éxito utilizando el Servicio de Validación del propio W3C, obteniendo la calificación de "CSS Versión 3 válido", sin ningún error encontrado (véase Figura 6.1).



Fig. 6.1 – Aplicación del W3C para validación CSS



Para realizar la prueba de validación anterior, se ha recurrido a un servicio de alojamiento gratuito, transfiriendo allí todos los contenidos del portal y validándolo a continuación empleando el Servicio ofrecido por el W3C, como sigue :

<http://jigsaw.w3.org/css-validator/validator?uri=http://www.trackman.site11.com&profile=css3>

Asimismo, se ha ubicado en el portal, el icono que el propio consorcio W3C pone a disposición de cualquier desarrollador web, el cual representa un indicativo para que los potenciales usuarios del portal web comprueben que el mismo cumple con el ejemplo, abogando por la interoperabilidad y la accesibilidad.

Cabe mencionar sobre el código CSS implementado para el portal objeto de estudio que, a pesar de cumplir con el estándar CSS Versión 3, es compatible sólo parcialmente con el estándar CSS Versión 2.1, debido únicamente a la utilización de la propiedad "text-shadow" (efecto sombra sobre un texto). No obstante, se ha decidido conservar esta propiedad por la notable efectividad del resultado conseguido, que no es otra que la de aumentar la legibilidad de un texto cuyo color apenas contrasta con el color del fondo sobre el que presenta. Por otro lado, esta propiedad es ya contemplada por una amplia mayoría de navegadores, como ahora Firefox, Chrome, Opera o Safari.

Verificación de enlaces.

La World Wide Web es volátil, un lugar en constantes cambios. Páginas y Websites enteros cambian, se mueven y desaparecen a diario, con el potencial riesgo de enlaces "muertos" que esto conlleva.

Existen diversas aplicaciones web (online) que chequean y verifican la cantidad de enlaces inválidos (que ya no "apuntan" a una página válida), indicando además cuáles es su ruta. Una de estas aplicaciones de gran popularidad es que puede encontrarse en el website www.anybrowser.com. En la Figuras 6.2 y 6.3 pueden apreciarse unas capturas de pantalla del interface de uso así como del resultado arrojado por esta aplicación online. En resumen, un total de 86 enlaces referenciados , todos ellos válidos, apuntados desde el portal web objeto de estudio.

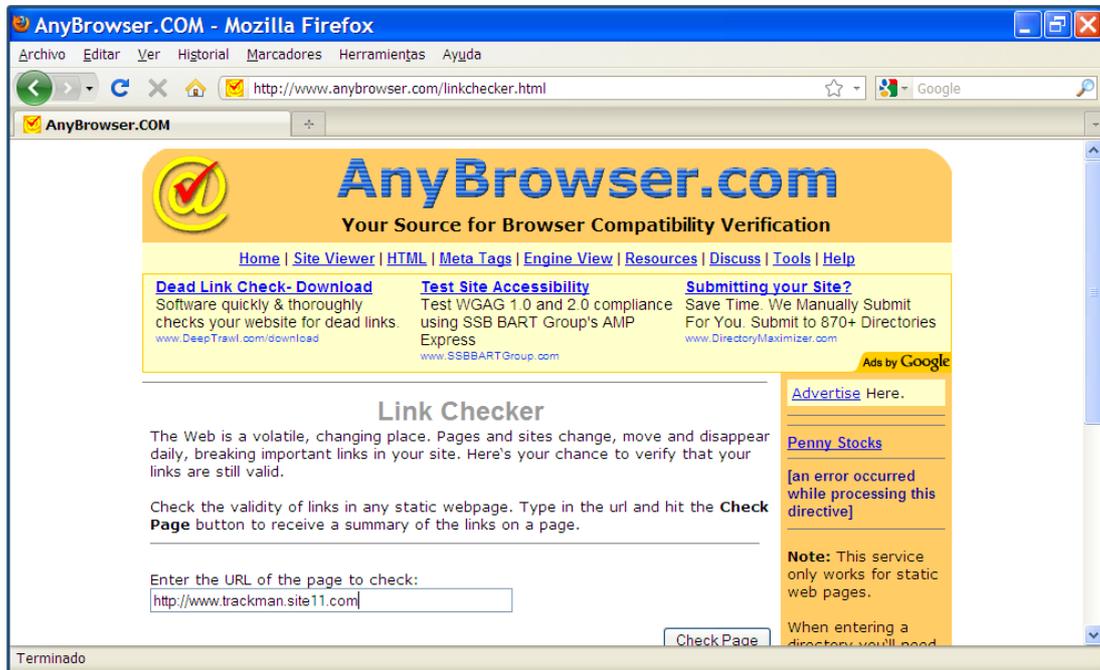


Fig. 6.2 – Interface de la aplicación online www.anybrowser.com, servicio de chequeo de enlaces

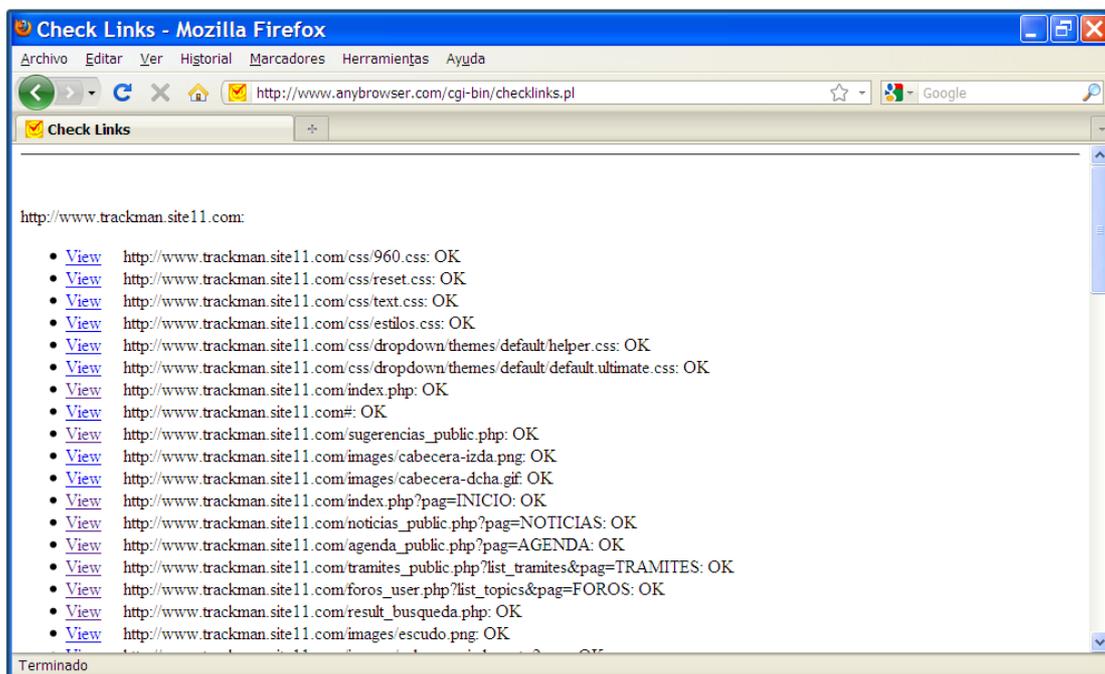


Fig. 6.3 – Resultado del chequeo de enlaces, ofrecido por la aplicación online www.anybrowser.com



Análisis de resoluciones de pantalla.

Durante el desarrollo de la aplicación, se buscó en todo momento que la misma pudiera ser visualizada de forma correcta y apta para su uso en diferentes resoluciones de pantalla o escritorio. El cumplimiento de esta peculiaridad recobra una especial relevancia en el momento preciso en el que nos encontramos, debido a la proliferación de ordenadores portátiles de tamaños reducidos, popularmente conocidos como netbooks y sus análogos, donde el tamaño de pantalla no acostumbra a superar las 10 pulgadas y una resolución de 1024x768 ppp.

En el momento de la presente edición, el ranking mundial de resoluciones de pantalla más extendidos queda como sigue:

1. 1024x768:	24.65%
2. 1280x800:	18.64%
3. 1280x1024:	10.51%
4. 1440x900:	8.41%
5. 1366x768:	6.20%
6. 1680x1050:	5.45%
7. 800x600:	2.60%
8. 1920x1200:	2.17%
9. 1152x864:	2.11%
10. 1920x1080:	2.01%

La aplicación ha sido utilizada y testeada intensamente para las resoluciones que ostentan las 5 primeras posiciones, siendo el resultado satisfactorio y según lo previsto. En gran medida, el éxito en este resultado se debe a la utilización del framework CSS 960 Grid System, cuyas principales virtudes ya fueron expuestas en un capítulo anterior.

A continuación se adjuntan algunas capturas de cómo es visualizada la aplicación para distintas resoluciones de pantalla (Figuras 6.4, 6.5 y 6.6).



Fig. 6.4 – Visualización de la aplicación en una resolución de 1366x768



Fig. 6.5 – Visualización de la aplicación en una resolución de 1280x800



Fig. 6.6 – Visualización de la aplicación en una resolución de 1024x768

Estudio de navegadores web.

Otro aspecto fundamental a evaluar redonda en la importancia de que nuestra aplicación sea visualizada y se comporte con normalidad y de forma análoga por cualquier navegador web disponible, preferentemente por aquéllos que ostentan un mayor porcentaje de cuota de utilización por parte de usuarios.

En el momento de edición de esta memoria, el ranking mundial de navegadores más utilizados queda como sigue:



- 1º) Internet Explorer: 55,73%
- 2º) Mozilla Firefox 30,98%
- 3º) Google Chrome 11,44%
- 4º) Opera 0,79%
- 5º) Safari 0,69%

La aplicación ha sido utilizada y testeada intensamente en todos los casos anteriores, siendo el resultado satisfactorio y según lo previsto. Se adjuntan capturas de los 3 primeros, los cuales ostentan una cuota significativamente superior a los restantes. Las Figuras 6.4 y 6.5 hacen referencia a los navegadores Chrome y Firefox respectivamente, mientras que la Figura 6.7 se refiere al navegador Internet Explorer.



Fig. 6.7 – Visualización de la aplicación mediante el navegador Internet Explorer



7. Conclusiones .-

Todo principio tiene un final. Si bien esta popular cita puede ser contemplada en casi cualquier proyecto o contexto, estoy convencido de que para el caso que ahora me ocupa, no me hayo en el final sino en el principio. No en vano es ahora cuando se abre un sinfín de posibilidades. Disponer de los conocimientos necesarios para crear nuevos contenidos para la "red de redes", indudablemente te hace sentir privilegiado, casi talentoso me atrevería a añadir, sobre todo después de comprobar la diversidad de tecnologías web y lenguajes afines que se precisan conocer a la par que aunar, de modo que la solución final discorra según las pretensiones iniciales.

La programación web es algo que durante el desarrollo de la carrera siempre me había llamado poderosamente la atención, si bien otras disciplinas relacionadas con la informática también lo hicieron, profiriendo a éstas últimas mayor prioridad que a la primera. Como no encontraba el momento de experimentar con esta materia que me atraía desde hacía ya algún tiempo, decidí que una lógica forma de abordarla sería canalizándolo a través de la última asignatura pendiente de mis estudios, el Proyecto Final de Carrera. De este modo busqué un tema para el mismo que contemplase la manipulación de la mayor cantidad posible de tecnologías web, como ahora el proyecto que nos ocupa, el de un completo y funcional portal, orientado en este caso a las gestiones de un Ayuntamiento.

Durante el desarrollo del proyecto, no faltaron los días, incluso en ocasiones semanas, de gran desconcierto e inclusive de desesperación, donde el desarrollo del mismo parecía haber entrado en una fase de estancamiento, sin salida por momentos, donde fue necesario aplicar grandes dosis de disciplina y rigurosidad para continuar adelante y recuperar el ritmo y el camino inicial. También fueron de vital ayuda los conocimientos adquiridos durante la carrera como los relacionados con el diseño y gestión de Bases de Datos o ahora la ingeniería del software.

Me gustaría asimismo plantear una serie de ampliaciones posibles y que quedan fuera de la extensión y objetivo de este proyecto. Me refiero a ampliaciones orientadas a aumentar la interactividad para con el usuario, utilizando para ello más recursos Javascript. Asimismo, otras ampliaciones posibles estarían orientadas a aumentar la funcionalidad de la sección Oficina Virtual, como ahora el envío de mensajes de móvil



(SMS) o bien mejorar la comunicación Funcionario-Usuario de forma que fuera más directa, casi en tiempo real, empleando tecnologías análogas a lo que se conoce como Chat o también Mensajería instantánea.

Concluyendo y con total convicción, puedo afirmar que todo lo aprendido hasta el día hoy, ha superado ampliamente mis expectativas iniciales y que me siento feliz y afortunado de haber seleccionado esta temática y no otra para mi Proyecto Final de Carrera. Por otro lado, mi satisfacción se ve reforzada sólo con la idea de que un amplio abanico de opciones se me abren en estos momentos, tanto para mi desarrollo personal como profesional.



8. Bibliografía .-

- Álvarez, M.A.
Desarrollo Web, Tu mejor ayuda para aprender a hacer webs.
(<http://www.desarrolloweb.com>).
Accedido en sept.2009.
- Buendía, F.
Una guía para la realización y supervisión de proyectos final de carrera (PFC) en el ámbito de la web.
Ed. UPV, 2008.
- Factoría de Internet, S.L.
WebTaller
(<http://www.webtaller.com>).
Accedido en oct.2009.
- Sanger L., Wales J.
Wikipedia, La enciclopedia libre.
(<http://es.wikipedia.org>).
Accedido en sept.2009.
- Welling L., Thomson L.
Programación Desarrollo Web con PHP y MYSQL.
Ed. Anaya Multimedia, S.A., 2009.