

Portal Web para la gestión de Ayuntamientos

Proyecto Final de Carrera

[Ingeniería Técnica Informática de Gestión]

Autores: Jose Manuel Francés López

Vicent Palau Arnau

Director: Félix Buendía García.

Septiembre de 2012.

Índice

1.	Introducción	6
1.1.	Objetivos.	6
1.2.	Contexto.	6
1.3.	Trabajo realizado.....	8
1.4.	Estructura memoria.....	8
2.	Especificación de requisitos.....	10
2.1.	Introducción.....	10
2.1.1.	Propósito.	10
2.1.2.	Ámbito.	10
2.1.3.	Definiciones, siglas y abreviaturas.....	11
2.1.4.	Referencias.	12
2.1.5.	Visión global.	12
2.2.	Descripción general.	13
2.2.1.	Perspectivas del producto.....	13
2.2.2.	Funciones del producto.....	13
2.2.3.	Características del usuario.....	14
2.2.4.	Restricciones.	15
2.2.5.	Supuestos y dependencias.....	16
2.3.	Requisitos específicos.....	16
2.3.1.	Requisitos de interfaces externos.....	16
2.3.2.	Requisitos funcionales.....	19
3.	Análisis.....	48
3.1.	Introducción.....	48
3.2.	Casos de uso.	48
3.3.	Diagramas de clase.....	53
3.4.	Diagramas de secuencia.	57
3.4.1.	Escenario Autenticarse.	58
3.4.2.	Escenario Recuperar Contraseña.	59
3.4.3.	Escenario Buscar Oposición.....	60
3.4.4.	Escenario Subir Imagen.	61



3.4.5.	Escenario Eliminar Evento.....	62
3.4.6.	Escenario Crear Usuario.	63
	<i>Fig. 15: Diagrama de secuencia Crear usuario.....</i>	<i>63</i>
4.	Diseño.....	64
4.1.	Introducción.....	64
4.2.	Nivel de presentación.....	65
4.3.	Nivel de negocio.....	67
4.4.	Nivel de datos o almacenamiento.....	69
4.4.1.	Diagrama entidad – relación.....	69
4.4.2.	Diseño lógico.....	71
5.	Implementación.	73
5.1.	Tecnologías.....	73
5.1.1.	HTML y HTML 5.0.	73
5.1.2.	CSS y CSS 3.....	75
5.1.3.	JavaScript.....	79
5.1.4.	PHP.....	80
5.1.5.	MySQL.	82
5.2.	Implementación del portal.....	83
5.2.1.	Parte pública.....	83
5.2.2.	Parte privada.....	100
5.2.3.	Implementación de la base de datos.....	112
6.	Evaluación.	113
6.1.	Visualización en diferentes navegadores.	113
6.1.1.	Mozilla Firefox 15.0.....	114
6.1.2.	Internet Explorer 8.	114
6.1.3.	Google Chrome.....	115
6.1.4.	Safari.	116
6.2.	Visualización en diferentes resoluciones.	116
6.3.	Validación de enlaces.	118
6.4.	Validación de estándares.....	119
7.	Conclusiones.	120
8.	Bibliografía.	122

1. Introducción

Este documento describe el trabajo realizado en el proyecto final de carrera de Ingeniería Técnica de Informática. Este consiste en la implementación de una aplicación web para la gestión de la información de la localidad de la comarca de l'horta sud, Alcàsser. Se pretende crear un portal que permita acercar dicha localidad tanto a visitantes como a las personas residentes en dicha localidad.

1.1. Objetivos.

La finalidad principal del proyecto es mediante este portal, acercar a la población toda la información referente a la localidad y el ayuntamiento, así como aumentar la capacidad de interacción con la aplicación por parte del usuario. La nueva aplicación dispondrá de 3 perfiles de usuario junto con una serie de características acerca de sus funcionalidades que tendrá a su disposición.

- Usuario Administrador: personal del ayuntamiento encargada de realizar las gestiones de las distintas funcionalidades que tendrá el portal, como por ejemplo: crear, eliminar y editar.
- Usuario Interino: El rol que cumple es el propio de un funcionario del ayuntamiento, es decir, dispondrá de una interfaz dedicada con el objetivo de gestionar los contenidos de una parte del portal.
- Usuario Registrado: será aquella persona que previamente haya completado el formulario de registro satisfactoriamente y se haya autenticado en el portal. Tendrá una serie de privilegios que no dispondrá un usuario anónimo, como por ejemplo: realizar reservas de las instalaciones deportivas.

1.2. Contexto.

El proyecto tiene su ámbito de aplicación inicial en el Ajuntament d'Alcàsser (Valencia). Dada la escasez de información ofrecida a través de la página web del Ayuntamiento y muchas veces información desactualizada, nos vimos obligados a intentar mejorar en aquellos aspectos que fuera posible esta situación.



Fig. 1: www.alcasser.es

Principalmente, la aplicación anterior solamente contiene información para el ciudadano, y en ningún caso permite la posibilidad de interactuar con la aplicación debido a que solamente existen dos tipos de usuario, siendo usuarios/as todas aquellas personas que visiten la página y usuario administrador/a, aquella persona trabajador/a del ayuntamiento que tenga acceso a insertar, modificar o eliminar noticias, eventos, etc.

Desde nuestro proyecto se han mantenido los apartados que muestran la información referente al municipio o al ayuntamiento pero además se ha hecho pensando en que tenga una mayor afluencia dicho portal. Para conseguirlo se han realizado varios perfiles de usuarios que han sido explicados en el apartado anterior.

Una mejora que hemos introducido respecto al portal inicial es la posibilidad de registrarse mediante un formulario de registro. Aquellas personas que cumplimenten con éxito el formulario pasarán a formar parte de otro grupo de usuario (usuarios registrados) siempre que estos se identifiquen a través de su nombre de usuario y su contraseña.

Una vez identificados, podrán visualizar toda la información que anteriormente como usuarios podían visualizar, además de poder hacer uso del sistema de reserva on-line de las instalaciones deportivas. Otra cosa que se permite a los

usuarios registrados, es la posibilidad de poder contribuir con la galería de imágenes subiendo aquellas imágenes que deseen y clasificándolas por los distintos ítems (Fiestas, deportes, Historia, etc).

También hemos insistido en la delegación de tareas ya que en la aplicación actual tampoco existe, y para esto hemos creado un usuario interino. Este tipo de usuario se encargará de introducir y/o modificar aquella parte del contenido que este encargado (noticias, eventos, etc), pero nunca tendrá la posibilidad de poder eliminar.

1.3. Trabajo realizado.

Para la realización del proyecto no hemos tenido problemas a la hora de dividirnos las tareas entre el grupo. Aunque se ha dividido en partes, no significa que cada uno solamente se haya dedicado a su parte, sino que también nos hemos ayudado en la parte que en principio no teníamos asignada.

El trabajo ha sido dividido como se explica a continuación.

José Manuel Francés López se ha encargado de la parte más técnica, de la implementación del lenguaje PHP, junto con la interacción con la base de datos, es decir, MySQL.

Vicent Palau Arnau, se ha encargado de la parte estética, es decir, el lenguaje HTML, junto con el JavaScript y de las hojas de estilo CSS que dan forma a la estructura de este portal.

1.4. Estructura memoria.

En esta memoria se detallan todas las fases que se han desarrollado para la realización del proyecto que se ha llevado a cabo para la elaboración del Portal Web del Ayuntamiento de Alcàsser. La estructura de la misma se divide en un capítulo específico por cada etapa realizada.

A continuación vamos a presentar cada una de las fases.

En primer lugar, se describe la especificación de requisitos software (ERS) en la que se va hacer una descripción completa del comportamiento del sistema que se va a desarrollar.

En los siguientes capítulos se van a documentar las etapas de análisis y diseño de la aplicación. Para poder realizar estas etapas hemos tenido que realizar diferentes tipos de diagramas para describir de forma gráfica las funcionalidades establecidas durante la especificación de requisitos como la base de datos o aspectos relacionados con los procesos que se desarrollan en el proyecto.

A continuación, pasaremos a la fase de implementación donde destacaremos las diferentes tecnologías utilizadas para llevar a cabo el proyecto, así como las diferentes estructuras de las páginas. También se comentarán aquellas partes del proyecto que presenten mayor interés para comprender su correcto funcionamiento.

Una vez explicada la implementación, pasaremos a la fase de pruebas, donde mostraremos el comportamiento de la aplicación en diferentes situaciones, así como la validación de estándares web que cumple el proyecto.

2. Especificación de requisitos

La especificación de requisitos contiene una descripción completa del comportamiento del portal web, que en nuestro caso va a ser el Ayuntamiento de Alcàsser, así como las características específicas de dicho portal.

2.1. Introducción.

Con este proyecto se busca intentar acercar el ayuntamiento a los ciudadanos utilizando la posibilidad que un portal puede ofrecer, como por ejemplo, mirar, participar, gestionar, compartir, etc.

2.1.1. Propósito.

En el proyecto que vamos a realizar, se ha optado por emplear una Especificación de Requisitos de Software (ERS) estándar IEEE- 830- 1998, de esta forma, obtendremos una especificación consistente y no ambigua.

Esta especificación nos servirá de enlace de comunicación entre el cliente (Ayuntamiento de Alcàsser) y el equipo de desarrollo.

2.1.2.Ámbito.

El producto para el cual se crea esta especificación de requisitos, será la creación del portal web del Ayuntamiento de la localidad de Alcàsser (Valencia).

El portal cumplirá como mínimo los siguientes requisitos:

- *Consistente*: el usuario tendrá los elementos siempre localizados en la misma posición, de manera que no lleve a error localizar las distintas secciones.
- *Organizado*: el sistema estará siempre organizado para que el usuario acceda con facilidad a los datos que busca.
- *Preciso*: mediante formularios de búsqueda se pretende que el usuario encuentre lo que busca.
- *Intuitivo*: ofrecerá al usuario un acceso rápido y eficaz a la zona requerida.

El portal contará con todo tipo de información referente al ayuntamiento. En él distinguiremos dos partes: una parte más estática y otra más dinámica.

La parte estática contará con información referente a la estructura jerárquica del ayuntamiento. La parte dinámica será cambiante, es decir que se actualizará y cambiará según la fecha, esto incluirá: noticias, trámites, oposiciones, eventos...

2.1.3. Definiciones, siglas y abreviaturas.

- **W3C:** El **World Wide Web Consortium**, es un consorcio internacional que produce recomendaciones para la World Wide Web.
- **E.R.S.:** La **Especificación de Requisitos Software** es una descripción completa del comportamiento del sistema que se va a desarrollar.
- **IEEE:** corresponde a las siglas de (Institute of Electrical and Electronics Engineers) en español **Instituto de Ingenieros Eléctricos y Electrónicos**, una asociación técnico-profesional mundial dedicada a la estandarización.
- **XAMPP:** es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.
- **HTTP:** (en español *protocolo de transferencia de hipertexto*) es el protocolo usado en cada transacción de la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force.
- **HTML:** siglas de **HyperText Markup Language** (*Lenguaje de Marcado de Hipertexto*), es el lenguaje de marcado predominante para la elaboración de páginas web.
- **APACHE:** El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

- Javascript: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.
- PHP: es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.
- MySQL: es un sistema de gestión de base de datos relacional, multihilo y multiusuario.
- CSS: Las hojas de estilo en cascada (en inglés Cascading Style Sheets). CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilos que servirán de estándar para los agentes de usuario o navegadores.

2.1.4.Referencias.

Las fuentes de información consultadas para llevar a cabo esta especificación de requisitos han sido:

- www.wikipedia.org
- Buendía, Félix.

Una guía para la realización y supervisión de proyectos de final de carrera (PFC) en el ámbito de la web.

Ed. UPV, 2008.

2.1.5.Visión global.

La especificación de requisitos, va a ofrecer al cliente una visión global de las diferentes funcionalidades con la que contará la aplicación, para posteriormente, llevar a cabo la descripción de cada uno de los requisitos específicos.

2.2. Descripción general.

El objetivo de este apartado es ofrecer una descripción global del portal que vamos a desarrollar. De esta forma, evitaremos posibles modificaciones en el futuro de este documento.

2.2.1. Perspectivas del producto.

El producto funcionará con cualquier plataforma o navegador web de actualidad, siempre y cuando se tenga una conexión a internet.

Las tecnologías con las que contará para su implementación serán:

- Lenguaje HTML, PHP, MySQL...

Para todo esto, necesitaremos contar con un servidor donde podamos alojar el producto que soporte las tecnologías arriba descritas, siendo recomendable un dominio que sea fácil de recordar para los usuarios.

2.2.2. Funciones del producto.

El portal web que vamos a realizar, ofrecerá funciones diferentes según el tipo de usuario.

- Acceso a la información referente al municipio y al ayuntamiento.
- 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 usuarios.
 - Visualización de usuarios y toda la información asociada a su registro.
 - Registro y autenticación de usuarios.
 - Modificación de la información del registro.
 - Modificación del password y envío de uno nuevo en caso de olvido.
 - Creación, edición y eliminación de usuarios.

- Cierre de sesión de usuarios.
- 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 nuevos elementos.
- Implementación de un motor de búsqueda interno.
 - Definición de uno y hasta dos términos simultáneos para la búsqueda.
 - Opción de búsqueda en todo en website o sólo en la página actual.
- Operaciones sobre las instalaciones del polideportivo.
 - Visualización de reservas.
 - Creación, edición y eliminación de una reserva de la instalación del polideportivo.
- Operaciones con imágenes.
 - Visualizar, subir y eliminar imágenes.

2.2.3. Características del usuario.

Dependiendo del tipo de usuarios que va a hacer uso de esta aplicación, podemos clasificarlos de la siguiente manera:

- Usuario Anónimo.
 - Un usuario anónimo será aquel visitante esporádico o no que no se haya autenticado.
 - Trataremos permanentemente de persuadirlo para que termine registrándose en el portal.
 - Tendrá acceso total a algunos apartados del portal, como por ejemplo, al apartado de noticias, agenda, información municipio, etc. Pero no podrá recibir boletines informativos.
- Usuario Registrado.
 - Un usuario registrado será aquella persona que previamente haya completado el formulario de registro satisfactoriamente y se haya autenticado en el portal.
 - Los usuarios registrados tendrán acceso a los mismos apartados que un usuario anónimo, además de los siguientes:

- Dispondrán de la opción de suscribirse al servicio de Noticias.
 - Podrá visualizar y subir imágenes.
 - Realizar reservar y anulaciones de instalaciones deportivas.
- Usuario Interino.
 - Un usuario Interino sólo podrá ser creado por un administrador.

El rol que cumple es el propio de un funcionario del ayuntamiento, es decir, dispondrá de una interfaz dedicada con el objetivo de gestionar los contenidos de una parte del portal.

- Visualizar, dar de alta y editar noticias, eventos en la agenda, etc.
 - 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.
 - Eliminar imágenes, el usuario interino podrá eliminar imágenes si así fuera necesario.
- Usuario Administrador.
 - Un usuario Administrador 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 en la agenda, etc.
 - Crear y eliminar instalaciones deportivas.

2.2.4. Restricciones.

Según hemos comentado anteriormente, solamente bastará una computadora con un navegador web y una conexión a internet para tener acceso al portal.

En dicho proyecto, se adoptarán 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.

De todos modos, es recomendable alojar el website en un servidor con HTTPS.

2.2.5. Supuestos y dependencias.

El portal web 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.

Como también se indica 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.

Los requisitos específicos vamos a describirlos atendiendo al nivel de detalle preciso que permita abordar con éxito las siguientes etapas del proyecto:

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

2.3.1. Requisitos de interfaces externos.

Según la tipología de usuario, pueden diferenciarse 2 tipos de interfaces de usuario:

- Usuarios no administradores (anónimos y registrados)

Los usuarios no administradores dispondrán de una página web, la cual estará distribuida de la siguiente manera:

- Sendas barras de navegación situadas en la parte superior como en el pie de página, incluyendo repetición de enlaces directos a algunas de los apartados principales.
- También en la parte superior, barra de navegación principal con acceso a los apartados principales del portal.
- Debajo de la barra de navegación principal, se encuentra la caja de búsquedas, la cual ofrecerá la opción de buscar en todo el portal.
- La zona central, estará dedicada a mostrar un resumen de los últimos elementos incorporados a cada uno de los apartados principales.

- La zona lateral derecha estará dedicada a mini-secciones de funciones de visualización rápida.
- La zona lateral derecha superior, estará dedicada al área de autenticación de usuarios, modificación del perfil de usuario y su contraseña.



Fig. 2: Formato de la página para los usuarios no administradores.

- Usuarios administradores (Administrador e interinos)

Los usuarios administradores dispondrán de una página web, la cual estará distribuida de la siguiente manera:

- En la zona más alta de la cabecera, se dispondrá de sendos enlaces: uno para poder visualizar el website como si de un 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 del portal.
- El área lateral izquierda superior estará reservada para las diversas opciones de gestión y que irán variando según el apartado principal seleccionado en la parte de navegación principal superior.

- El área central principal estará dedicado a mostrar las informaciones propias de cada gestión en curso.
- Sendas barras de navegación situadas en la parte superior como en el pie de página, incluyendo repetición de enlaces directos a algunas de los apartados principales.



Fig. 3: Formato de la página para los usuarios administradores.

- Interfaces de hardware

Como ya se ha mencionado anteriormente, los requisitos hardware necesario 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 7, sobre el cual se ha instalado y configurado un servidor Apache.

Los lenguajes de programación que serán utilizados para la implementación de esta aplicación web son PHP y HTML, junto con plantillas CSS y JavaScript.

El sistema de gestión de bases de datos correrá a cargo del sistema MySQL.

2.3.2.Requisitos funcionales.

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

Usuario Anónimo.

Un usuario anónimo será aquel visitante esporádico o no que no se haya autenticado.

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.
Entradas	Cada uno de los campos del formulario de registro.
Proceso	El sistema muestra el formulario de registro de nuevo usuario. Si el usuario confirma el envío, el sistema verificará que los campos cumplimentados por el usuario marcado como obligatorios, hayan sido cumplimentados satisfactoriamente, así como la corrección de formato de algunos de ellos. Si las condiciones anteriores se cumplen, el nuevo usuario es insertado en la base de datos, asegurándose previamente el sistema de que no exista duplicidad en los campos Username y Email.
Salidas	El sistema devolverá un texto al usuario informando de que el proceso fue satisfactorio.

2.3.2.2.	Autenticarse en el portal
Introducción	Para identificarse en el portal e iniciar una sesión, es imprescindible con que el usuario ingrese su nombre usuario y contraseña en el área de Usuarios.
Entradas	Los datos Username y Password ingresados por el usuario.
Proceso	El sistema contrastará la información recibida del cliente con la base de datos. En caso de encontrar una tupla que coincida, se validará la información e iniciará una sesión.
Salidas	En caso de haber introducido correctamente los campos de autenticación, el sistema le dará un mensaje de bienvenida, redirigiendo al usuario a la página correspondiente, en virtud de los permisos que dicho usuario tiene. En caso de fallar en el proceso de autenticación, el sistema sugiere al usuario a volver a intentarlo.

2.3.2.3	Recuperar la contraseña en caso de olvido.
Introducción	Si el usuario no se autentifica con éxito, debido a un olvido de la contraseña, el sistema le facilitará un pequeño formulario para que ingrese su Username y Email. Posteriormente, el sistema le generará una contraseña automática que le enviará a su Email que podrá ser modificada posteriormente.
Entradas	Los datos de Username y Email ingresados por el usuario.
Proceso	Si los datos existen en una misma tupla, el sistema generará automáticamente una contraseña aleatoria. A continuación, la nueva contraseña es almacenada en la base de datos y enviada por email al usuario para su conocimiento. Si los datos ingresados no coinciden con una tupla, se informará al usuario y se le invitará 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 a intentarlo con motivo de que algunos de los datos ingresados no son correctos.

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 haciendo click sobre un día del calendario, el cual deberá estar marcado de forma destacada como no vacío. Los días sin evento no dispondrán de enlace alguno. Los usuarios dispondrán de la posibilidad de poder cambiar el mes y el 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 base de datos toda la información referente a los actos o eventos existentes para el día seleccionado.
Salidas.	Mostrará una ventana con los detalles 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 visión ampliada del mismo, ubicada fuera del portal.
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 preseleccionado.

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 a 7 días e información detallada) haciendo click en la imagen del tiempo integrada en la página principal del portal.
Entradas.	Click en la imagen del tiempo.
Proceso	El usuario es redirigido al website de www.eltiempo.es
Salidas.	Se muestra el website de www.eltiempo.es , con la información ampliada referida a la población de Alcàsser.

2.3.2.7.	Contactar con el Ayuntamiento.
Introducción	El usuario dispone de un apartado de sugerencias para hacer llegar al ayuntamiento cualquier tipo de comentario.
Entradas.	Datos del formulario enviados por el usuario.
Proceso	El sistema verificará que hayan sido cumplimentados los campos del formulario marcados como obligatorios de forma correcta. Si todo ha sido correcto, se enviará un correo al servicio de atención al cliente del ayuntamiento, incluyendo los datos del usuario y el texto de la sugerencia.
Salidas.	El sistema mostrará al usuario que envía la sugerencia un mensaje diciendo que el envío ha tenido éxito. En caso de producirse algún error, el sistema invitará al usuario a volver a intentarlo.

2.3.2.8.	Iniciar búsquedas selectivas dentro del portal.
Introducción	El usuario dispone en el portal de un buscador de cualquier tema referente a cualquiera de las secciones del portal. También podrá restringir la búsqueda a la sección en la que se encuentre.
Entradas.	La palabra o palabras introducidas por el usuario, así como la opción de búsqueda en todo el portal o en la sección en la que se encuentre.
Proceso	El sistema buscará dentro del portal las páginas donde aparece la palabra especificada.
Salidas.	Se visualizará una página con los enlaces a las páginas que contienen esa palabra.

2.3.2.9.	Visualizar las últimas noticias.
Introducción	El usuario puede visualizar un resumen con las últimas noticias más destacadas. Para ello, solamente seleccionará la sección noticias.
Entradas.	Enlace seleccionado por el usuario.
Proceso	El sistema busca las últimas noticias introducidas en la base de datos, extrayendo toda la información y ordenando el resultado desde la más actual a la más antigua.
Salidas.	Páginas con las últimas noticias, visualizando toda la información.

2.3.2.10.	Visualizar en detalle una noticia.
Introducción	El usuario puede visualizar una sola noticia con todos sus detalles pulsando sobre el titular de la noticia.
Entradas.	Identificador de la noticia seleccionada por el usuario.
Proceso	El sistema busca en la base de datos la noticia y extrae la información.
Salidas.	Muestra toda la información disponible sobre la noticia.

2.3.2.11.	Visualizar todos los eventos del mes.
Introducción	El usuario tiene la opción de visualizar todos los eventos programados en la agenda para el mes actual o bien para cualquier otro que desee consultar.
Entradas.	Enlace en 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 base de datos todos los detalles de los eventos para el mes actual. Si la entrada es el formulario, el sistema se ceñirá la búsqueda a los eventos programados para la fecha seleccionada en el formulario.
Salidas.	Página con los detalles de cada uno de los eventos programados del mes actual o bien de los eventos programados en las fechas escogidas en el formulario.

2.3.2.12.	Visualizar en detalle un evento.
Introducción	El usuario puede visualizar un evento con sus respectivos detalles, pulsando sobre el título del evento.
Entradas.	Identificador del evento seleccionado por el usuario.
Proceso	El sistema busca en la base de datos y extrae la información del evento seleccionado.
Salidas.	Muestra toda la información disponible sobre el evento escogido.

2.3.2.13.	Visualizar trámites disponibles.
Introducción	El usuario puede buscar los trámites que desde el portal puede realizar. Se podrá buscar un trámite en concreto utilizando filtros de búsqueda.
Entradas.	Formulario de búsqueda enviado por el usuario, donde contendrá los filtros de la búsqueda seleccionados por el mismo
Proceso	El sistema muestra el formulario de búsqueda con los criterios seleccionables por parte del usuario.
Salidas.	Se mostrará un listado con los trámites que cumplen los criterios introducidos por el usuario.

2.3.2.14.	Visualizar en detalle un trámite.
Introducción	Se permitirá al usuario visualizar los detalles de un trámite en concreto al clicar sobre el mismo, desde la lista de trámites.
Entradas.	Identificador del trámite seleccionado.
Proceso	El sistema busca el trámite seleccionado en la base de datos del sistema y extrae la información del trámite en cuestión.
Salidas.	Se mostrará por pantalla la información del trámite que el usuario ha seleccionado.

2.3.2.15.	Mostrar alguna de las secciones sobre el ayuntamiento.
Introducción	El usuario puede consultar la información referente al funcionamiento interno del ayuntamiento (horarios, teléfonos, Saluda del alcalde, concejales y sus cargos, etc...)
Entradas.	Se accederá desde el menú lateral dedicado al Ayuntamiento y su funcionamiento.
Proceso	El sistema accede a la página de la información requerida y la muestra.
Salidas.	Página estática con la información seleccionada.

2.3.2.16.	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 actual.
Entradas.	Ninguna.
Proceso	El sistema actualiza la información sobre la fecha y el día de la semana cada vez que se carga la página.
Salidas.	La fecha y el día, estarán visualizados permanentemente en la parte superior derecha del portal.

2.3.2.17.	Visualizar últimos usuarios registrados.
Introducción	En el apartado de visitas en la página principal, se muestra el Username de los últimos usuarios que se han registrado y la fecha en la que lo hicieron.
Entradas.	Ninguna.
Proceso	El sistema buscará en la B.D. los usuarios registrados en los últimos 30 días (no se mostrarán los usuarios interinos ni los Administradores).
Salidas.	Se mostrará por pantalla un listado con los últimos usuarios registrados.

2.3.2.18.	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 abrirá una ventana imprimiendo el texto del Aviso Legal.
Salidas.	Muestra el texto del Aviso Legal.

Usuario Registrado.

Un usuario registrado será aquella persona que previamente haya completado el formulario de registro satisfactoriamente y se haya autenticado en el portal.

2.3.2.19.	Cambiar contraseña.
Introducción	El usuario puede cambiar su contraseña en cualquier momento 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 nos muestra un formulario para poder realizar el cambio de contraseña. Si el usuario lo rellena y confirma su envío, el sistema realizará una consulta a la base de datos para verificar que la contraseña antigua es correcta. Al mismo tiempo, actualiza la nueva contraseña en la base de datos e informa al usuario que el proceso ha finalizado con éxito.
Salidas.	Muestra un mensaje al usuario indicándole que el cambio de contraseña sea realizado correctamente (en el caso que todo sea correcto) y por el contrario, nos informará que hemos introducido algún campo erróneamente.

2.3.2.20.	Modificar perfil y preferencias de usuario
Introducción	El usuario podrá modificar sus opciones de perfil, sus datos de registro y sus preferencias.
Entradas.	Formulario con los datos de registro del usuario.
Proceso	El sistema comprobará si las modificaciones son posibles, buscando en la base de datos si ya existe el mail y/o el Username que han sido sustituidos. Si los datos son correctos se actualizará la tupla de la base de datos.
Salidas.	Se informa al usuario que las modificaciones han sido realizadas o si por el contrario no se pueden realizar.

2.3.2.21.	Subscribirse al servicio de Noticias.
Introducción	El usuario puede subscribirse al servicio de envío de boletines.
Entradas.	Formulario con las opciones de suscripción seleccionadas por el usuario.
Proceso	El sistema muestra el formulario de suscripción. Cuando el usuario envía el formulario, el sistema comprueba que tenga seleccionada alguna opción para la suscripción del servicio. Si todo ha sido correcto, se actualiza en la base de datos.
Salidas.	Se muestra un mensaje informando que la suscripción se ha realizado con éxito o en caso contrario, el proceso ha sido cancelado.

2.3.2.22.	Visualizar pistas y horarios del polideportivo.
Introducción	El usuario puede visualizar las pistas libres y ocupadas en un calendario y los horarios y instalaciones del polideportivo.
Entradas.	Desde el menú principal habrá un enlace llamado Deportes, desde el cual podremos acceder.
Proceso	Se seleccionará la pestaña “Polideportivo” dentro del menú “Deportes”.
Salidas.	El sistema buscará en la base de datos las pistas del polideportivo y mostrará en una página las pistas y otras instalaciones del polideportivo.

2.3.2.23.	Reservar una pista del polideportivo.
Introducción	El usuario podrá reservar pistas y mirar si están libres o ocupadas.
Entradas.	Clicando en la instalación a reservar desde la página que muestra el menú Polideportivo se mostrará la información de la pista y el usuario podrá realizar la reserva pulsando sobre el calendario, en una zona libre.
Proceso	El sistema buscará en la base de datos la información de la pista a reservar y mostrará un horario semanal de la pista seleccionada y si está reservada o no, en rangos de una hora.
Salidas.	Se mostrará una ventana emergente diciendo si la pista se ha podido reservar o si por el contrario la pista ya estaba reservada.

2.3.2.24.	Anular la reserva de una pista del polideportivo
Introducción	El usuario podrá anular su reserva realizada.
Entradas.	Clicando en la instalación a anular desde la página que muestra el menú Polideportivo se mostrará la información de la pista y el usuario podrá realizar la anulación pulsando sobre el calendario.
Proceso	El sistema comprobará en la base de datos si esa franja horaria había sido reservada por el usuario registrado y procederá a la anulación de la reserva si fuera posible, es decir, si el usuario es el mismo que realizó la reserva y si quedan más de dos horas para la hora reservada, en caso contrario no será posible la anulación.
Salidas.	Se mostrará una ventana emergente que indicará si se ha podido anular o no la reserva.

2.3.2.25.	Visualizar tramitaciones o gestiones
Introducción	El usuario puede visualizar cada tramitación o gestión que ha realizado y ver en qué estado se hallan.
Entradas.	Enlace a la “Oficina virtual”.
Proceso	Se obtendrán las tuplas de las tramitaciones realizadas por el usuario de la base de datos.
Salidas.	Se visualizará una lista con las tramitaciones del usuario así como su estado, su fecha de iniciación y si fue Aprobada o Denegada la fecha de aprobación o denegación.

2.3.2.26.	Visualizar en detalle una tramitación o gestión
Introducción	El usuario podrá visualizar los detalles de la tramitación.
Entradas.	Clicando sobre la propia tramitación.
Proceso	Se obtendrá de la base de datos, la información del trámite, donde se incluirán los datos de la solicitud.
Salidas.	Se muestra en una nueva página los detalles de la tramitación.

2.3.2.27.	Eliminar una tramitación o gestión
Introducción	El usuario podrá eliminar una tramitación siempre y cuando su estado no fuera Iniciado o Finalizado (Aprobado o Denegado).
Entradas.	Desde un botón en la lista de trámites.
Proceso	Se borrará de la base de datos, la tupla correspondiente al trámite seleccionado siempre que se cumplan los requisitos para ello.
Salidas.	Se muestra una ventana emergente que nos indica si estamos seguros que queremos borrar el trámite seleccionado o si por el contrario no podemos hacerlo.

2.3.2.28.	Editar una tramitación o gestión
Introducción	El usuario podrá añadir notificaciones a una tramitación siempre y cuando su estado no sea Finalizado (Aprobado o Denegado).
Entradas.	Desde un botón en la lista de trámites.
Proceso	El sistema buscará en la base de datos las notificaciones de dicha gestión y las mostrará en una página nueva, una vez modificadas insertará en la base de datos los cambios realizados.
Salidas.	Se muestra una página con las notificaciones actuales y se da la posibilidad de modificarlas o completarlas, mediante el botón guardar se permitirá tal acción.

2.3.2.29.	Iniciar una tramitación o gestión
Introducción	El usuario podrá buscar las tramitaciones disponibles mediante un formulario de búsqueda.
Entradas.	Un formulario con los criterios de búsqueda, y una vez seleccionada la tramitación se iniciará otra página con el formulario a rellenar con las notificaciones correspondientes y el botón confirmar.
Proceso	Se insertará en la base de datos la tupla correspondiente a la nueva tramitación.
Salidas.	Se informa al usuario que la tramitación se ha confirmado y está pendiente de iniciación.

2.3.2.30.	Visualizar imágenes
Introducción	El usuario puede visualizar las imágenes que han sido subidas por él o por otros usuarios.
Entradas.	Enlace a través del menú principal.
Proceso	El sistema buscará las imágenes subidas por los usuarios en la base de datos.
Salidas.	El usuario visualizará en miniatura las imágenes.

2.3.2.31.	Visualizar en detalle una imagen.
Introducción	El usuario podrá visualizar en detalle una imagen seleccionada.
Entradas.	Clicando sobre la propia imagen.
Proceso	Se obtendrá de la base de datos, la imagen en un tamaño superior, donde se incluirán datos tales como, el uploader, el nombre, la resolución...
Salidas.	Se muestra en una nueva página con la imagen y los detalles de la misma.

2.3.2.32.	Subir imagen.
Introducción	El usuario registrado podrá subir imágenes relativas a la localidad desde su equipo a la página web.
Entradas.	Una ventana emergente permitirá seleccionar las imágenes.
Proceso	El sistema hará una copia de la imagen del usuario en el servidor.
Salidas.	Se informa al usuario que la imagen ha sido subida correctamente o si por el contrario no se ha podido subir.

2.3.2.33.	Visualizar oposiciones.
Introducción	El usuario mediante el correspondiente formulario de búsqueda, tendrá la posibilidad de obtener un listado de todas las oposiciones disponibles en ese momento. El listado que visualizará el usuario, permitirá para cada oposición la opción de edición en detalle de dicha oposición, así como la posibilidad de editarla.
Entradas.	Enlace Gestión oposiciones, seleccionado por el usuario.
Proceso	El sistema obtendrá de la base de datos toda la información disponible hasta ese momento sobre las oposiciones.
Salidas.	Listado con todas las oposiciones disponibles hasta ese momento.

2.3.2.34.	Visualizar en detalle una oposición.
Introducción	Dada una oposición, el usuario podrá visualizar todos los detalles, así como la posibilidad de descargar sus bases en caso de que éstas existiesen.
Entradas.	Identificador de la oposición seleccionada por el usuario.
Proceso	El sistema buscará el evento en la base de datos y extraerá la información que contiene dicha oposición.
Salidas.	La información disponible sobre la oposición seleccionada para visualizar con detalle, así como un enlace para poder descargar las bases de dicha oposición en caso de que exista este documento.

2.3.2.35.	Cerrar sesión
Introducción	El usuario dispone de la posibilidad de cerrar la sesión activa en cualquier instante desde el área de Usuarios.
Entradas.	El usuario dispone de un enlace “Cerrar sesión”.
Proceso	El identificador de la sesión será destruido así como las variables de sesión.
Salidas.	Al cerrar la sesión, el usuario volverá a la página principal como si de un usuario anónimo se tratara.

Usuario interino.

El rol que cumple es el propio de un funcionario del ayuntamiento, es decir, dispondrá de una interfaz dedicada con el objetivo de gestionar los contenidos de una parte del portal.

2.3.2.36.	Visualizar tramitaciones pendientes de resolución.
Introducción	El usuario podrá visualizar todas las tramitaciones pendientes de resolución haciendo una búsqueda por el nombre de usuario.
Entradas.	Formulario de búsqueda con el username seleccionado por el usuario.
Proceso	El sistema mostrará de la base de datos aquellas tramitaciones cuyo estado no sea finalizado.
Salidas.	Listado con las tramitaciones pendientes de resolución del usuario previamente seleccionado. Mostrando el nombre, el estado de la tramitación y la fecha de iniciación.



2.3.2.37.	Modificar una tramitación.
Introducción	El usuario puede modificar el estado de una tramitación, entre los estados posibles, así como introducir o indexar una notificación.
Entradas.	Formulario con el nuevo valor del estado de la tramitación, así como el texto a notificar.
Proceso	El sistema obtiene las tramitaciones, el usuario realizará las modificaciones correspondientes y al clicar en aceptar se actualizará la base de datos añadiendo las modificaciones a las tuplas correspondientes.
Salidas.	Se informará mediante una ventana emergente que se han realizado las modificaciones correctamente.

2.3.2.38.	Visualizar trámites disponibles.
Introducción	El usuario puede obtener un listado de todos los trámites disponibles mediante un formulario. El listado mostrará para cada trámite sus correspondientes 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 mostrará al usuario el formulario de búsqueda. Si el usuario confirma el envío, consulta con la base de datos en función de los filtros introducidos en el formulario y obteniendo los resultados de aquellos trámites que cumplen con dichos filtros.
Salidas.	Listado de los trámites que reúnen las condiciones establecidas en el formulario de búsqueda seleccionado por el usuario.

2.3.2.39.	Visualizar en detalle un trámite.
Introducción	El usuario puede visualizar en detalle un trámite en concreto, así como descargar el impreso asociado.
Entradas.	Identificador del trámite seleccionado por el usuario.
Proceso	El sistema buscará dicho trámite en la base de datos y extraerá toda la información de dicho trámite.
Salidas.	Mostrará la información sobre el trámite seleccionado, así como un enlace al impreso para poder descargarlo en caso de que exista un impreso asociado al mismo.

2.3.2.40.	Crear un nuevo trámite.
Introducción	EL usuario dispondrá en todo momento de la opción de dar de alta un nuevo trámite, cumplimentando los campos del correspondiente formulario. También podrá anexar un impreso para posteriormente poder ser descargado.
Entradas.	Formulario cumplimentado correctamente y enviado por el usuario.
Proceso	El usuario cumplimentará el formulario de un nuevo trámite. Si el usuario confirma el envío, el sistema comprobará que el formulario ha sido cumplimentado con éxito. Además, si anexó algún documento, comprobará que haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema insertará el trámite en la base de datos.
Salidas.	Se informará al usuario que el trámite que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.41.	Editar un trámite
Introducción	El usuario podrá modificar un trámite existente, así como cambiar el impreso asociado al mismo.
Entradas.	Formulario de modificación enviado por el usuario.
Proceso	El sistema mostrará el formulario de edición y si el usuario confirma su envío, el sistema comprobará que estén todos los datos cumplimentados correctamente. Además, si anexó algún documento, comprobará que haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema insertará el trámite en la base de datos.
Salidas.	Se informará al usuario que el trámite que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.42.	Visualizar y editar noticias.
Introducción	El usuario mediante el correspondiente formulario de búsqueda, tendrá la posibilidad de obtener un listado de todas las noticias disponibles. El listado que visualizará el usuario, permitirá para cada noticia la opción de edición en detalle de dicha noticia, así como la posibilidad de editarla.
Entradas.	Formulario de búsqueda cumplimentado correctamente aplicando los correspondientes filtros de búsqueda.
Proceso	El sistema mostrará el correspondiente formulario de búsqueda. Si el usuario confirma el envío una vez cumplimentado el formulario, el sistema consultará en la base de datos las noticias que cumplen con los filtros aplicados por el usuario y devolverá aquellas noticias que cumplen con dichas condiciones.
Salidas.	Listado con las noticias que han cumplido con los filtros establecidos en el formulario de búsqueda seleccionadas por el usuario.

2.3.2.43.	Crear una noticia.
Introducción	El usuario dispondrá de la opción de dar de alta una nueva noticia mediante la cumplimentación de los campos del formulario correspondiente, así como anexar alguna fotografía si fuera necesario.
Entradas.	Formulario cumplimentado y enviado por el usuario.
Proceso	El usuario cumplimentará el formulario de una nueva noticia. Si el usuario confirma el envío, el sistema comprobará que el formulario ha sido cumplimentado con éxito. Además, si anexó alguna fotografía, comprobará que haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema insertará el trámite en la base de datos.
Salidas.	Se informará al usuario que la noticia que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.44.	Editar una noticia.
Introducción	El usuario podrá modificar una noticia existente, así como cambiar la fotografía asociado al mismo.
Entradas.	Formulario de modificación enviado por el usuario.
Proceso	El sistema mostrará el formulario de edición y si el usuario confirma su envío, el sistema comprobará que estén todos los datos cumplimentados correctamente. Además, si anexó alguna fotografía, comprobará que haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema actualizará la noticia en la base de datos.
Salidas.	Se informará al usuario que la noticia que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.45.	Crear un nuevo evento.
Introducción	El usuario dispondrá de la opción de dar de alta un nuevo evento mediante la cumplimentación de los campos del formulario correspondiente.
Entradas.	Formulario cumplimentado y enviado por el usuario.
Proceso	El usuario cumplimentará el formulario de un nuevo evento. Si el usuario confirma el envío, el sistema comprobará que el formulario ha sido cumplimentado con éxito. Si se cumplen las condiciones anteriores, el sistema insertará el evento en la base de datos.
Salidas.	Se informará al usuario que el evento que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.46.	Editar un evento.
Introducción	El usuario podrá modificar un evento existente.
Entradas.	Formulario de modificación enviado por el usuario.
Proceso	El sistema mostrará el formulario de edición y si el usuario confirma su envío, el sistema comprobará que estén todos los datos cumplimentados correctamente. Si se cumplen las condiciones anteriores, el sistema actualizará el evento en la base de datos.
Salidas.	Se informará al usuario que el evento que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.47.	Crear una nueva oposición.
Introducción	El usuario dispondrá de la opción de dar de alta una nueva oposición mediante la cumplimentación de los campos del formulario correspondiente.
Entradas.	Formulario cumplimentado y enviado por el usuario.
Proceso	El usuario cumplimentará el formulario de una nueva oposición. Si el usuario confirma el envío, el sistema comprobará que el formulario ha sido cumplimentado con éxito. Además, tendrá la posibilidad de anexar un documento con las bases de la convocatoria de dicha oposición. Si se cumplen las condiciones anteriores, el sistema insertará el evento en la base de datos.
Salidas.	Se informará al usuario que la oposición que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.48.	Editar una oposición.
Introducción	El usuario podrá modificar una oposición existente, así como la posibilidad de cambiar el documento de la base de la convocatoria de dicha oposición.
Entradas.	Formulario de modificación enviado por el usuario.
Proceso	El sistema mostrará el formulario de edición y si el usuario confirma su envío, el sistema comprobará que estén todos los datos cumplimentados correctamente. Además, si anexó algún documento de la base de la convocatoria, comprobará que haya sido recibido correctamente. Si se cumplen las condiciones anteriores, el sistema actualizará el evento en la base de datos.
Salidas.	Se informará al usuario que la oposición que ha introducido se ha realizado con éxito, o bien, que el proceso ha sido cancelado.

2.3.2.49.	Enviar boletín noticias a los subscriptores.
Introducción	Se envía a los usuarios registrados un mail con los eventos y noticias, que son introducidos semanalmente, los usuarios pueden recibir solo la información de eventos, solo las noticias o ambos a la vez
Entradas.	Ninguna.
Proceso	Se obtienen los emails de los usuarios registrados de la base de datos y se enviaran las noticias o eventos que han sido introducidos en la base de datos esa semana.
Salidas.	Aparecerá un mensaje en la pantalla del usuario interino diciéndole que ha sido enviado el boletín de noticias.

2.3.2.50.	Eliminar una imagen.
Introducción	El usuario interino tiene la posibilidad de quitar de la base de datos la imagen subida si incumple alguna norma.
Entradas.	Ninguna.
Proceso	Desde la página de visualización de imágenes se permite al usuario mediante un botón eliminar la imagen seleccionada.
Salidas.	Aparecerá un mensaje en la pantalla del usuario interino diciéndole si está seguro de querer eliminar la imagen.

2.3.2.51.	Crear instalación.
Introducción	El interino puede dar de alta en la base de datos nuevas instalaciones que vayan construyéndose en el polideportivo.
Entradas.	Formulario cumplimentado y enviado por el usuario.
Proceso	El sistema muestra el formulario de creación de nueva instalación. Si el usuario hace clic sobre el botón guardar, el sistema comprobará que estén todos los campos obligatorios rellenados. Si además se puso una imagen de referencia, el sistema comprueba que haya sido recibida correctamente. Si todo es correcto el sistema añadirá la nueva instalación a la base de datos.
Salidas.	Se informa al usuario que la instalación fue introducida con éxito, o bien que no se ha podido realizar.

Usuario administrador.

2.3.2.52.	Visualizar usuarios registrados en el portal.
Introducción	El administrador puede obtener el listado de los usuarios registrados en el portal, ordenados alfabéticamente por el apellido, mostrando los siguientes datos: los apellidos, el nombre, la fecha de registro, el Username y su categoría, así como dos botones de eliminación y otro de visualizar en detalle.
Entradas.	Podrá acceder desde el menú principal en la opción Usuarios, solo visible para administradores.
Proceso	El sistema obtiene la información de la base de datos.
Salidas.	Listado ordenado alfabéticamente por apellido de todos los usuarios junto con las opciones disponibles en dos botones, Eliminar y Visualizar.

2.3.2.53.	Visualizar en detalle un usuario.
Introducción	El administrador podrá visualizar los detalles del usuario seleccionado a excepción de su clave de acceso.
Entradas.	Clicando sobre el botón visualizar.
Proceso	El sistema busca al usuario en la base de datos y extrae su información.
Salidas.	Se abre una página nueva con toda la información sobre el usuario a excepción de su contraseña.

2.3.2.54.	Crear un nuevo usuario.
Introducción	El administrador puede dar de alta a nuevos usuarios, igual que el registro de cualquier usuario pero además puede definir el tipo de usuario del que se trata, Administrador, Interino o Registrado.
Entradas.	Formulario de registro.
Proceso	El sistema mostrará el formulario de registro a cumplimentar y una vez completado correctamente permitirá la inserción en la base de datos mediante un botón de guardar.
Salidas.	Se informa mediante una ventana emergente si el usuario fue introducido correctamente o que no se puede realizar.

2.3.2.55.	Eliminar usuario.
Introducción	El administrador podrá eliminar a un usuario de la base de datos.
Entradas.	Clicando sobre el botón eliminar.
Proceso	El sistema busca al usuario en la base de datos y elimina la tupla correspondiente si es posible.
Salidas.	Se muestra una ventana emergente indicando si se está seguro de la acción, si se clica sobre el si se mostrará otra que nos indicará que se ha realizado correctamente.

2.3.2.56.	Eliminar un trámite.
Introducción	El administrador podrá eliminar un trámite de la base de datos.
Entradas.	En la opción de visualizar tramites, tendrá un botón para tales efectos.
Proceso	El sistema mostrará los trámites filtrados mediante un formulario de búsqueda.
Salidas.	Se muestra una ventana emergente indicando si se está seguro de la acción, si se clica sobre el si se mostrará otra que nos indicará que se ha realizado correctamente.

2.3.2.57.	Eliminar una noticia.
Introducción	El administrador puede eliminar noticias de la base de datos.
Entradas.	Donde se visualizan las noticias, un botón para tales efectos.
Proceso	El sistema mostrará las noticias como a cualquier usuario, pero habrá un botón para poder eliminar la noticia.
Salidas.	Se muestra una ventana emergente indicando si se está seguro de la acción, si se clica sobre el si se mostrará otra que nos indicará que se ha realizado correctamente.

2.3.2.58.	Eliminar un evento.
Introducción	El administrador puede eliminar eventos de la base de datos.
Entradas.	Donde se visualizan los eventos, un botón para tales efectos.
Proceso	El sistema mostrará los eventos como a cualquier usuario, pero habrá un botón para poder eliminar el evento.
Salidas.	Se muestra una ventana emergente indicando si se está seguro de la acción, si se clica sobre el si se mostrará otra que nos indicará que se ha realizado correctamente.

2.3.2.59.	Eliminar una oposición.
Introducción	El administrador puede eliminar oposiciones de la base de datos.
Entradas.	Donde se visualizan las oposiciones, un botón para tales efectos.
Proceso	El sistema mostrará las oposiciones como a cualquier usuario, pero habrá un botón para poder eliminar la noticia.
Salidas.	Se muestra una ventana emergente indicando si se está seguro de la acción, si se clica sobre el si se mostrará otra que nos indicará que se ha realizado correctamente.

2.3.2.60.	Eliminar instalación.
Introducción	El administrador puede eliminar de la base de datos instalaciones antiguas o que se encuentren en proceso de remodelación en el polideportivo.
Entradas.	Botón que se encuentra en la opción de visualizar que permite al administrador eliminar la instalación.
Proceso	El sistema muestra el listado de las instalaciones. Si el usuario hace clic sobre el botón eliminar, el sistema comprobará que es posible eliminar. Si todo es correcto el sistema eliminará la instalación a la base de datos.
Salidas.	Se informa al usuario que la instalación fue eliminada con éxito, o bien que no se ha podido realizar.

Todos los usuarios con un rango superior tendrán disponibles las funciones de un usuario de rango inferior, ordenándose los rangos en el siguiente orden de mayor a menor:

Usuario administrador > Usuario interino > Usuario registrado > Usuario anónimo.

3. Análisis.

Después de haber descrito las funcionalidades del sistema en la especificación de requisitos, vamos a profundizar con alguna metodología que nos permita detectar aquellos apartados generales que guardan una relación directa con los requisitos descritos en el apartado anterior. La metodología que usaremos debe ser una metodología capaz de conseguir mantener la independencia con los recursos disponibles para su implementación.

Para ello utilizaremos RUP (Rational Unified Process) que es un proceso de desarrollo de software y que se apoya en modelos UML (Unified Modeling Language) que ayudan a describir la función, la arquitectura o el diseño del software. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

3.1. Introducción.

UML es el lenguaje de modelado de sistemas software más utilizado en la actualidad, que sirve para poder describir procesos basándose en elementos gráficos. Permite visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo, incluyendo aspectos conceptuales, como por ejemplo, procesos de negocio y funciones del sistema, expresiones del lenguaje de programación, esquemas de bases de datos y componentes reutilizables.

UML dispone de diversos tipos de diagramas para identificar entidades del sistema que vamos a modelar. En este proyecto, nos vamos a centrar en los Diagramas de Casos de Uso así como también en el Diagrama de Clases y en los Diagramas de Secuencia.

3.2. Casos de uso.

Los Diagramas de Casos de Uso nos muestran cómo debe reaccionar el sistema con un usuario, o bien con otro sistema para poder conseguir su objetivo. Se trata de representar la descripción escrita del comportamiento del sistema al afrontar

una tarea de negocio o un requisito de negocio, esta descripción se centra en el valor suministrado por el sistema a entidades externas tales como usuarios humanos u otros sistemas.

El uso de diagramas de casos de uso está muy extendido, ya que promueve una imagen fácil y sencilla del comportamiento del sistema, un entendimiento común entre el cliente/usuario/propietario y el equipo de desarrollo.

A continuación, en la Ilustración 1 se muestran los distintos actores que intervienen en el sistema, organizados jerárquicamente, en el podemos observar la especialización de algunos usuarios, lo que se denomina generalización, es decir, identificar elementos en común entre conceptos y definir las relaciones de una superclase (concepto general) y subclasses (concepto especializado).

Tendremos los siguientes actores en el sistema:

Usuario Anónimo será cualquier usuario que entre en la página web.

Usuario Registrado será aquel que ha rellenado el formulario para registrarse y que mediante su username y contraseña accede al portal a contenidos exclusivos.

Usuario Interino es el que realiza las tareas de documentación, colgar noticias, crear eventos, subir oposiciones, controlar el buen uso del portal....

El Usuario Administrador tendrá los mismos privilegios que el anterior y además podrá eliminar eventos, noticias, etc, y crear usuarios.

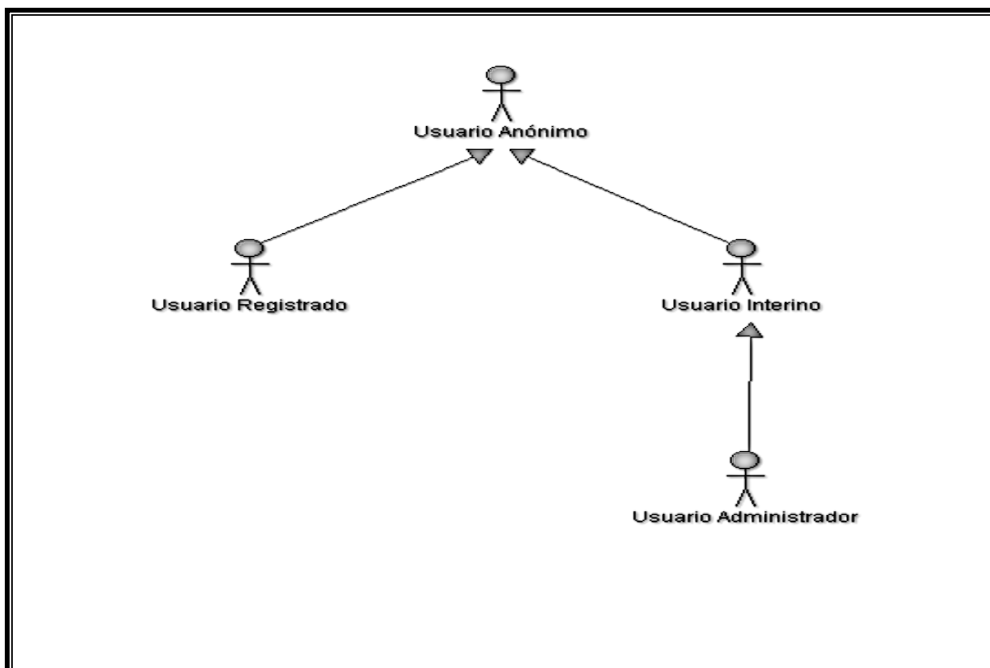


Fig. 4: Actores del sistema.

Los siguientes diagramas de casos de uso contienen, como ya se ha especificado anteriormente, las funciones que puede realizar cada “tipo” de usuario en el sistema. Las funciones contienen relaciones entre ellas expresadas mediante inclusión (<<include>>), que representa que un caso de uso puede incluir otro, o extensión (<<extend>>), que representa que un caso de uso dado puede extender otro, es decir, que a partir de una función puede desencadenarse otra en función de la condición que se cumpla. Las condiciones en nuestro caso, aparecerán escritas debajo de la etiqueta <<extend>>.

A continuación, en la Fig. 5, se muestra el Diagrama de Casos de uso del Usuario Anónimo. Entendemos que la función autenticarse debe estar asociada al Usuario Anónimo puesto que, un usuario hasta que no se autentique solo puede realizar funciones de este tipo de usuarios. Una vez realizado esto, si pasaría a ser un Usuario Registrado.

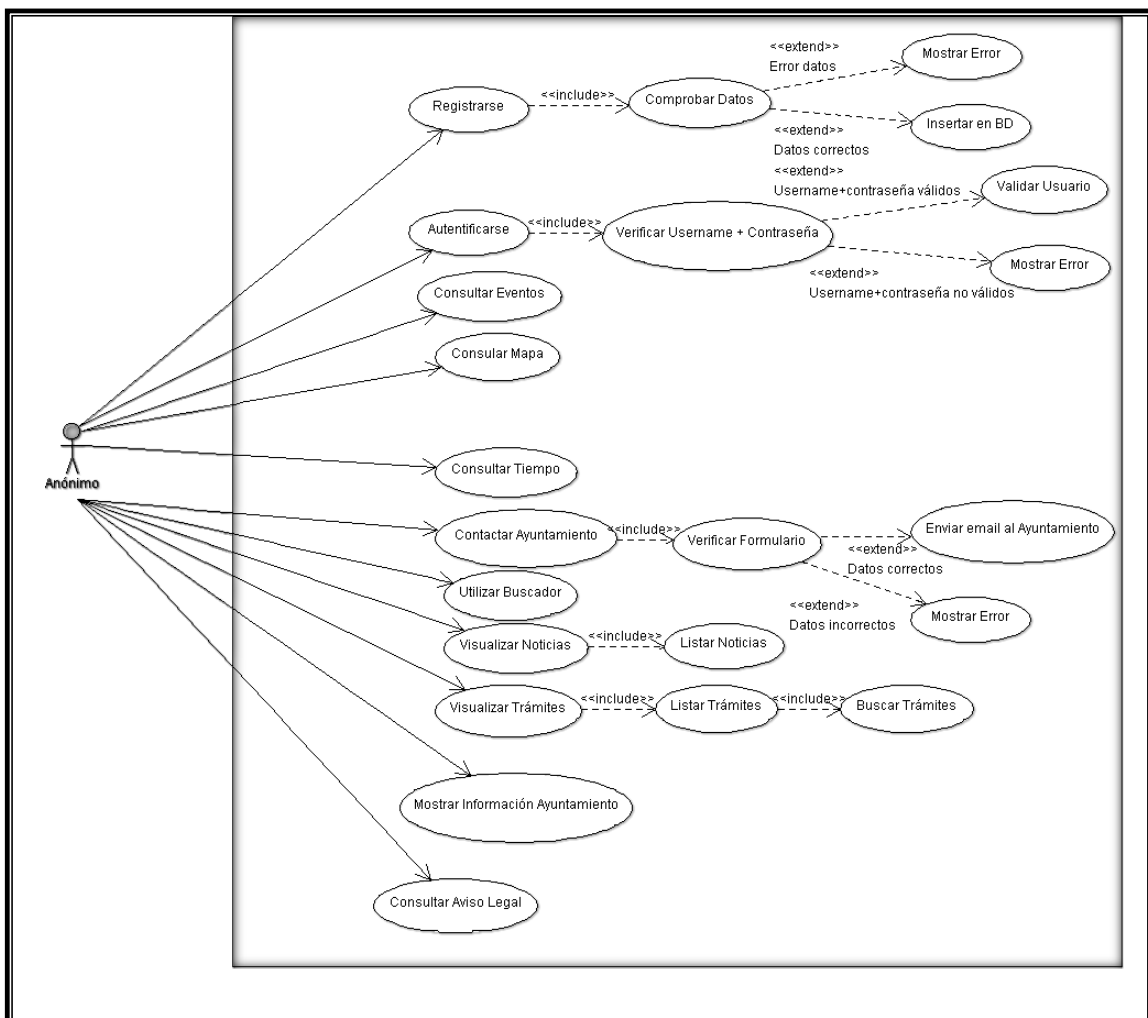


Fig. 5: Casos de uso del Usuario Anónimo.

En la siguiente figura, Fig. 6. mostramos los casos de uso del Usuario Registrado.



Fig. 6: Casos de uso del Usuario Registrado.

En la Fig. 7, tenemos representados los casos de uso del Usuario Interino. Omitiremos en la representación los casos “Gestionar Tramitaciones”, “Gestionar Noticias” y “Gestionar Oposición”, dado que siguen el mismo patrón que “Gestionar Eventos”, que si ha sido representado.

Para las opciones “Editar Evento” y “Crear Evento” (extensible a noticia, tramite y oposición), será necesario verificar (“Verificar Evento”) que todos los datos han sido completados correctamente, si no fuera el caso, se realizaría la acción de “Mostrar Error”.

El Usuario interino tendrá la capacidad de poder añadir una instalación deportiva a la base de datos (“Crear instalación”). Esta incluiría una imagen representativa.



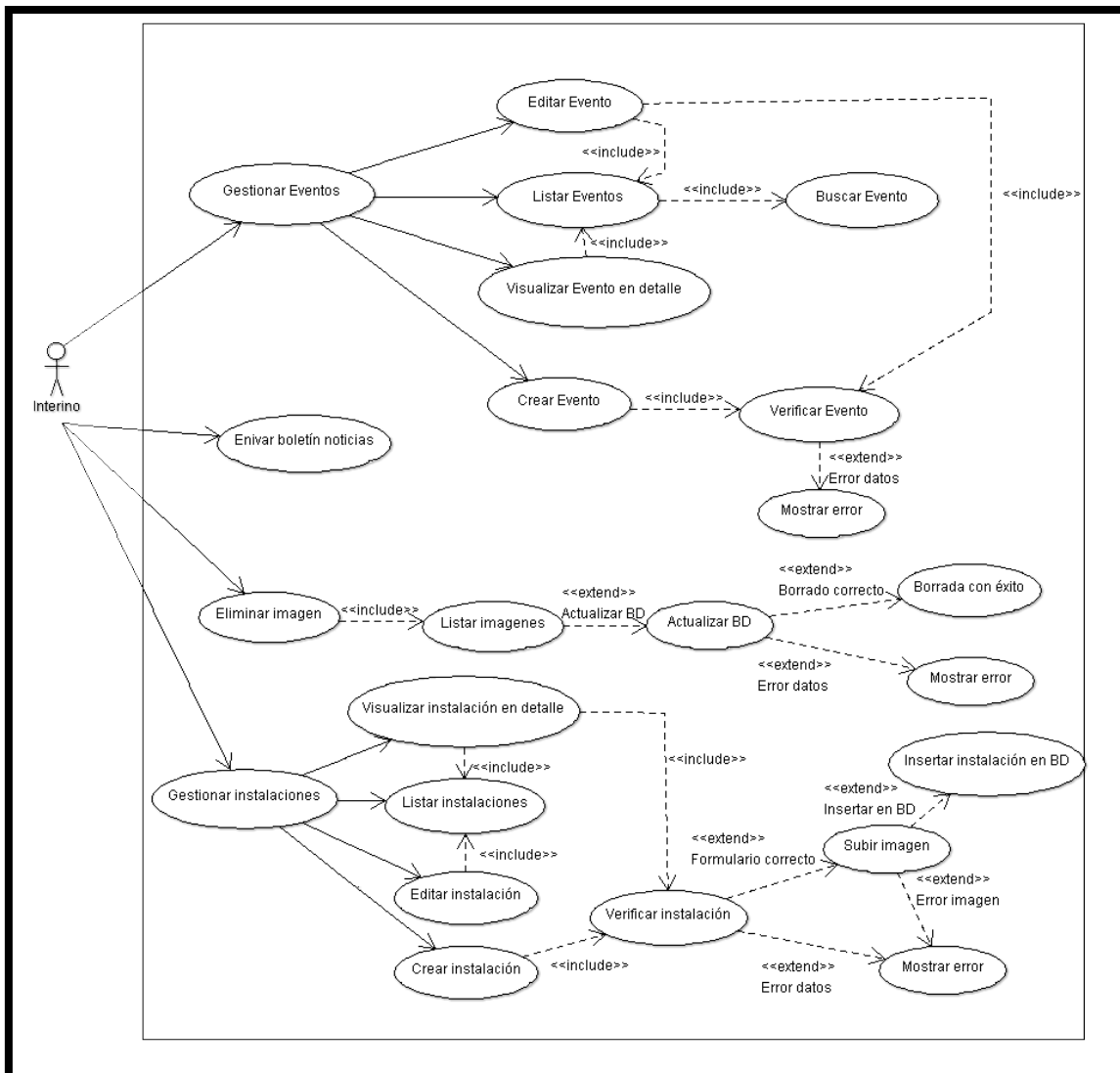


Fig. 7: Casos de uso del Usuario Interino.

A continuación, tenemos los Casos de Uso para el Usuario Administrador (Fig. 8.) que a diferencia del Usuario Interino, tiene la posibilidad de eliminar “objetos” de la base de datos, así como gestionar los usuarios del portal.

Igual que en el caso anterior, omitiremos los casos “Eliminar Noticia”, “Eliminar Tramite”, “Eliminar Oposición” y también “Eliminar Instalación”.

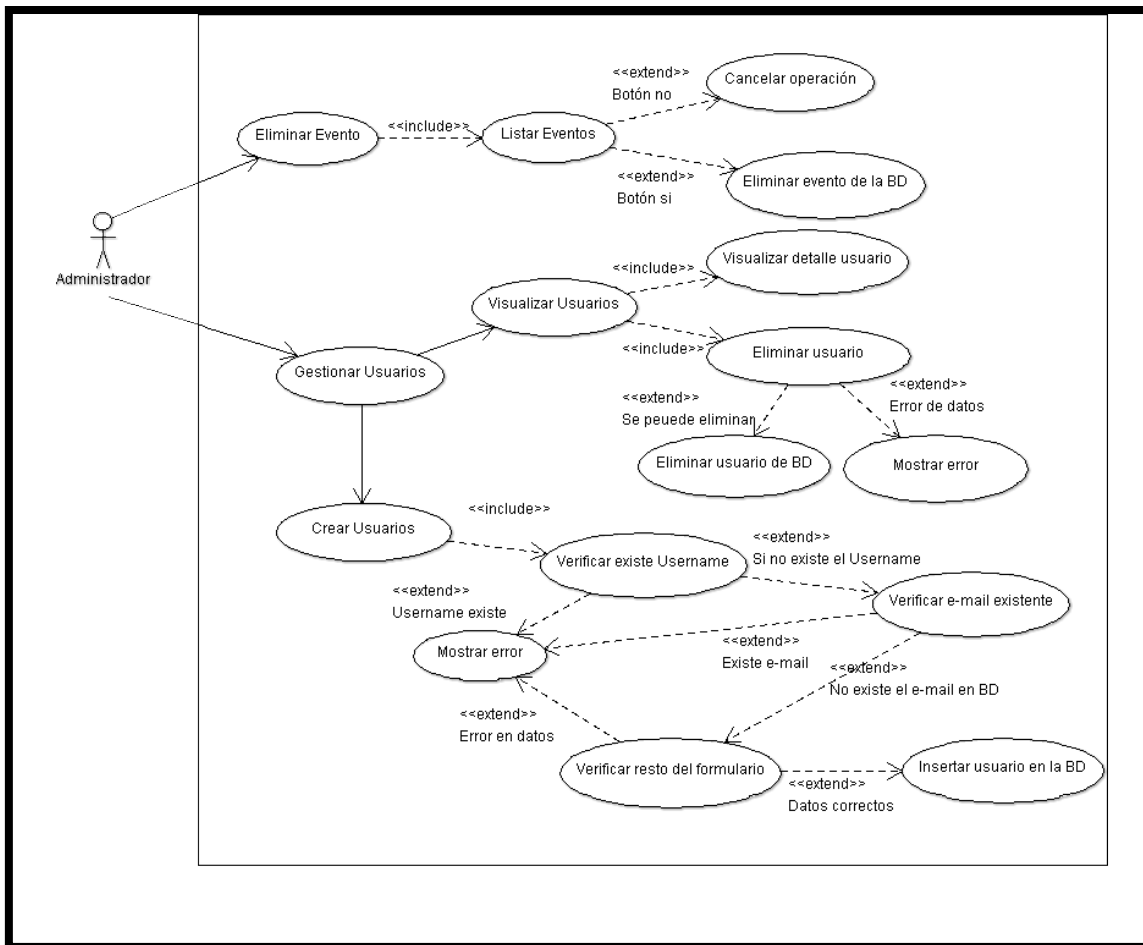


Fig. 8: Casos de uso para el Usuario Administrador

3.3. Diagramas de clase.

Los Diagramas de Clases describen la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Una clase define las propiedades y operaciones de una serie de objetos que pertenecen a esa misma clase, de forma que todos los objetos de una clase determinada tendrán por definición los mismos atributos y métodos, ya que estos son definidos en la clase a la que pertenecen. En nuestro caso, el sistema se sustenta en los siguientes tres tipos de clases: servicio, información y usuario.

En los Diagramas de Clases podemos ver también sub-clases o clases hijas más especializadas que pueden incorporar nuevos atributos y métodos. Esto se



denomina herencia. Estas sub-classes, pueden tener nuevos atributos y métodos además de mantener los que heredan de la clase superior o clase padre. Por ejemplo, la clase CONTACTO que hereda de la clase SERVICIO, tendrá sus atributos pero además deberá tener los siguientes atributos, *nom_usuario*, *mail* y *descripción*, y como también podemos observar la diferencia con sus clases *hermanas*, por ejemplo con BOLETÍN NOTICIAS, esta clase, “solo” tiene un atributo *tipo*, pero también los de la clase padre y también contiene objetos de tipo NOTICIA, cosa que CONTACTO no, y ambas dos son clases que tienen como clase padre SERVICIO.

La Fig. 9. muestra el Diagrama de Clases de nuestro sistema, donde se muestran las relaciones y las cardinalidades de las distintas clases (se omiten las de relación 1 a 1), también sus atributos propios. Hemos omitido los distintos métodos hasta tener el sistema totalmente finalizado.

A continuación, vamos a argumentar el por qué de la construcción de este Diagrama de clases. Como hemos comentado anteriormente, el sistema se basa en tres clases principales de las cuales heredarán el resto de clases, esta son:

- ***Servicio:*** esta clase engloba todas aquellos apartados del portal que son utilizados de alguna manera por cualquiera de los usuarios del mismo, es decir aquellos que requerirán la participación de algún usuario, son los siguientes: *Trámite, Inst. Deportivas, Boletín noticias, Subir Imag., Contacto y Oficina Virt..*
- ***Información:*** en esta clase se organizan todas aquellas clases en las que el usuario solo puede ver información pero no interactuar con ella o hacerlo mínimamente, esto incluye *Callejero, Tiempo, Noticia, Evento, Oposición y Ayto. y municipio.*
- ***Usuario:*** esta clase contiene la jerarquía de clases de usuario ordenadas por su comportamiento y nivel de privilegios de cara al portal. Estará dividido en dos subclases, *Anónimo* y *Autenticado*, esto es, los usuarios que previamente han introducido los datos de registro y se hallan en la base de datos y el resto de usuario. Los usuarios *Autenticado* se dividirán en *Registrado* y en *P. Ayuntamiento*, los usuarios registrados serán aquellos que tienen los privilegios de interactuar con el portal, pero no podrán crear ni modificar nada que no pertenezca a sí mismos, mientras que *P. Ayuntamiento* (personal ayuntamiento) englobará a los trabajadores de tipo *Interino*, es decir, aquellos que modifican la información del portal regularmente y a los de tipo *Administrador*, que son los que manejan en su totalidad el portal, estos últimos pueden eliminar y crear nuevos usuarios y cambiar los privilegios de los ya creados. *Pref. Usuario* contiene las preferencias de un usuario.

Con respecto a las relaciones, podemos observar muchas con la etiqueta “administra”, casi todas ellas excepto la que va de *Administrador* a *Usuario*, ya que este es el único que puede “modificar” usuarios, el resto de etiquetas “administra” van a *P. Ayuntamiento*, puesto que el personal del ayuntamiento son los que modificarán los objetos de las otras clases.

Al usuario *Registrado* llegan tres relaciones, “tramita”, “sube” y “pertenece”.

- La relación “tramita” es la que existe entre este tipo de usuario y la clase *Oficina virt.* Esta relación describe la interacción del usuario a la hora de pedir trámites.
- La relación “sube” relaciona al usuario con la subida de imágenes al portal, es decir, con la clase *Subir imag.*
- Y la relación “pertenece” describe a que usuario pertenecen ciertas *Pref. Usuario* (preferencias del usuario).

Tenemos una clase asociación llamada *Reserva* que se crea de la asociación entre los usuarios autenticados y las instalaciones deportivas, de manera que almacenará cuando cierto usuario autenticado hace una reserva de una instalación deportiva.

Por último, cabe también reseñar que un boletín de noticias (*Boletín noticias*) contiene noticias (*Noticia*), formando la relación “contiene”. Y que las clases *Inst. Deportivas*, *Subir imag.*, *Oposición* y *Trámite* tienen una clase agregada mediante una relación de agregación, es decir, que cuando se cree un objeto de un tipo de estos también se creará un objeto de los tipos *Imagen instalación*, *Imagen*, *Base* y *Impreso* respectivamente.

3.4. Diagramas de secuencia.

Los Diagramas de Secuencia se utilizan frecuentemente ya que modelan la interacción entre objetos del sistema.

Este tipo de diagramas, muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

También cabe destacar que estos diagramas muestran los mensajes en orden cronológico según van sucediéndose, desde arriba hacia abajo, a través de líneas discontinuas verticales, las cuales están asociadas a algún objeto en concreto.

Solo hemos construido algunos de estos diagramas a modo de ejemplo, teniendo en cuenta aquellos que son más importantes para el sistema y evitando la reiteración puesto que algunos serán básicamente iguales.



En este caso los usuarios han sido representados como clases.

3.4.1. Escenario Autenticarse.

El usuario anónimo como ya hemos comentado anteriormente y se refleja en su diagrama de casos de uso, puede autenticarse en el portal. A continuación se muestran los pasos que se realizan hasta que el usuario cambia de anónimo a autenticado.

1. El usuario introduce en un pequeño formulario su Username y contraseña y lo envía.
2. El portal tras recibir el formulario, comprueba la existencia de la información del usuario en la base de datos de usuarios.
3. Si la información no se encuentra en la base de datos, se informará del error al usuario.

Si la información es válida, automáticamente el usuario cambiará de *Anónimo* a *Autenticado*, mostrándose la nueva página en el portal.

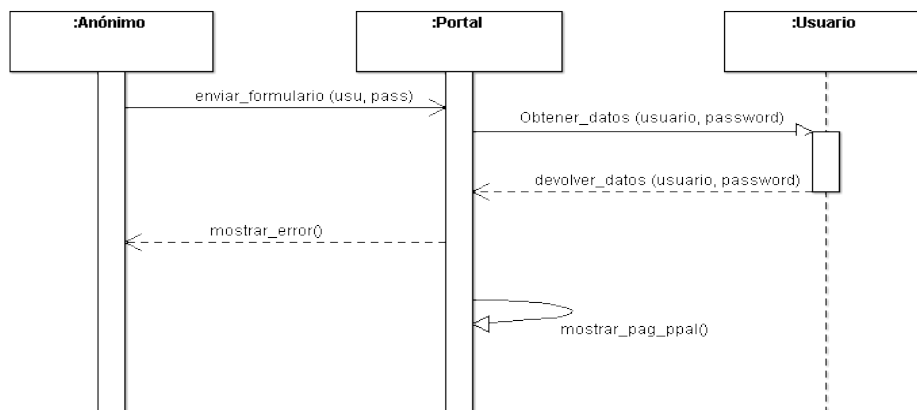


Fig. 10: Diagrama de secuencia Autenticarse.

3.4.2. Escenario Recuperar Contraseña.

En el siguiente diagrama mostramos la acción a realizar por un usuario que se ha olvidado de su contraseña. Para poder realizarlo deberá introducir su Username y email.

1. El usuario envía mediante un formulario su nombre de usuario y email.
2. El portal envía a la base de datos las entradas que acaba de escribir el usuario.
3. Se comprueba que los datos son correctos si se encuentran en la base de datos. En caso contrario, se enviará al usuario un mensaje de error.
4. Si los datos son correctos el portal generará automáticamente una nueva contraseña
5. La nueva contraseña es enviada al email proporcionado por el usuario.

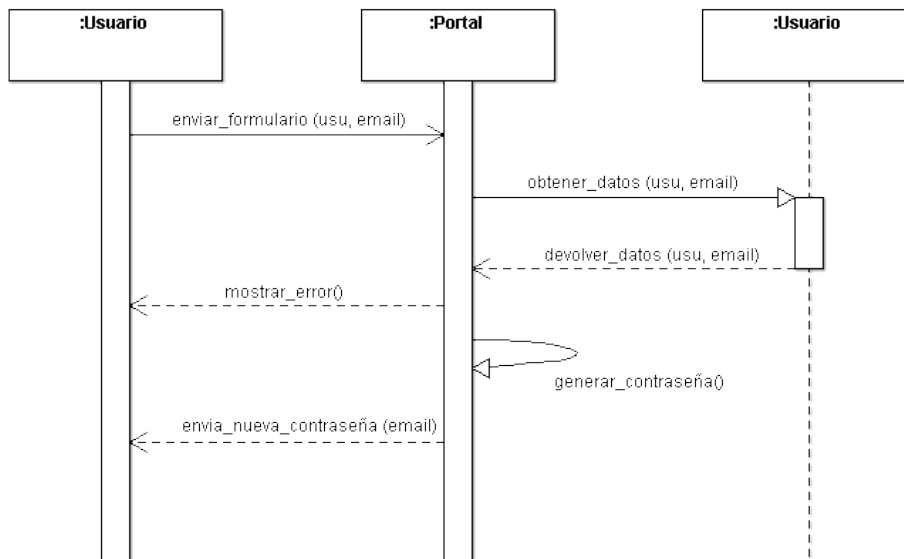


Fig. 11: Diagrama de secuencia Recuperar contraseña.

3.4.3. Escenario Buscar Oposición.

El usuario mediante el uso de un formulario indicará las claves de la oposición que desea buscar.

1. El usuario introduce en un formulario los datos de la oposición que busca.
2. El portal envía a la base de datos los datos recibidos.
3. Mediante un bucle se van recibiendo todas las oposiciones que concuerden con los datos enviados por el usuario.
4. El portal da formato a los datos para que puedan ser visualizados y los devuelve al usuario.

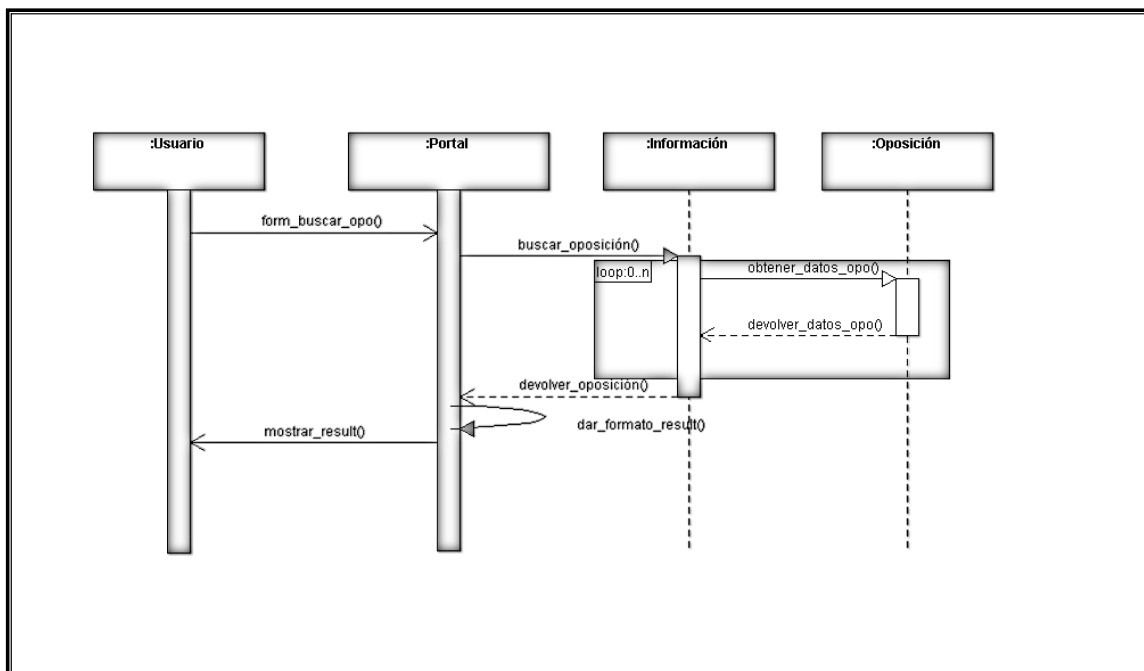


Fig. 12: Diagrama Buscar oposición.

3.4.4. Escenario Subir Imagen.

Aquí se muestran los pasos que sigue un usuario registrado para subir una imagen al portal.

1. El usuario *Registrado* hará clic en el botón subir imagen.
2. Se le muestra una ventana emergente donde debe seleccionar la imagen a subir.
3. El usuario debe escribir un Nombre y una Descripción y enviarlo al portal.
4. El portal comprobará que no exista ninguna imagen con el mismo identificador o nombre en la base de datos de imagen.
5. Envía error al usuario en el caso que existan los datos ya en la base de datos.
6. En caso contrario, el portal insertará la imagen en la base de datos y devolverá al usuario un mensaje comunicándole que la imagen se ha subido correctamente.

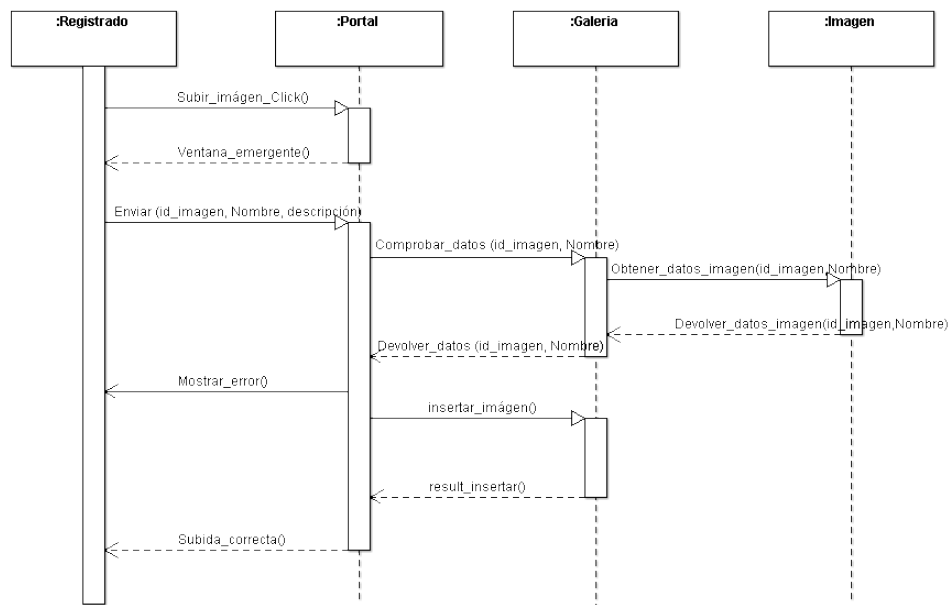


Fig. 13: Diagrama de secuencia Subir imagen.

3.4.5. Escenario Eliminar Evento.

A continuación, se muestra la secuencia que se realiza para eliminar un evento. Se ha omitido la visualización de eventos, y se ha empezado a partir de donde el Administrador pulsa el botón eliminar.

1. El administrador desde un botón alojado en el listado de eventos mediante el uso del ratón, hace clic en dicho botón.
2. El portal intentará realizar el borrado de la base de datos.
3. En caso de que no se pudiese realizar, se mostraría un error o en caso contrario, se informa que se ha realizado correctamente.

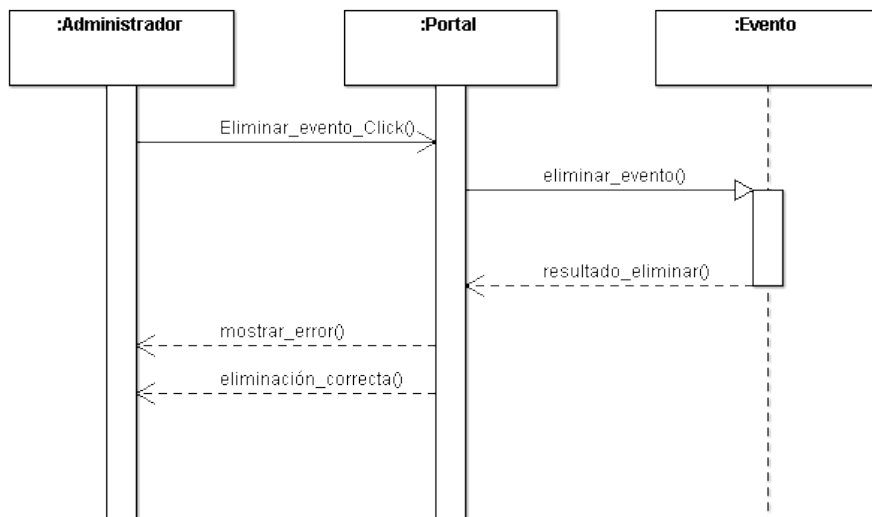


Fig. 14: Diagrama de secuencia Eliminar evento

3.4.6. Escenario Crear Usuario.

Como ya se ha comentado anteriormente, el usuario *Administrador* puede crear nuevos usuarios. A continuación, se muestra la secuencia de cómo se realiza:

1. El administrador rellena el formulario con los datos del nuevo usuario y lo envía al portal.
2. El portal comprobará que en la base de datos no existe ningún usuario con el Username y email proporcionado.
3. Si existiera, se mostrará un error.
4. El portal verificará que el resto de datos son correctos, por ejemplo que la fecha de nacimiento sea con el formato dd/mm/aaaa.
5. Si no fuera correcto, se mostrará el error.
6. Se insertará el nuevo usuario en la base de datos.
7. Si es posible, se muestra un mensaje comunicándolo, en caso contrario, se mostrará un mensaje de error.

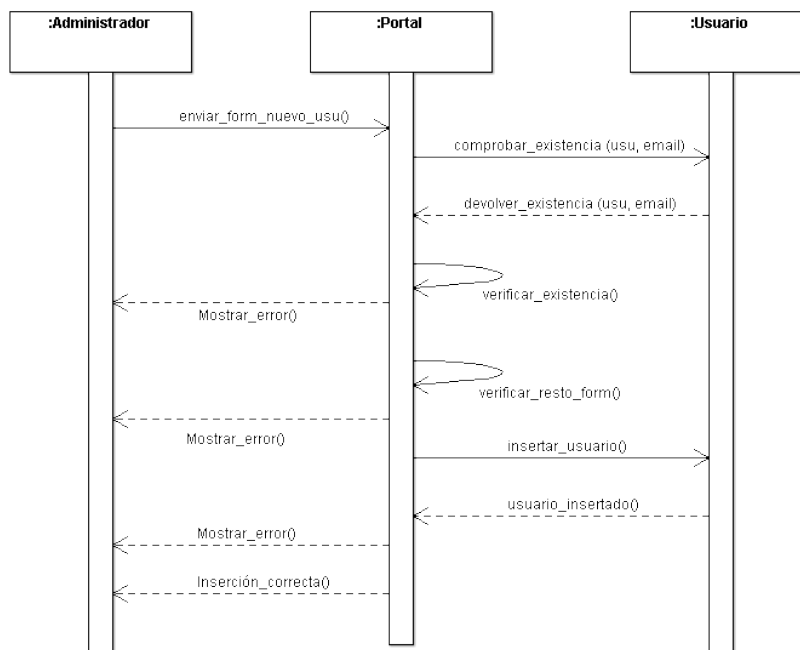


Fig. 15: Diagrama de secuencia Crear usuario.



4. Diseño.

En este punto vamos a especificar con más detalle el diseño que utilizaremos para realizar el portal. Vamos a usar la arquitectura de tres niveles que se describe a continuación.

4.1. Introducción.

El modelo de 3 niveles, así como cualquier arquitectura multinivel, tiene una ventaja muy importante que la diferencia de cualquiera de las otras arquitecturas de programación, y es que permite realizar cambios únicamente en el nivel que sea necesario, sin la necesidad de modificar los otros 2. Otra ventaja importante es la abstracción, es decir, cada grupo de trabajo trabajará con un nivel aislado de los otros, únicamente será necesario conocer la API (interfaz de programación de aplicaciones) que existe entre niveles.

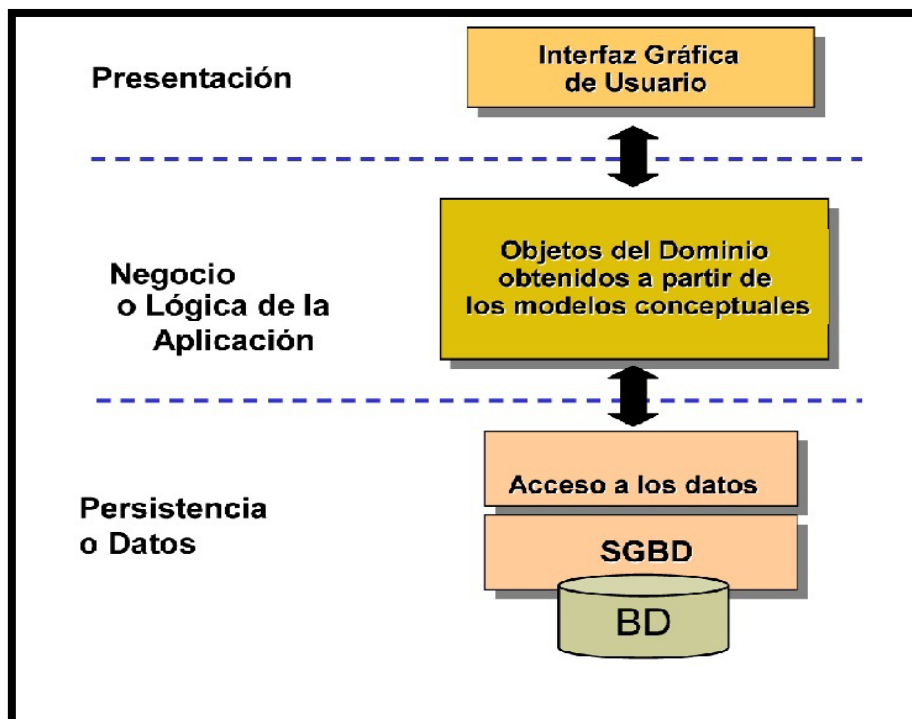


Fig. 16: Arquitectura de 3 capas

Ahora, describiremos brevemente cada uno de los niveles que conforman esta arquitectura:

- **Nivel de Presentación:** este nivel está formado por todo aquello que permite al usuario tratar con el sistema, es decir, todo lo que el usuario común podrá ver, páginas, mensajes, formularios..., le comunica la información y captura la información del usuario.

Una de sus características principales es que debe ser “amigable” (entendible y fácil de usar). Este nivel se relaciona únicamente con el nivel de negocio.

- **Nivel de Negocio:** este nivel contiene las funciones y métodos donde esta implementado el comportamiento de las clases. También es llamado muchas veces nivel de lógica puesto que es el encargado de establecer las reglas que deben cumplirse para el funcionamiento correcto del sistema. Se trata del nivel intermedio, es el que comunica los otros dos niveles.
- **Nivel de Datos:** También llamado nivel de almacenamiento, es el que contiene la base de datos y el sistema que lo gestiona. Recibe de la capa intermedia (Nivel de Negocio) las solicitudes de los datos que necesite el sistema.

4.2. Nivel de presentación.

Como hemos comentado anteriormente, la interfaz del portal debe ser intuitiva y amigable, es decir, que cualquier usuario neófito consiga manejarse con cierta facilidad y consiga localizar fácilmente y en el menor tiempo posible los elementos que tiene delante de él.

Todas las páginas contendrán un banner inferior donde tendremos el Copyright y el aviso legal, así como dos secciones superiores a izquierda y derecha donde tendremos fecha e idiomas.

En la Figura 16 tenemos la presentación de cómo será el aspecto de la página principal. Debe contener las secciones que sean indispensables. Tenemos un banner superior que contendrá el título, es decir, en este caso el nombre del pueblo, una zona principal que contiene los contenidos principales, en el lateral derecho tendremos cosas tales como menús de navegación, calendario de eventos, usuarios recientemente registrado y otras secciones.



Fig. 17: Formato de la Página principal.

En la Figura 17 observamos representado como será el resto de páginas del portal, para cualquier tipo de usuarios. Su aspecto es similar al de la anterior, pero esta menos sobrecargada, ya que hay elementos que sólo son necesarios en la página principal. La zona central, será más grande puesto que estará ocupado por la información que el usuario ha requerido. En la parte derecha tendremos un menú con las opciones del usuario, que variarán para cada usuario.



Fig. 18: Presentación resto de páginas.

4.3. Nivel de negocio.

El nivel de negocio está formado por los módulos que utilizará el portal y que no serán visibles para el usuario común. Este nivel realiza también el papel de nexo entre los otros dos niveles del sistema, obtiene los datos del nivel inferior y les da formato para que sean visibles en el nivel superior, así como la opción inversa, obtiene información del nivel de interfaz y si fuese necesario la inserta, siempre que cumpla las restricciones, en el nivel de datos.

El papel de enlace de este nivel es el que permite la independencia de los niveles, puesto que permite efectuar cambios en los otros niveles, sin que haya que realizar excesivas modificaciones en los 2 restantes.

Podemos observar en los diagramas de casos de uso de la fase de análisis, la separación que existe entre los procesos dependientes de la interfaz (Nivel de presentación) y los que se encargan de procesar y darle formato a la información para realizar una acción concreta (Nivel de Negocio).

A continuación, en la Fig. 18 se muestra el diagrama de tareas donde vemos representados los procesos externos como puntos de entrada de la información a la aplicación de nuestro portal, así como también quedan representados los conjuntos de procesos internos que se desencadenaran respondiendo a la interacción de los procesos externos.

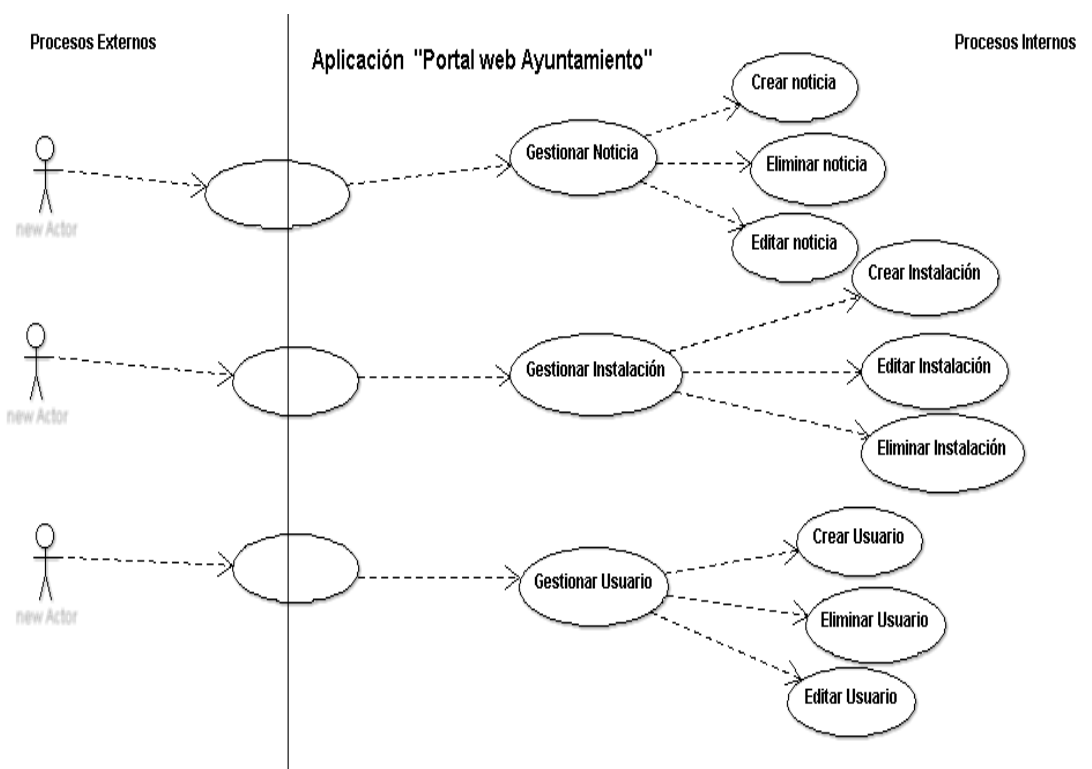


Fig. 19: Diagrama de tareas

4.4. Nivel de datos o almacenamiento.

El nivel de datos o almacenamiento es el último nivel de la arquitectura de 3 niveles y contendrá los datos que el sistema utilizará. Está relacionado únicamente con el nivel superior (Negocio), el cual accederá a los datos mediante un sistema de gestión de bases de datos.

4.4.1. Diagrama entidad – relación.

A continuación vamos a explicar y mostrar el Diagrama Entidad-Relación, el cual está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos. Ahora pasaremos a explicar los componentes de este diagrama:

- Entidades: las entidades se representan mediante un rectángulo y hace referencia a un objeto del mundo real que es distinguible del resto.
- Relación: se representa mediante un rombo en su interior con un verbo. Este rombo se debe unir mediante líneas con las entidades que relaciona.
- Atributos: se representa mediante una elipse etiquetada mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se subraya dicha etiqueta.

En la Figura 19 tenemos la representación del modelo Entidad-Relación que vamos a seguir en la creación de los datos de nuestro portal.

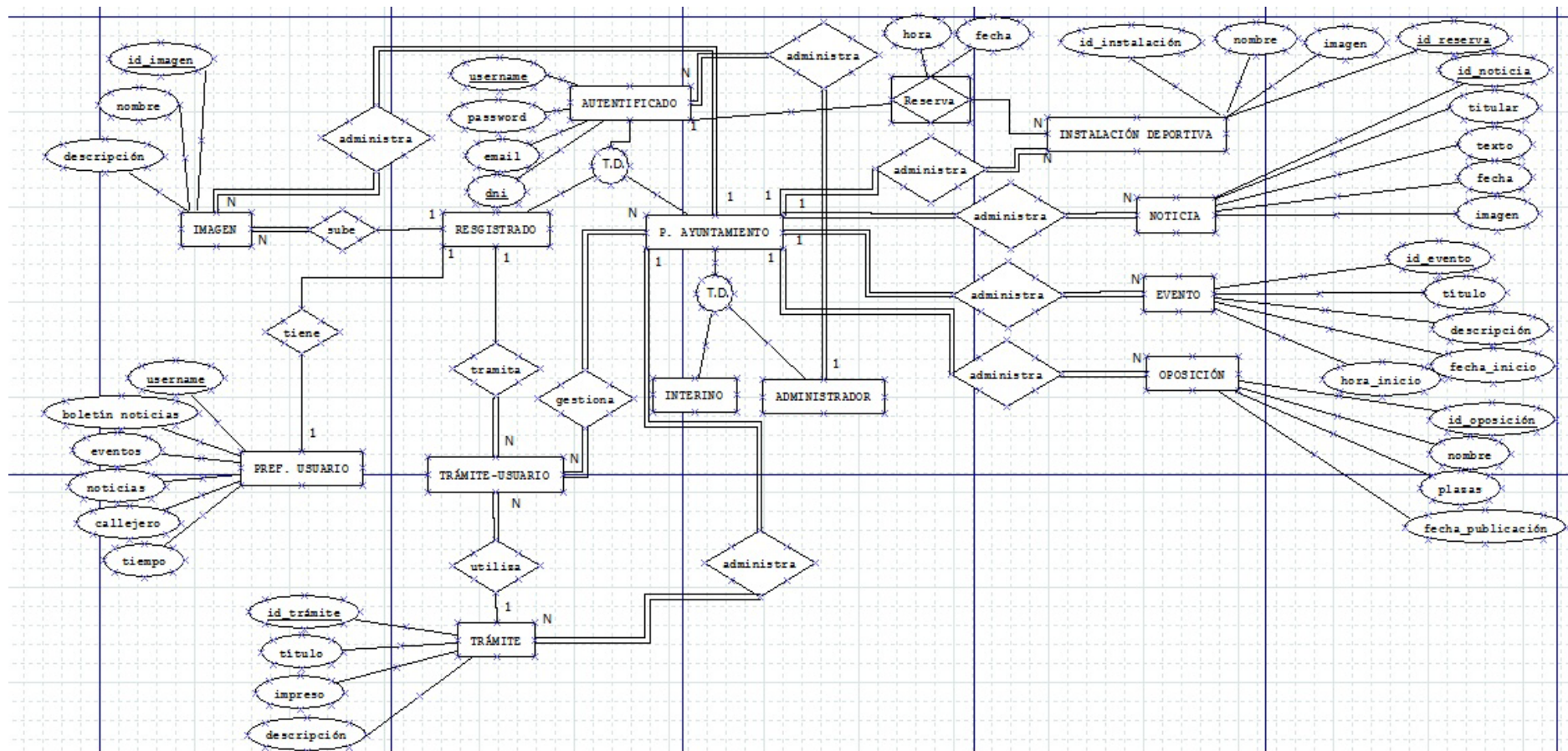


Fig. 20: Diagrama Entidad - Relación.

4.4.2.Diseño lógico.

En este punto tenemos la estructura de la base de datos representada en lenguaje relacional.

AUTENTIFICADO: {username, password, email, nombre, apellido1, apellido2, sexo, dni, telefono, fecha_nacimiento, fecha_registro}

C.P.: {username, dni}

VNN: {password, email, nombre, apellido1, apellido2, sexo, dni, telefono, fecha_nacimiento, fecha_registro}

TRÁMITE:{id_trámite, título, impreso, descripción, importe, lugar_presentación}

C.P.: {id_trámite}

VNN: {título, impreso, descripción, importe, lugar_presentación}

C. AJ: {id_trámite} → TRÁMITE_USUARIO

TRÁMITE-USUARIO: {id_trámite, username}

C.P.: {id_trámite, username}

C.AJ: {id_trámite} → TRÁMITE

C.AJ: {username} → AUTENTIFICADO

INSTALACIONES DEPORTIVAS:{id_instalación, nombre, imagen, descripción}

C.P.: {id_instalación}

VNN: {nombre}

NOTICIA: {id_noticia, titular, texto, tipo, fecha, imagen}

C.P.: {id_noticia}

VNN: {titular, texto, fecha, tipo}

EVENTO: {id_evento, título, descripción, fecha_inicio, hora_inicio}

C.P.: {id_evento}

VNN: {título, descripción, fecha_inicio, hora_inicio}

OPOSICIÓN: {id_oposición, nombre, plazas, fecha_publicación}

C.P. : {id_oposición}

VNN: {nombre, plazas, fecha_publicación}

IMAGEN: {id_imagen, username, titulo, nombre, descripción, destino, fecha_publicación}

C.P.: {id_imagen}

VNN: {titulo, nombre, destino, fecha_publicación }

C.AJ: {username} → AUTENTIFICADO

PREF. USUARIO: {username, boletín_noticias, eventos, noticias, callejero, tiempo}

C.P.: {username}

VNN: {boletín_noticias, eventos, noticias, callejero, tiempo}

C.AJ: {username} → AUTENTIFICADO

RESERVA: {id_reserva, dni, username, id_instalación, fecha, hora}

C.P.: {id_reserva}

VNN: {fecha, hora}

C.AJ: {dni, username} → AUTENTIFICADO

C.AJ: {id_instalación} → INSTALACIÓN DEPORTIVA

5. Implementación.

En este punto pasaremos a explicar detalladamente, las tecnologías usadas para la implementación del portal, así como los códigos de las diferentes partes que componen la arquitectura de 3 niveles utilizada para realizar este proyecto: presentación, negocio y datos.

5.1. Tecnologías.

Para el desarrollo de la implementación de la aplicación partimos de la base del lenguaje HTML. Adaptándonos a los estándares establecidos actualmente para el desarrollo web se decidió por utilizar la variante HTML 5.0 en cuanto a la estructura de los documentos hipertexto. Para la visualización se optó por utilizar hojas de estilo en cascada o CSS. Con su uso se perseguía obtener unos resultados más cómodos para el usuario que iba a visualizar el distinto contenido. Además, como soporte a esta presentación y como ayuda para realizar diferentes operaciones para la interacción entre el usuario y el portal, se ha utilizado el lenguaje Javascript, haciendo uso de estándares como el DOM y accediendo a diferentes librerías de este lenguaje con el fin de proporcionar dinamismo y accesibilidad a los distintos contenidos del portal, Login, presentación de imágenes....

Para la capa de datos y debido a la necesidad de interactuar desde estos documentos con la base de datos, se optó por desarrollar la aplicación en el lenguaje de programación PHP y con un sistema de gestión de bases de datos MySQL como soporte a la propia base de datos.

La elección de estas tecnologías para el desarrollo de la parte dinámica de la aplicación fue debida a que combinadas ambas tecnologías, logran de una manera sencilla que se puedan realizar múltiples posibilidades para la creación de aplicaciones web. Otro aspecto que se ha tenido en cuenta a la hora de escoger estas tecnologías es que son de carácter libre y poseen un gran soporte en la red.

5.1.1.HTML y HTML 5.0.

HTML, acrónimo de *HyperText Markup Language* (lenguaje de marcado de hipertexto), es un lenguaje que se utiliza para el desarrollo de páginas web, es usado para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML



se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

A principios de 1990, Tim Berners-Lee define por fin el HTML como un subconjunto del conocido SGML (Standard Generalized Markup Language), y crea algo más valioso aún, el World Wide Web. En 1991, Tim Berners-Lee crea el primer navegador web, ViolaWWW, que funcionaría en modo texto y sobre un sistema operativo UNIX.

Su uso y desarrollo está basado en una serie de recomendaciones emitidas por una entidad llamada W3C (World Wide Web Consortium), dirigida por el propio Berners-Lee.

El HTML se encarga de gestionar los distintos objetos que aparecen como textos, imágenes, URL..., y la estructura que el contenedor de estos objetos deberá poseer. Es un lenguaje genérico que es usado como base para otros lenguajes web. Los objetos son introducidos e utilizados mediante el uso de etiquetas llamadas tags, que permiten la interacción de estos objetos entre sí. Estas etiquetas siempre van metidas dentro de los símbolos < y >. Los elementos contenidos van dando forma a la estructura, y además tienen dos elementos identificables, el propio contenido y sus respectivos atributos, aunque en versiones posteriores a la creación del lenguaje estos atributos se han ido quedando obsoletos o han sido eliminados.

HTML 5.0 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como *HTML5* y una variante XHTML conocida como sintaxis *XHTML5* que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

En esta nueva versión se han incluido elementos como, por ejemplo, <audio> etiqueta para la inclusión de audio, <canvas> que permite la inclusión de objetos en 3D o <datalist> que permite realizar listas predefinidas de opciones para un elemento <input>.

5.1.2.CSS y CSS 3.

El nombre hojas de estilo en cascada viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento (etiquetas html y otros lenguajes) de su presentación (interfaz visual: tamaños, colores, márgenes...).

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso, podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>".

Un ejemplo de su utilización se puede ver, por ejemplo, en la etiqueta HTML <h1> que indica que un bloque de texto es un encabezamiento más importante que las etiquetas que la suceden (<h2>, <h3>...). En versiones más antiguas de HTML se permitía la inclusión de algunos atributos dentro de la etiqueta que han quedado obsoletos y han dejado de utilizarse. No obstante la etiqueta debía presentar ciertos atributos para definir su aspecto y diseño. Con la aplicación de CSS la etiqueta <h1> no debería proporcionar información sobre cómo será visualizado, solamente marca la estructura del documento. La información de estilo, separada en una hoja de estilo, especifica cómo se ha de mostrar <h1>: color, fuente, alineación del texto, tamaño, etc...

Por otro lado, antes de que estuviera disponible CSS, la única forma de componer espacialmente una página era el uso de tablas <table>. Aunque esta era una técnica cómoda y versátil, ello conllevaba el uso de un elemento con una semántica particular, y en el que la distribución de los datos no se ajustaban al flujo de la información que se obtenía en la vista desde los navegadores habituales, lo que redundaba en una merma en la accesibilidad a la página por parte de otros navegadores (orientados a personas con alguna deficiencia sensorial, o a ciertos dispositivos electrónicos).

Mediante el uso de CSS, se ha permitido eliminar el uso de tablas para el diseño, usándolas solamente para la muestra de datos tabulados, si bien es cierto que obtener la versatilidad que ofrecía el diseño con tablas, es algo más complicado si no se usan.



La primera especificación oficial de CSS también llamada CSS1 fue publicada en diciembre del año 1996 y dejó de utilizarse definitivamente en el 2008.

Existen diferentes modos de utilización de las hojas de estilo CSS

1. *Un estilo en línea (online)* es un método para insertar el lenguaje de estilo de página directamente dentro de una etiqueta HTML. Esta manera de proceder no es totalmente adecuada. El incrustar la descripción del formateo dentro del documento de la página Web, a nivel de código, se convierte en una manera larga, tediosa y poco elegante de resolver el problema de la programación de la página. Además presenta el inconveniente de que no se puede aplicar reusabilidad de los estilos. Este modo de empleo es útil en los clientes de correo electrónico ya que no soportan las hojas de estilos externas, y que no existen estándares que los fabricantes de clientes de correo respeten para utilizar CSS en este contexto, la solución más recomendable para maquetar correos electrónicos, es utilizar este modo.
2. Una hoja de estilo interna, que es una hoja de estilo que está incrustada dentro de un documento HTML, dentro del elemento `<head>`, como hemos nombrado anteriormente mediante el uso de la etiqueta `<style>`. De esta manera, se obtiene el beneficio de separar la información del estilo del código HTML propiamente dicho. Se puede optar por copiar la hoja de estilo incrustada de una página a otra (esta posibilidad es difícil de ejecutar si se desea para guardar las copias sincronizadas). En general, se usa únicamente cuando se quiere proporcionar alguna característica a una página Web en un simple fichero, por ejemplo, si se envía algo a la página Web.
3. Una hoja de estilo externa, es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web. Esta es la manera de programar más potente y más eficiente, porque separa completamente las reglas de formateo para la página HTML de la estructura básica de la página.

Ventajas del uso de CSS:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.

- Separación del contenido de la presentación, lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, sólo modificando algunos parámetros del CSS.
- Optimización del *ancho de banda* de la conexión, pues pueden definirse los mismos estilos para muchos elementos con un sólo selector; o porque un mismo archivo CSS puede servir para una multitud de documentos.
- Mejora en la *accesibilidad* del documento, pues con el uso del CSS se evitan antiguas prácticas necesarias para el control del diseño (como las tablas), y que iban en perjuicio de ciertos usos de los documentos, por parte de navegadores orientados a personas con algunas limitaciones sensoriales.

A pesar de parecer que el uso del CSS solo presenta ventajas no es así, puesto que también tiene sus limitaciones como por ejemplo, algunas que se aplican hasta la versión 2.1, ya que en la versión 3 se están intentando solucionar.

- Los selectores no pueden usarse en orden ascendente según la jerarquía del DOM (hacia padres u otros ancestros). La razón que se ha usado para justificar esta carencia por parte de la W3C, es para proteger el rendimiento del navegador, que de otra manera, podría verse comprometido. XSLT soporta en la actualidad un mayor número de sistemas operativos. Así mismo, también es mejor para trabajar con la mayoría de buscadores de Internet.
- Dificultad para el alineamiento vertical; así como el centrado horizontal se hace de manera evidente en CSS2.1, el centrado vertical requiere de diferentes reglas en combinaciones no evidentes, o no estándares.
- Ausencia de expresiones de cálculo numérico para especificar valores (por ejemplo: `margin-left: 10% - 3em + 4px;`). Un borrador de la W3C para CSS3, propone `calc()` para solventar esta limitación.
- Las pseudo-clases dinámicas (como: `hover`) no se pueden controlar o deshabilitar desde el navegador, lo que las hace susceptibles de abuso por parte de los diseñadores en banners, o ventanas emergentes.



En cuanto al CSS3, es el último estándar de CSS que a diferencia de CSS2, que fue una gran especificación que definía varias funcionalidades, está dividida en varios documentos separados, llamados módulos. Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad.

Los trabajos en el CSS3, comenzaron a la vez que se publicó la recomendación oficial de CSS2, y los primeros borradores de CSS3 fueron liberados en junio de 1999.

Debido a la modularización del CSS3, los módulos se encuentran en diferentes estados de su desarrollo, de forma que en noviembre de 2011, hay alrededor de cincuenta módulos publicados, tres de ellos se convirtieron en recomendaciones oficiales de la W3C en 2011: "*Selectores*", "*Espacios de nombres*" y "*Color*".

Algunos módulos, como "*Fondos y colores*", "*Consultas de medios*" o "*Diseños multicolumna*" están en fase de "candidatos", y considerados como razonablemente estables, a finales de 2011, y sus implementaciones en los diferentes navegadores son señaladas con los prefijos del motor del mismo.

En nuestro caso hemos utilizado la plantilla 960 Grid System para la estructura básica de la página. Este sistema da un formato a la página web como si de una rejilla se tratará. Divide la página en distintas columnas, en nuestro caso 12, todas ellas del mismo tamaño, agrupándose todas en un tamaño de 960 pixeles de ahí su nombre.

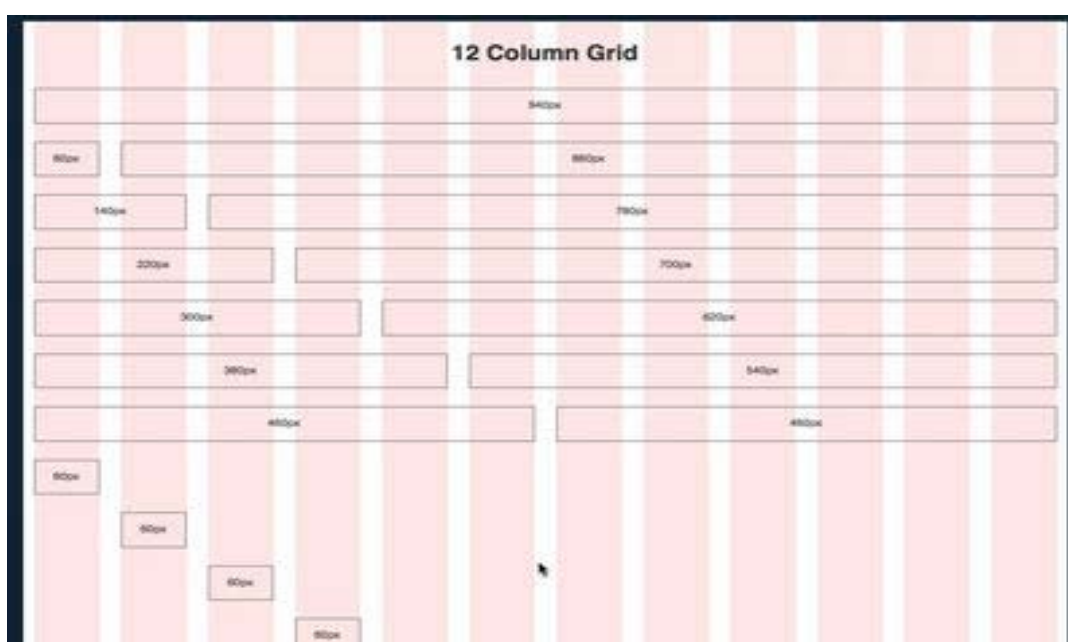


Fig.21: Modelo de 12 columnas.

5.1.3.JavaScript.

JavaScript es un lenguaje de programación interpretado, orientado a objetos y de uso dinámico no estático. Se utiliza principalmente como parte de un navegador web y que permite mejorar la interfaz de usuario, las páginas web dinámicas, el acceso local a bases de datos, etc.

JavaScript fue diseñado con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del nombrado anteriormente DOM (Document Object Model).

Tradicionalmente se ha utilizado en páginas HTML para realizar operaciones y únicamente en el lado del cliente, sin acceso a funciones del servidor. JavaScript es interpretado por el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

JavaScript fue desarrollado por Brendan Eich con el nombre de *Mocha*, el cuál fue renombrado posteriormente a *LiveScript*, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

Con los años fue adoptándose como un estándar, primero por el ECMA (European Computer Manufacturers Association) en 1997 y poco después por ISO (International Organization for Standardization).

Se puede incluir JavaScript tanto de manera interna como externa es decir, dentro del propio fichero web como en un fichero externo que es como el W3C define que es lo correcto.

En nuestro caso básicamente hemos incluido casi todas las funciones en archivos Javascript (.js) que son invocados mediante la etiqueta <script>, aunque también hemos utilizado scripts de manera interna.



5.1.4.PHP.

PHP es un lenguaje de programación, diseñado originalmente para la creación de páginas web de forma dinámica. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente en 1994 por Rasmus Lerdorf. Sin embargo la implementación principal de PHP es realizada por The PHP Group y es la utilizada como estándar al no haber una especificación formal. Es considerado como software libre.

Puede ser utilizado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas. Es el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo devuelve al cliente.

Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Sus principales características son las siguientes:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje de fácil aprendizaje, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.

- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de bases de datos que se utilizan en la actualidad.
- Capacidad de expandir su potencial utilizando extensiones.
- Amplia documentación en su web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos (POO). Incluso aplicaciones como Zend framework, empresa que desarrolla PHP, están totalmente desarrolladas mediante esta metodología.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.

También tiene algunos pequeños inconvenientes:

- Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado.
- Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel.
- Las variables al no ser tipadas dificulta a los diferentes IDEs para ofrecer asistencias para el tipeado del código, aunque esto no es realmente un inconveniente del lenguaje en sí.

En cuanto a las versiones se refiere, la primera fue lanzada en junio de 1995 y fue nombrada PHP tools, la más extendida de las versiones es la 5.X que es la que más mejoras ha recibido, actualmente la versión en uso es la 5.4.3 que fue lanzada en Mayo de 2012, en cada nueva versión se han ido solucionando pequeños bugs o fallos de vulnerabilidad, pero además se han ido facilitando



usos de variables para los programadores, sintaxis abreviadas, variables superglobales, etc. Actualmente se encuentra en desarrollo la salida de la versión 6, pero aún no tiene una fecha especificada de salida, ya que algunas de las mejoras que se iban a utilizar en esta versión fueron implementadas en versiones anteriores.

5.1.5.MySQL.

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB (perteneciente actualmente a la empresa Oracle) desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de imágenes o la gran cantidad de información en una red corporativa. Dado que los ordenadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

Por un lado se ofrece bajo la GNU GPL (Licencia pública general) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C (estándar para programación en lenguaje C).

Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

MySQL es muy utilizado en aplicaciones web, en plataformas, y por herramientas de seguimiento de errores. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Sus principales características son las siguientes:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.

- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves ajenas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Actualmente permite el uso de muchos tipos de motores de almacenamiento distintos, ofreciendo amplias posibilidades, dependiendo del tipo de datos que se quieran guardar y del tiempo de acceso que estos requieran. Algunas de las características diferenciales entre los distintos motores son las siguientes, velocidad de consultas, almacenamiento en memoria, tolerancia a fallos, compatibilidad de datos, etc.

La última versión y más estable es la 5.5.27 fechada en agosto de 2012, que incluye algunos de los siguientes cambios:

- El tipo de datos year(2) de dos caracteres pasará a estar obsoleto puesto que es problemático y en versiones posteriores será eliminado.
- Para algunas subconsultas que se deben ejecutar con un recorrido de intervalo en un índice no primario y el uso necesario de filesort, sólo la primera ejecución de la subconsulta se hará como una exploración de distancia. En todas las ejecuciones se realizarán las siguientes exploraciones de la tabla completa, lo que dará un rendimiento inferior.
- Un evento cuya longitud exceda del tamaño del master dump (max_allowed_packet) causa errores en la repetición.

5.2. Implementación del portal.

5.2.1. Parte pública.

A continuación describiremos todas aquellas partes que son comunes a todos los usuarios tanto anónimos como registrados. Para el desarrollo de la implementación, como ya se ha comentado en el apartado anterior, utilizaremos las siguientes tecnologías HTML, hojas de estilo CSS y Javascript en el lado del cliente y PHP con MySQL en la parte del servidor.



Como ya hemos comentado anteriormente la estructura del portal está basada en la plantilla 960 Grid System, que define los atributos del ancho de las columnas y está alojada en la hoja de estilo 960.css.

```
.container_12{
  margin-left:auto;
  margin-right:auto;
  width:960px
}
.grid_1,.grid_2,.grid_3,.grid_4,.grid_5,.grid_6,.grid_7,.grid_8,.grid_9,.grid_10,.grid_11,.grid_12,.grid_13,.grid_14,.grid_15,.grid_16{
  display:inline;
  float:left;
  margin-left:10px;
  margin-right:10px
}
.container_12 .grid_3{width:220px}
.container_12 .grid_6{width:460px}
.container_12 .grid_9{width:690px}
.container_12 .grid_12{width:940px}
.alpha{margin-left:0}
.omega{margin-right:0}
.container_12 .grid_1{width:60px}
.container_12 .grid_2{width:140px}
.container_12 .grid_4{width:300px}
.container_12 .grid_5{width:380px}
.container_12 .grid_7{width:540px}
.container_12 .grid_8{width:620px}
.container_12 .grid_10{width:780px}
.container_12 .grid_11{width:860px}

.container_12 .prefix_3{padding-left:240px}
.container_12 .prefix_6{padding-left:480px}
.container_12 .prefix_9{padding-left:720px}
.container_12 .prefix_1{padding-left:80px}
.container_12 .prefix_2{padding-left:160px}
.container_12 .prefix_4{padding-left:320px}
.container_12 .prefix_5{padding-left:400px}
```

Por facilidad de uso, aunque se utiliza el lenguaje HTML en gran parte del portal, nuestros ficheros accedidos son de tipo PHP.

Hemos creado el árbol de directorios basándonos en el tipo de archivo que contiene, así pues todos los archivos en lenguaje PHP se encuentran en el directorio raíz, los de tipo Javascript en el directorio js, los de la hoja de estilos en la carpeta estilos, y las imágenes en la carpeta images, a excepción de los iconos de crear, eliminar y nueva que se encuentran en la carpeta img.

Mediante las funciones include, include_once, require y require_once hemos incluido los archivos php que se van a utilizar en distintos sitios. Por ejemplo:

```
require("llamadasCabecera.php");
```

5.2.1.1 Archivos comunes.

Tenemos 5 archivos que son comunes a todo el bloque del portal. A continuación vamos a clasificar los archivos según el nivel al que pertenecen:

- Nivel de presentación:
 - Cabecera.php
 - Pie.php
 - Detalle.php
- Nivel de negocio:
 - llamadasCabecera.php
- Nivel de persistencia o datos:

- BD.php

- o BD.php, este archivo contiene funciones para la conexión a la base de datos y está concebida como una clase, lo que facilita su invocación, al crear objetos de su propio tipo. Aunque en la parte pública su acceso es mínimo, es importante nombrarlo puesto que es el que saca los datos de la base de datos. El código para la invocación y creación del objeto es el siguiente:

```
<?php require_once("bd.php");
$objBD=new BDConectar;
$objBD->Conectar();
?>
```

A continuación el código completo de bd.php:

```
<?php
class BDConectar{
    var $connect;
    function Conectar(){
        if (!($link = mysql_connect("localhost","admin","nimda")))
        {
            echo "Error conectando a la base de datos.";
            exit();
        }
        if (!mysql_select_db("ayuntamiento",$link))
        {
            echo "Error seleccionando la base de datos.";
            exit();
        }
        $this->connect=$link;
        return true;
    }
    function consulta($sql, $c)
    {
        $res = mysql_query($sql, $c);
        return $res;
    }
    function siguiente($res)
    {
        return mysql_fetch_assoc($res);
    }
    function Desconectar($link)
    {
        mysql_close($link); //cierra la conexion
    }
}
?>
```

- o El fichero cabecera.php, está contenido en un grid_12, es decir que ocupa 12 columnas, incluye toda la cabecera del portal, el buscador, el login, la imagen de enlace a la página principal, un “slideshow” de imágenes relacionadas con el pueblo, la selección de idioma y el menú. La información esta almacenada en una función llamada mostrarcabecera() que requiere ser llamada para su funcionamiento. Más adelante pasaremos a desglosarlo en sus distintas partes.

```
<?php
    include_once ("cabecera.php");
    mostrarcabecera ();
?>
```

Aquí vemos su aspecto:



Fig. 22: Aspecto de cabecera

Y aquí parte de su código, donde esté la invocación de las imágenes para el cambio de idioma:

```
function mostrarcabecera() {
    echo '<div class="container_12">';
    echo '<div id="cabecera" class="grid_12">';
    ?>

    <div class="grid_5 alpha">
    <div class="grid_1 alpha" id="banderes">
    <a href="selidioma.php?idioma=va" title="Valencia" ></a>
    <a href="selidioma.php?idioma=es" title="Castellano"></a>
    <div class="clear"></div>
    </div>
```

- o LlamadasCabecera.php es el contenedor de todas las invocaciones a las hojas de estilo y a los ficheros JavaScript, contenidos en una función de nombre llamadas().

```
<?php
function llamadas() {

    echo '<link rel="shortcut icon" type="image/x-icon" href="images/escudo-ico.jpg" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/estandar.css"/>';
    echo '<link rel="stylesheet" type="text/css" href="estilos/960.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/reset.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/text.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/menuprincipal.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/slideUp.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/estilos_noticias.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/estilos_imagenes.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/estilos_eventos.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/formRegistro.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/estilos_buscadore.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/menuAdmin.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/slideshow1.css" />';
    echo '<link type="text/css" rel="stylesheet" href="estilos/estilo.css" />';
    echo '<link type="text/css" rel="stylesheet" href="estilos/style.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/calendario.css" />';
    echo '<link rel="stylesheet" type="text/css" href="estilos/responsiveslides.css" />';

    echo '<script type="text/javascript" src="js/slideshow.js"></script>';
    echo '<script type="text/javascript" src="js/ieHover.js"></script>';
    echo '<script type="text/javascript" src="js/jquery.js"></script>';
    echo '<script type="text/javascript" src="js/reloj.js"></script>';
    echo '<script type="text/javascript" src="js/xfade2.js"></script>';
    echo '<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>';
    echo '<script type="text/javascript" src="js/responsiveslides.min.js"></script>';

}
?>
```

- Detalle.php, igual que cabecera.php también está contenida en un grid de tamaño 12 (grid_12), es el menú principal al detalle, desglosado en sub-apartados, enlazados con su respectivo contenido, y va colocado al final de cada página, después de la información central.

Ayuntamiento	Información	Servicios Municipales	Agenda	Galería
Bienvenida alcaldesa	Historia del municipio	Policia local	Agenda	Ver imágenes
Corporación municipal	Fiestas	Urbanismo	Noticias	Subir imagen
Servicios Públicos	Situación geográfica	Educación y cultura		
Bando	El tiempo	Juventud		
Actas pleno	Asociaciones	Deportes		
	Trámites	Sanidad y mujer		
		Medio Ambiente		
		Servicios Sociales		
		Fomento		

Fig. 23: Aspecto del detalle.

- Pie.php

Igual que detalle tiene un tamaño de 12 columnas, es el último bloque del portal e incluye una pequeña imagen de enlace a la página principal, la dirección del ayuntamiento, así como un enlace al Aviso legal.



Fig. 24: Aspecto del pie.

5.2.1.2 Servicios implementados.

A continuación se detallan otros ficheros y funcionalidades que son de uso público pero que no son comunes a todas las páginas, utilizados principalmente en el archivo index.php que engloba todas las anteriores y además otras informaciones que podrán visualizarse por cualquier visitante.

Como ya hemos comentado y haciendo uso de la plantilla 960 Grid System, hemos dividido el aspecto en una parte central que ocupa el grueso de 9 columnas y que contendrá lo más relevante, y una subsección lateral de 3 columnas de tamaño, ubicada a la derecha que contiene la fecha y hora, los últimos usuarios registrados y el calendario de eventos. Además de incluir dos secciones inferiores debajo de la parte principal donde tenemos un slideshow de imágenes de “Actualidad”, que enlazan con su respectiva noticia, y otro de “Fiestas y tradiciones”, donde vemos imágenes del pueblo.

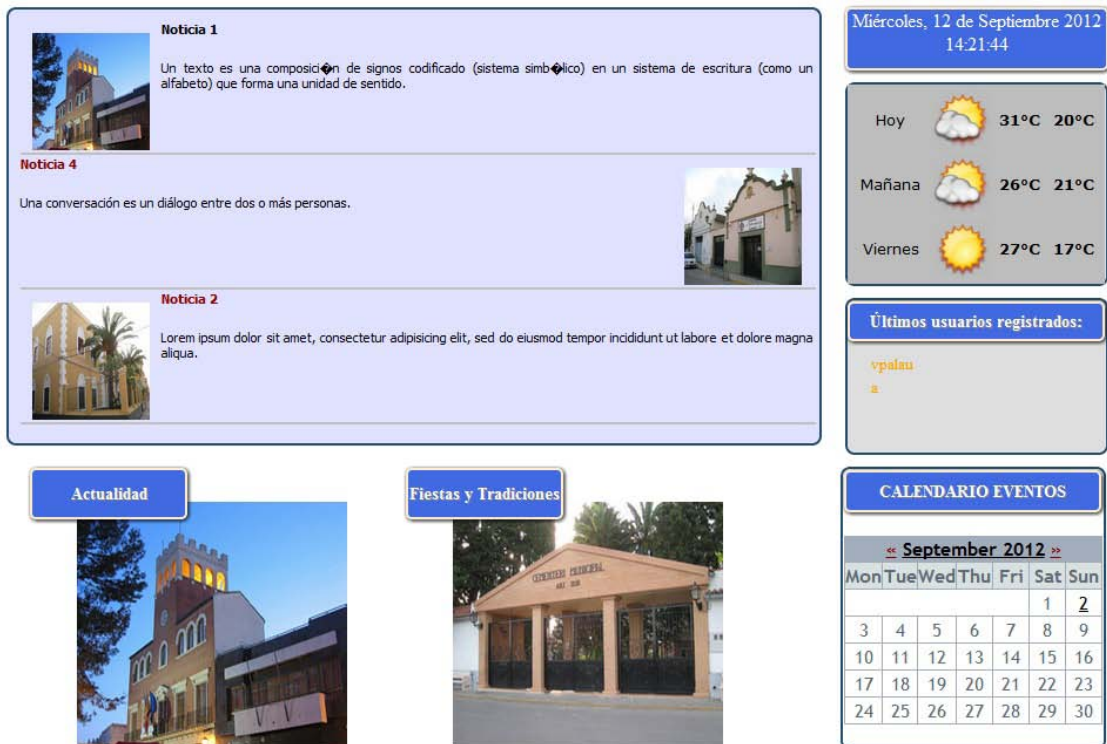


Fig. 25: Aspecto del slideshow de imágenes.

o Noticias.php

En la página del índice (index.php) esta parte principal está basada en la llamada al fichero noticias.php, que contiene las 3 últimas noticias de la tabla noticias de la base de datos.

Invocación de noticias.php dentro de un div de tamaño grid_9 que está definido en 960.css

```
<div class="grid_9" id="izq">
  <?php
    require("noticias.php");
  ?>
  <div class="clear"></div>
</div>
```

Parte del código de noticias.php, donde tenemos la consulta a la base de datos, nos devuelve el contenido, teniendo en cuenta según la fecha las 3 últimas noticias. La variable \$texto recoge el contenido de la columna texto de la noticia y separado por puntos, así se puede escoger solo hasta el primer punto. Esto se realiza con la función PHP explode, a continuación, mediante la variable \$lado elegimos el lado en el que se mostrará el contenido. Mediante la etiqueta de html definimos una miniatura del enlace al archivo de la imagen guardada en

el servidor y enlazada en la base de datos, a su lado encontramos el titular de la noticia que enlaza con el archivo mostrar_noticia.php que recibe el identificador de la misma y la muestra en toda su extensión y a continuación la llamada al array \$texto, con el subíndice 0 que contiene como ya hemos dicho una parte de la noticia. Las noticias están separadas por una línea horizontal mediante la etiqueta HTML <hr>.

```
$query = "SELECT * FROM noticia ORDER BY fecha DESC LIMIT 3";
$resultado = mysql_query($query);
$lado = izq;
while($columna = mysql_fetch_assoc($resultado)){
    $texto = explode(".", $columna["texto"]);
    echo '<div id="listar">';
    if($lado == izq) {
        echo '<div id="contenedor_img_noticia">';

        echo '<div id="img_noticia_izq" >';
        echo '';
        echo '</div>';

        echo '</div>';
        echo '<h1><a href="mostrar_noticia.php?id_noticia=' . $columna["id_noticia"] . '" title="Leer m&aacute;s...">' . $columna["titular"] . '</a></h1>';
        echo '<p>' . $texto[0] . '</p>';
        echo '<hr>';
        $lado=der;
    }
    else{
        echo '<div id="contenedor_img_noticia">';

        echo '<div id="img_noticia_der" >';
        echo '';
        echo '</div>';

        echo '</div>';
        echo '<h1><a href="mostrar_noticia.php?id_noticia=' . $columna["id_noticia"] . '" title="Leer m&aacute;s...">' . $columna["titular"] . '</a></h1>';
        echo '<p>' . $texto[0] . '</p>';
        echo '<hr>';
        $lado=izq;
    }
}
echo '</div>';
```

o Reloj.js

El campo de la hora se encuentra dentro de un <div> de tamaño 3 como ya hemos dicho anteriormente y en este caso no se ha utilizado un recurso php sino un recurso de tipo JavaScript que devuelve la fecha y la hora del sistema, llamado reloj.js, la invocación del cual se hace mediante un identificador HTML, llamado campohora.

```
<div id="fecha" class="grid_3">
    <div id="campoHora"></div>
</div>
```

Dividido en funciones, muestraFecha(), donde creamos dos arrays que contendrán los meses en español y el día de la semana en español, luego consultaremos mediante la función “Date” el día en el que nos encontramos. La función devuelve el día de la semana en letra, el día en número, el mes en letra y el año.




```
function muestraFecha(){
    var mes = new makeArray(12);
    mes[0] = "Enero";
    mes[1] = "Febrero";
    mes[2] = "Marzo";
    mes[3] = "Abril";
    mes[4] = "Mayo";
    mes[5] = "Junio";
    mes[6] = "Julio";
    mes[7] = "Agosto";
    mes[8] = "Septiembre";
    mes[9] = "Octubre";
    mes[10] = "Noviembre";
    mes[11] = "Diciembre";

    var dia_sem = new makeArray(7);
    dia_sem[0] = "Domingo";
    dia_sem[1] = "Lunes";
    dia_sem[2] = "Martes";
    dia_sem[3] = "Miércoles";
    dia_sem[4] = "Jueves";
    dia_sem[5] = "Viernes";
    dia_sem[6] = "Sábado";

    var hoy = new Date();
    var dia = hoy.getDate();
    var num_mes = hoy.getMonth();
    var anyo = hoy.getYear();
    var num_dia = hoy.getDay();
    if (anyo < 1000) {
        anyo += 1900; }
    return( " " + dia_sem[num_dia] + ", " + dia + " de " + mes[num_mes] + " " + anyo);
}
```

La función `muevereloj()`, como su nombre indica es la encargada de mostrar la hora actual dinámicamente además de, mediante la función `getElementById` devolver la concatenación de la fecha devuelta por la función anterior junto con la hora al campo que tiene como identificador `campoHora`.

```
function mueveReloj(){
    var momentoActual = new Date();
    var hora = momentoActual.getHours();
    if (hora<10) hora = "0" + hora;
    var minuto = momentoActual.getMinutes();
    if (minuto<10) minuto = "0" + minuto;
    var segundo = momentoActual.getSeconds();
    if (segundo<10) segundo = "0" + segundo ;

    horaImprimible = muestraFecha() + "</br>" + hora + ":" + minuto + ":" + segundo;
    document.getElementById('campoHora').innerHTML = horaImprimible;
    //document.form_reloj.reloj.value = horaImprimible
    tiempo=setTimeout("mueveReloj()",1000);
}
```

o Tiempo.php

Este fichero es el encargado de devolver el tiempo de los próximos 3 días en Alcàsser, utilizando un script proporcionado por la web www.tutiempo.net. Omitimos el código de invocación puesto que es similar a otros ya mostrados anteriormente. Su código completo es el siguiente:

```

<?php
function tiempo_alcasser(){
    echo '<div class="grid 3" id="der1">';
        echo '<div id="TI_eo353I3j4j14KAxzzzDjzYuQ9Kp">';
            echo '<h2><a href="http://www.tutiempo.net/Tiempo-Espana.html">El Tiempo</a></h2>';
            echo '<a href="http://www.tutiempo.net/Tiempo-Alcasser-E46045.html">El tiempo en Alcágrave;sser</a>';
            echo '</div>';
        echo '<script type="text/javascript" src="http://www.tutiempo.net/TIapiV2/eo353I3j4j14KAxzzzDjzYuQ9Kp"></script>';
    echo '</div>';
}
?>

```

- o localizacion.php

En este fichero tenemos el mapa de la situación geográfica de la localidad, se puede acceder a él tanto desde el menú principal como desde la secciones al detalle. El mapa ha sido extraído de google maps, que además proporciona un enlace para poder colocar en páginas web, mediante la etiqueta iframe. A continuación vemos la parte del código donde vemos como esta insertado en la página.

```

<div class="grid 9" id="izqscrollable2">
    <h6 class="title">Situación geográfica</h6>
    <iframe id="mapa" src="https://maps.google.es/maps?f=q&source=s_q&hl=es&geocode=&q=Alc&C3&Alcer,+Alc&C3&A0sser&aq=1&q=Alcacer&am
</div>

```

Y ahora tenemos como se muestra al usuario.



Fig. 26: Mapa de la localización.

o `ultimosregistrados.php`

Este archivo realiza una simple consulta que muestra los 3 últimos usuarios registrados según fecha.

```
<?php
require_once ("bd.php");
$objBD=new BDConectar;
$objBD->Conectar();

$ultimosUsuario = 'SELECT username FROM autenticado ORDER BY fecha_registro DESC LIMIT 3';
$resultadoUsuarios = mysql_query($ultimosUsuario) or die(mysql_error());
while($columna = mysql_fetch_assoc($resultadoUsuarios)){
    foreach($columna as $users){
        echo "<p>".$users."</p>";
    }
}
?>
```

o `calendario_eventos.php`

Genera un calendario del mes actual con todos los días, que enlaza a eventos, según el día, según el mes y además también enlaza el mes anterior y el mes posterior. Se ha realizado modificando un calendario ya generado anteriormente por el autor Keith Devens.

```
$objDB=new BDConectar;
$objDB->Conectar();
$mes=date('m');
$consulta=mysql_query("SELECT day(fecha_inicio) as fecha_inicio FROM evento WHERE month(fecha_inicio)='".$mes."' ORDER BY fecha_inicio ASC");
$dias=array();
while($row=mysql_fetch_array($consulta))
{
    $dias[]=$row['fecha_inicio'];
}

$time = time();
$oldlocale = setlocale(LC_TIME, NULL); #save current locale
setlocale(LC_TIME, 'es_ES'); #Spanish

$days = array();
foreach($dias as $clave=>$valor){
    //echo $dias[$clave];
    $days[$dias[$clave]] = array("mostrar_eventos.php?dia_evento=".$dias[$clave]."&mes_evento=".$mes,"linked-day");
}
$pn = array();
$mes2=date('n', $time);
$mesanterior=$mes2-1;
$mesposterior=$mes2+1;
$pn = array('&laquo;=>"mostrar_eventos.php?mes_evento='.$mesanterior.", '&raquo;=>"mostrar_eventos.php?mes_evento='.$mesposterior");
echo generate_calendar(date('Y', $time), date('n', $time), $days, 3, "mostrar_eventos.php?mes_evento=".$mes2, 1, $pn);
```

En este trozo de código vemos las modificaciones realizadas, la consulta que se realiza según el mes del sistema y seleccionando la fecha de inicio. Se recorre la consulta almacenando los días que hay eventos en un array llamado `$dias`. Para que no haya errores especificaremos que la hora local es la de España `es_ES`. A continuación llenaremos un vector llamado `$days` que lo que hace es enlazar los días que contienen eventos con la página `mostrar_eventos.php` que recibe por URL el mes y el día de los eventos. A continuación se realiza una operación similar, pero para los meses, anterior y posterior, que nos permite visualizar todos los eventos del mes pasado y del mes siguiente.

o `galeriaNoticias.php` y `galeriaEventos.php`

Estos dos archivos contienen las galerías de imágenes que están bajo y que van pasando en el caso de galeríaNoticias.php (Actualidad), las imágenes serán enlaces a las 3 últimas noticias a las que pertenece dicha imagen, en el otro caso, galeríaEventos.php (Fiestas y tradiciones) muestra todas las imágenes, el efecto de giro se consigue mediante JavaScript.

```

<?php
require_once("bd.php");
$objBD=new BDConectar;
$objBD->Conectar();
$sql = "SELECT * FROM noticia ORDER BY fecha DESC LIMIT 3";
$resultado = mysql_query($sql) or die(mysql_error());
$arrayImag = array();
$arrayId = array();
while($columna = mysql_fetch_array($resultado)){
    $numCol = mysql_num_rows($resultado);
    $arrayImag[] = $columna["imagen"];
    $arrayId[] = $columna["id_noticia"];
}
echo '<div id="rotator">';
for($i=0; $i<count($arrayId); $i++){
    echo '<a href="mostrar_noticia.php?id_noticia='.$arrayId[$i].'"></a>';
    echo '<a href="mostrar_noticia.php?id_noticia='.$arrayId[$i+1].'"></a>';
    echo '<a href="mostrar_noticia.php?id_noticia='.$arrayId[$i+2].'"></a>';
}
echo '</div>';
?>

```

Aquí tenemos el código completo de galeríaNoticias.php, donde tenemos la consulta a las 3 últimas noticias. En un array llamado arrayImag[] almacenamos el nombre de las imágenes correspondientes a dichas noticias, y sus identificadores en un array llamado arrayId[], que los utilizaremos para enviarlos por URL a mostrar_noticia.php.

- o imagen.class.php

Es una clase implementada para poder realizar operaciones con imágenes, utilizada principalmente en visualizarImagen.php de la parte pública y en subir_imagen.php de la parte privada.

- o visualizarImagen.php

Para acceder a la funcionalidad de este fichero deberemos ir al menú a la sección Galería y luego Ver imágenes. Se trata de la galería de imágenes que han ido subiendo los usuarios registrados al servidor y que permite seleccionar las imágenes que queremos ver según su tipo.



```

<div id="divvisualizarimagen">
<form id="imagen" name="imagen" method="post" action="visualizarImagen.php">
  <div id="formulario">
    <label> Elige el tipo: </label>
    <select name="destino" size="1">
      <option>Fiestas</option>
      <option>Historia</option>
      <option>Deportes</option>
      <option>General</option>
      <option>Todas</option>
    </select>
  </div>
  <div id="formularioboton">
    <input type="submit" name="submit" id="button" value="Mostrar" />
  </div>
</form></div>

```

Y nos las muestra ordenadas en una tabla de 4 en 4 según una variable llamada tope que nos dice cuando se realiza el salto a la siguiente fila de imágenes. Como la localización de las imágenes se hace en la base de datos en la columna destino, puede que esa imagen no se encuentre alojada en el servidor, lo que comprobaremos mediante la función de PHP file_exists(), como vemos en el siguiente código que muestra las imágenes de todos los tipos excepto de la opción Todas.

```

if($destino<>'Todas')
{
  $i=0;
  $tope=4;
  if($consultaImagen)
  {
    echo "<table>";
    echo "<tbody>";
    while($imagen=mysql_fetch_array($consultaImagen))
    {
      if($destino==$imagen['destino'] && file_exists('images/'.$imagen["destino"].'/'.$imagen["nombre"]))
      {
        echo '<td>';
        echo '<div id="divvisualizarimagen">'.ucfirst($imagen['titulo'])."<br />";
        echo '<a href="mostrar_imagen.php?id_imagen='.$imagen['id_imagen'].'">';
        echo '</td>';
        $i++;
        if(($i%$tope)==0){
          echo "</tr>";
          echo "<tr>";
        }
      }
    }
  }
  echo "</tbody>";
  echo "</table>";
}
}

```

Así se verían las imágenes. Cada una de ellas tendrá un enlace a mostrar_imagen.php.



Fig. 27: Galería de imágenes.

- o `mostrar_noticia.php`, `mostrar_imagen.php`, `mostrar_evento.php` y `mostrar_eventos.php`

Estos archivos son la visualización al detalle de, como su nombre indica, el tipo de dato que alojen. En el caso de `mostrar_noticia`, `mostrar_imagen` y `mostrar_evento`, muestra los atributos visualizables de cada elemento según su identificador. En el caso de `mostrar_eventos` se trata de los eventos accedidos desde el calendario de eventos, y recoge las variables del mes actual y del día que se ha seleccionado. Si solo recibe el mes, muestra todos los eventos de dicho mes, si recibe el día y el mes, muestra los eventos de ese día en concreto. A continuación vemos el código de `mostrar_eventos.php`.

```

if(isset($_GET['dia_evento'])) {
    $dia_evento = $_GET['dia_evento'];
    $mes_evento = $_GET['mes_evento'];

    $sql = "SELECT * FROM evento WHERE day(fecha_inicio)='$dia_evento' AND month(fecha_inicio)='$mes_evento' ORDER BY fecha_
$resultado = mysql_query($sql) or die(mysql_error());
    foreach($mes_espanol as $clave=>$valor) {
        if($clave+1==$mes_evento) {
            echo '<h1 class="title2">Eventos dia '.$dia_evento.' de '.$valor.</h1>';
        }
    }
    while($columna = mysql_fetch_array($resultado)) {
        echo '<h1 id="titulo_evento">'.$columna["titulo"].' a las '.$substr($columna["hora_inicio"],0,-3).</h1>';
        echo '<hr>';
        echo '<p id="texto_evento">'.$columna["descripcion"].</p>';
        echo '<hr>';
    }
    if(mysql_num_rows($resultado)==0) {
        echo '<h1 id="texto_evento">No hay eventos.</h1>';
    }
}
elseif(isset($_GET['mes_evento'])) {
    $mes_evento2 = $_GET['mes_evento'];
    $sql = "SELECT * FROM evento WHERE month(fecha_inicio)='$mes_evento2' ORDER BY fecha_inicio ASC";
    $resultado = mysql_query($sql) or die(mysql_error());
    foreach($mes_espanol as $clave=>$valor) {
        if($clave+1==$mes_evento2) {
            echo '<h1 class="title2">Eventos de '.$valor.</h1>';
        }
    }
    while($columna = mysql_fetch_array($resultado)) {
        $dia = explode("-", $columna["fecha_inicio"]);
        echo '<h1 id="titulo_evento">D&iacute;a '.$dia[2].</h1>';
        echo '<h1 id="titulo_evento">'.$columna["titulo"].' a las '.$substr($columna["hora_inicio"],0,-3).</h1>';
        echo '<hr>';
    }
}

```

Aquí vemos como se ve mostrar_noticia.php.

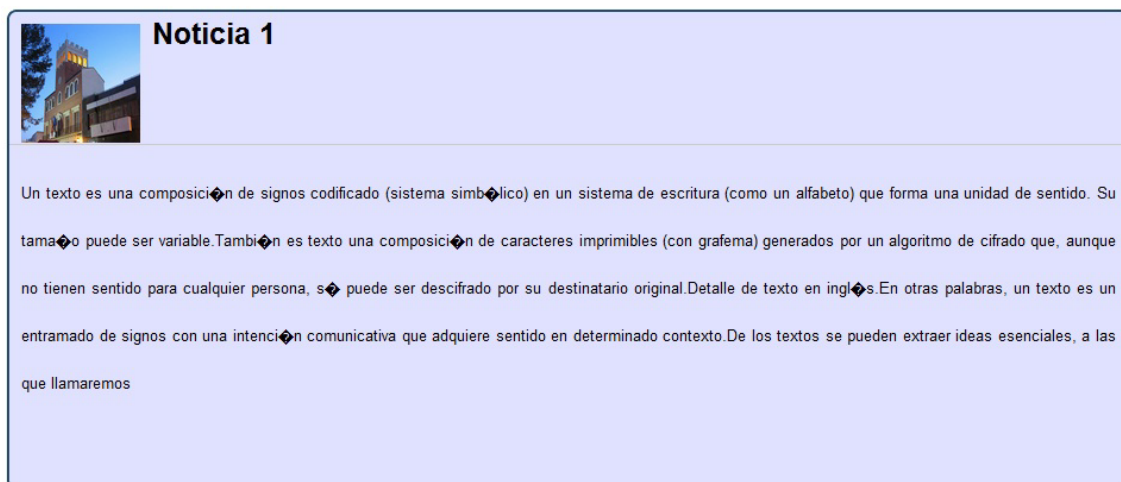


Fig. 28: Mostrar noticia.

o FormResgistro.php

Se trata del formulario de registro, accesible desde la cabecera, mediante, el enlace llamado Registrarse. Se encarga de insertar los datos de registro de un usuario anónimo en la base de datos. Si el resultado es positivo, entonces este pasaría a ser un usuario registrado. El formulario controla cosas como, por ejemplo, que el email sea un email, utilizando el archivo ValidacionEmail.php, que el DNI tenga 9 caracteres, que no es un usuario que ya está registrado, etc. Para realizar tanto las operaciones de existencia como la de inserción, utilizaremos las funciones de una clase llamada Autenticado.class.php, que realiza las funciones de los objetos de tipo usuario.

Podemos observar en la parte del código que se muestra a continuación algunos de los errores que controlamos.

```
if(empty($username))
{
    $error[] = 'El campo USERNAME es obligatorio.';
}
if($objAutenticado->comprobar_existeUsername($username)==true)
{
    $error[]='Ya existe un usuario con username "'. $username.'".';
}
if(empty($dni))
{
    $error[] = 'El campo DNI es obligatorio.';
}
if($objAutenticado->comprobar_usernameDni($username,$dni)==true)
{
    $error[] = 'Ya existe un usuario con username "'. $username.'" y dni "'. $dni.'".';
}
if(strlen($dni)<9 || strlen($dni)>9){
    $error[]='El campo DNI debe tener 9 caracteres.';
}
if(filter_input(INPUT_POST, substr($dni,0,-1), FILTER_VALIDATE_INT)===false){
    $error[] = 'Los primeros 8 caracteres del DNI deben ser numéricos.';
}
if($objAutenticado->comprobar_existeDni($dni)==true)
{
    $error[] = 'Ya existe un usuario con dni "'. $dni.'".';
}
if(empty($password1) || empty($password2))
{
    $error[] = 'El PASSWORD es obligatorio.';
}elseif(strlen($password1)<6 || strlen($password2)<6)
{
    $error[] = 'El password debe tener minimo 6 caracteres.';
}
```

o Buscador.php

Este archivo contiene las funciones que se realizan en el nivel de negocio, internamente, y devuelve los resultados de la búsqueda, se accede desde la cabecera, donde tenemos un área de texto y un icono que al apretarlo envía a este archivo la palabra o palabras que queremos buscar. Hemos restringido la búsqueda únicamente a las noticias y los eventos. No se mostrarán completamente, sino una pequeña parte, y mediante un enlace a `mostrar_noticia.php` y a `mostrar_evento.php` podremos acceder a leerlas. A continuación vemos su código completo:



```

<?php
session_start();
ob_start();
?>
<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca-es" lang="ca-es" dir="ltr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Ayuntamiento de Alcázar de San Juan</title>
</head>
<?php
require("bd.php");
$objBD=new BDConectar;
$objBD->Conectar();

$buscar = (isset($_POST['buscador'])) ? trim($_POST['buscador']) : '';

if (!empty ($buscar))
{
    require_once("llamadasCabecera.php");
    llamadas();
    include_once ("cabecera.php");
    mostrarcabecera();

    echo '<div class="grid_12" id="buscado">';

    $search=$_POST['buscador'];
    $keywords_array = explode(" ", $search);

    $sql1 = "SELECT * FROM noticia n WHERE n.texto LIKE '%" . $keywords_array[0] . "%'";
    $resultado1 = mysql_query($sql1) or die(mysql_error());
    $numCol1 = mysql_num_rows($resultado1);

    if($numCol1==0 && $numCol2==0)
    {
        echo "<script type='text/javascript'>alert('No se han encontrado resultados.');" . window.history.back();</script>";
    }
    elseif($numCol1>0 && $numCol2==0)
    {
        for($i=1;$i<count($keywords_array);$i++)
        {
            $sql1 .= " OR texto LIKE '%" . $keywords_array[$i] . "%'";
        }

        if($resource1 = mysql_query($sql1))
        {
            echo '<div id="contenido_buscado">';
            while($row = mysql_fetch_assoc($resource1))
            {
                $titulo = $row["titular"];
                $texto = explode(".", $row["texto"]);
                echo "<b>". $titulo . "</b>";
                echo '<p>'. $texto[0] . '</p>';
                echo '<a href="mostrar_noticia.php?id_noticia=' . $row["id_noticia"] . '>Leer más...</a>';
                echo "<br />";
                echo "<br />";
            }
            echo '</div>';
        }
    }
    elseif($numCol1==0 && $numCol2>0)
    {
        for($i=1;$i<count($keywords_array);$i++)
        {
            $sql2 .= " OR texto LIKE '%" . $keywords_array[$i] . "%'";
        }
    }
}

```



```

if($resource2 = mysql_query($sql2))
{
    echo '<div id="contenido_buscado">';
    while($row = mysql_fetch_assoc($resource2))
    {
        $titulo = $row["titulo"];
        $descripcion = explode(".", htmlentities($row["descripcion]));
        echo "<b>".$titulo."</b>";
        echo '<p>'.$descripcion[0].'.</p>';
        echo '<a href="mostrar_evento.php?id_evento='.$row["id_evento"].'">Leer m aacute;s...</a>';
        echo "<br />";
        echo "<br />";
    }
    echo '</div>';
}
}
else
{
    for($i=1;$i<count($keywords_array);$i++)
    {
        $sql1 .= " OR texto LIKE '%$keywords_array[$i].%'";
        $sql2 .= " OR texto LIKE '%$keywords_array[$i].%'";
    }
    $resource1 = mysql_query($sql1);
    $resource2 = mysql_query($sql2);
    if($resource1 > 0 && $resource2 > 0)
    {
        echo '<div id="contenido_buscado">';
        while($row1 = mysql_fetch_assoc($resource1))
        {
            $titulo = $row1["titular"];
            $texto = explode(".", htmlentities($row1["texto]));
            echo "<b>".$titulo."</b>";
            echo '<p>'.$texto[0].'.</p>';
            echo '<a href="mostrar_noticia.php?id_noticia='.$row1["id_noticia"].'">Leer m aacute;s...</a>';
            echo "<br />";
            echo "<br />";
        }
        echo "<br />";
        echo "<br />";
        while($row2 = mysql_fetch_assoc($resource2))
        {
            $titulo = $row2["titulo"];
            $descripcion = explode(".", htmlentities($row2["descripcion]));
            echo "<b>".$titulo."</b>";
            echo '<p>'.$descripcion[0].'.</p>';
            echo '<a href="mostrar_evento.php?id_evento='.$row2["id_evento"].'">Leer m aacute;s...</a>';
            echo "<br />";
            echo "<br />";
        }
        echo '</div>';
    }
}
echo '</div>';

require("detalle.php");
seccion_detalle();
require("pie.php");
mostrarpie();
}
else
{
    echo "<script type='text/javascript'>alert('Debes introducir una palabra.');

```

Recibe el string de texto que se ha introducido en el cuadro del buscador, que como hemos dicho encontramos en la cabecera, lo separa por palabras mediante la funci3n *explode* y a continuaci3n realizamos dos b3squedas, una para las noticias y otra para los eventos, y mediante clausulas if y elseif vamos separando las b3squedas en noticias, eventos y noticias y eventos, ya que el formato de vuelta ser a diferente.

- o selidioma.php

Guarda las variables de idioma en una variable SESSION para que pueda ser utilizada en todo el portal. Como ya hemos visto el idioma se selecciona desde las “flags” que aparecen arriba a la izquierda.

5.2.1.3 Otros archivos.

- o alcaldessa.php, fiestas.php y deportes.php

Son páginas estáticas donde apreciamos, la bienvenida de la alcaldesa del pueblo, las fiestas del pueblo y un breve resumen de los deportes. En alcaldessa.php y fiestas.php se aplica la conversión de idioma a modo de ejemplo. Tienen una estructura similar a la de la página principal. A continuación vemos la página alcaldessa.php con el selector de idioma en “va” de valenciano.



Fig. 29: Bienvenida alcaldesa.

5.2.2. Parte privada.

La técnica utilizada para diferenciar la parte privada de la pública, ha sido la opción de “loguearse”, mediante el apartado login situado en cabecera.php en la parte superior derecha, y que abre un cuadro de dialogo utilizando JavaScript que permite al usuario recuperar su password y loguearse mediante un formulario. El

archivo que permite recuperar el password es un formulario llamado FormRecuperarPassword.php que nos permite mediante el username y el email recuperar el password, para loguearse basta con introducir nuestro username y password, que serán analizados por el archivo analizadorLogin.php.

Fig. 30: Formulario para loguearse

Fig. 31: Usuario logueado

Vemos el código de analizadorLogin.php, se busca al usuario en la base de datos a partir de los datos introducidos en el formulario de Login, si el usuario existe y la clave concuerda, se guardan las variables username, categoría y dni recibidas de la base de datos en variables de sesión (\$_SESSION). Estas serán utilizadas a posteriori en varios lugares de la web, y serán movidas entre las páginas con la función session_start(), que se encarga de mantener la sesión abierta de manera persistente.

```

$username = (isset($_POST['username'])) ? trim($_POST['username']) : '';
$password = (isset($_POST['password'])) ? $_POST['password'] : '';
$redirect = (isset($_REQUEST['redirect'])) ? $_REQUEST['redirect'] : 'index.php';

if(isset($_POST['submit']) && $_POST['submit'] == "Acceder"){
    $query = "SELECT * FROM autenticado WHERE username = '$username' AND password = '$password'";
    $resultado_busqueda = mysql_query($query) or die(mysql_error());
    if(mysql_num_rows($resultado_busqueda) > 0){
        $_SESSION['logged'] = 1;
        while($row = mysql_fetch_assoc($resultado_busqueda))
        {
            $_SESSION['username'] = $row['username'];
            $_SESSION['categoria'] = $row['categoria'];
            $_SESSION['dni'] = $row['dni'];
        }
        if($_SESSION['categoria']=="Usu")
        {
            header("Location: index.php");
        }
        else
        {
            header("Location: index_admin.php");
        }
        mysql_free_result($resultado_busqueda);
        mysql_close();
        die();
    }
    else{
        $_SESSION['username'] = '';
        $_SESSION['logged'] = 0;
        echo "<script type='text/javascript'>alert('El usuario o la contraseña es incorrecta.');

```

En el caso de que el usuario haya puesto erróneamente el username o password se mostrará un Popup advirtiéndole.



También se distingue a los usuarios registrados según su categoría, “Usu” para los usuarios registrados y “Adm” para los administradores, según esto, el usuario será enviado de nuevo al índice en el caso de los usuarios registrados sin rango y la sección de administrador en el caso de los administradores.

5.2.2.1. Usuarios registrados.

Estos usuarios tienen la posibilidad de realizar varias acciones que los usuarios anónimos no pueden realizar.

- o subirimagen.php

Para acceder a este archivo lo haremos desde el menú principal o desde el menú de detalle en la sección galería, haciendo clic en subir imagen. De nuevo utilizamos la técnica del popup para advertir al usuario no logueado de que no podrá realizar esta acción y será devuelto a index.php.

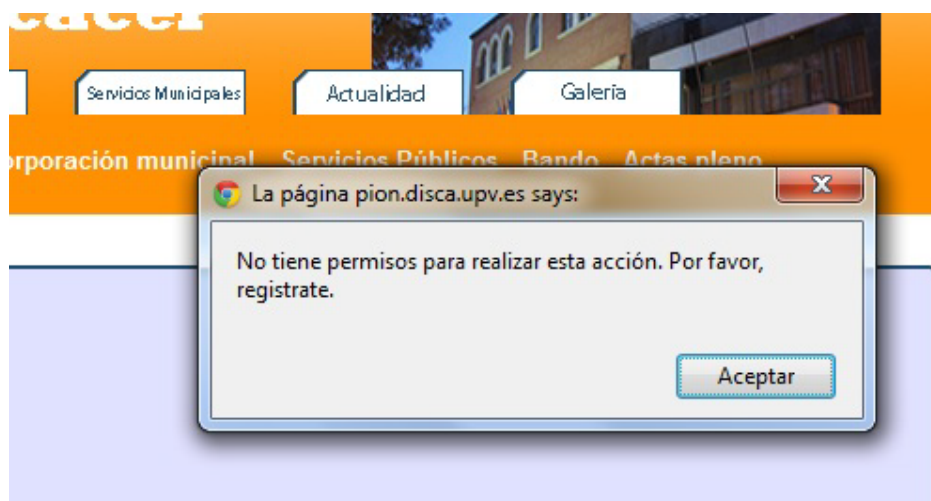


Fig. 32: Popup de advertencia en subir imagen.

En el caso de que estemos logueados, veremos un formulario que nos permitirá subir una imagen desde nuestro pc al servidor, y que además de subirla, insertará información en la base de datos, siempre que hayamos completado todos los datos del formulario, siendo el campo “descripcion” el único que podremos dejar en blanco.

Fig. 33: Formulario subir imagen.

A continuación vemos la estructura de la tabla imagen en la base de datos. Podemos observar que tiene más campos que los requeridos en el formulario, como id_imagen que es un campo autoincrementable, que la propia base de datos se encarga de poner, o el campo username, que será recogido de la variable de sesión (\$_SESSION['username']), ya que así la imagen estará asociada al usuario que la ha subido, y el campo fecha_public es cogido del sistema cuando el usuario sube la imagen.

	Campo	Tipo	Cotej
<input type="checkbox"/>	<u>id_imagen</u>	int(11)	
<input type="checkbox"/>	username	varchar(20)	utf8_sp
<input type="checkbox"/>	titulo	varchar(255)	utf8_sp
<input type="checkbox"/>	nombre	varchar(20)	utf8_sp
<input type="checkbox"/>	descripcion	varchar(255)	utf8_sp
<input type="checkbox"/>	destino	varchar(20)	utf8_sp
<input type="checkbox"/>	fecha_public	date	

Fig. 34: Atributos de la tabla imagen.

- o consultaReservaUsuario.php, realizarreserva.php y reservar.php

El usuario que este logueado podrá consultar sus reservas si las tuviera, accediendo desde la sección deportes. De nuevo hemos creado una clase para realizar las tareas del portal a la base de datos, reservaUsuario.class.php.

Además de poder consultar sus propias reservas, podrá anularlas con antelación y realizar nueva reserva, estas funcionalidades son interesantes y además pueden distinguir este portal de otros portales de ayuntamientos.



Fig. 35: Consulta reserva de un usuario.

Para realizar una nueva reserva accederíamos mediante el botón que vemos en la imagen anterior, llamado NuevaReserva que nos enviara a realizarreserva.php, donde encontraremos un selector de opciones con las distintas pistas que se pueden reservar. Cuando seleccionamos una, se mostrará una tabla con las fechas y los horarios, en rojo las horas ya pasada o que han sido reservadas, y en verde las que podremos reservar.



Fig. 36: Tabla de reserva.

En la ilustración anterior vemos como la hora que aparece reservada en la tabla de consulta esta en rojo. Su código sería el siguiente.

```

echo "<h1 class='title2'>DISPONIBILIDAD HORARIA".$instal."</h1>";
echo "</br>";
echo "<table border='1px'>";
echo "<th colspan=15>Díacutec;as</th>";

foreach($dias as $clavedia=>$valordia)
{
    echo "<tr border='1px'>";
    echo "<td>";
    echo implode ( "-", array_reverse ( preg_split ( "/\D/", $valordia)));
    echo "</br>";
    echo "</td>";

    foreach($array_horas as $clavehoras=>$valorhoras)
    {
        if(($objReservaUsuario->comprobar_reserva($dias[$clavedia], $array_horas[$clavehoras], $instal)==true) || ((implode ( "-", array_reverse (
        {
            echo "<td bgcolor='red' border='1px' title='Reservado'>";
            echo $clavehoras;
            echo "</td>";
        }
        )else
        {
            echo "<td bgcolor='green' border='1px' title='Libre'>";
            echo "<a href=reservar.php?hora=".$clavehoras."&fecha=".$dias[$clavedia]."&username=".$_SESSION['username']."&dni=".$_SESSION['dni']
            echo "</td>";
        }
    }
    echo "</tr>";
}
echo "</table>";

```

Al seleccionar una hora libre nos aparecerá un formulario (reservar.php) ya rellenado con nuestros datos, que no serán modificables y que nos permite reservar o cancelar.

CONFIRMAR DATOS RESERVA	
DNI:	<input type="text" value="29198089A"/>
INSTALACION:	<input type="text" value="bas002"/>
FECHA:	<input type="text" value="19-09-2012"/>
HORA:	<input type="text" value="08:00"/>
USUARIO:	<input type="text" value="vpalau"/>
<input type="button" value="Reservar"/> <input type="button" value="Cancelar"/>	

Fig. 37: Confirmación datos reserva.

5.2.2.2. Usuario administrador.

El usuario administrador tendrá una sección especial, con una página índice propia llamada index_admin.php, y que le permitirá controlar los diferentes objetos del sistema, mediante clases creadas para cada uno de ellos, noticia.class.php para las noticias, evento.class.php para los eventos,

autenticado.class.php para los usuarios autenticados, instalaciones.class.php para instalaciones deportivas y reserva.class.php para las reservas. Index_admin.php tiene un menú lateral que devuelve la opción correspondiente que es analizada mediante un switch.

```
function insertar_noticia($campos){
    if($this->link->Conectar()==true){
        //print_r($campos);
        //echo "INSERT INTO noticia (id_noticia, titular, texto, imagen, fecha) VALUES ('".$campos[0]."', '".$campos[1]."', '".$campos[2]."', '".$campos[3]."', '".$campos[4]."', '".$campos[5]."' );
        return mysql_query("INSERT INTO noticia (id_noticia, titular, texto, imagen, fecha, tipo) VALUES ('".$campos[0]."', '".$campos[1]."', '".$campos[2]."', '".$campos[3]."', '".$campos[4]."', '".$campos[5]."' );
    }
}

function actualizar_noticia($campos,$id_noticia){
    if($this->link->Conectar()==true){
        //print_r($campos);
        return mysql_query("UPDATE noticia SET id_noticia = '".$campos[0]."', titular = '".$campos[1]."', texto = '".$campos[2]."', imagen = '".$campos[3]."', fecha = '".$campos[4]."', tipo = '".$campos[5]."' WHERE id_noticia='".$id_noticia."'");
    }
}

function mostrar_noticia($id_noticia){
    if($this->link->Conectar()==true){
        return mysql_query("SELECT id_noticia, titular, texto, imagen, DATE_FORMAT(fecha,'%d-%m-%Y') AS fecha, tipo FROM noticia WHERE id_noticia='".$id_noticia."'");
    }
}

function mostrar_noticias(){
    if($this->link->Conectar()==true){
        return mysql_query("SELECT * FROM noticia ORDER BY fecha ASC");
    }
}

function eliminar_noticia($id_noticia){
    if($this->link->Conectar()==true){
        return mysql_query("DELETE FROM noticia WHERE id_noticia='".$id_noticia."'");
    }
}

function comprobar_id_noticia($id_noticia){
    if($this->link->Conectar()==true){
        return mysql_query("SELECT id_noticia FROM noticia WHERE id_noticia='".$id_noticia."'");
    }
}
```

Parte del código de noticia.class.php con las distintas funciones que realiza hacía la base de datos.

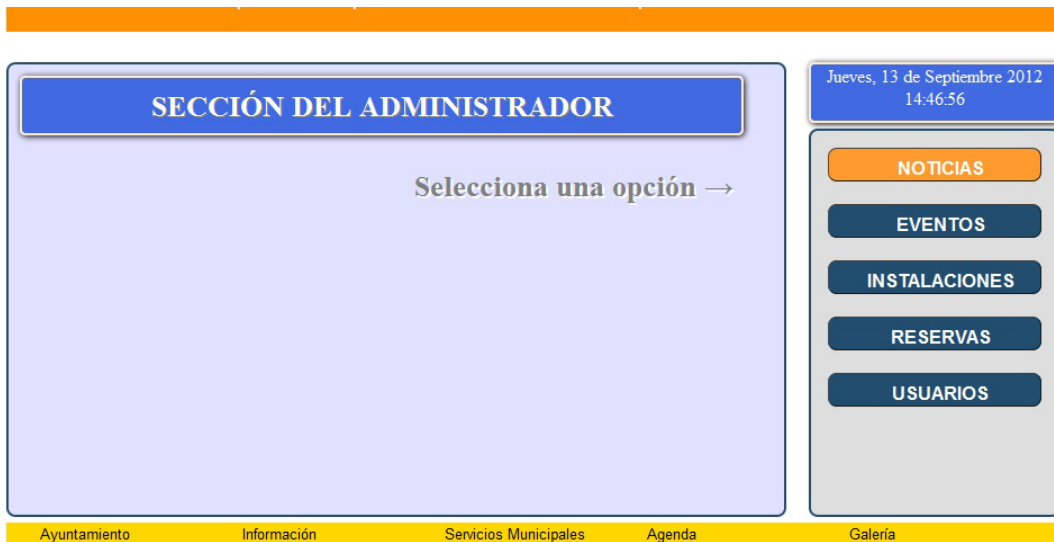


Fig. 38: index_admin.php.

- o consulta.php, consultaEvento.php, consultaInstalacion.php, consultaReserva.php y consultaAutenticado.php

El switch nombrado anteriormente nos dice que opción se ha escogido y realiza la llamada a uno de estos archivos que nos muestran una tabla con todos los elementos de la base de datos, son invocados en el div central, siendo la tabla distinta para cada tipo de objeto, ya que sus atributos difieren. Desde esta tabla, además de poder visualizar la cantidad de elementos de un tipo que tenemos, podremos, eliminar, actualizar y crear nuevos objetos de ese tipo.



Fig. 39: Vista de consultaInstalación.php.

A continuación vemos el código completo de consultaInstalacion.php

```

<?php
require('instalaciones.class.php');
$objInstalacion=new Instalacion;
$consulta=$objInstalacion->mostrar_instalaciones();
??
<div class="bar">
<a id="nuevaInstalacion" class="button" href="nuevaInstalacion.php"> Nueva Instalacion</a>
</div>
<table>
<tr>
<th>ID INSTALACION</th>
<th>NOMBRE</th>
<th>IMAGEN</th>
<th>DESCRIPCIÓN</th>
<th></th>
<th></th>
</tr>
<tr>
<td><?php echo $instalacion['id_instalacion'] ?></td>
<td><?php echo $instalacion['nombre'] ?></td>
<td><?php echo $instalacion['imagen'] ?></td>
<td><?php echo $instalacion['descripcion'] ?></td>
<td><span class="modi"><a href="actualizarInstalacion.php?id_instalacion=<?php echo $instalacion['id_instalacion'] ?>"><a onClick="EliminarDatoInstalacion(<?php echo $instalacion['id_instalacion'] ?>); return false" href="eliminarInstalaci
</td>
</tr>
<tr>
<td><?php echo $instalacion['id_instalacion'] ?></td>
<td><?php echo $instalacion['nombre'] ?></td>
<td><?php echo $instalacion['imagen'] ?></td>
<td><?php echo $instalacion['descripcion'] ?></td>
<td><span class="modi"><a href="actualizarInstalacion.php?id_instalacion=<?php echo $instalacion['id_instalacion'] ?>"><a onClick="EliminarDatoInstalacion(<?php echo $instalacion['id_instalacion'] ?>); return false" href="eliminarInstalaci
</td>
</tr>
</table>

```



- o nueva.php, nuevoEvento.php, nuevaInstalacion.php, nuevaReserva.php, nuevoAutenticado.php

Se acceden desde el botón superior de cada sección y nos permitirán crear un nuevo objeto del tipo deseado.



Fig. 40: Botón nueva instalación.

Se trata de formularios que nos permitirán introducir los datos del objeto, los cuales serán insertados en la base de datos, antes se comprobaba una serie de requisitos distintos para cada elemento, por ejemplo, los campos que sean obligatorios, en caso contrario no se nos permitirá realizar la operación. En esta parte del código vemos los errores junto con una parte del formulario.

```

if(empty($nombre)){
    $error[] = 'El campo NOMBRE es obligatorio.';
}
?>
<div id="formularioErrores">
<?php
if(count($error)>0){
    foreach($error as $clave=>$valor){
        echo '<p><span style=color:red>' . $valor. '</p>';
    }
}
?>
</div>
<?php
if(empty($error)){
    if ( $objInstalacion->insertar_instalacion(array($id_instalacion,$nombre,$imagen,$descripcion)) == true)
    {
        echo "<script type='text/javascript'>alert('La instalaci\u00f3n ha sido creada con exito'); document.location='./index_admin.php';</script>";
    }
}
?>
<form id="frmInstalacionNueva" name="frmInstalacionNueva" method="post" action="nuevaInstalacion.php">
<div id="formulario">
<label>ID INSTALACION</label>
<input class="text" type="text" name="id_instalacion" id="id_instalacion" />
</div>
<div id="formulario">
<label>NOMBRE</label>
<input class="text" type="text" name="nombre" id="nombre_instalacion" />
</div>
<div id="formulario">
<label>IMAGEN</label>

```


Fig. 41: Mensajes de errores.

En el caso de que la inserción se realizará correctamente se nos mostraría un popup indicándolo.

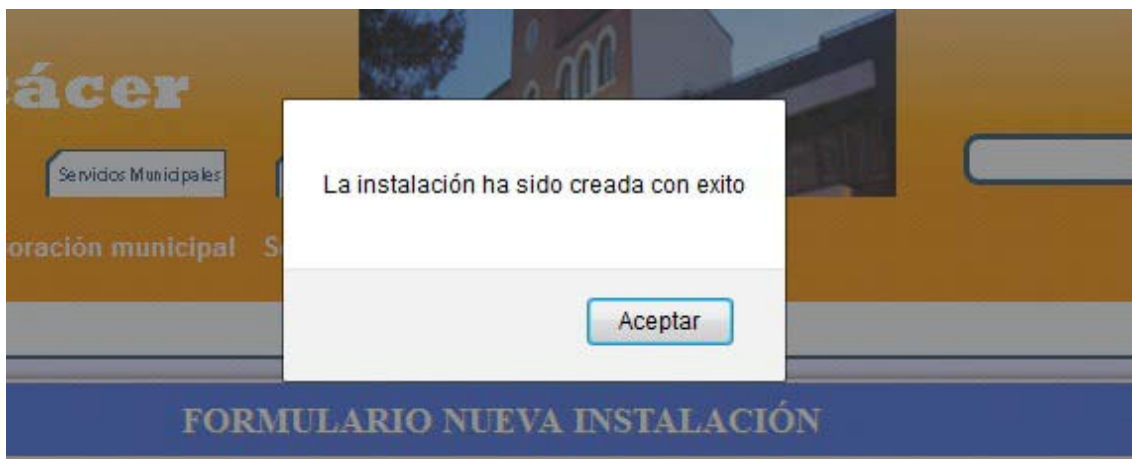


Fig. 42: Creación correcta.

- o actualizar.php, actualizarEvento.php, actualizarInstalacion.php, actualizarReserva.php y actualizarAutenticado.php

Mediante el icono editar que aparece al lado de cada elemento podremos editar algunos de los campos de dicho elemento, mediante una nueva página (actualizar.php) que contendrá un formulario con sus campos rellenos con los datos que contienen dicho elemento extraídos de la base de datos.



Fig. 43: Icono editar.

Fig. 44: Formulario de actualizar instalación.

Igual que en la opción de “Nuevo” hemos creado un array para almacenar los errores (\$error[]) que se pudiesen producir a la hora de actualizar el elemento y que se mostrarán encima de otro formulario que mantendrá los datos enviados anteriormente en el primer formulario de actualizar, y los recibirá mediante “POST”. Aquí tenemos el primer formulario con los datos leídos de la base de datos, correspondiente a la imagen anterior.

```

if(isset($_GET['id_instalacion'])){
    require('instalaciones.class.php');
    $objInstalacion = new Instalacion;
    $consulta = $objInstalacion->mostrar_instalacion($_GET['id_instalacion']);
    $instalacion = mysql_fetch_array($consulta);
}
?>
<form id="frmInstalacionActualizar" name="frmInstalacionActualizar" method="post" action="actualizarInstalacion.php" >
    <input type="hidden" name="id_instalacion" id="id_instalacion" value="<?php echo $instalacion['id_instalacion']?>" />
    <div id="formulario">
        <label>NOMBRE<em>*</em></label>
        <input class="text" type="text" name="nombre" id="nombre_instalacion" value="<?php echo $instalacion['nombre']?>" />
    </div>
    <div id="formulario">
        <label>IMAGEN</label>
        <input class="text" type="text" name="imagen" id="imagen_instalacion" value="<?php echo $instalacion['imagen']?>" />
    </div>
    <div id="formulario">
        <label>DESCRIPCIÓN</label>
        <textarea class="text" name="descripcion" id="descripcion_instalacion" cols="40" rows="3"><?php echo $instalacion['descripcion']?></textarea>
    </div>
    <div id="formularioboton">
        <input type="submit" name="submit" id="button" value="Guardar" />
        <input type="button" name="Cancelar" value="Cancelar" onClick="location.href='index_admin.php'" />
    </div>
</form>

```

Y aquí el código del segundo formulario, que mostrará los errores si los hubiese.

```

$id_instalacion = (isset($_POST['id_instalacion'])) ? trim($_POST['id_instalacion']) : '';
$nombre = (isset($_POST['nombre'])) ? trim($_POST['nombre']) : '';
$imagen = (isset($_POST['imagen'])) ? trim($_POST['imagen']) : '';
$descripcion = (isset($_POST['descripcion'])) ? trim($_POST['descripcion']) : '';

$error = array();
$objInstalacion=new Instalacion;

if(empty($nombre)){
    $error[] = 'El campo NOMBRE es obligatorio.';
}
?>
<div id="formularioErrores">
<?php
if(count($error)>0){
    foreach($error as $clave=>$valor){
        echo '<p><span style=color:red>'. $valor. '</p>';
    }
}
?>

<form id="frmInstalacionActualizar" name="frmInstalacionActualizar" method="post" action="actualizarInstalacion.php" >
<input type="hidden" name="id_instalacion" id="id_instalacion" value="<?php echo $id_instalacion?>" />
<div id="formulario">
<label>NOMBRE<em>*</em></label>
<input class="text" type="text" name="nombre" id="nombre_instalacion" value="<?php echo $nombre?>" />
</div>
<div id="formulario">
<label>IMAGEN</label>
<input class="text" type="text" name="imagen" id="imagen_instalacion" value="<?php echo $imagen?>" />
</div>
<div id="formulario">
<label>DESCRIPCION<em>*</em></label>
<textarea class="text" name="descripcion" id="descripcion_instalacion" cols="40" rows="3"><?php echo $descripcion?></textarea>
</div>
<div id="formularioboton">

```

En caso de que no hubiese errores, es decir que el array \$error estuviese vacío, este segundo formulario no se mostraría, se actualizaría directamente y se avisaría de la correcta actualización.

```

if(empty($error))
{
    if ( $objInstalacion->actualizar_instalacion(array($id_instalacion,$nombre,$imagen,$descripcion),$id_instalacion) == true)
    {
        echo "<script type='text/javascript'>alert('La instalacion ha sido actualizada con exito'); document.location='./index_admin.php';</script>";
    }
}

```

- o eliminar.php, eliminarEvento.php, eliminarInstalacion.php, eliminarReserva.php y eliminarAutenticado.php

Para poder eliminar un elemento deberemos presionar encima del icono que tiene como title (etiqueta flotante) Eliminar.



Fig. 45: Icono eliminar.

El código, igual que los anteriores, requiere de un objeto del tipo que vayamos a borrar, para poder utilizar la opción de borrado, esta se eliminará a partir de su clave primaria, en casi todos los casos su identificador. Seremos avisados que el borrado se ha conseguido satisfactoriamente o en caso contrario que no se ha podido borrar, como vemos en su código completo.



```

<?php
require('instalaciones.class.php');

$id_instalacion=$_GET['id_instalacion'];
$objInstalacion=new Instalacion;
if( $objInstalacion->eliminar_instalacion($id_instalacion) == true){
    echo "<script type='text/javascript'>alert('La instalación ha sido eliminada con éxito'); document.location='./index_admin.php';</script>";
}else{
    echo "<script type='text/javascript'>alert('Ocurrió un error.');

```

5.2.3. Implementación de la base de datos.

La implementación de la base de datos la hemos realizado mediante MySQL, debido a que viene integrado en el XAMPP y como es sabido tiene muy buena combinación junto con PHP y nos permite la utilización de su gestor de MySQL, PhpMyAdmin, programa desde el que gestionamos la creación y administración de la base de datos de manera intuitiva.

Hemos utilizado para el almacenamiento de las tablas el tipo “InnoDB” porque nos ofrece muchas ventajas, ya que cumple con el soporte ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), y es mejor a la hora de utilizar claves ajenas.

A continuación tenemos las diferentes tablas que componen nuestra base de datos, teniendo en cuenta lo indicado en la especificación de requisitos, realizando solo pequeñas modificaciones en algunas de ellas, para poder enlazar correctamente las claves ajenas y para un mejor uso de las claves primarias.

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> autenticado	[Icons]	~2	InnoDB	utf8_spanish2_ci	32.0 KB	-
<input type="checkbox"/> boletin	[Icons]	~0	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> evento	[Icons]	~4	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> imagen	[Icons]	~6	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> instalaciones_deportivas	[Icons]	~4	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> noticia	[Icons]	~3	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> oposicion	[Icons]	~0	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> pref_usuario	[Icons]	~0	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> reserva	[Icons]	~10	InnoDB	utf8_spanish2_ci	64.0 KB	-
<input type="checkbox"/> tramite	[Icons]	~0	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> tramite_autenticado	[Icons]	~0	InnoDB	utf8_spanish2_ci	48.0 KB	-

Fig. 46: Tablas de la base de datos.

6. Evaluación.

Para la evaluación y las pruebas de la implementación desarrollada para la aplicación se han llevado a cabo una serie de procedimientos para validar el correcto funcionamiento de la web a nivel general y también para comprobar en diferentes escenarios el comportamiento de la misma.

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

Hemos insistido en la visualización de los datos en diferentes dispositivos y resoluciones para comprobar un correcto funcionamiento en estas variaciones.

Principalmente hemos realizado las siguientes pruebas validando los siguientes aspectos de la aplicación:

- Visualización de la aplicación en diferentes navegadores y sistemas operativos.
- Visualización de la aplicación en diferentes resoluciones de pantalla y tipos de dispositivo.
- Validación de los enlaces de la aplicación.

6.1. Visualización en diferentes navegadores.

Hemos realizado pruebas en diferentes navegadores para observar el comportamiento de la aplicación en la visualización en cada uno de ellos. Los principales navegadores donde hemos realizado las pruebas son los siguientes:

- Mozilla Firefox 15
- Internet Explorer
- Google Chrome
- Safari

6.1.1. Mozilla Firefox 15.0.

Mozilla Firefox ha sido el navegador en el que desde un primer momento hemos ido realizando el desarrollo de la aplicación y en el que se ha adecuando a su visualización. Podemos afirmar que desde este navegador podremos visualizar correctamente la aplicación.



Fig. 47: Visualización en Mozilla Firefox.

6.1.2. Internet Explorer 8.

Con Internet Explorer hubo problemas al principio del desarrollo, los cuales fueron causa de la aplicación de las hojas de estilo CSS, ya que el navegador no reconocía correctamente algunas propiedades de las mismas.

Estos problemas fueron resueltos, por lo que la visualización de la web desde el navegador de Microsoft no difiere de la observada desde Firefox.

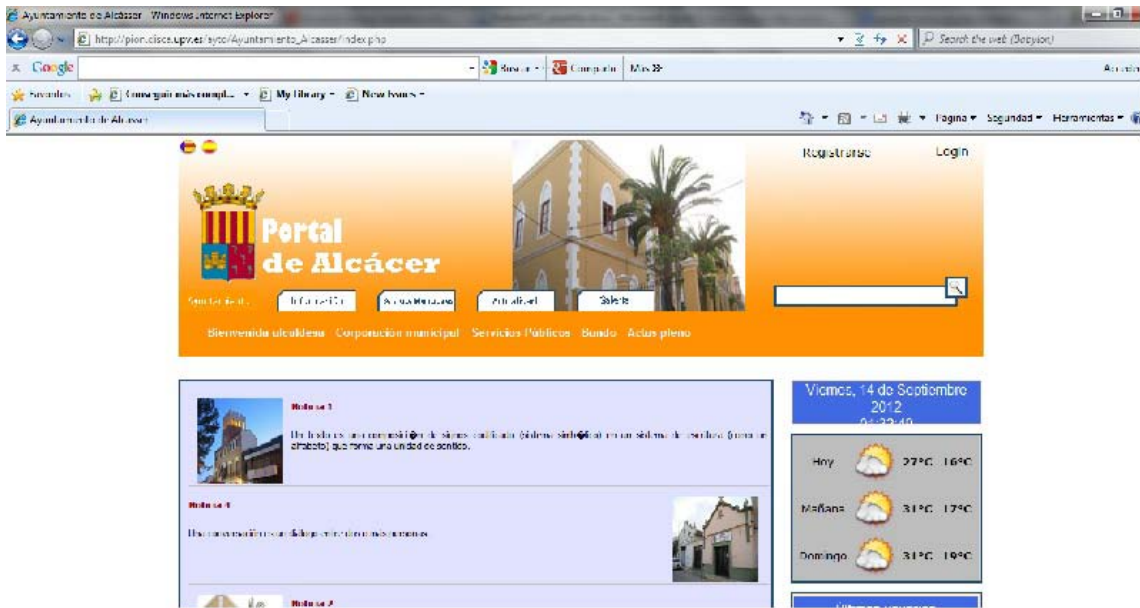


Fig. 48: Visualización con Internet Explorer 8.

6.1.3. Google Chrome.

Con Google Chrome también se pasaron satisfactoriamente las pruebas, integrando a la perfección el comportamiento de la aplicación dentro del navegador, ya que junto con el navegador Firefox fueron los navegadores en que a medida que la aplicación avanzaba, se iban realizando las pruebas.



Fig. 49: Visualización en Google Chrome.

6.1.4. Safari.

Con el navegador Safari tampoco observamos ningún comportamiento diferente en la aplicación respecto a los otros navegadores.

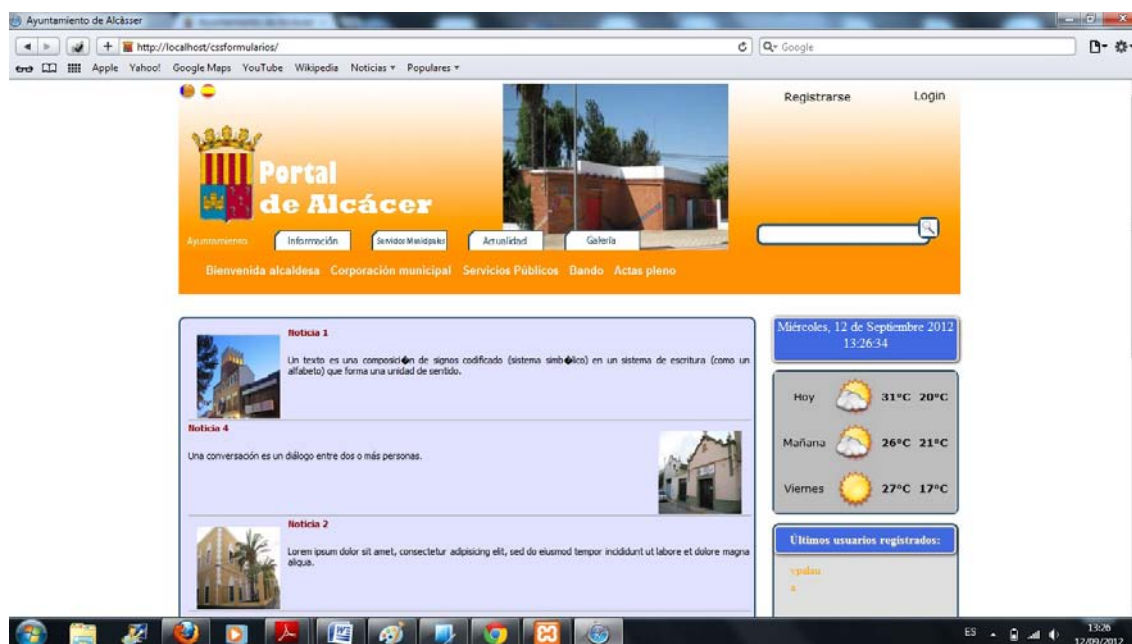


Fig. 50: Visualización en navegador Safari.

6.2. Visualización en diferentes resoluciones.

Debido a las resoluciones de pantalla que existen en la actualidad, esta aplicación se puede visualizar de forma óptima a partir de 1024 x768 en adelante. Con resoluciones inferiores se visualizaría de forma automática un scroll horizontal para poder navegar por la parte que no se visualiza debido a que no cumple con la anchura mínima de 960 píxeles.

Para poder realizar esta prueba hemos recurrido a la siguiente dirección “<http://testsize.com/>”.

El resultado que vamos a mostrar a continuación es el que corresponde a una resolución de 800 x 600, donde podemos observar en la parte inferior el scroll horizontal.



Fig. 51: Resolución 800 x 600.

Para poder visualizar la aplicación de forma correcta tendremos que tener una resolución de 1024 x 768, ya que la anchura del contenido de la web quedará por debajo del límite de la resolución horizontal y por tanto el usuario visualizará su contenido sin ningún scroll horizontal. Aplicando una resolución de 1024 x 768 se visualizará de la siguiente manera:



Fig. 52: Resolución 1024 x 768.

6.3. Validación de enlaces.

Para validar los enlaces, nos hemos encontrado con la problemática de no poder validar los enlaces de la parte privada de los usuarios registrados, ya que al validar los enlaces con el validador del W3C, <http://validator.w3.org/checklink> no tiene los permisos para visitar estas páginas, con lo cual solamente podrá comprobar los enlaces de la parte pública.

En la parte pública, analizó todos los enlaces para comprobar si había enlaces rotos o corruptos y el resultado de la validación fue el siguiente:

Results

List of broken links and other issues

There are issues with the HTML which break the web's structure. The table summarizes the issues and suggests actions by HTML developers. Show more

Code	Occurrences	What to do
500	2	This is a server side problem. Check the URI

Line: 69 <http://www.tuempoe.net/Tiempo-Alcasser-E4604a.html>
 Status: 302 -> 200 Server closed connection without sending any data back
 This is a server side problem. Check the URI

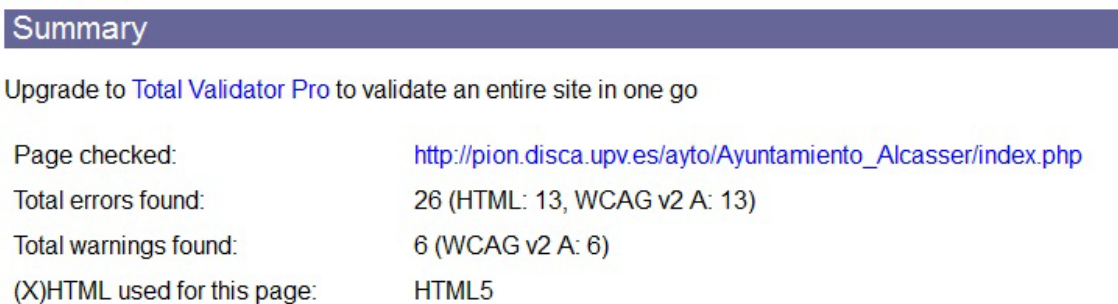
Line: 69 <http://www.tuempoe.net/Tiempo-Cspaña.htm>
 Status: 500 Server closed connection without sending any data back
 This is a server side problem. Check the URI

Fig. 53: Resultado de la validación de enlaces.

6.4. Validación de estándares.

Según lo comentado en el apartado de las tecnologías utilizadas a lo largo de la implementación de este proyecto, para la validación del código HTML y CSS se ha llevado a cabo a través de los estándares HTML5.0 y CSS.

Mediante el complemento para Mozilla Firefox llamado “Total Validator Tools” se ha podido ver si cumple o no con los estándares. Este complemento nos muestra en una pestaña si cumple o no con el estándar. El resultado de validación de nuestra aplicación ha sido la siguiente:



Summary	
Upgrade to Total Validator Pro to validate an entire site in one go	
Page checked:	http://pion.disca.upv.es/ayto/Ayuntamiento_Alcasser/index.php
Total errors found:	26 (HTML: 13, WCAG v2 A: 13)
Total warnings found:	6 (WCAG v2 A: 6)
(X)HTML used for this page:	HTML5

Fig. 54: Resultado de la validación de estándares..

7. Conclusiones.

El desarrollo de este proyecto en el ámbito de la web ha supuesto enfrentarse a nuevos retos que hasta entonces desconocíamos, como era el desarrollo de una web dinámica con inclusión de base de datos. Todo este proceso ha servido para ampliar los conocimientos en una parte que prácticamente no se da durante la carrera.

También ha supuesto tener que resolver muchos problemas a la hora de realizar tareas que anteriormente no habíamos realizado, como puede ser el aprender nuevos lenguajes de programación (PHP y MySQL) y la utilización de nuevas herramientas.

Otro de los problemas en que nos encontramos al principio fue cuando tratábamos de redactar la especificación de requisitos, debido a que no tuvimos ningún tipo de ayuda por parte del ayuntamiento para clarificar las posibles necesidades del portal y por este motivo tuvimos que visitar uno por uno los diferentes portales de los pueblos vecinos e inspirarnos en algunos de ellos.

Destacar que una vez finalizado el proyecto quedan facetas de la aplicación pendientes de aplicar en un futuro, ya que por motivos de tiempo no se han podido incluir en esta implementación. Uno de los aspectos pendientes sería la implementación del usuario interino para poder tener una correcta delegación de tareas, ya que este tipo de usuarios serían trabajadores del ayuntamiento pero cada uno se encargaría de una sección, por ejemplo, juventud, deportes, etc.

Para finalizar tenemos que decir que este ha sido el trabajo más complejo durante la carrera, ya que partíamos de escasos conocimientos de los lenguajes de programación que en la parte de implementación tuvimos que utilizar y tras pelear día tras día ha sido posible poner fin a este proyecto.

También nos gustaría decir que ha sido una lástima por parte del ayuntamiento haber mostrado tan poco interés en este proyecto, ya que hubiera sido más

satisfactorio para nosotros poder llegar a ponerlo en marcha y que los ciudadanos de la población tuvieran alguna motivación para poder visitar el portal web del ayuntamiento.

Si una cosa nos ha quedado clara una vez terminado el proyecto, es que hace falta mucho esfuerzo y dedicación para conseguir la finalidad que buscábamos. Para conseguirlo hemos tenido que documentarse mucho, bien vía web y también mediante algunos libros, para poder realizar en condiciones este proyecto. Ha sido un reto difícil, pero pensamos que ha valido la pena, ya que sobretodo hemos aprendido mucho con esta experiencia.

8. Bibliografía.

- Buendía, F.
Una guía para la realización y supervisión de proyectos final de carrera en el ámbito de la web
Ed. UPV, 2008
- Timothy Boronczyk, Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass.
Desarrollo Web con PHP 6, Apache y MySQL.
Anaya Multimedia/Wrox, 2009.
- Eguíluz, J.
Introducción a CSS
<http://www.librosweb.es>, 2009
- Eguíluz, J.
CSS avanzado
<http://www.librosweb.es>, 2009
- Eguíluz, J.
Introducción a JavaScript
<http://www.librosweb.es>, 2009
- Dan Wellman
JQuery 1.4 Animation Techniques
Packt Publishing Ltd., 2011
- Jonathan Chaffer, Karl Swedberg
Aprende JQuery 1.3.
Anaya Multimedia, 2010.
- Curso de introducción a PHP y MySQL
Apuntes del curso
CFP-UPV, 2012