**[0 HERRAMIENTAS:](#T0)**

[**1 PRIMEROS PASOS:**](#T1)

[**1.1 BASE DE DATOS:**](#T11)

[**1.1.1 ENTRADAS:**](#T111)

[**1.1.2 IMÁGENES:**](#T112)

[**1.1.3 COMENTARIOS:**](#T113)

[**1.1.4 LOGIN:**](#T114)

[**1.1.5 ENTRADAS\_ETIQUETAS:**](#T115)

[**1.1.6 ETIQUETAS:**](#T116)

[**1.1.7 MEDIANTE IMPORTACIÓN DE ARCHIVO SQL:**](#T117)

[**1.2 USUARIOS SQL:**](#T12)

[**1.2.1 USUARIO ARTMANIA:**](#T121)

[**1.2.2 USUARIO MINDUNDI:**](#T122)

**[1.3 ESTRUCTURA DE DIRECTORIOS Y ARCHIVOS:](#T13)**

**[1.3.1 CARPETA USUARIOS:](#T131)**

**[1.3.2 CARPETA FRAMEWORKS:](#T132)**

**[1.3.3 CARPETA CROQUETTE99:](#T133)**

**[1.3.4 CARPETA 00BBDDs:](#T134)**

[**2 COMIENZO CON EL BACKEND:**](#T2)

[**2.1 HEAD DE LAS PÁGINAS BACKEND:**](#T21)

[**2.2 INDEX.PHP BACKEND:**](#T22)

[**2.3 OLVIDO.PHP BACKEND:**](#T23)

[**2.4 ADMINISTRACION.PHP BACKEND:**](#T24)

[**2.5 ADMINISTRACION2.PHP BACKEND:**](#T25)

[**3 FRONTEND:**](#T3)

[**3.1 HEAD DE LAS PÁGINAS FRONTEND:**](#T31)

[**3.2 INDEX.PHP DEL FRONTEND:**](#T32)

[**3.3 GALERIA.PHP DEL FRONTEND:**](#T33)

[**3.4 PORFOLIO.PHP DEL FRONTEND:**](#T34)

[**3.5 SOBREMI.PHP DEL FRONTEND:**](#T35)

[**3.6 LOGIN.PHP Y REGISTRAR.PHP DEL FRONTEND:**](#T36)

**4 NO CONTEMPLADO:**

**0 HERRAMIENTAS:**

Para el trabajo he creado una nueva máquina de Windows7.

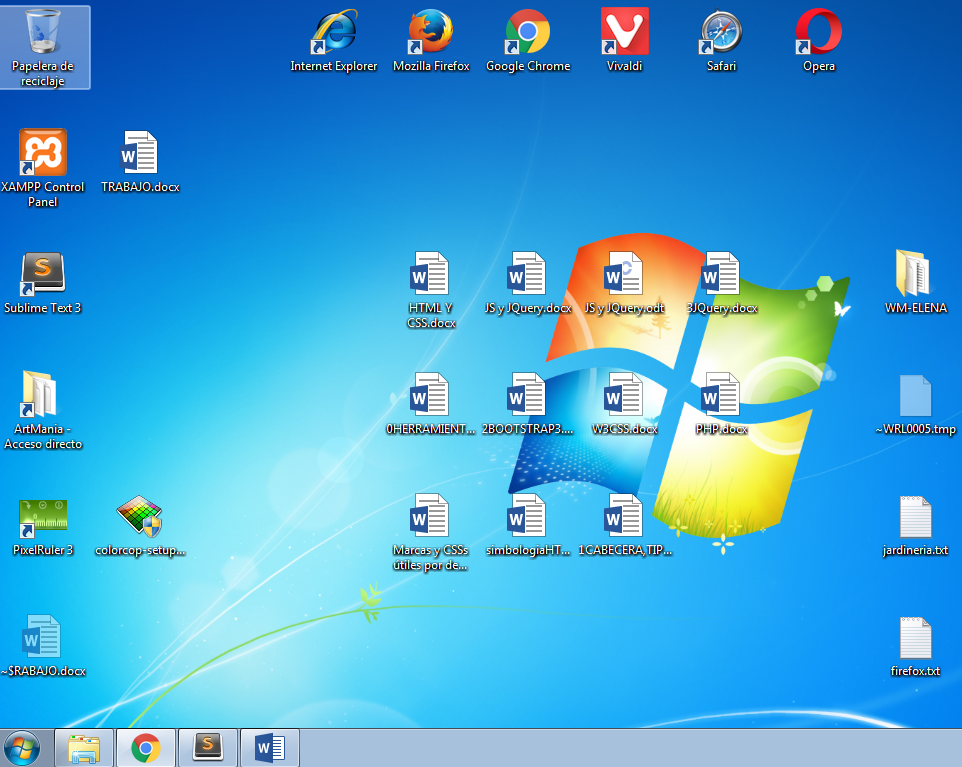
Como obligatorio:

* He instalado XAMPP y los navegadores más utilizados.

Opcionalmente he instalado:

* Notepad++ y sublime text para crear el código.
* Winrar y el office 2013 para poder ir documentando todo.
* PixelRuler y ColorCop que son herramientas de diseño, una para pixeles y otra para colores.

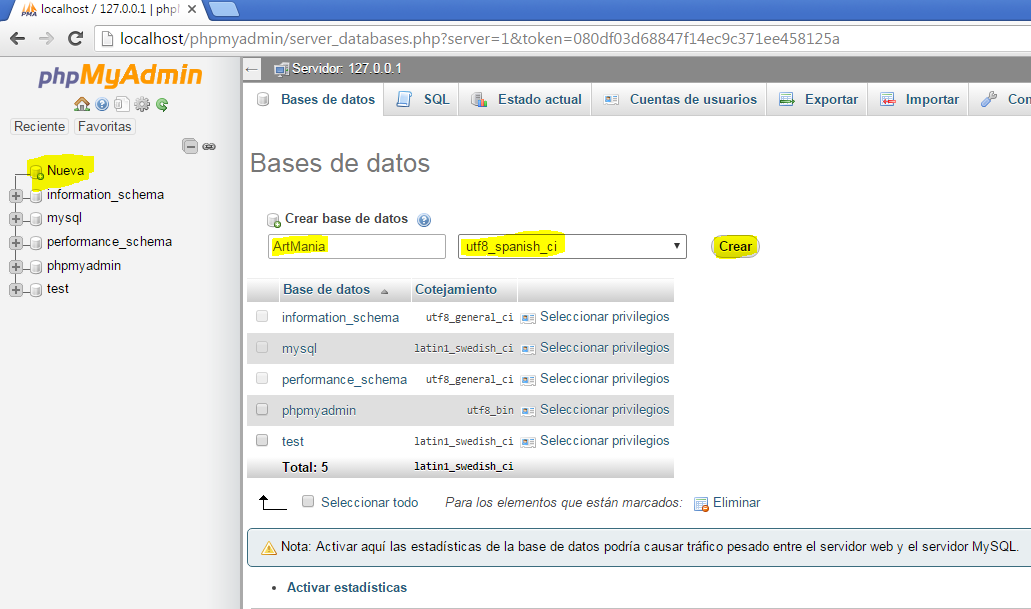
El resultado es el siguiente:



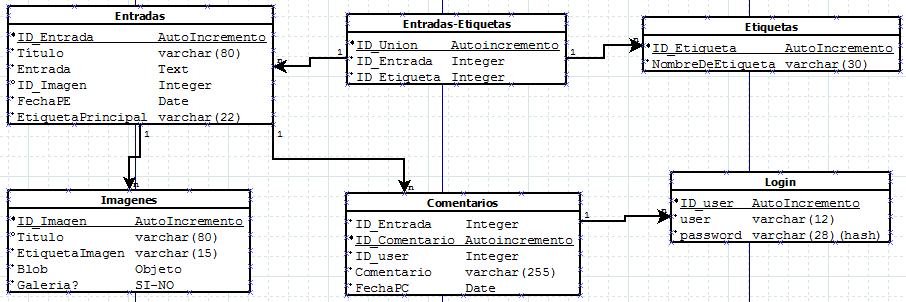
**1 PRIMEROS PASOS:**

**1.1 BASE DE DATOS Y USUARIOS:**

Entrando en localhost/phpmyadmin voy a crear la base de datos ArtMania que va a ser el nombre provisional de la aplicación web.

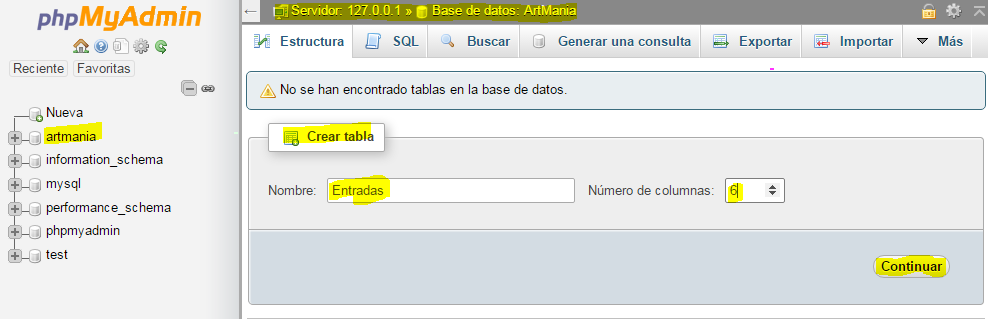


Ahora tengo que crear todas las tablas con sus campos hasta lograr tener el esquema que tengo del proyecto que es el siguiente:

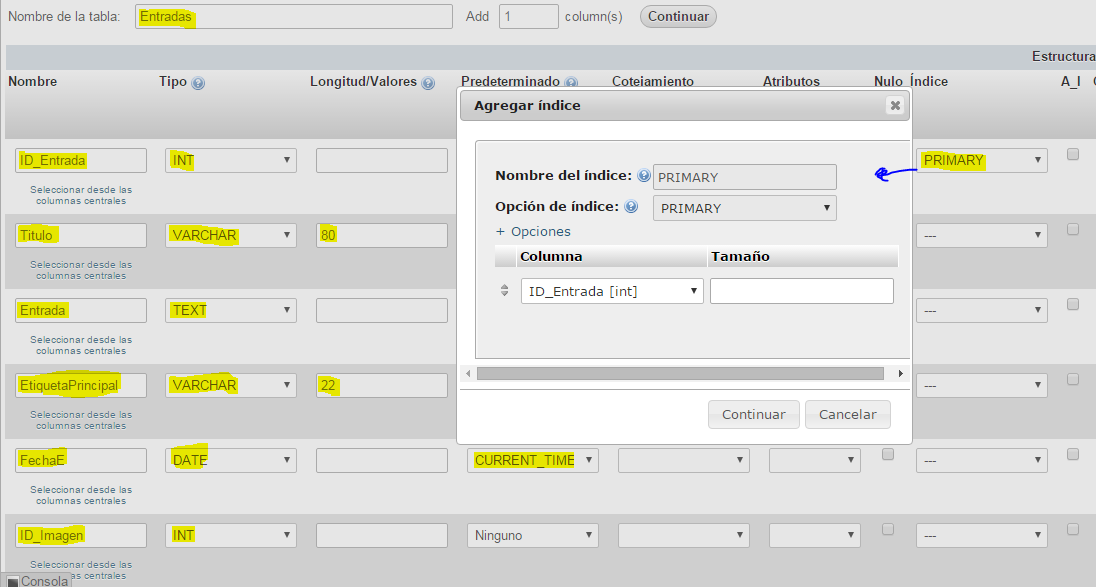


**1.1.1 ENTRADAS:**

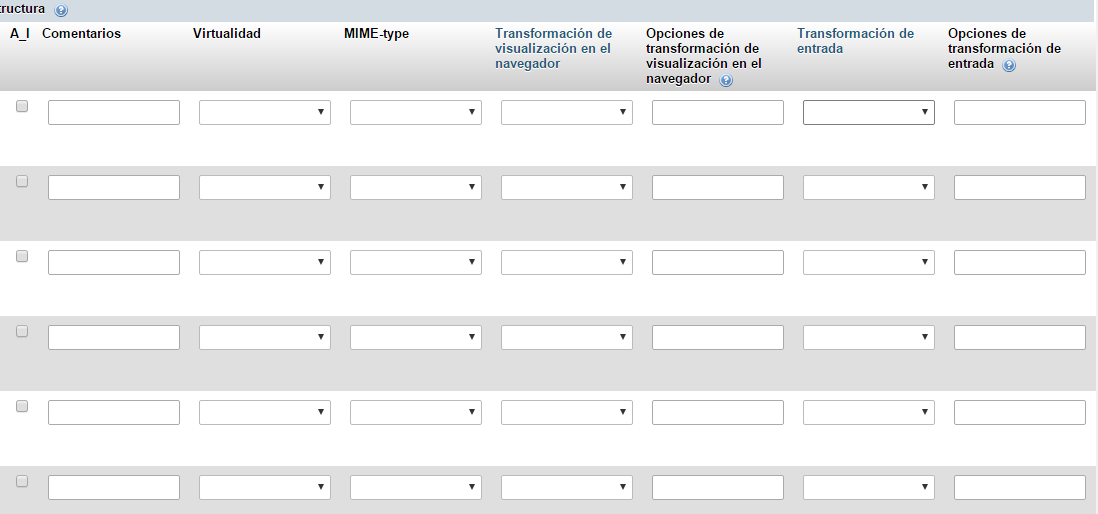
Primero le decimos la tabla que vamos a crear y los campos que contiene.



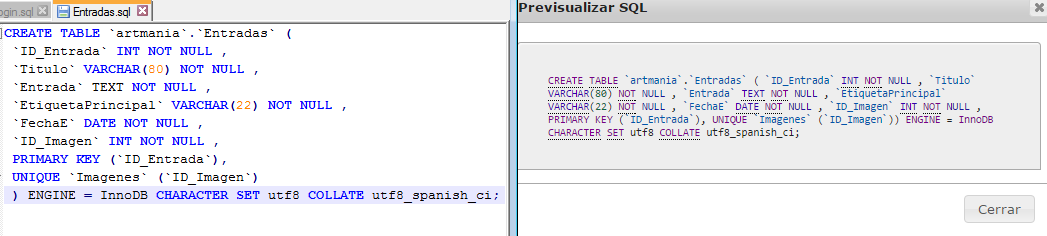
Después definimos el nombre de los campos, el tipo de dato que manejan, su longitud y establecemos las demás condiciones, como establecer el ID\_Entrada como campo primario y el ID\_Imagen como único, dado que solo va a haber una imagen por entrada.



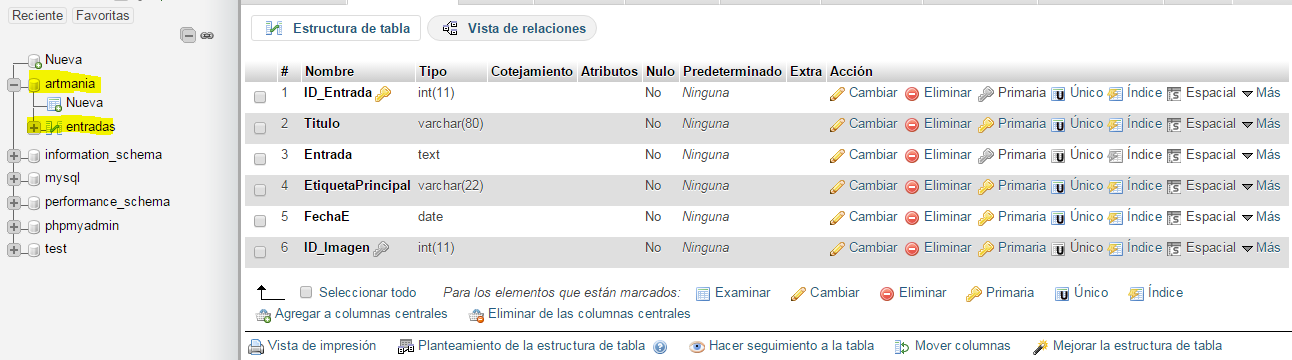
He dejado los demás campos en blanco:



Este es el código SQL que se generaría para crear la Tabla Entradas dentro de la Base de Datos de artmania.

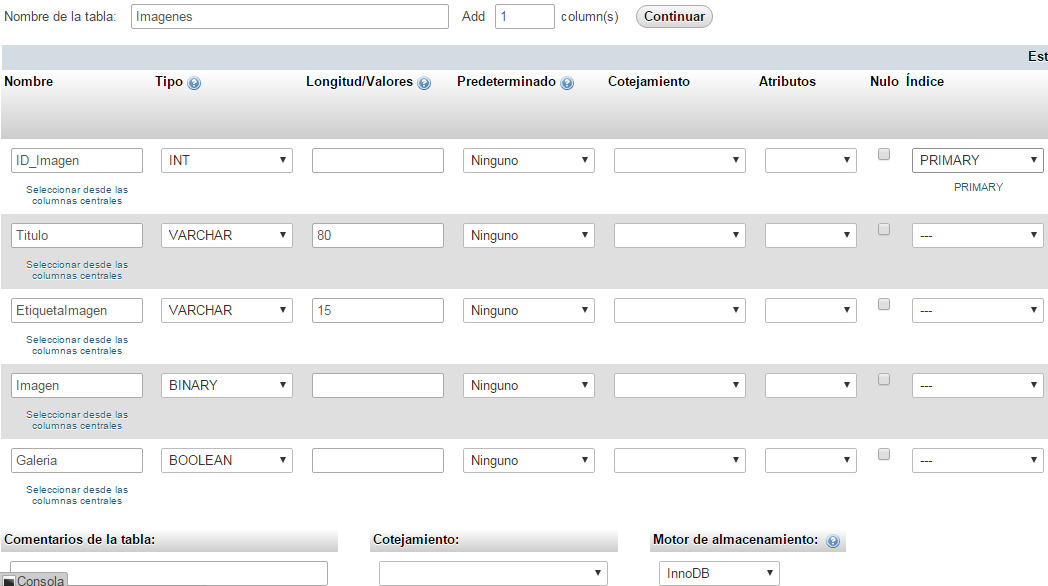


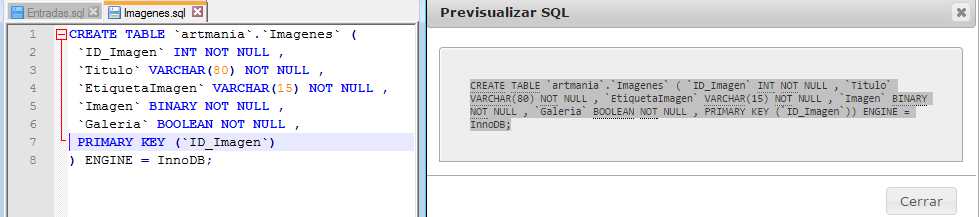
Una vez le demos a guardar nos aparecerá ya la tabla creada con todos los campos.

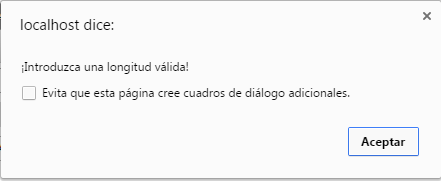


**1.1.2 IMAGENES**

Creamos también imágenes con la clave primaria del ID\_Imagen.

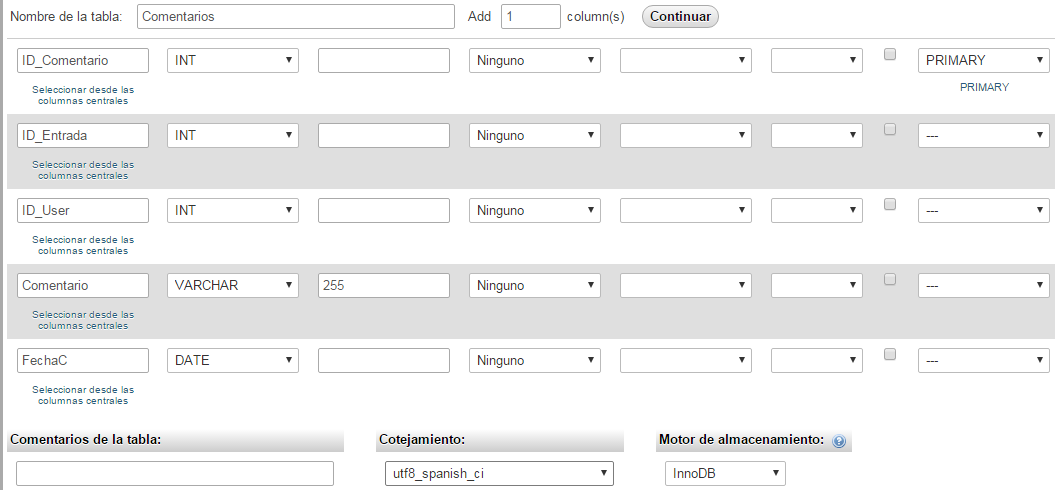


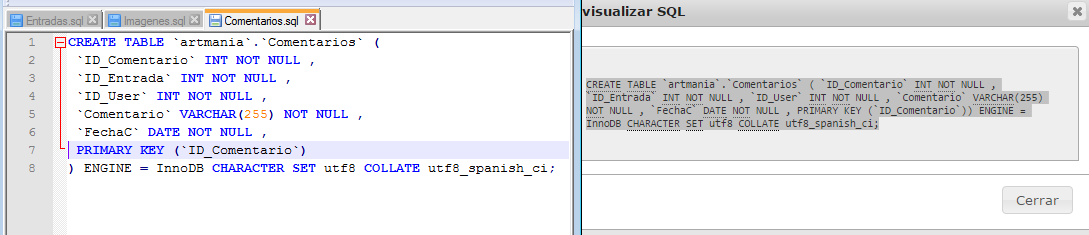




**1.1.3 COMENTARIOS:**

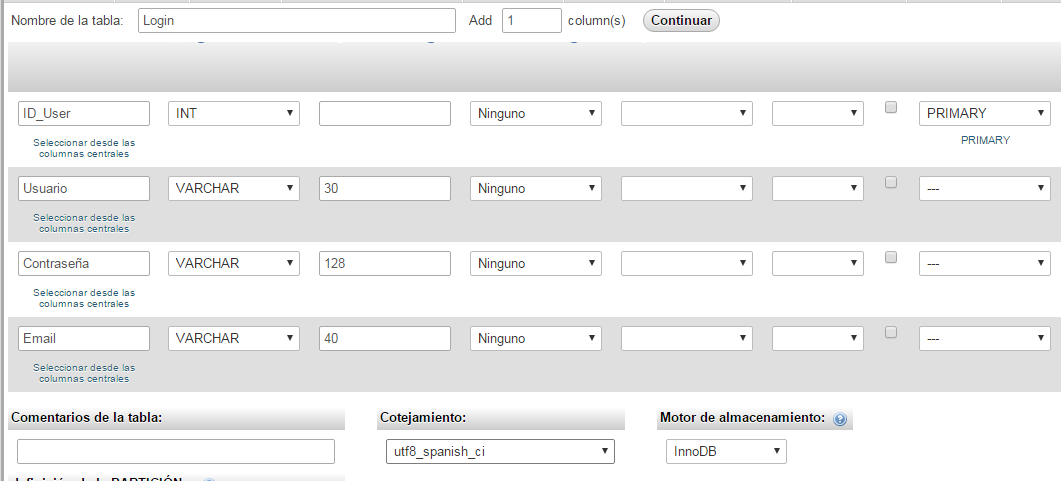
Comentarios es una tabla con más claves que campos propios, pero nos sirve para relacionar los usuarios que los escriben y nos da la capacidad de poner varios comentarios en una sola entrada y ordenarlos por la fecha en la que han sido creados, el campo ID\_Comentario identifica cada comentario de forma única, el campo ID\_Entrada lo relaciona con la entrada que se ha comentado y el campo ID\_User lo relaciona con el usuario que ha escrito y publicado el comentario.

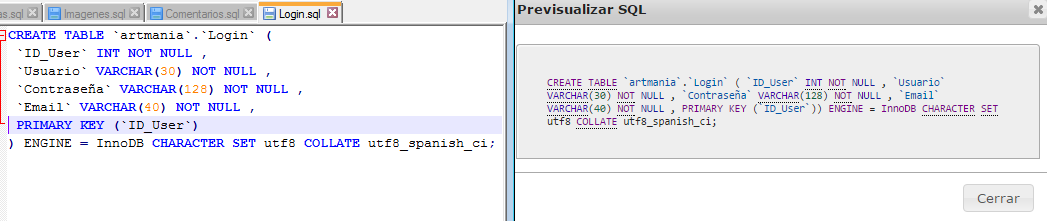




**1.1.4 LOGIN:**

Esta será la tabla de validar a los usuarios, en ella habrá un ID\_User que identifica de forma única a cada usuario, el nombre de ese usuario y su contraseña con la que se identifica y el email que deja para recuperar la contraseña o informarle de alguna novedad. La contraseña intentaré cifrarla con el algoritmo resumen SHA512 que devuelve justo 128 caracteres, sino tendré que intentarlo con SHA128 que devuelve 64 y modificar el tamaño del campo.

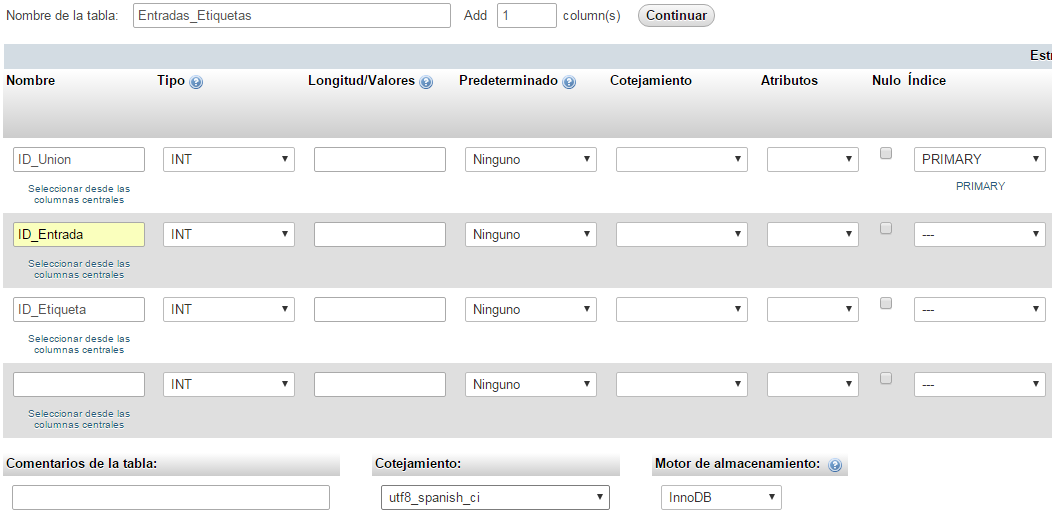


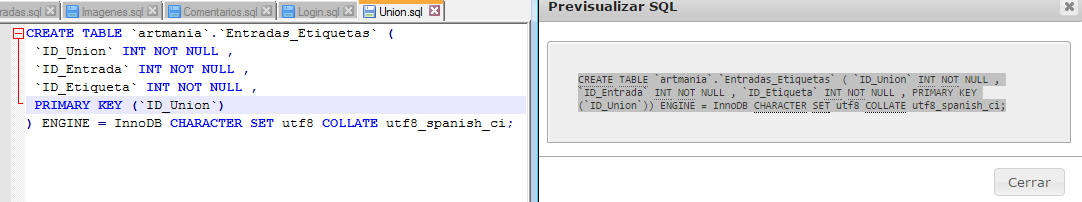


**1.1.5 ENTRADAS\_ETIQUETAS:**

Esta tabla solo servirá de unión entre las dos de las que se compone su nombre, Entradas y Etiquetas, por ello solo tiene un campo ID\_union que identifica cada unión entre Entrada y Etiqueta de forma única y dos campos ID\_Entrada e ID\_Etiqueta que son las claves primarias de sus respectivas tablas que unen uno a otro.

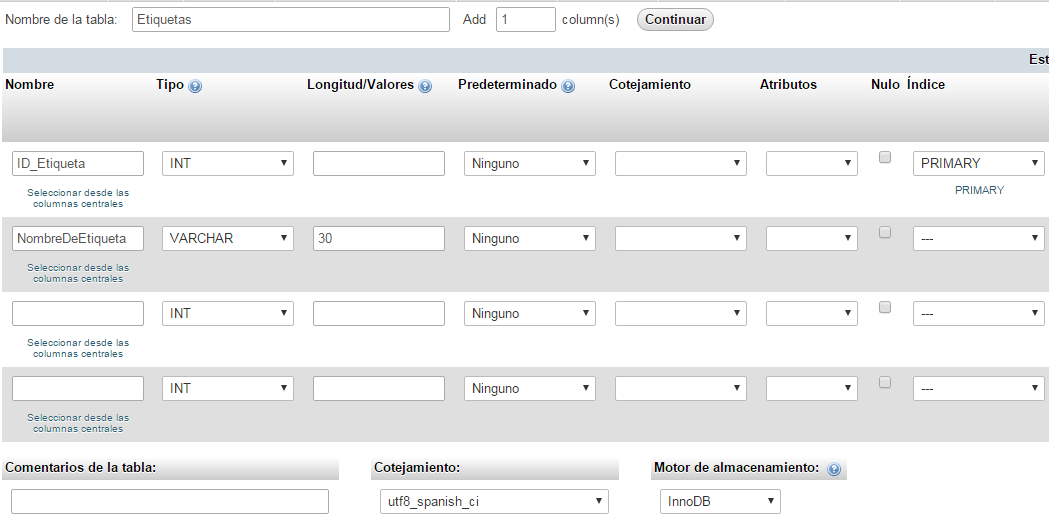
Esta tabla permite una relación muchos a muchos en la cual una entrada puede tener varias etiquetas y una etiqueta puede enlazar varias entradas. Dado que estas etiquetas las crea el administrador de la aplicación, esta tabla de unión es necesaria.

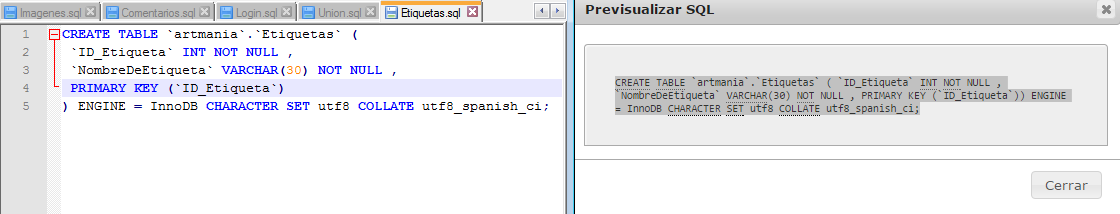




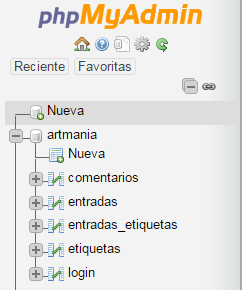
**1.1.6 ETIQUETAS:**

Esta tabla tiene el identificador único de cada Etiqueta junto con su nombre, al fin y al cabo las etiquetas solo van a servir para filtrar las entradas, no necesitan más información adicional.



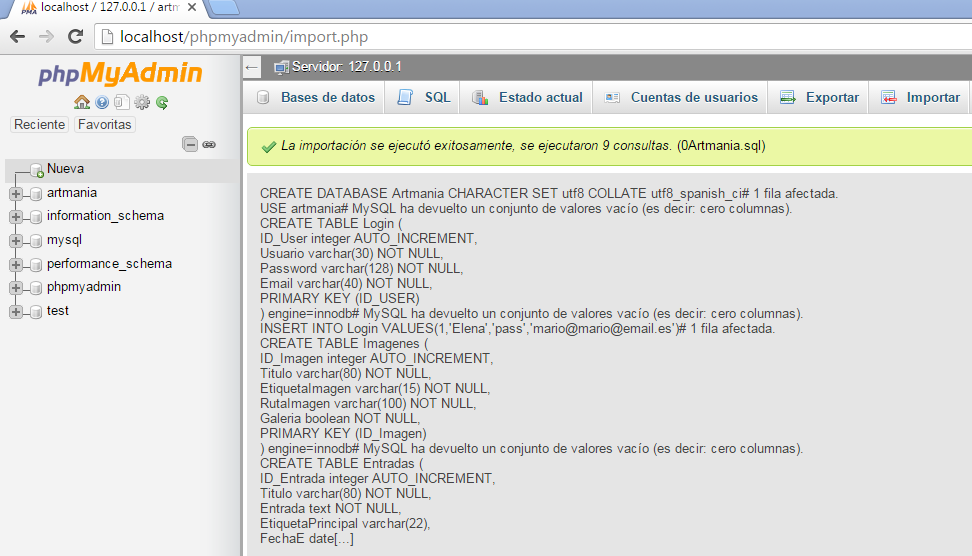


Se han creado todas excepto la de Imágenes:



**1.1.7 MEDIANTE IMPORTACIÓN DE ARCHIVO SQL:**

Como he tenido problemas con la tabla imágenes y además al poner las relaciones entre las tablas he borrado la base de datos y la he creado en código, ahora solo queda importar el archivo 0Artmania.sql que se encuentra en la carpeta 00BBDDs para que nos cree la base de datos, las tablas y sus relaciones:



Como podemos ver ahora tenemos la base de datos con todas las tablas.

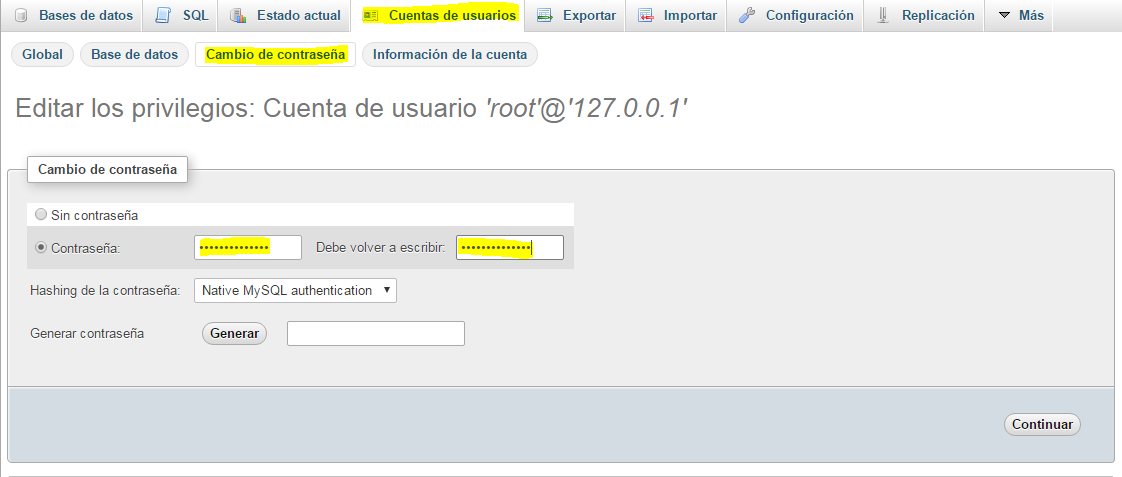


**1.2 USUARIOS SQL:**

Como podemos ver todos los usuarios de la base de datos no tienen contraseña, incluido root, lo cual es una gran vulnerabilidad.



Antes de crear los usuarios para la base de datos de la aplicación, vamos a ponerle a root una contraseña fuerte, puesto que no será el usuario que utilizaremos y así evitamos que alguien adquiera permisos para realizar actividades maliciosas. La contraseña que he puesto es JARdineria@#00.



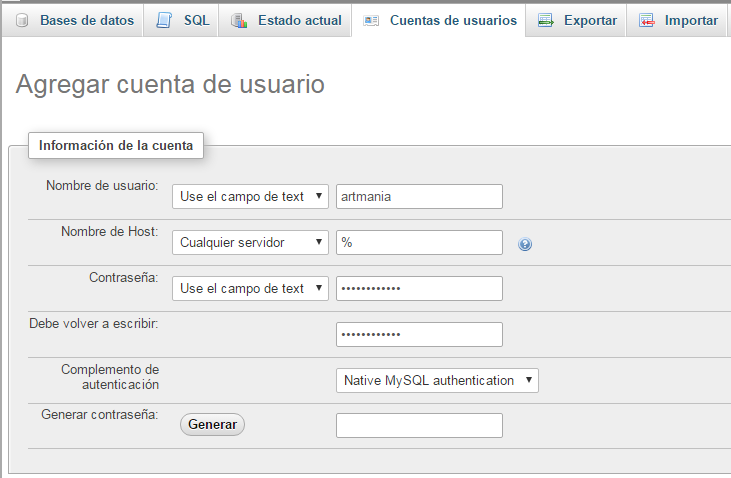
Ahora todos los usuarios root tienen contraseña, es poco recomendable darle contraseña al que solo se puede conectar desde localhost, porque tienes que configurar el archivo C:\XAMPP\phpMyAdmin\config.inc.php para que vuelva a funcionar phpmyadmin indicándole la nueva contraseña de root.



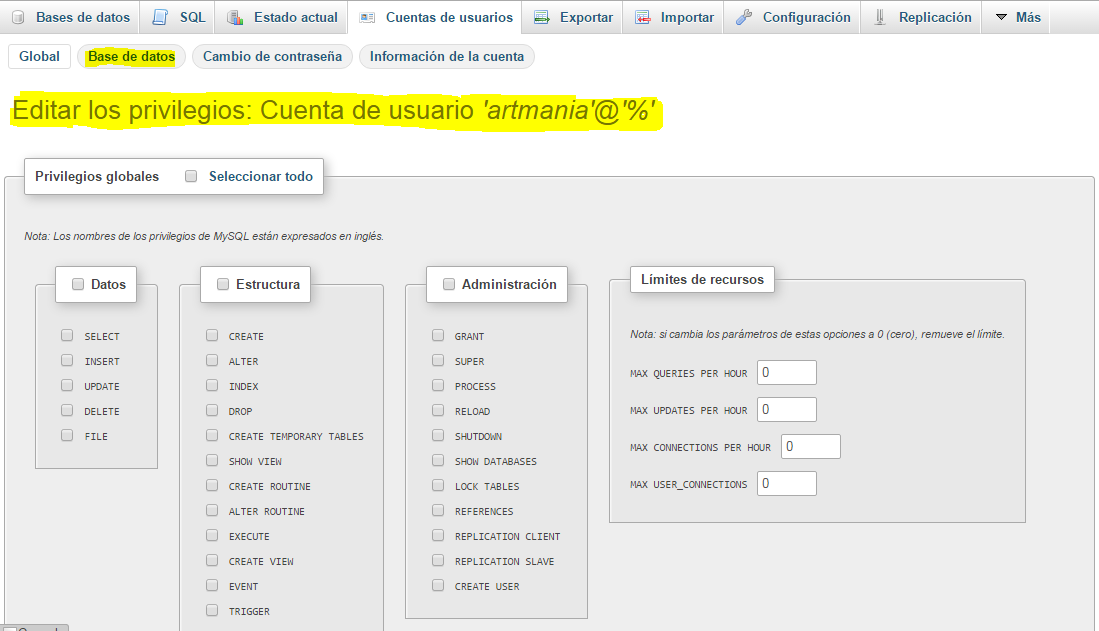
A los usuarios anónimos no les he dado contraseña porque tienen privilegios mínimos.

**1.2.1 USUARIO ARTMANIA:**

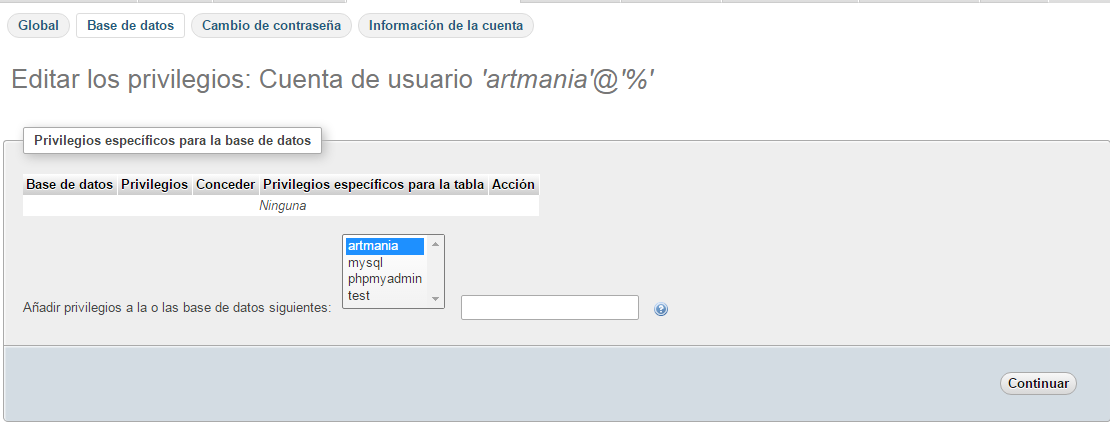
El usuario artmania será el usuario que tendrá todos los privilegios sobre la base de datos de artmania, será el que utilice la parte de la aplicación del backend. La contraseña será MANIAart#@55.



Una vez creada la cuenta, sin privilegios globales, vamos a editar los privilegios como hemos hecho con root. Le daremos todos los privilegios sobre los datos, es decir, select, insert, update y delete, pero no globales, sino solo en una base de datos en concreto, para ello vamos a Base de datos encima de los privilegios.



Seleccionamos la base de datos de la aplicación, artmania.

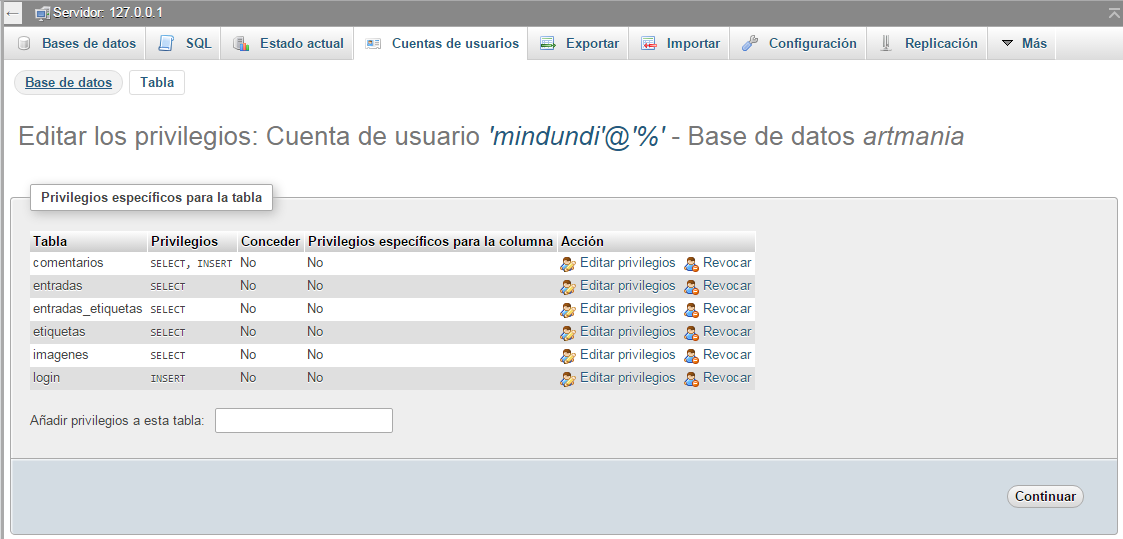


Y seleccionamos los privilegios que necesitamos, como hemos dicho antes, todos los privilegios sobre los datos, ya que insertará imágenes, entradas, comentarios…, los listará y los borrará.



**1.2.2 USUARIO MINDUNDI:**

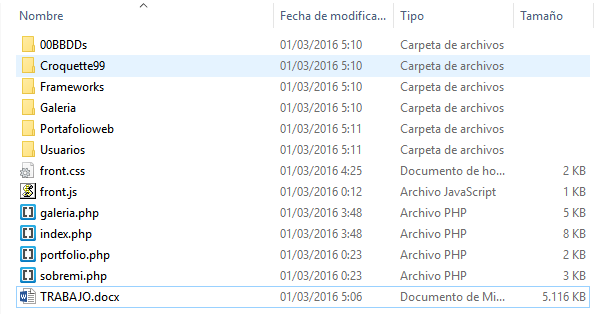
El usuario mindundi tendrá privilegios de select para casi todas las tablas y de insert sobre las tablas comentarios y login, por supuesto este usuario será el utilizado en el frontend de nuestra aplicación y los insert son para que los usuarios se den de alta y puedan comentar las entradas, pero como vemos es un usuario muy restringido. Su pass es mindu99.



Por último, he exportado la configuración de los dos usuarios por si necesito restaurarlos.

**1.3 ESTRUCTURA DE DIRECTORIOS Y ARCHIVOS:**

El contenido del proyecto se encuentra en la carpeta **00ArtMania** dentro del directorio raíz del servidor **htdocs** o lo que es lo mismo en la ruta **C:\xampp\htdocs\00ArtMania** encontramos:

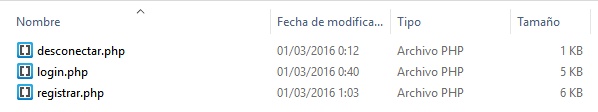


En la carpeta raíz del proyecto encontramos las 4 partes relativas a la parte pública de la aplicación, al front-end, exceptuando lo relativo al login y registro de los usuarios:

* **Index.php**: que contendrá las el menú principal y las entradas, ya descrita en el documento explicativo del proyecto.
* **Galeria.php**: la galería de imágenes, ya descrita en el documento explicativo del proyecto, muestra las imágenes de la carpeta galería controladas desde la BBDD.
* **Portfolio.php**: será una página estática con un iframe a la presentación web ya creada en la carpeta Portafolioweb.
* **Sobremi.php**: será una página estática con información sobre el administrador de la página.

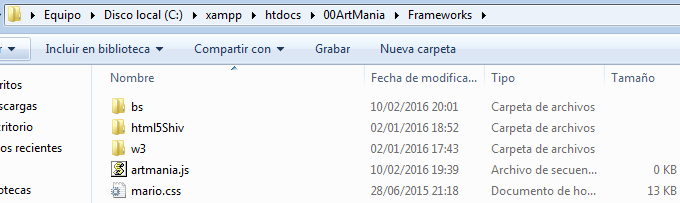
**1.3.1 CARPETA USUARIOS:**

En la carpeta Usuarios tenemos los ficheros que manejan el login, el registro de los usuarios y la desconexión de la sesion.



**1.3.2 CARPETA FRAMEWORKS:**

Tenemos los distintos frameworks y archivos css y js que necesita la aplicación, tanto la parte del back-end como la parte del front-end. MARIO NO USADO AL FINAL.



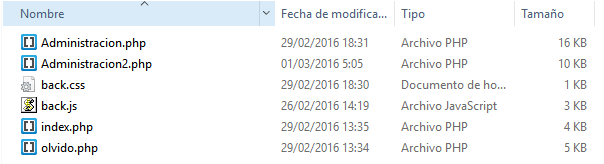
* **Mario.css**: contiene varias utilidades y el código css de la aplicación;
  + normalize.css sirve para equilibrar el css en los distintos navegadores, es parecido a reset.css, pero en vez de dejar los valores a 0 por defecto, intenta acercarse más a los estándares. Tengo el archivo normalize.css original en la raíz del servidor htdocs.
  + Tiene mi propio framework, con un sistema de grillas y colores responsive que he eliminado porque no los usaré en este proyecto y algunas clases reutilizables (centrado:centra, floatl:el elemento flota a la izquierda, floatr:el elemento flota a la derecha, etc…) que esas no las he eliminado.

Hasta aquí está mimificado, no obstante, he dejado el archivo original mario.styl, que aunque está codificado con el preprocesador stylus, la sintaxis es similar al css tradicional y se entiende.

* + Por último tiene las clases relativas a los elementos de la aplicación, las que aportan el estilo final a esos elemento.
* **Artmania.js**: aprovechando que bootstrap incluye Jquery, el archivo contiene el código javascript escrito con la librería de Jquery necesario para que los elementos de la aplicación interactúen con el usuario y sean funcionales.
* **Html5Shiv**: carpeta que contiene un framework que mediante hacks permite ejecutar algunas de las funciones de html5 en navegadores que no implementaban este estándar en un principio.
* **Bs**: carpeta que contiene los archivos del framework de bootstrap para trabajar con él en la aplicación.
* **W3**: carpeta que contiene los archivos del framework de w3school para trabajar con él en la aplicación.

**1.3.3 CARPETA CROQUETTE99:**

En la carpeta Croquette99 se encuentra los archivos .PHP referentes al backend, la zona de administración de la aplicación.



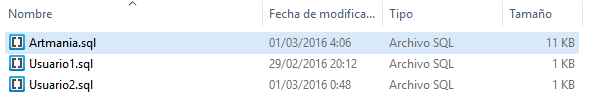
Index y olvido se encargan de la autenticación del administrador de la página, Administracion.php contiene todos los formularios para realizar acciones sobre las tablas que modifican el frontend y Aministracion2.php contiene el código para resolver los datos que le pasan esos formularios.

**1.3.4 CARPETA 00BBDDs:**

En la carpeta 00BBDDs se encuentran los ficheros .SQL que crean las tablas de la base de datos de la aplicación.

El archivo Artmania.sql que es el que importe finalmente para crear la base de datos. Contiene las tablas y unos cuantos registros para probar el resultado del frontend sin tocar la parte adminsitrativa.

A parte están las órdenes para crear y dar privilegios a los dos usuarios de la BBDD de la aplicación, uno para el backend con máximos privilegios sobre la misma y otro del frontend con privilegios de select y algún insert sobre las tablas necesarias.

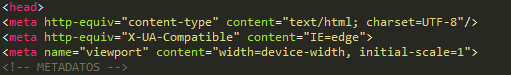


**2 COMIENZO CON EL BACKEND:**

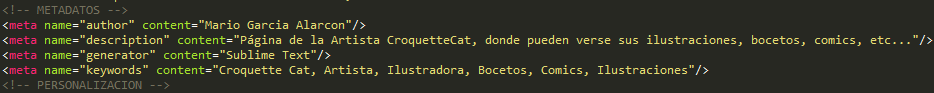
**2.1 HEAD DE LAS PÁGINAS BACKEND:**

Las etiquetas que necesitaré para incluir los frameworks y utilidades a las páginas son las siguientes, aunque variará la ruta de la que obtienen los ficheros.



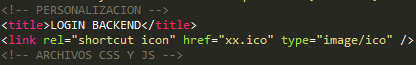


En las primeras etiquetas meta definimos el comportamiento del archivo, como por ejemplo la codificación que será UTF-8.

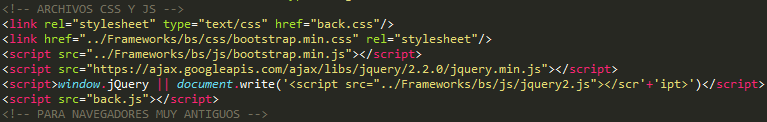


En los metadatos definimos los datos sobre la página, son muy importantes a la hora de que google y otros buscadores indexen la página:

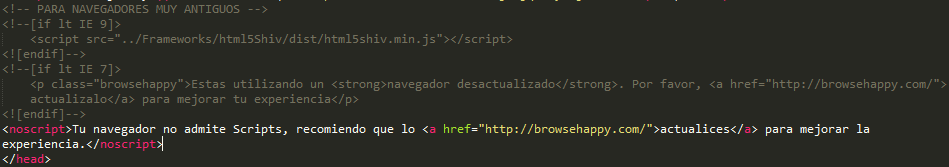
* Author: define el autor de la página.
* Description: Pone la descripción de la página, suele ser la que aparece debajo del nombre en las búsquedas con google.
* Generator: Indicamos con que software hemos realizado la página, normalmente se utiliza para editores que generan ellos el código.
* Keywords: incluye palabras clave para cuando las utilice alguien en sus búsquedas.



En la personalización incluiremos el título de cada página y un icono para que aparezca en la pestaña del navegador que identifique nuestra página.



Después incluimos los archivos de bootstrap, el del w3school, el css de la propia página con back.css, cargamos jquery desde un CDN y si no funciona le decimos que lo cargue desde local y back.js que es el javascript de la propia página.

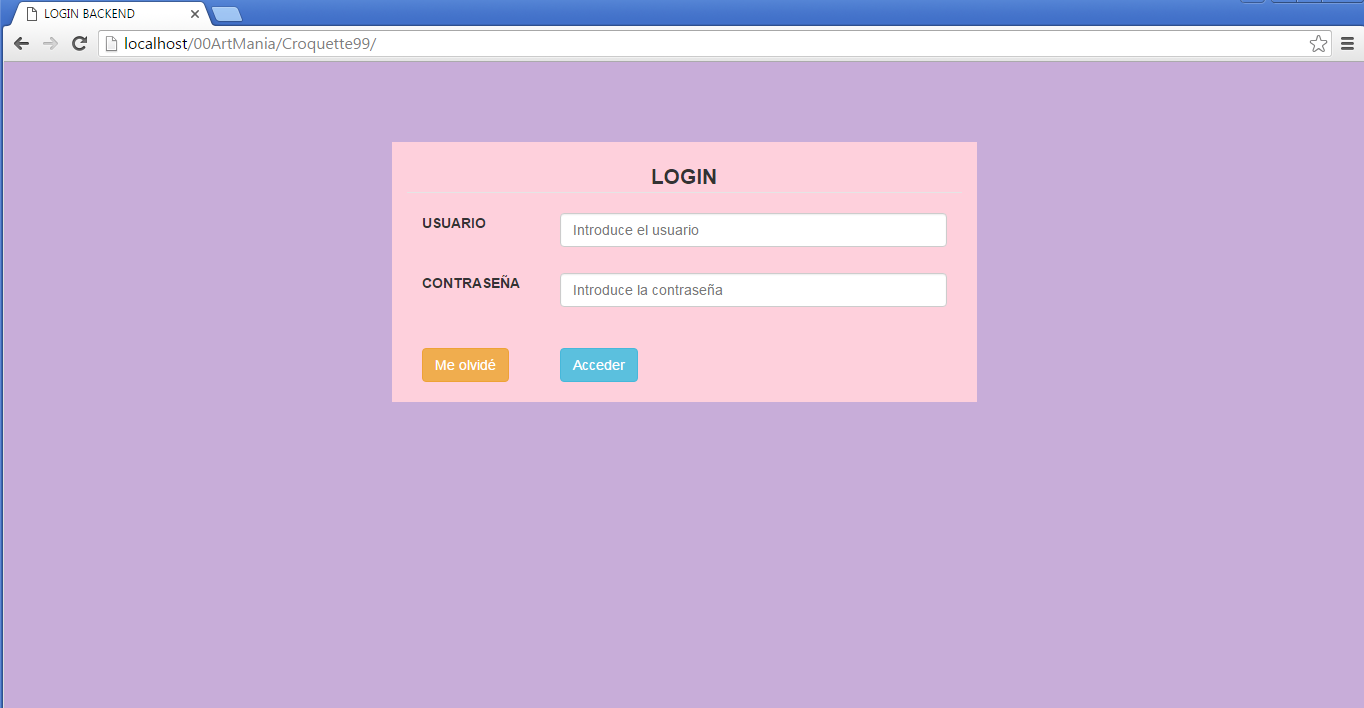


Por ultimo incluimos la utilidad html5shiv, el hack para cargar html5 en navegadores que no lo soportan, para Internet Explorer más antiguo de la versión 9.

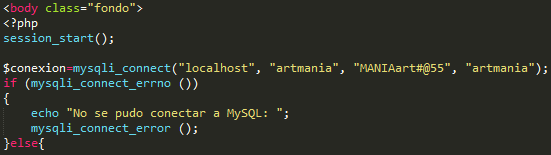
Para navegadores más antiguos que la versión 7 de Internet Explorer o aquellos navegadores que no admitan scripts (de los menos), le recomendamos que actualice el navegador con la página de browsehappy, que contiene los navegadores actualizados listos para descargar y utilizar.

**2.2 INDEX.PHP BACKEND:**

El backend se encontraría en la ruta de acceso/Croquette99. El index consiste en el login de acceso a la página de administrar. Como solo va a entrar el administrador a esta página, solo admite el usuario elena y la contraseña pass, que estará definido con el ID=1 ya definido en la base de datos.



Como es el propio index el que comprueba los parámetros pasados, lo primero que hacemos es iniciar sesión para quedarnos con el usuario y si logra autenticarse y la conexión con la base de datos.



Comprobamos si nos han pasado datos no vacíos por el formulario, guardamos el usuario y la contraseña.

Hacemos una consulta para obtener el usuario y administrador almacenado en la Base de Datos, que es aquel con el ID:User=’1’.

Comparamos que ambos coincidan (usuario pasado con usuario de la BBDD y password pasada con password de la BBDD) y si coinciden, guarda el usuario y un switch de validado en la sesión para la página de administración a la que te manda directamente.



Debajo de ese código se encuentra el código html de la página necesario para que se muestra la caja con el login, que carga en el caso de que no hayas introducido datos o los datos introducidos sean incorrectos y por tanto no te envíe a la página Administracion.php.

Dado que la única persona que usará esta parte de la aplicación es el administrador, no doy ningún tipo de mensaje de error.

**2.3 OLVIDO.PHP BACKEND:**

A la página olvido se accede desde el index pulsando sobre el botón de Me olvidé. Tiene un aspecto similar, pero el código es distinto.



Comprueba que le has pasado los datos no vacíos, hace una consulta sobre el primer usuario de la base de datos (solo el administrador), pidiendo su nombre e email para compararlo con lo que le has pasado y si coinciden ambos, manda un email con la contraseña.

Aunque no me envía el mensaje, no me llega al correo, ni siquiera a spam, estando habilitado que lleguen correos de fuentes externas no confiables.



**2.4 ADMINISTRACION.PHP BACKEND:**

La página de administración es a la que se accede desde el index.php cuando das las credenciales correctas: elena y pass. En ella tenemos un header que muestra una bienvenida, un menú con botones a la izquierda y el formulario en el resto de la página.

Los botones modifican el contenedor de los formularios según el que necesitemos, a través de una función de javascript en el archivo back.js. Cada formulario tiene sus campos necesarios, el de modificar entrada y las de borrar, listan las entradas que posee la base de datos.

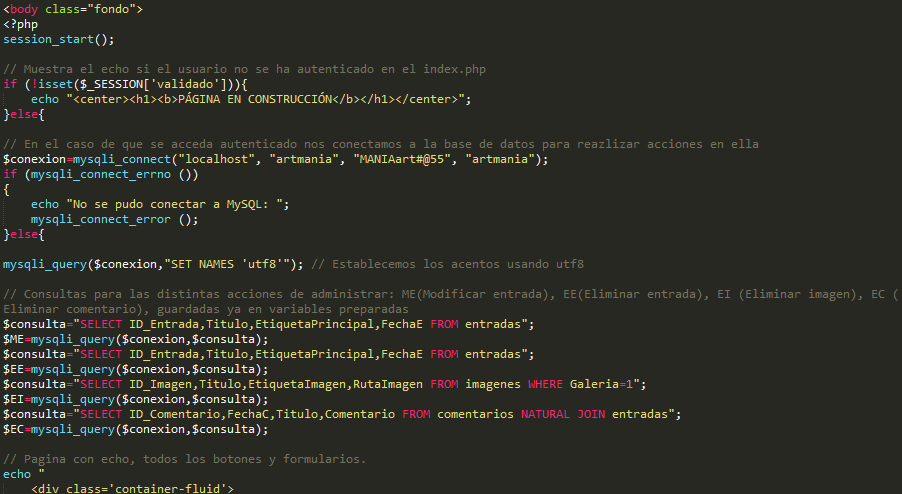


Lo primero que hace la página es comprobar que existe la variable de sesión validado, que solo está si nos hemos validado en index.php, si no muestra un mensaje de página en construcción para evitar accesos no deseados.

Una vez comprobado se conecta a la base de datos, hace las consultas necesarias para mostrar las entradas en los formularios necesarios y luego con un echo escribe el código html.

Las Etiquetas principales, de entrada y de imágenes se pasan a través de listas, porque se usarán como filtros en la parte del frontend y están definidos para que no nos confundamos.

Las listas de entradas permiten seleccionar varios registros a la hora de borrar entradas, imágenes y comentarios, pero para modificar no.



**2.5 ADMINISTRACION2.PHP BACKEND:**

Está página es la que trata los formularios de la página anterior.

Funcionan todas las funciones menos insertar una entrada, que no me ha dado tiempo a implementarla, no obstante, la BBDD cuenta con entradas de prueba para poder modificarlas, borrarlas y verlas en el frontend.

**3 FRONTEND:**

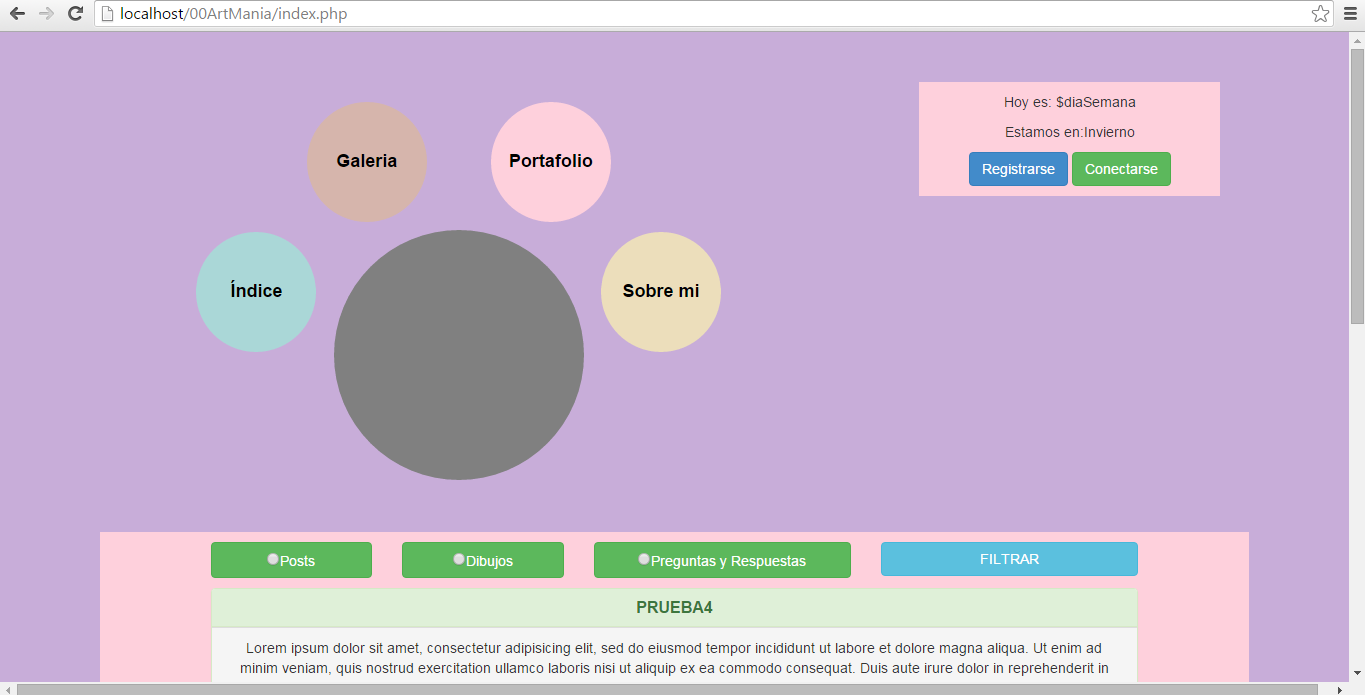
**3.1 HEAD DE LAS PÁGINAS FRONTEND:**

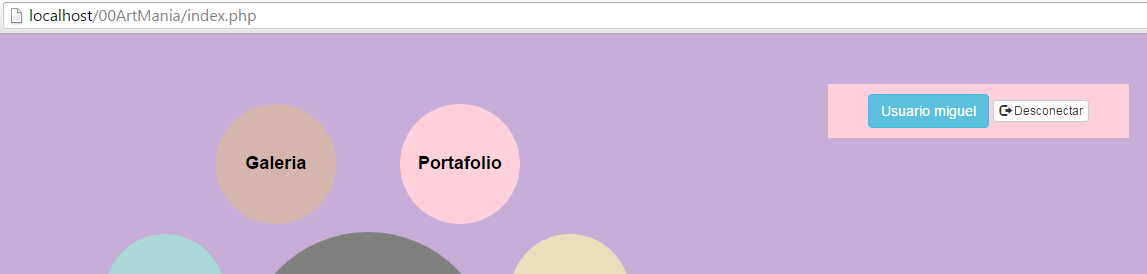
El head de las páginas del frontend es similar al del backend, solo cambian las rutas de los archivos y back.css y back.js, que en el frontend son sustituidos por front.css y front.js.



**3.2 INDEX.PHP DEL FRONTEND:**

En esta página nos encontramos un menú en la parte superior izquierda que lleva a las demás páginas del frontend y un panel de autenticación en la parte superior derecha que cambia al iniciar sesión pidiéndonos desconexión.





En la parte media tenemos una caja con las entradas, también tenemos filtros, como hemos podido observar en la imagen que permiten filtrar según el campo EtiquetaPrincipal de la tabla entradas de la BBDD.

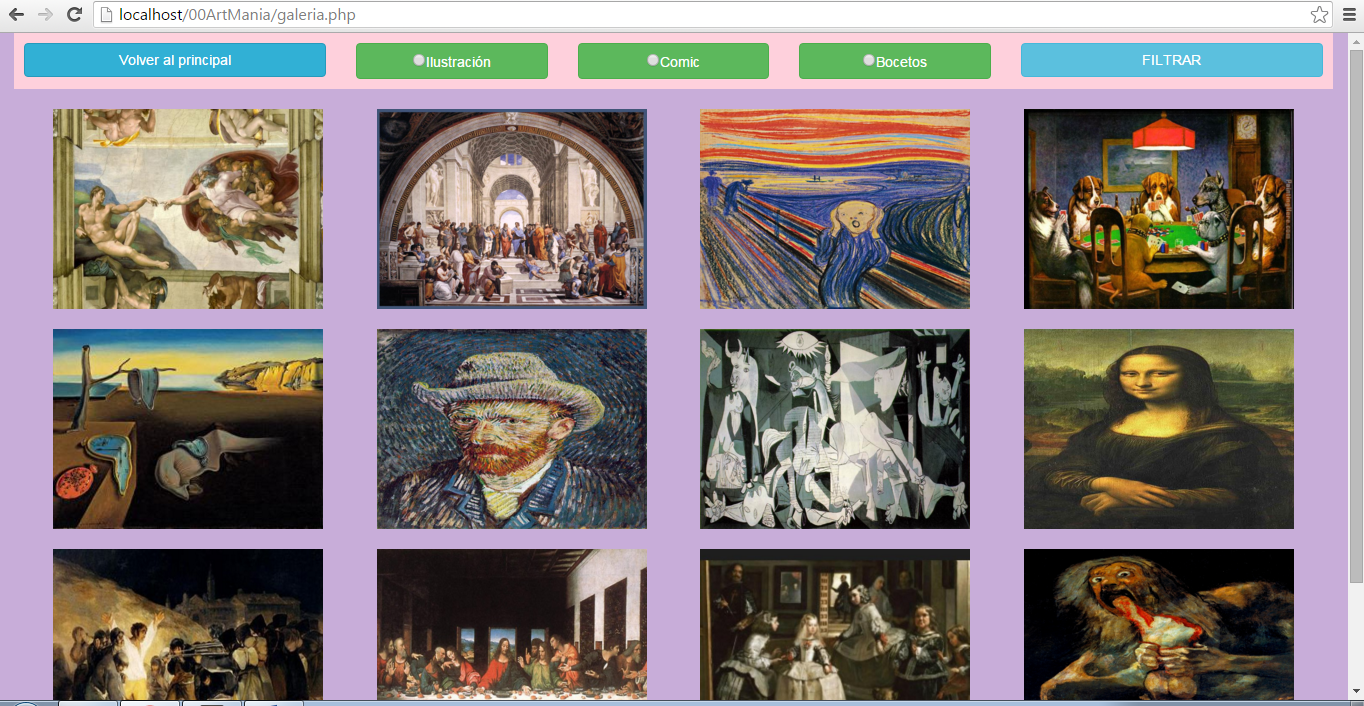


Tengo que aclarar que se muestran 5 entradas por página, podemos ver las 5 siguientes pulsando sobre el radio SiguientesEntradas y dando en MOSTRAR.

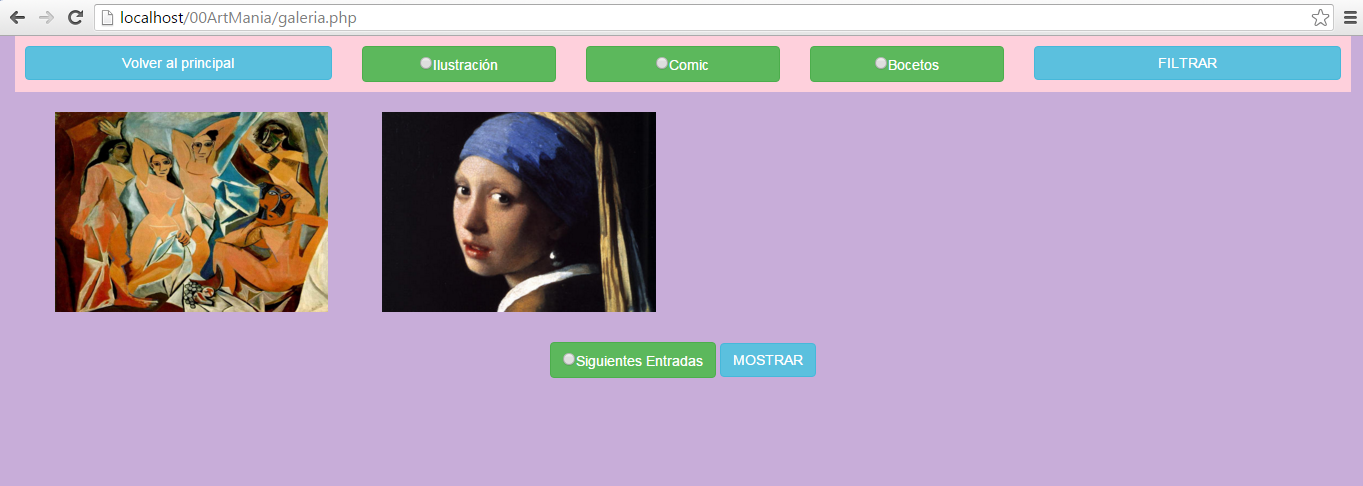
A parte las entradas solo muestran imágenes si hay una imagen asociada en la tabla de Imágenes a esa entrada.

**3.3 GALERIA.PHP DEL FRONTEND:**

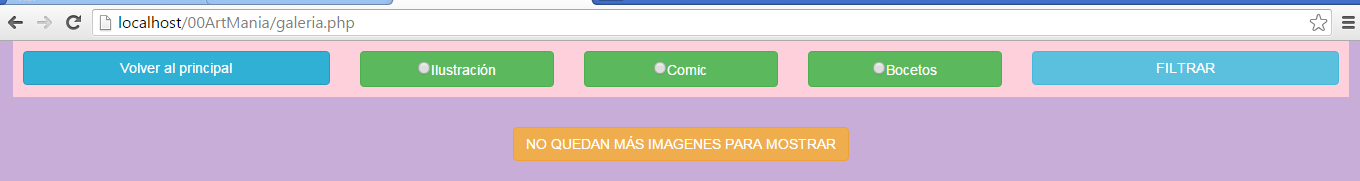
La página de galería a la que se accede desde el index del frontend, nos muestra en la parte superior un botón para regresar al index y unos botones con radio, al igual que en index, seleccionamos 1 y podemos filtrar según el campo EtiquetaImagen de la tabla Imágenes.



Tiene 12 entradas por página, divididas en 4x3, genera imágenes hasta que no tiene más. Podemos cargar las siguientes con el radio de Siguientes Entradas y acto seguido dando a buscar, igual que en el index.



En el caso de no encontrar más entradas producirá el mensaje de NO QUEDAN MÁS IMÁGENES PARA MOSTRAR y si pinchamos sobre ellos nos devolverá a la galería sin ningún filtro con las 12 imágenes iniciales.



**3.4 PORFOLIO.PHP DEL FRONTEND:**

Está página a la que se accede desde el index presenta una presentación web alojada en la carpeta portafolioweb, también tiene un botón fijo que nos devolverá al index cuando queramos.



**3.5 SOBREMI.PHP DEL FRONTEND:**

Tiene simplemente una caja central con información sobre la dueña del blog. Tiene la misma que el portafolio porque se pensó antes del proyecto que entre esta y la página de galería, reemplazarían al portafolio, dado que contendrán los mismos elementos. También con un botón con el que volver al índex.



**3.6 LOGIN.PHP Y REGISTRAR.PHP DEL FRONTEND:**

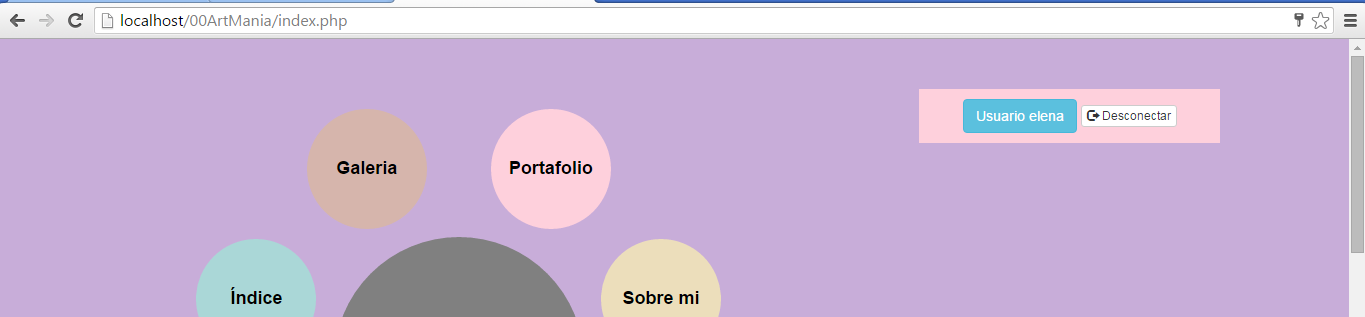
A ambos se accede desde los botones del menú de la derecha del index y ambos están en la carpeta Usuarios, junto con desconectar.php, que solo desconecta el usuario de la sesión.



Como vemos registro pide tres campos, comprobará que tanto la contraseña como el email no se encuentran en uso por otro usuario y que los campos se han mandado con contenido y en el caso de que sea así insertará al usuario en la tabla correspondiente de la BBDD. También nos da mensajes de fallo y tiene un botón de acceso al login.



El login es muy parecido, pide un usuario y una contraseña y tras comprobar que le introduces datos en el formulario, comprueba también que ambos coinciden con un registro de la tabla de usuarios y si es así te manda autenticado al index. Además todas las páginas del frontend te respetan la sesión, para desconectarte tendrás que pinchar sobre desconectar.



**4 NO CONTEMPLADO:**

* El aspecto visual deja que desear, querría haber puesto un fondo mejor en el frontend, la caja con opacidad, mejorar la combinación de colores y meter los formularios del backend de otra forma más visualmente agradable.
* Comprobar que el usuario mete bien la contraseña en el registro a través de otro campo para que la repita.
* Los comentarios no me ha dado tiempo a implementarlos, simplemente igual que hace con las imágenes leería aquellos que van asociados a una entrada y los pondría a continuación de está. A parte comprobaría si hay algún usuario con sesión activa y si es así, le mostraría un campo de textarea después de los demás comentarios para que pudiese dar su opinión en cada entrada.
* Las imágenes no se suben a la carpeta de Galeria que es donde están todas, en la parte del backend simplemente actualiza la tabla de Imágenes.
* La inserción de entradas, puesto que había que comprobar muchos factores y no me ha dado tiempo.