

# Documentación El Localero

## PHP

La estructura de ficheros se organiza de la siguiente manera:

index.php: incluye la página de portada, y se sirve de llamadas para imprimir las noticias.

Concretamente hace uso de la función `imprimeNoticia()` del fichero `imprimeNoticia.php`

- **noticia.php**: Contiene una plantilla que será rellena con el `idNoticia` que se le pasará por GET. Además incluye el código para ejecutar Ajax, encargado de procesar los comentarios de forma asíncrona, y JQuery con tooltips para que muestre la ventana flotante con las últimas noticias relacionadas.
- **seccion.php**: Según el `idSeccion` que se le pase por Get, así cambiará su apariencia e imprimirá las noticias correspondientes a dicha sección.  
En últimas noticias, seguirán saliendo de otras secciones, ya que las que salen por seccion están ordenadas por antigüedad.
- **nuevoUsuario.php**: Contiene el formulario para el alta de un nuevo usuario. Además contiene la validación de ficheros mediante JQuery Validator.
- **registrarUsuario.php**: Es el encargado de registrar el usuario en la base de datos, y mostrar un mensaje de bienvenida a la nueva incorporación.
- **nuevaNoticia.php**: Contiene el formulario para la creación de la nueva noticia, y permite la subida de imágenes.
- **registrarNoticia.php**: Guarda los datos en la base de datos, y muestra una vista de como ha quedado la noticia.
- **modificarNoticia.php**: recibirá por GET el `idNoticia` que modificar, y rellenará los campos del formulario tal que puedan editarse. Al enviar el formulario, llama a `registrarNoticia.php` de nuevo y vuelve a mostrar como quedó finalmente la noticia. También permite cambiar la imagen.
- **borrarNoticia.php**: recibirá por POST el `idNoticia` de la noticia a borrar, y la borra de la BD.

- **validarLogin.php**: Es el encargado de validar los datos enviados por el menu emergente de logeo. Si son correctos inicia sesión, sino, notifica el fallo.
- **cerrarLogin.php**: Cierra la sesión del usuario, eliminando las variables de sesión, y vuelve a usuario a la página donde se situaba.
- **imprimeNoticia.php**: Contiene 4 funciones encargadas de imprimir noticias:
  - `imprimirNoticias($idNoticia)`: imprime la noticia pasada en formato portada.
  - `imprimirNoticiasSeccion($idNoticia)`: imprime la noticia en formato sección
  - `imprimirUltimasNoticias()`: imprime las 5 últimas noticias insertadas en la BD.
  - `imprimirNoticiasRelacionadas($nombreSeccion)`: imprime las 5 últimas noticias de una sección concreta.
- **imprimeSecciones.php**: Contiene una función encargada de imprimir todas las secciones en forma de lista no ordenada, permitiendo así introducir nuevas secciones en cualquier momento.

## Cargado de noticias dinámico

He modificado la estructura de directorios que venía indicada en la práctica 1, ya que ahora, genero todas las noticias de forma dinámica, es decir, tengo una plantilla de noticias, y sobre esa, cargó la noticia correspondiente, indicada por su "idNoticia" único para cada uno de ellas.

En la página principal siempre van a cargarse las noticias de forma dinámica, es decir, se ordenan de mayor a menor antigüedad. Por tanto se mostrarán las noticias más actuales. Una de cada sección, y para mantener la estructura hemos tenido que de todas las noticias extraídas, vamos buscando cual de ellas es la que nos falta para completar nuestro array.

En las páginas de categorías pasará exactamente igual. Al comienzo del cargado de la página, se cargan las noticias correspondientes a dicha sección por orden de más a menos actuales, y se rellenará con las 5 primeras páginas que se hayan introducido.

### DISTRIBUCIÓN DE LAS NOTICIAS EN EL CARGADO

Cuando se hace la consulta para las noticias que se van a cargar, en la página principal mostramos la última noticia de cada sección. En las secciones, mostramos la última noticia como destacada, y el resto ordenadas en fecha descendente.

## Diseño de la BD

Con nuestro objetivo de generar todo el contenido de forma dinámica, hemos creado varias tablas dedicadas a nuestro propósito. Esto nos permite una plataforma ya pensada en ampliar sus funciones y contenido.

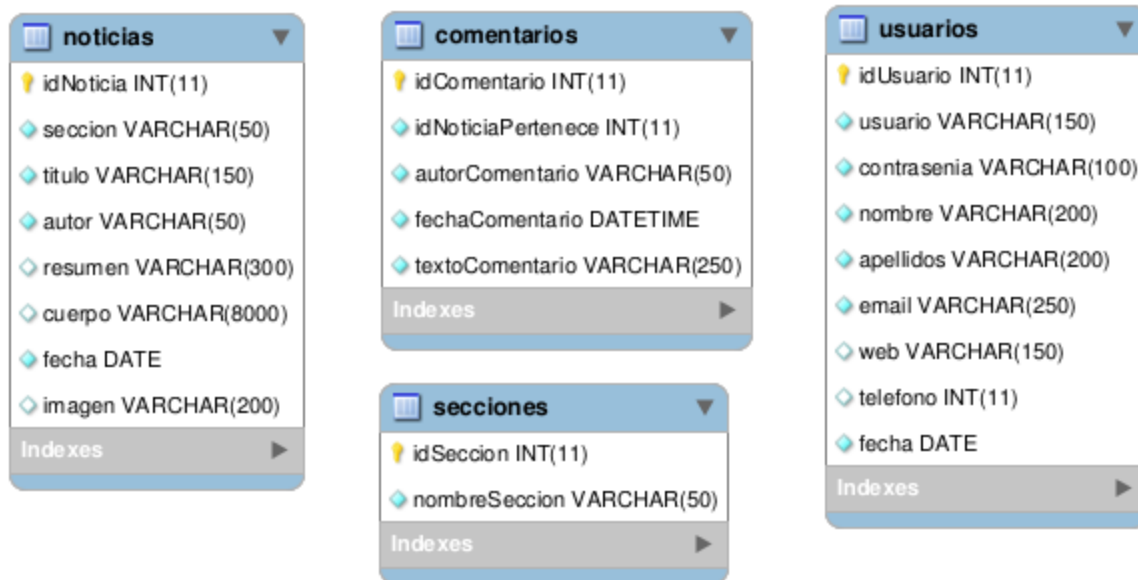


Table	Columns
noticias	idNoticia INT(11), seccion VARCHAR(50), titulo VARCHAR(150), autor VARCHAR(50), resumen VARCHAR(300), cuerpo VARCHAR(8000), fecha DATE, imagen VARCHAR(200)
comentarios	idComentario INT(11), idNoticiaPertenece INT(11), autorComentario VARCHAR(50), fechaComentario DATETIME, textoComentario VARCHAR(250)
usuarios	idUsuario INT(11), usuario VARCHAR(150), contrasenia VARCHAR(100), nombre VARCHAR(200), apellidos VARCHAR(200), email VARCHAR(250), web VARCHAR(150), telefono INT(11), fecha DATE
secciones	idSeccion INT(11), nombreSeccion VARCHAR(50)

Las noticias, los usuarios, los comentarios, y las secciones tienen id autoincrementables. El usuario se identifica con usuario, y no con id. El id es para la identificación en la BD, pero las comprobaciones se hacen con usuario.

A las noticias se les relacionan los comentarios incluidos. Igualmente los comentarios se asocia con el usuario que los escribe.

**class ConexionDB**, facilitando las cosas, con la conexión a base de datos.

Para facilitar la interacción con la BD he creado un objeto llamado ConexionDB, implementado en el archivo "conexionDB.php", que consta de varias funciones para simplificar un poco las consultas e inserciones.

Principalmente hay dos tipos de métodos en mi clase: los de inserción, modificación y borrado; y por otra parte están los de consulta. Es decir, los que esperan y los que no resultados.

### getConsulta(String \$consulta)

Este método recibe una cadena con la consulta, capta los datos y los devuelve en forma de PDO\_Statment. Sería similar a query().

#### insertarConsulta(String \$consulta)

Este método recibe una cadena con la consulta y devuelve true o false si la consulta ha sido realizada con éxito. Es conveniente solo usarlo con las consulta de tipo INSERT ya que de otra forma no devolvería resultado.

#### getConsultaArray(String \$consulta)

Es similar a getConsulta, solo que el resultado lo devuelve en forma de array con una tupla por fila.

Esta consulta internamente se hace mediante prepare() y execute(), y transformando mediante fetchAll().

## JavaScript:

Principalmente, para JavaScript he usado JQuery, por objeto de estudio para el seminario, y por facilidad y rendimiento en su uso.

Ademas de los .js algunos script estan incrustados en el código html, por no ser muy extensos y por su mejor compresión, como es el caso del procesamiento de AJAX, o el despliegue de algunas ventanas flotantes como el tooltips últimas noticias.

También está incrustado el código de validación del formulario.

Los ficheros son:

- **aspectoAdmin.js**: Encargado de adecuar el entorno a la vista de administrador, como aparecer botones y dar permisos de edición y borrado. Además de habilitar la opción de nueva noticia.
- **aspectoUser.js**: Encargado de adecuar la vista, a la de usuario, permitiéndole comentar, por ejemplo.
- **ventanaLogin.js**: Encargado de hacer correctamente el despliegue del login.

## Novedades:

### Login desplegable

Para el Login desplegable he usado un aside oculto, que cuando se hace un click sobre el enlace "Login" se llama a un JavaScript, el cual, con JQuery visualiza la caja con toggle() o la seconde con hide().

### Cargado de imagenes

La creación de noticias permite la subida de imagenes a la carpeta upload del directorio en bahía.

Además cada noticia guardada en BD lleva asociada su imagen correspondiente, la cual se le asigna en el momento de su creación.

En modificarNoticia.php podemos cambiarla a nuestro antojo, viendo antes una visión preliminar de que imagen es la que se está usando actualmente.

## JQuery

Con jQuery consigo visualizar y ocultar todos los elementos que deseo, lo que me permite cambiar el aspecto de la página según el usuario que haga login.

Todos los elementos suelo tenerlos ya en la página, solo que los muestro cuando deseo. Esto no es así para los comentarios, ya que los genera JQuery conforme vamos insertando, todo gracias a Ajax, como veremos más adelante.

Además me sirvo del plugin de JQuery, validator() para la validación de formularios.

## HTML

Hemos usado un nuevo marcador incorporado en html5, el marcador <button> que nos da más versatilidad a la hora de diseñar y adaptar botones a nuestras necesidades

## Redirección hacia atrás.

Cuando se hace el envío para el procesado de algún formulario, se vuelve a la página donde estaba gracias a `header("Location:".$_SERVER['HTTP_REFERER']);`

## AJAX.

La inserción de comentarios se hace de forma asíncrona, es decir, puedes insertar comentarios sin recargar la página. Uso la función ajax() de JQuery que me permite mandar el comentario a base de datos con otro php sin moverme de la página donde me sitúo.

Tooltips.

Me he servido de una función en JQuery UI llamada tooltips() que te permite generar un tooltips de forma sencilla con los enlaces de las últimas noticias.

## Icono en pestaña.

He hecho y añadido un icono para que aparezca en la pestaña de El Localero.

## CURIOSIDADES

El AdBlock me bloquea la publicidad de mi propia página.

He encontrado una biblioteca llamada FPDF muy fácil de usar, con la que iba a hacer un formulario para generar la documentación. Queda como posible mejora. <http://www.fpdf.org/>

## MEJORAS A IMPLANTAR

- Responsive design mobile.
- Generar las noticias, en función a cambiar de sección con Ajax (fácil de hacer, pero no se si será muy eficiente).
- Vista preliminar de una noticia solo con situar el ratón (Me gusta más que poner las noticias relacionadas).

- Panel de control y visualización de estadísticas para el usuario.
- Borrado de noticias con ajax.
- Login con ajax.

## SENSACIONES

Pués creo que como la mayoría de los compañeros piensan, ha sido una de las asignaturas más enriquecedoras en lo que llevo de grado. He disfrutado mucho programando esta plataforma y me ha dado pie ni más ni menos, que a enfocar mi proyecto fin de carrera a estas tecnologías.

Me ha abierto un gran abanico de posibilidades en el ámbito personal y profesional, capacitandome a hacer muchas cosas. Pero lo más importante es que he aprendido a aprender, es decir, he aprendido a coger una herramienta o un lenguaje nuevo, y poder hacer cosas con ellos con relativa facilidad, siguiendo unos criterios que antes no creí importantes, como la consulta a las API, el uso de Frameworks etc.

Siguiendo con los frameworks he aprendido herramientas como CSS, FPDF y por supuesto JQuery, que me han hecho ver, que es importante saber de que herramientas disponemos ya los informáticos para trabajar de forma eficiente, y evitar así “matar moscas a cañonazos”.

Por supuesto, esta plataforma la compartiré con mucho orgullo en mi GitHub (algo que he aprendido también) y que me va a hacer empezar con muchas más ganas mi primer contacto con el mundo laboral como informático. ya que mi principal tarea va a ser desarrollo web.

## REFERENCIAS

1. Manual de JQuery: <http://www.desarrolloweb.com/manuales/manual-jquery.html>
2. Tooltips: <http://jquerytools.org/documentation/tooltip/index.html#configuration>
3. JQuery Validatio: <http://jqueryvalidation.org/>
4. Algunos diseños CSS de la comunidad: <http://cssdeck.com/>
5. Tendencias en diseño CSS: <http://www.cssblog.es/>
6. AJAX: <http://elpoli.delphiaccess.com/php-validar-usuario-y-contrasena-usando-ajax/>
7. LibreriaFPDF para generar pdf a partir de funciones php(para funturo desarrollo):
8. <http://aprendeaprogramarenphpya.com/blog/generando-facilmente-pdf-en-linea-con-php.html>
9. Para una buena documentación de la WEB: <http://www.phpdoc.org/>