Proyecto Final

ProgramacionWeb 1

y

Bases de datos

# Una breve Introduccion al documento

El siguiente trabajo expone la solucion a un problema en concreto con el uso de los conocimientos adquiridos durante el semestre en las materias de Programacion 1 y Base de datos. Este documento tiene la funcion de respaldar todas las acciones durante la elaboracion del producto final.

El trabajo realizado y el docuemento son con fines meramente educacionales y se prohibe su uso para fines comerciales.

# El problema a resolver

El problema que se tiene que resolver de una manera que satisfaga todas las necesidades es:

“Hacer una pagina que contenga usuarios, los usuarios deberan de iniciar session para acceder al contenido de la pagina.

El contenido del website debera ser dinamico y el contenido de los elementos de cada pagina sera determinado por una base de datos”

# Analizando la problemática

El problema es muy claro y especifico “una pagina que contenga usuarios”, “iniciar session para acceder”, “contenido del website debera ser dinamico”, “el contenido de los elementos sera determinado por una base de datos.”

En otras palabras seria : “Crear una pagina web dinamica , que el contenido de la pagina sea dependiendo los datos de una base de datos y el contenido de la pagina solo lo podran ver los usuarios registrados.”

# Herramientas a usar

Editor de Texto/Codigo:Visual Studio Code 2017

Editor de Graficos Rasterizados: Photoshop CSS

Entorno de Desarrollo web : WampServer

Systema de Gestion de Bases de Datos: MySQL

Lenguaje Principal de Programacion : PHP

Lenguaje de Marcado : HTML5

Dentro de el editor de Texto/Codigo “Visual Studio Code” sera el corazon de el proyecto , donde se realizara casi todas las tareas como… la programacion o el etiquetado, ya que ese editor es muy versatil y permite hacer tantas cosas como sea posible al editor.

Dentro de el editor de Graficos “Photoshop” se realizaran tareas sensillas de edicion de imágenes , converciones a formatos optimizados para tecnologias web, etc…

Dentro de nuestro entorno de desarrollo web “WampServer” el cual nos permite simular un servidor en el cual realizaremos las pruebas pertinentes del proyecto y donde veremos el resultado final gracias a esta poderosa herramienta de desarrollo web.

El sistema de base de datos a usar sera MySQL por varias razones como el echo que es de codigo abierto y no se nesecita ningun tipo de licencia para operarlo o por el echo de que es el sistema de gestion de base de datos con amplia documentacion y sin mencionar que es la mas accesible.

# Diagramacion de la base de datos

La diagramacion de base de datos usaremos las referencias de un servicio en linea llamado : “MindMeister”.

La url de donde estar alojado el diagrama para su mejor comprencion es :

“<https://mm.tt/924328876?t=H6RnfaaryK>”

Señalisacion :

La señal de un marcador(Correcto|Check) es una Entidad.

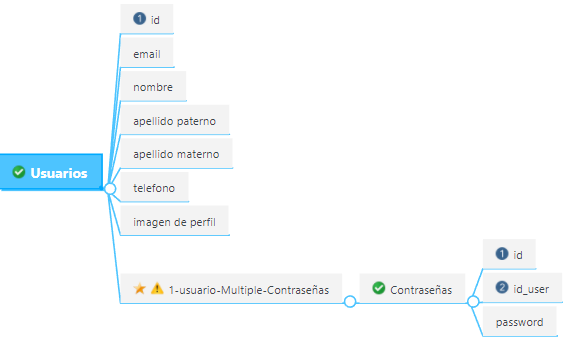
La señal de una estrella indica una relacion y el texto es el tipo de relacion que tiene.

La señal de un numero “1” indica que dentro de la entidad correspondiente ese atributo es el identificador unico.

La señal de un numero “2” indica que dentro de la entidad correspondiente ese atributo es el identificador foraneo.(Hace referencia a un identificador unico de otra entidad).

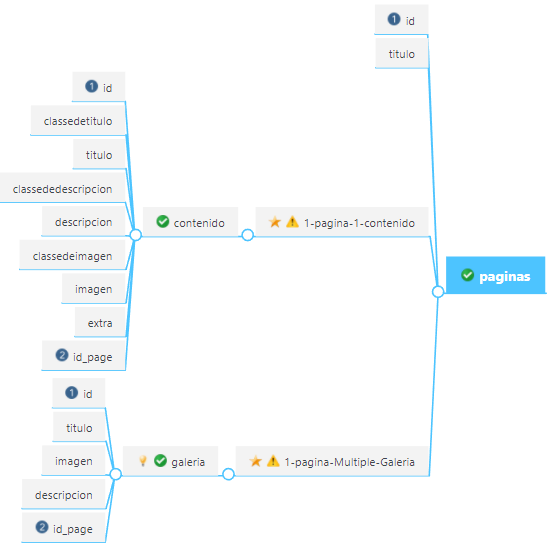
La señal de precaucion indica que son relaciones que posiblemente puedan cambiar … digamos que no son permanentes y conforme la elaboracion del proyecto pueden definirse como estaticas o puede que cambien de valor.

**La tabla de usuario con su relacion con la tabla de Contraseñas:**



Esta tabla nos indica que un usuario registrado con x numero de caracterizticas puede o no tener multiples contraseñas y que estas contraseñas solo podran corresponder a un usuario pero sin embargo esa contraseña no es monopolizada por un usuario , puede llegar el caso en el que 2 usuarios tengan 2 contraselas diferentes pero! Con el mismo valor dentro de las contraseñas.

**La tabla de paginas con sus relaciones con la tabla de Contenido y de Galeria:**

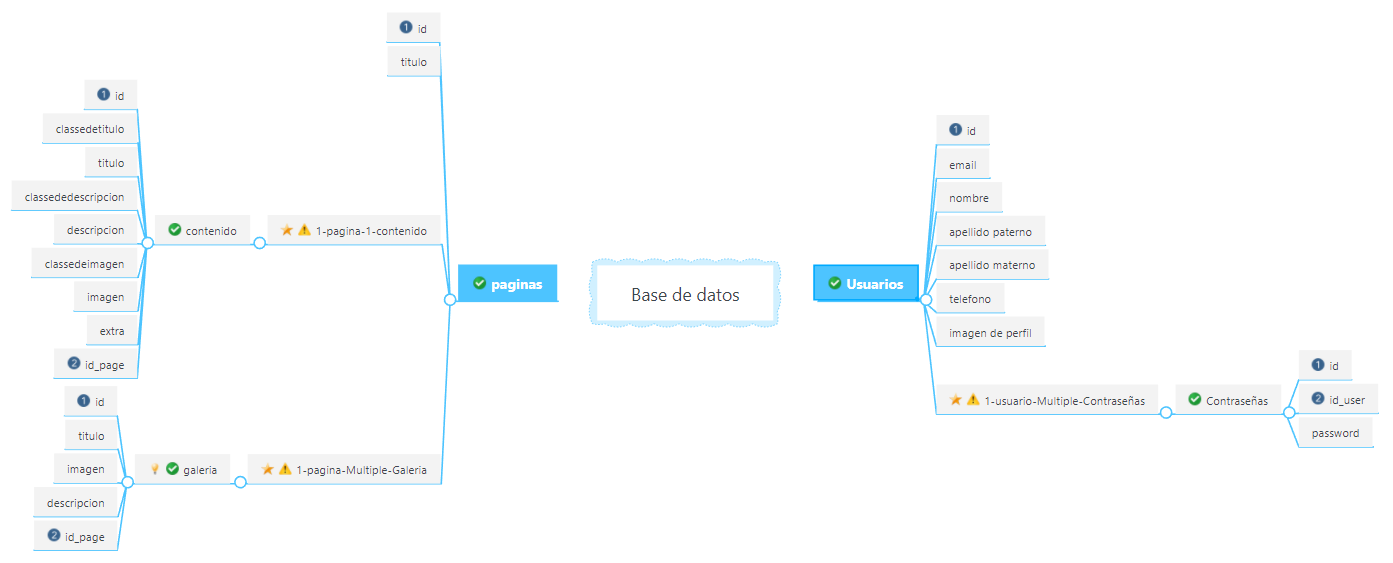


Esta tabla de paginas nos hace referencia a el numero de pagina y el titulo de la pagina.

Mientras que la relacion de Paginas🡪Galeria nos indica que en “X” pagina existen “Y” cantidad de imágenes acomodadas como galeria o pertenecientes a galeria.

Y por ultimo la relacion de la Paginas🡪Contenido nos indica que para una pagina “X” debe de corresponderle un contenido “Y” y solo un contenido.

**El diseño de la base de datos completa:**



Sorendentemente la base de datos solo contiene 5 Entidades y 3 Relaciones.

En mi no experta opinion y puede que erronea : es una base de datos sensilla pero muy optima para este caso en particular.

Talvez durante el desarrollo pueda encontrar maneras de optimizar mas el flujo de datos, pero por el momento es con lo que trabajaremos.

Anotaciones: --Por el momento ninguna--

# Diagrama de flujo Estructuracion Web

Recurso online en : “https://mm.tt/920122613?t=I1KUthXoEp”

**Simbologia** :

**Bandera verde**: inicio de el flujo.

**Bandera roja** : Fin del flujo.

**Estrella** : Conexión a la base de datos | Elemento clave en la creacion de la pagina.

**Simbolo de advertencia** : Posible existencia de datos.

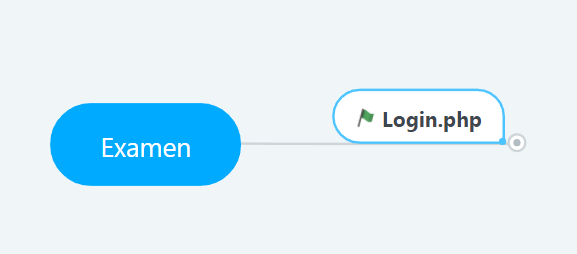
**Simbolo de repeticion**: indica que los datos obtenidos por el usuario se estaran enviando al mismo archivo y en el caso que la validacion en algun punto sea erronea entonces los volvera a pedir hasta que sean validos , comparables y correctos.

**Simbolo de palomita verde** : credenciales y/o datos comparables con la base de datos correctos y continua el flujo de datos.

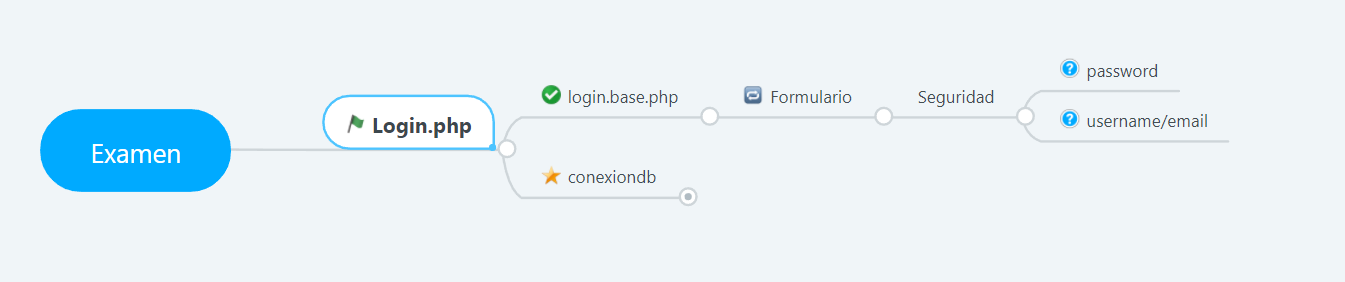
**Simbolo de cruz roja**: por algun motivo las credenciales y/o los datos comparables con la base de datos son erroneos y no puede continuar el flujo hasta que sean validos , comparables y correctos.

**Simbolo de foco** : Redireccion hacia dicha pagina y/o estado

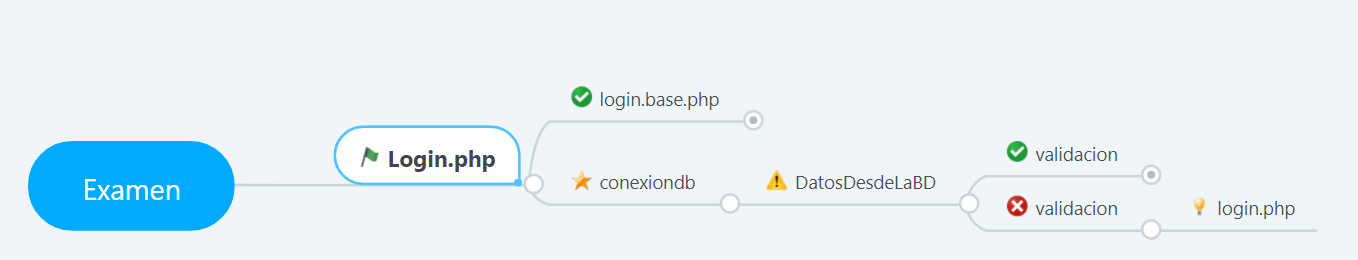
**Primero debemos de pensar una manera en la que podamos interactuar con el usuario**.



**Asi podremos obtener datos del usuario**



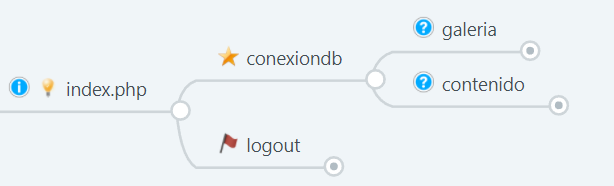
**Deberemos no suponer que hay usuarios normales. Hay que proteger nuestra base de datos y asi mismo el codigo de fallos , errores , inconsistencias.**



En caso que el usuario proporcione datos diferentes a sus credenciales… entonces lo redirecionaremos al inicio de session.

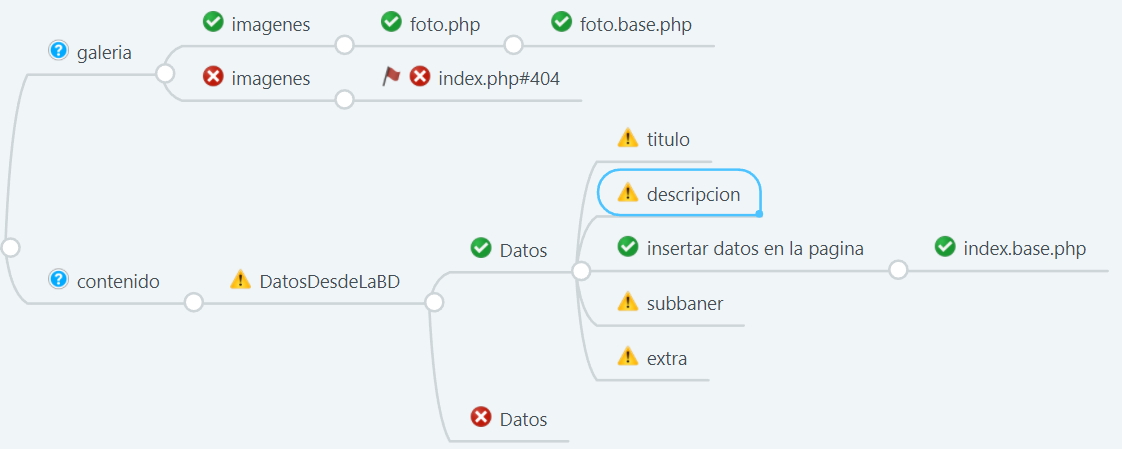
Anotaciones: Podriamos poner un boton de recuperar cuenta etc… si es que queda tiempo.

**Suponemos que el usuario ingreso correctamente a su cuenta y esta iniciado su session correctamente … entonces deberemos de preguntar a la base de datos cual es la pagina principal a mostrar**



Boton en cualquier momento podemos salir de nuestra session. (Suponemos que el usuario desea salir de la pagina y no quisiera que nadie use su cuenta mientras el no esta)

**Preguntamos a la base de datos si la pagina a cargar que tipo de contenido tiene y si tiene imágenes de la galeria o no.**



En el caso que no exista imágenes o alguna este corrompida entonces deberemos de dirijirlo a la seccion de errores.

En el caso que no existan datos en el contenido tambien deberemos de enviarlo a la seccion de errores.

Si todo sale como deberia entonces los datos y las imágenes de la galeria las mostraremos.

# Creacion de la base de datos

Sesión extra: una explicacion rapida de el por que la palabra reservada “LocalHost”

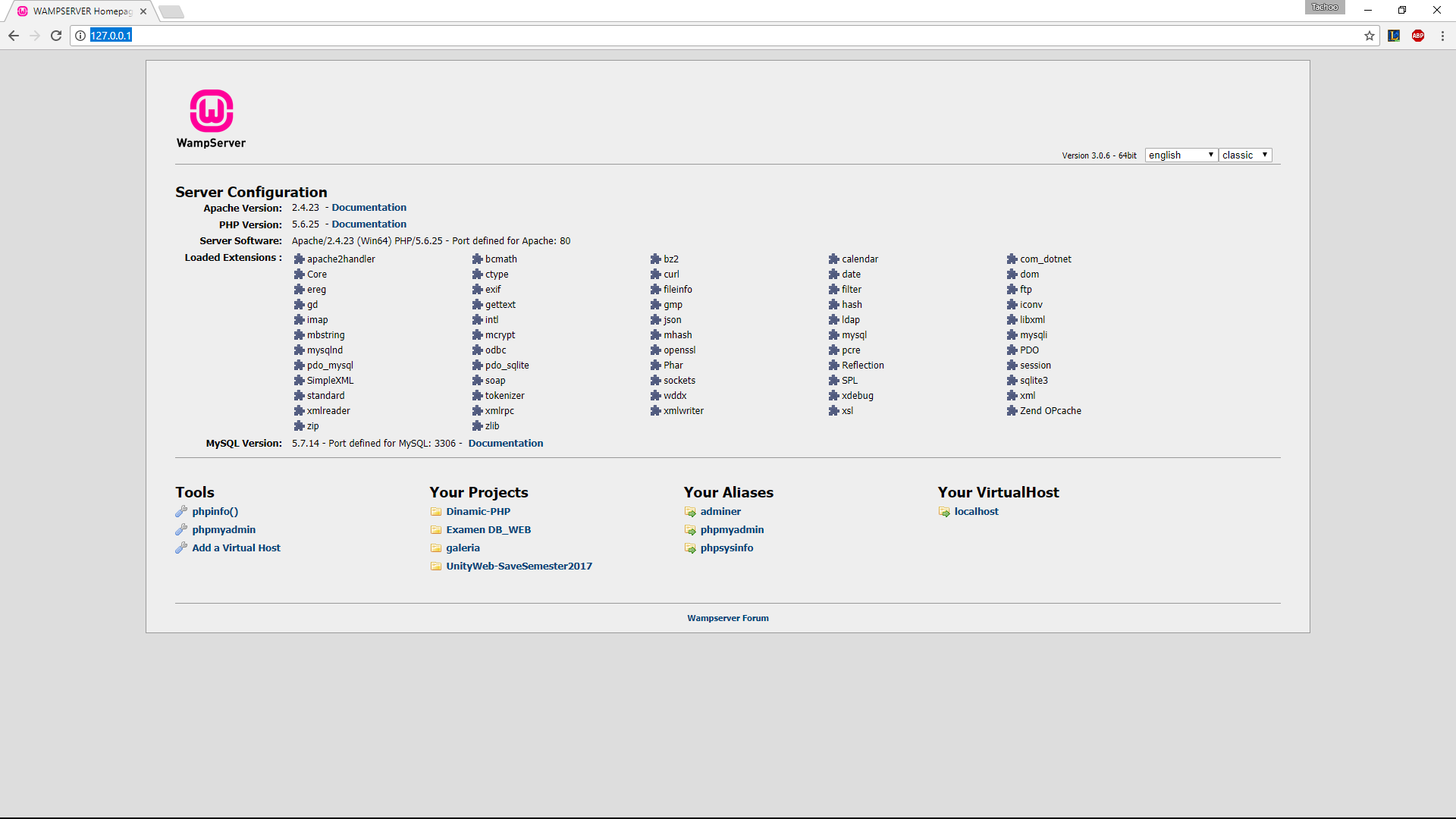
El explorador web es como un interprete realmente no entiende que no afueran de Direcciones de internet… realmente cuando nos dirigimos a nuestro sitio habitual

ejemplo : “ http://tachoo.xyz/ ” lo que estamos diciendole al explorador web es : “Por favor puedes ir a la direccion 185.28.21.205 y traeme toda la informacion que puedas obtener ”.

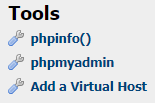
Por lo cual es dificil aprenderse esas direcciones no lo creees? . por eso mismo es que le dieron nombres o dominios , los cuales son lo que conocemos como [www.“Nombre.extencion](http://www.“Nombre.extencion)” . por ese motivo las paginas pueden cambiar de nombre muchas veces y conservar su “ip publica”.

Una vez ya instalada nuestra herramienta “WampServer” que nos permitira emular un servidor localmente sin la necesidad de tantos conocimientos tecnicos sobre servidores , internet etc…

En nuestro navegador web preferido en la direccion escribirmos : “http://127.0.0.1/” o bien podemos escribir : “ http://localhost/ ” para acceder a la pagina de inicio de WampServer



en la seccion de “Herramientas” seleccionamos “phpmyadmin” para acceder al systema de gestion de bases de datos mysql



Ahora nos direccionara a la pagina de 127.0.0.1/phpmyadmin/ y literalmente debemos de ingresar como administradores.

NOTA: Como los inicios de phpmyadmin y en general todas estas herramientas de codigo abierto tienen raiz en linux y por ende las credenciales son las mismas de dicho sistema por default.



Una vez dentro podremos observar que tenemos un monton de configuraciones

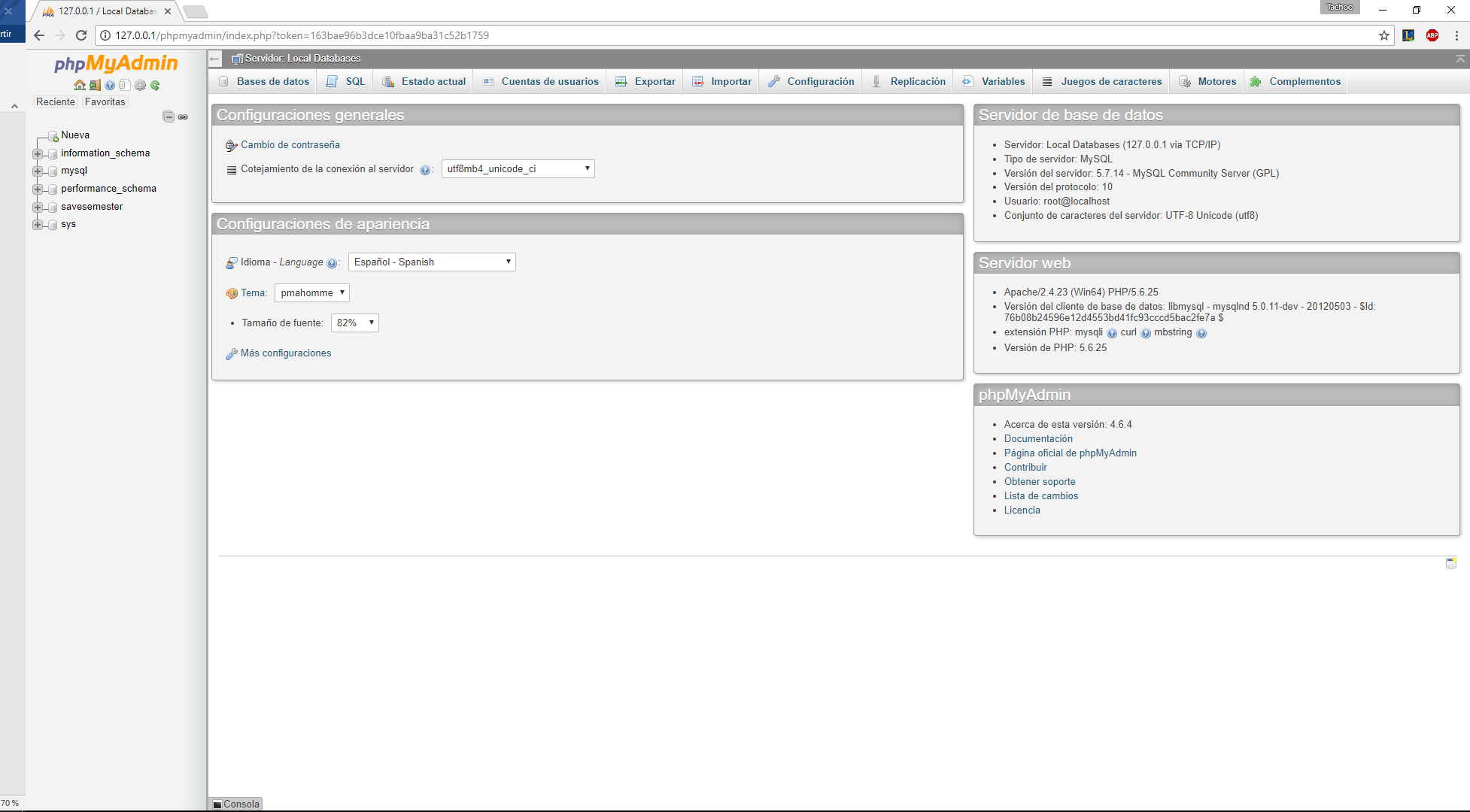
Los detalles del servidor de la base de datos

Detalles del servidor web

Detalles de phpmyadmin

Configuraciones generales y configuraciones de apariencia

Son cosas utiles para el super-usuario|gestionador de bases de datos tenga a la mano por cualquier error.

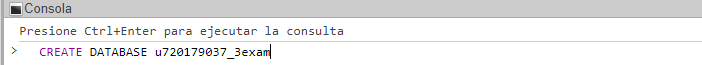


Pero lo que a nosotros nos interesa esta en la parte de abajo llamada “Consola”



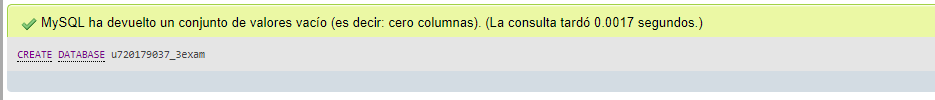
Como se supone que en el curso los temas se vieron muy superficiales y cada persona a base de su interes y tiempos de practica desarrollo diferente la capacidad en el uso de SQL. En este caso lo vamos a hacer todo basado en instruciones SQL.

**Lo primero que debemos de hacer es : Crear una nueva base de datos**

La instrucion seria 

Para ejecutarla instrucion Ctrl+Enter(Numerico);

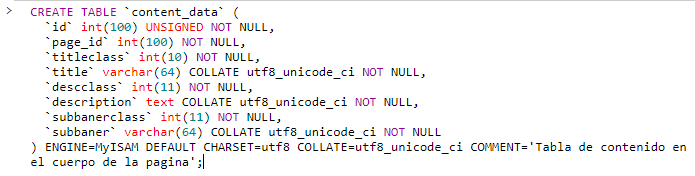
**Al ejecutarlo veremos un resultado como :**



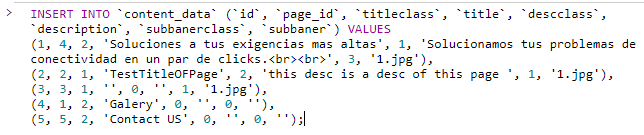
Donde podemos ver el tiempo que se tardo al realizar la consulta. Y aveces nos devuelve mas valores.

Ahora siguiendo nuestro diseño de base de datos crearemos las entidades (Tablas) correspondientes:

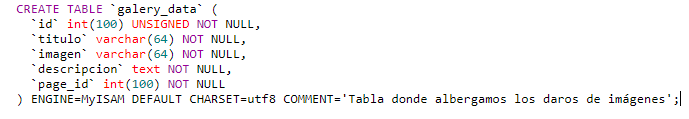
**Contenido:**



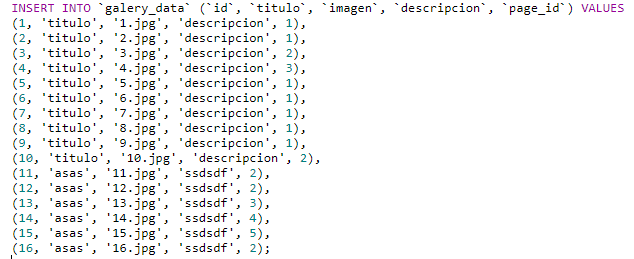
**Ahora que tenemos creada la tabla debemos de llenar con datos:**



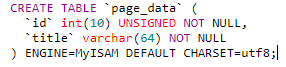
**Galeria :**



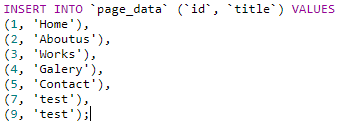
**Llenamos :**



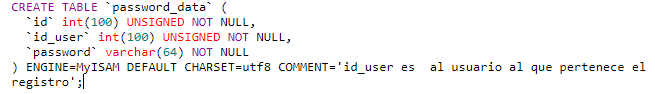
**Paginas:**



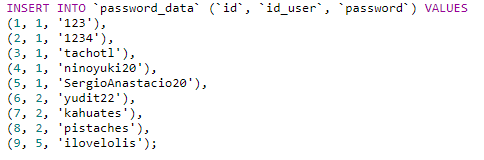
**Llenamos:**



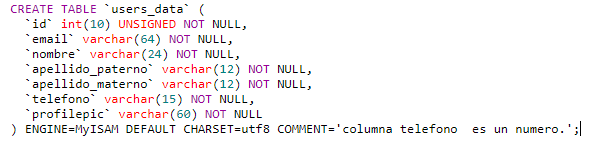
**Contraseñas:**



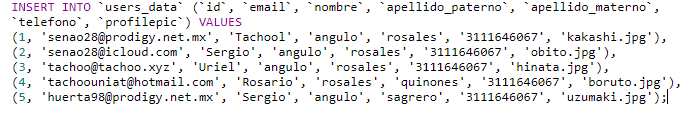
**Llenamos:**



**Usuarios:**

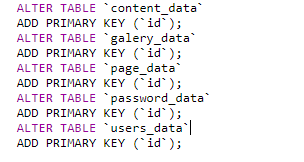


**Lleamos :**



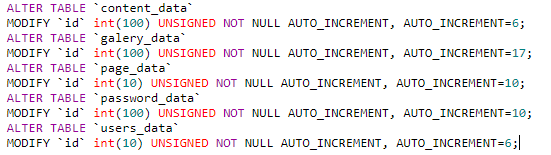
Bueno hasta el momento solo tenemos las tablas y tenemos datos adentro de ellas. Pero!! Debemos de indicar cual es el valor en cada tabla que sera el identificador unico o la llave primaria.

IDs:

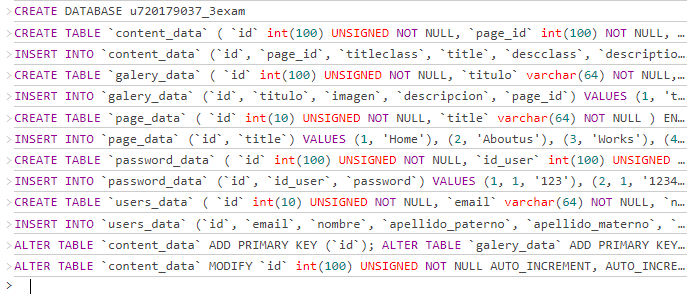


Perfecto , pero ahora debemos de indicarle que esos id son incrementales para simplificar algunos processos y que sea mas rapido ingresar datos.

AI:



Con tan solo 13 consultras tenemos nuestra base de datos completa creada , con datos y lista para usarse.



Creacion de la web

La creacion de la web sera creo que la parte mas entretenida pero lo bueno es que con el uso de repocitorios se puede compartir el codigo fuente mas facilmente.

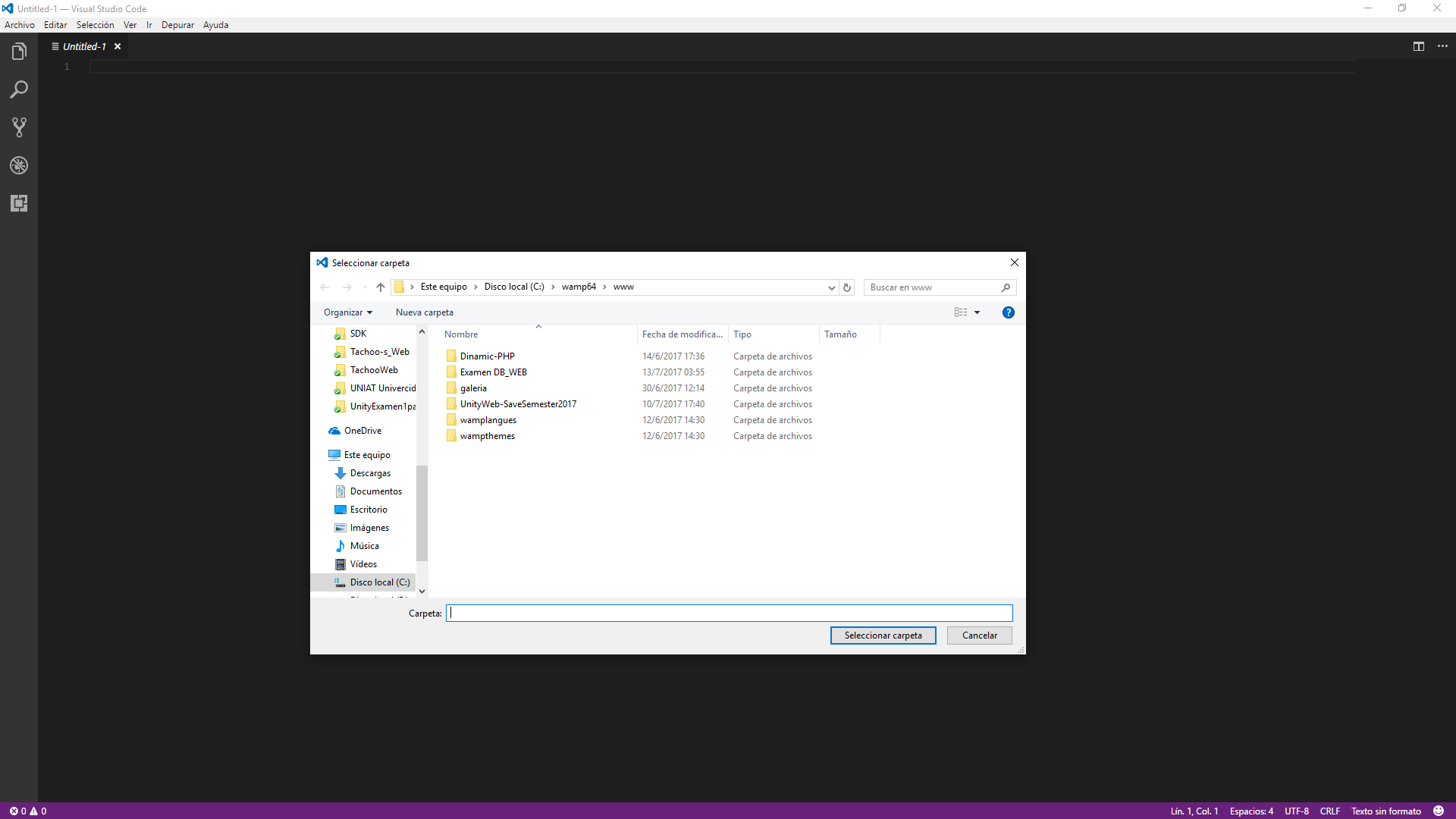
Url del repocitorio: “ https://github.com/Tachoo/Examen-DB\_WEB ”

En dicho repocitorio estara actualizado hasta el dia de entrega.

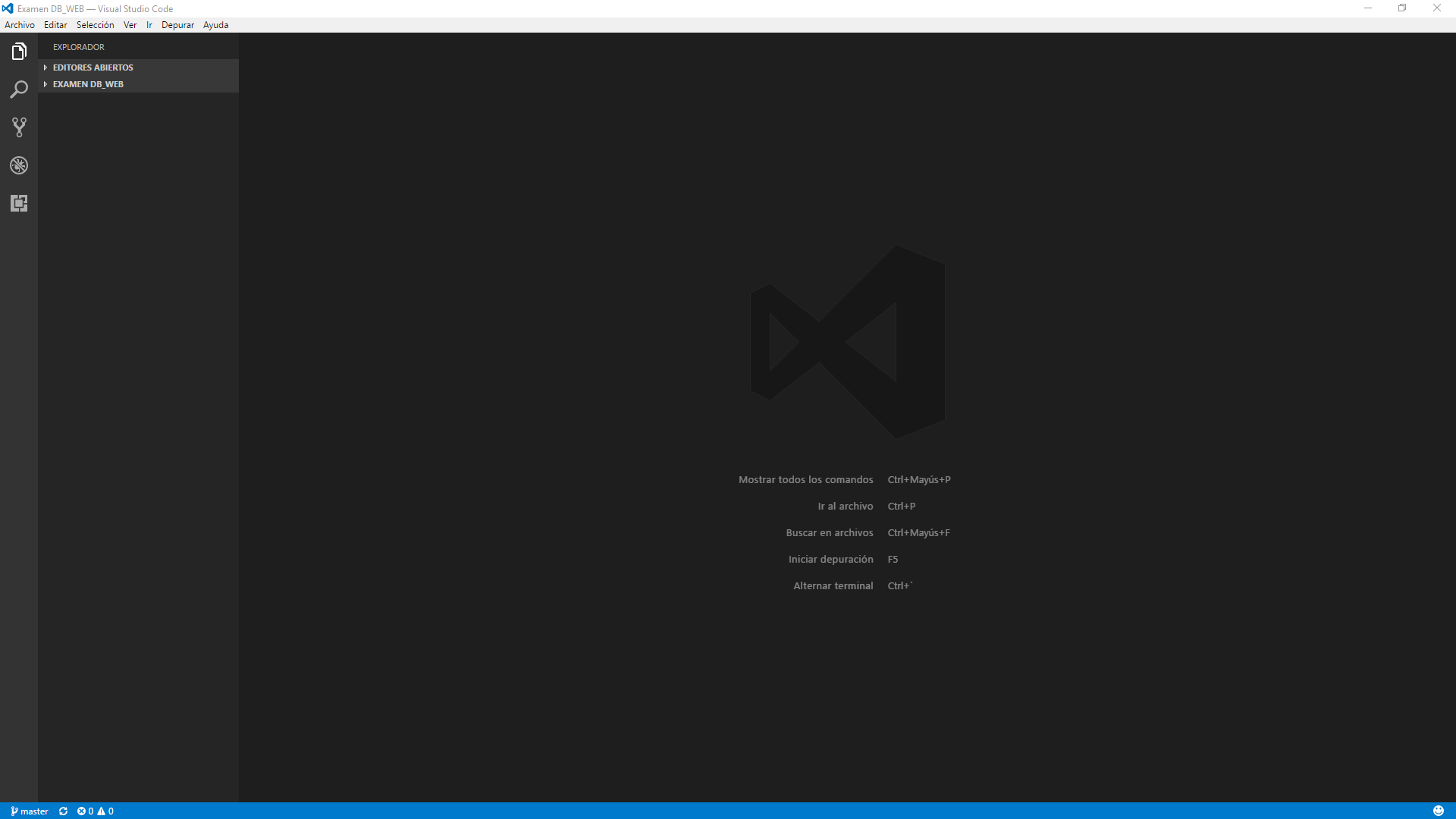
Primero lo primero VisualStudioCode

En nuestro editor de texto/codigo “VisualStudioCode ” cuenta con unas funciones que facilitan el trabajo como su explorador dentro del editor.

Al momento de abrir el VSC usamos la macro : “ Ctrl+K, Ctrl+O ” y nos abrira el explorador para seleccionar una carpeta :



Navegamos hasta donde se instalo Wamp en nuestro disco duro y adentro existe una carpeta llamada “www” donde deberemos crear una carpeta para iniciar/crear la web que estamos realizando. Ya cuando la seleccionemos/abrimos tendremos nuestro explorador dentro del editor



Es practicamente igual a navegar entre las carpetas de windows , Estamos listos para comenzar a escribir codigo!!.

Diferencias entre HTML y PHP

La diferencia mas notable es en la extencion del archivo y que uno es un lenguaje de etiquetas y el otro es un lenguaje deprogramacion como tal.

Php cuenta con un interprete de etiquetas incluido por lo que podemos poner etiquetas de html y funcionara igual que en un archivo.html.

Divide y Venceras

En esencia probablemente los archivos de php seran un poco extensos asi que los dividire en 2 partes o en “n” numero de partes que sean nesesarias.

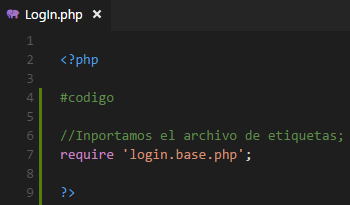
**Primera parte** : Todo el codigo robusto de php donde nos encargamos de hacer n cantidad de cosas relacionadas con programacion.

(Concentramos la mayor cantidad de codigo posible)

**Segunda parte** : Todas las etiquetas html y pequeños pedazos de codigo

y te preguntaras ¿como es que funcionara si estan en dos archivos diferentes las cosas?

Simplemente en php existe una manera de llamarlos … es con una funcion que se llama : “require :’archivo.extencion’”. de esa manera funcionaran con normalidad.



Asi de facil es.

POO(Programacion orientada a objetos)

y Conexiones PHP🡪Bases de datos MySQL

La programacion orientada a objetos no es complicada pero puede que tome tiempo entender algo para poder usarlo a la perfecccion.

Imaginemos el siguiente caso : “Estamos nosotros , esta mi madre , esta mi hermana y nosotros estamos enojados con mi madre por X motivo , entonces llamamos a mi hermana para que sea mi mensajera para poder comunicarme con mi madre. ” algo asi resulta ser la conexión a las bases de datos. Tenemos un objeto (PDO) que lo tenemos que nombrar en muchas ocaciones se suele poner una variable para el objeto… $conexion= new PDO(“Valores”); .

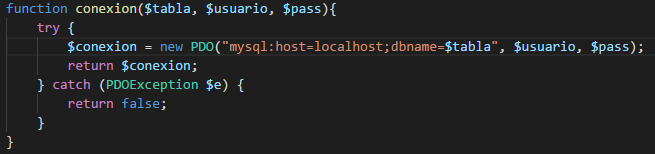
Ya que sabemos que conexión-chan es un objeto y que tiene una relacion muy sercana con nuestra base de datos nosotros solo le vamos a lanzar preguntas y $conexion-chan va a ir a preguntarle a la base de datos por nosotros y dependiendo de lo que la base de datos le responda llegara con nosotros a darnos el recado.

Si hacemos preguntas mal estructuradas o sin sentido … lo mas logico es que no sepa la base de datos la pregunta y no responda nada.

Si hacemos preguntas bien echas , bien estructuradas … respondera sin problemas.

Ahora que sabemos el principio del funcionamiento es momento de ponerlo en practica.

En la funcion pasamos solo el usuario , password , nombre de la tabla. La direccion donde se encuntra es la direccion local 127.0.0.1 o “Localhost”.



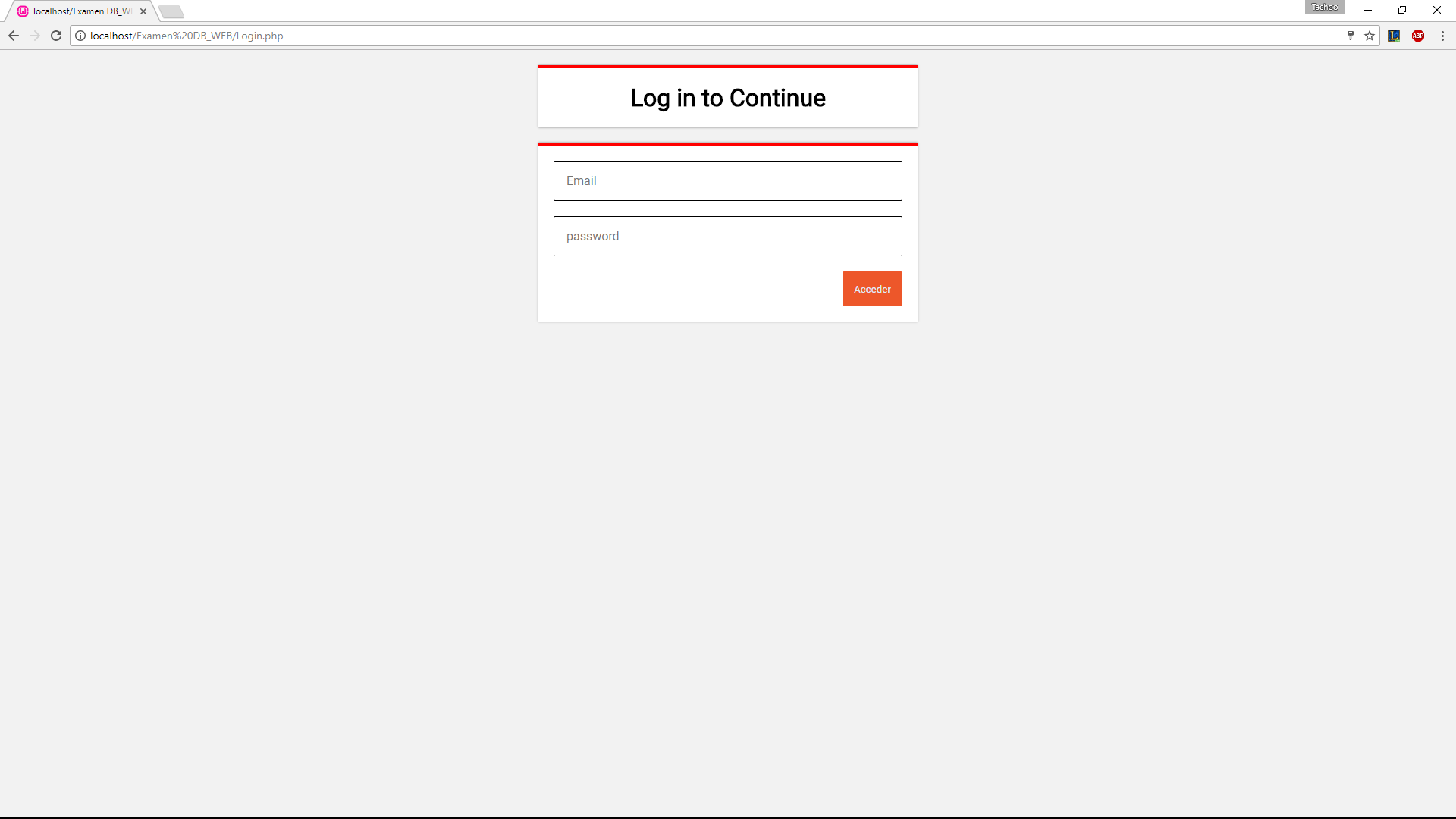
Si por alguna razon estamos equivocados en algo debemos de retornar falso para indicar que fallo algo en la conexión a la base de datos.

Es hora del codigo!!

Primero debemos de hacer nuestra pagina html la cual funcione bien y tenga sus estilos ya echos.

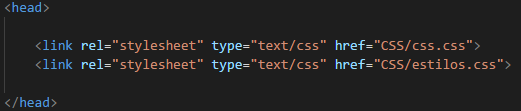
Asi que primero el diseño y despues le damos vida.

**Pagina de Login.php**



**Primero el html**

**Estilos:**



**Cuerpo:**



Despues los estilos css

**Css:**



Fuente RobotoSacada de la Api de googlefonts

**Estiloscss:**

\*

{

padding: 0px;

margin: 0px;

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

body

{

font-family: "Roboto" , Arial, Helvetica, sans-serif;

font-size: 16px;

background: #F2F2F2;

}

.warp

{

width: 90%;

max-width: 500px;

margin: auto;

}

#Title,form

{

width: 100%;

margin: 20 0;

padding: 20px;

background: #fff;

overflow: hidden;

box-shadow: 0 0 3px grey;

border-top: 4px solid red;

}

#Title h1

{

text-align: center;

vertical-align: middle;

}

form input[type="text"],

form input[type="password"]

{

border: 1px solid black;

border-radius: 2px;

padding: 16px;

width: 100%;

display: block;

margin-bottom: 20px;

font-family: "Roboto",Arial, Helvetica, sans-serif;

font-size: 1em;

color: #141938;

}

form input[type="text"]:focus,

form input[type="password"]:focus

{

border: 2px solid #3F51B1;

padding: 15px;

}

.alert

{

padding: 1em;

color: #fff;

border-radius: 2px;

margin-bottom: 20px;

font-size: 14px;

}

.alert.error

{

background: #F2DEDE;

border: 1px solid #A94442;

color: #A94442;

}

.alert.success

{

background: #4CAF50;

}

.submit,form input[type="submit"]

{

padding: 15px;

background: #ED572A;

color: #E1E9FA;

font-family: "Roboto",Arial, Helvetica, sans-serif;

border-radius: 2px;

border: none;

float: right;

cursor: pointer;

}

.submit,form input[type="submit"]:hover

{

background: #E64A19;

}

**Ahora el php:**

<?php

//Punto de Acceso

try

{ /\* 10.1.2.121 u720179037\_3exam u720179037\_ana SergioAnastacio20\*/

/\* 127.0.0.1 db\_web\_4\_test root \*/

$conexion=new PDO('mysql:host=127.0.0.1;dbname=u720179037\_3exam','root','');

}catch(PDOException $e)

{

echo "Error:".$e->getMessage();

}

$id="";

$name="";

$errores="";

$enviado="";

$login=false;

//Primero veo si la variable en post de submit existe

// digamos que es para validar desde este mismo php los datos antes de enviar a otro php

if(isset($\_POST['submit']))

{

//Si se envio el formulario entonces

// vamos a guardarlo en una variable tipo post para que no sea privada

$user=$\_POST['email'];

$password=$\_POST['password'];

//Tenemos algo pero no sabemos que es lo que tenemos... creo que es emomento de ver que es

//estara vacio ?

if(!empty($user))

{

//Pues ya sabemos que no esta vacio es momento de limpear lo que tenga adentro

$user=trim($user);

$user=filter\_var($user,FILTER\_SANITIZE\_EMAIL);

if(!filter\_var($user,FILTER\_VALIDATE\_EMAIL))

{

$errores.="Ingresa un correo valido <br\>";

}

}else

{

$errores.="";

}

if(!empty($password))

{

//Pues ya sabemos que no esta vacio es momento de limpear lo que tenga adentro

$password=trim($password);

$password=filter\_var($password,FILTER\_SANITIZE\_STRING);

}else

{

$errores.=" Rellene todos los campos de el Formulario";

}

/\*Si no encuentra errores quiere decir que todo esta bien y listo para hacer la query\*/

if(!$errores)

{

//Preparamos la Query

$statement=$conexion->prepare('SELECT id,nombre FROM users\_data WHERE email=:user');

//lanzamos la Query con el valor obtenido del formulario ( correo y contrase;a)

$statement->execute( array(':user'=>$user) );

$result=$statement->fetch();

//Debemos de ver si el arreglo es mayor a 0 de ser asi es que se lanzo la Query Bien y por consecuente si existe el correo en la base de datos

if($result>0)

{

/\*Sacamos algunos datos que ocuparemos despues\*/

$id=$result['id'];

$name=$result['nombre'];

/\*Sacamos algunos datos que ocuparemos despues\*/

$statement=$conexion->prepare("SELECT password\_data.password FROM password\_data JOIN users\_data ON password\_data.id\_user = users\_data.id WHERE password\_data.id\_user=:id");

$statement->execute( array(':id'=>$id) );

//Obtenemos todos los Resultados

$result=$statement->fetchAll();

if($result>0)

{

//Sabemos que el resultado es mayor a 0 entonces ya sabemos que tenemos todas las contrase;as

foreach ($result as $result => $value)

{

//Recorremos hasta saber cual es el correcto ( problema al tener 2 contrase;as iguales en una misma persona)

if($value['password']==$password)

{

//Mucha mas seguridad

$enviado="Bienvenido De Nuevo Ing.".$name;

$login==true;

break;

}

}

//Solo cuando enviado sea diferente de "" y login sige siendo falso

if(!$enviado&&$login==false)

{

$errores.="Usuario y/o Password incorrecto";

}

}else

{

}

}else

{

//de lo contrario sabemos que no hay nada en la base de datos. y lo

$enviado="";

//por motivos de seguridad deberiamos de mantener en anonimato que el correo es el que esta mal.

$errores.="Usuario y/o Password incorrecto";

}

}

}

//Inportamos el test.php

require 'login.base.php';

?>

La idea inicial es que no pase a otra pagina hasta que no sea validado al 100%

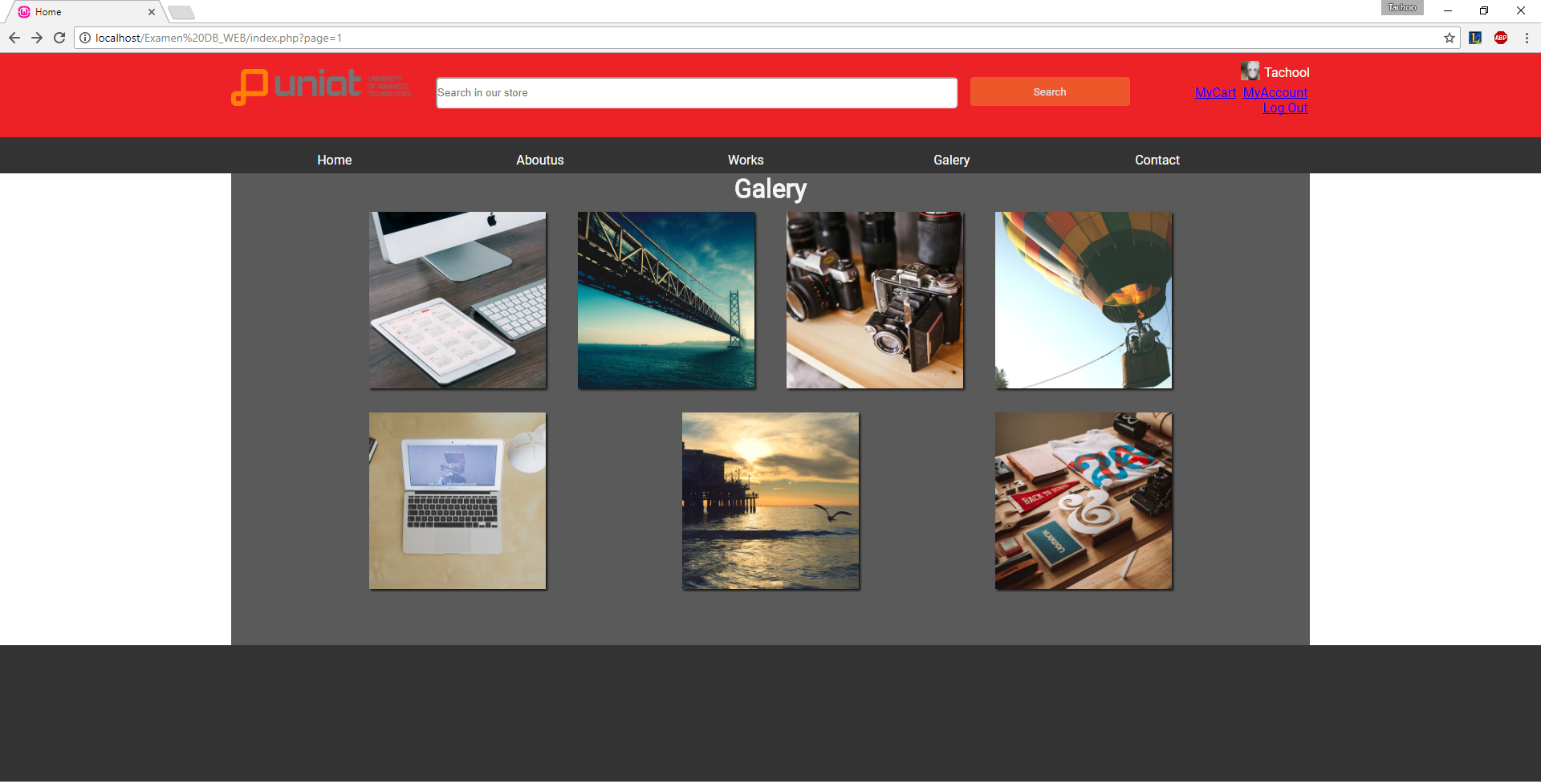
Jugar con los valores de $errores para hacer que despliege mensajes de errores en el html.

$enviado para cuando se tiene que mostrar el mensaje personalizado al momento de iniciar session

$login una bandera de paso.

Lo demas se encarga las condiciones del documento que tiene codigo html y php junto.

**Pagina de index.php**



**Primero el html:**

<!DOCTYPE>

<html>

<head>

<meta charset="UTF-8">

     <meta name="viewport" content="width=device-width, user-scalable=no,

     initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<!--Solo para obtener unas imagenes en forma de txt-->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css">

<!--Obtenemos una fuente desde google-->

<link href='https://fonts.googleapis.com/css?family=Slabo+27px' rel='stylesheet' type='text/css'>

<!--Estilosbase y una fuente Roboto-->

<link rel="stylesheet" type="text/css" href="CSS/base\_home.css">

<link rel="stylesheet" type="text/css" href="CSS/dinamic\_presets.css">

<link rel="stylesheet" type="text/css" href="CSS/css.css">

<title><?php echo$pagetitle;?></title>

</head>

<body>

<div id="warper">

<!--top-->

<div id="header">

<div class="posfixed">

<div id="TA"><!--Logo>SearchSection>AccountName-->

<figure id="Logosec">

<img id="logo" src="img/logo/uniat.png">

</figure>

<!--Search-->

<div id="searchbar">

<form id="SearchBrowser">

<input type="text" placeholder="Search in our store" id="TextArea">

<input type="submit" value="Search" class="submit" id="SubmitArea">

</form>

</div>

<!--ACC-->

<div id="accountname">

<div id="NameandPick">

<figure id="ProfilePic">

<?php

echo'<a href="#"><img src="img/profilepicks/'.$\_SESSION["user"][0]["profilepic"].'"></a>'; /\*Eureka !!! esta muy grande el arreglo 3 dimencional\*/

?>

</figure>

<div id="username">

<?php echo'<p>'.$\_SESSION["user"][0]["nombre"].'</p>'?>

</div>

</div>

<div id="opcions">

<ul>

<li><a href="#"class="links">MyAccount</a></li>

<li><a href="#"class="links">MyCart</a></li>

<li><a href="index.php?logout=3%lg$va%us235"class="links">Log Out</a></li>

</ul>

</div>

</div>

</div>

</div>

<!--NavBar-->

<div id="nabvar">

<div class="posfixed" >

<div id="TB"><!--Navbar-->

<ul id="navfix">

<?php

$index=1;

foreach ($Menu as $key => $value)

{

echo '<li><a href="index.php?page='.$index.'"><br><p>'.$value.'</p></a></li>';

$index++;

}

?>

</ul>

</div>

</div>

</div>

</div>

<!--mid-->

<div class="posfixed" id="mid">

<?php if(!empty($extra)):?>

<!--Titulo-->

<?php

if(!empty($title))

{

echo '<div class="class'.$title[0].'"><h1>'.$title[1].'</h1></div>';

}

?>

<!--Descripcion-->

<?php

if(!empty($Description))

{

echo '<div id="dinamic\_grid"><div class="class'.$Description[0].'"><dd>'.$Description[1].'</dd></div>';

}

?>

<!--Imagen-->

<?php

if(!empty($Subbaner[0])&&!empty($Subbaner[1]))

{

echo '<div class="class'.$Subbaner[0].'"><img src="img/subbaners/'.$Subbaner[1].'"></div></div>';

}else

{

}

?>

<!--Extra-->

<?php elseif(empty($extra)):?>

<!--Titulo-->

<?php

if(!empty($title))

{

echo '<div class="class'.$title[0].'"><h1>'.$title[1].'</h1></div>';

}

?>

<!--Descripcion-->

<?php

if(!empty($Description))

{

echo '<div id="dinamic\_grid"><div class="class'.$Description[0].'"><dd>'.$Description[1].'</dd></div>';

}

?>

<!--Imagen-->

<?php

if(!empty($Subbaner[0])&&!empty($Subbaner[1]))

{

echo '<div class="class'.$Subbaner[0].'"><img src="img/subbaners/'.$Subbaner[1].'"></div></div>';

}

?>

<!---->

<?php elseif(!empty($extra)):?>

<?php endif;?>

<!---->

<div class="contenedor">

<?php foreach($fotos as $foto):?>

                 <div class="thumb">

                     <a href="foto.php?id=<?php echo $foto['id']."&p="; echo $pagina\_actual; echo"&page=".$\_GET['page']; ?>">

                     <img src="img/fotos/<?php echo $foto['imagen'] ?>" alt="">

                     </a>

                 </div>

             <?php endforeach;?>

<div class="paginacion">

                 <?php if ($pagina\_actual > 1): ?>

                     <a href="index.php?<?php echo "page=".$\_GET['page']."&p="; echo $pagina\_actual - 1;?>" class="izquierda"><i class="fa fa-long-arrow-left"></i> Pagina Anterior</a>

                 <?php endif ?>

                 <?php if ($total\_paginas != $pagina\_actual): ?>

                     <a href="index.php?<?php echo "page=".$\_GET['page']."&p=";echo $pagina\_actual + 1;?>" class="derecha">Pagina Siguiente <i class="fa fa-long-arrow-right"></i></a>

                 <?php endif ?>

</div>

</div>

<!---->

</div>

</div>

<!--bot-->

<div id="footer" >

<div class="posfixed" id="bot">

<!-- Seecion de informacion -->

<!--Seccion de boletin informativo-->

<!--Seccion de patrocinadores-->

</div>

</div>

</body>

</html>

**Css:**

**Home\_base.css:**

\* {

    margin: 0;

    padding: 0;

    -webkit-box-sizing: border-box;

    -moz-box-sizing: border-box;

    box-sizing: border-box;

}

/\*Base SetUP\*/

body

{

position: relative;

font-family: "Roboto",Arial, Helvetica, sans-serif;

font-size: 16px;

font-style: normal;

color: black;

}

h4

{

font-size: 14px;

}

a

{

text-decoration: none;

text-align: center;

vertical-align: middle;

color: black;

}

li

{

list-style: none;

}

.links

{

text-decoration-line: underline;

color:blue;

}

.links:hover

{

text-decoration-line: none;

}

/\*Base SetUp\*/

/\*Base Structure\*/

#warper

{

width: 100%;

}

.posfixed

{

width: 70%;

margin: 0 auto;

}

/\*Header\*/

#header

{

top: 0px;

position: relative;

background: #ec2227;

}

/\*top\*/

#TA

{

width: 100%;

height: 105px;

display: flex;

}

/\*Logo\*/

#Logosec

{

position: left;

width:20%; /\*-->> Depens \*/

height: 100%;

}

#logo

{

width: 100%;

margin-top:20px;

}

/\*Logo\*/

/\*Search\*/

#searchbar

{

width: 80%; /\*-->> Depens \*/

height: 100%;

}

#SearchBrowser

{

margin-top: 30px;

margin-left: 30px; /\*should center as pixel perfect\*/

background: #ec2227;

}

#SearchBrowser input

{

margin: 0px;

padding: 0px;

vertical-align: middle;

padding-bottom: 10px;

padding-top:10px;

}

#SearchBrowser>#TextArea

{

width: 75%;

margin-left: 1px;

border-radius: 5px;

}

#SearchBrowser>#SubmitArea

{

width: 23%;

border-radius: 4px;

}

form input[type="text"],

{

border: 1px solid black;

border-radius: 2px;

border-radius: 2px;

font-family: "Roboto",Arial, Helvetica, sans-serif;

font-size: 1em;

color: #141938;

}

form input[type="text"]:focus,

{

border: 2px solid #3F51B1;

padding: 15px;

}

.submit,form input[type="submit"]

{

padding: 15px;

background: #ED572A;

color: #E1E9FA;

font-family: "Roboto",Arial, Helvetica, sans-serif;

border-radius: 2px;

border: none;

float: right;

cursor: pointer;

}

.submit,form input[type="submit"]:hover

{

background: #E64A19;

}

/\*Search\*/

/\*ACC\*/

#accountname

{

display: inline-block;

position: relative;

right: 0px;

width: 20%; /\*-->> Depens \*/

height: 100%;

}

#NameandPick

{

position: absolute;

margin-top:10px;

right: 0px;

display: flex;

}

#ProfilePic

{

width:24px;

height:24px;

}

#ProfilePic img

{

width: 100%;

height: 100%;

}

#username

{

margin-top:5px;margin-left:5px;

}

#username p

{

color:white;

}

/\*Caso cuando esta logeado\*/

#opcions

{

margin-top:35px;

width: 100%;

height: 70px;

}

#opcions ul

{

padding-top: 5px;

margin-left: 33%;

}

/\*Acomodo d las opciones de la cuenta\*/

#opcions ul li

{

margin-left: 5px;

margin-right: 3px;

display:block;

float: right;

}

/\*ACC\*/

/\*top\*/

/\*NavBar\*/

#nabvar

{

background:#333333;

}

#TB

{

width: 100%;

height: 45px;

display: inline-block;

}

#navfix

{

display: inline-block;

width: 100%;

height: 100%;

}

#navfix li

{

display: inline-block;

margin-left: 1px;

width: 19%; /\*-->> Perfecto\*/

height: 100%;

}

#navfix li:hover

{

background:#E64A19;

}

#navfix li a

{

width: 100%;

height: 100%;

}

#navfix li a p

{

color: white;

}

/\*NavBar\*/

/\*Header\*/

/\*mid\*/

#mid

{

background:rgba(20,20,20,0.7);

position: relative;

}

/\*mid\*/

/\*footer\*/

#footer

{

position: relative;

bottom: 0px;

width: 100%;

background: #333333;

}

/\*bot\*/

#bot

{

height: 170px;

}

/\*bot\*/

/\*footer\*/

/\*Base Structure\*/

**Dinamic\_presets.css:**

\* {

    margin: 0;

    padding: 0;

    -webkit-box-sizing: border-box;

    -moz-box-sizing: border-box;

    box-sizing: border-box;

}

/\*dinamic\_grid\*/

#dinamic\_grid

{

margin: 10px auto;

width: 100%;

display: inline-block;

}

/\*Left\*/

.class1

{

float: left;

width: 50%;

}

/\*Midle\*/

.class2

{

margin: 0px auto;

width: 33%;

text-align: center;

}

/\*Rigth\*/

.class3

{

float: right;

width: 50%;

}

.class1 dd ,.class2 dd ,.class3 dd

{

margin: 10px;

text-align: justify;

vertical-align: middle;

color: whitesmoke;

}

.class1 h1 ,.class2 h1 ,.class3 h1

{

color: whitesmoke;

}

/\*Para cualquier imagen adentro de esas classes\*/

.class1 img,.class2 img,.class3 img

{

width: 100%;

}

/\*Galeia\*/

.titlulo

{

margin: 0 auto;

}

a {

    text-decoration: none;

    color:#fff;

}

a:hover {

    text-decoration: none;

}

.contenedor {

    max-width: 1000px;

    width: 90%;

    margin: auto;

}

.izquierda {

    float: left;

}

.derecha {

    float: right;

}

/\* --- Header ---\*/

header {

    margin: 40px 0;

    font-size: 20px;

    text-align: center;

}

/\* --- imagenes --- \*/

.fotos, .contenedor

{

    display: flex;

    flex-direction: row;

    flex-wrap: wrap;

    justify-content: space-between;

}

.thumb {

    width: 22%;

    margin-bottom: 30px;

    outline:5px solid rgba(255,255,255,0);

    -webkit-box-shadow: 2px 2px 3px rgba(0,0,0,.8);

    box-shadow: 2px 2px 3px rgba(0,0,0,.8);

    -webkit-transition: all .3s ease;

    -o-transition: all .3s ease;

    transition: all .3s ease;

}

.thumb:hover {

    outline:5px solid rgba(255,255,255,1);

}

.thumb a {

    display: block;

}

.thumb img {

    vertical-align: top;

    width: 100%;

}

/\* --- Single Foto --- \*/

.foto {

    width: 100%;

    margin: auto;

    max-width: 500px;

    text-align: center;

}

.foto img {

    width: 100%;

    vertical-align: top;

    -webkit-box-shadow: 2px 2px 3px rgba(0,0,0,.8);

    box-shadow: 2px 2px 3px rgba(0,0,0,.8);

    margin-bottom: 30px;

}

.foto .texto {

    max-width: 500px;

    margin-bottom: 30px;

}

.regresar {

    padding: 5px 10px;

    margin-bottom: 30px;

    display: inline-block;

    border-radius: 50px;

    font-size: 16px;

    border:1px solid rgba(255,255,255,.2);

    -webkit-transition: all .3s ease;

    -o-transition: all .3s ease;

    transition: all .3s ease;

}

.regresar:hover {

    border:1px solid #fff;

    text-decoration: none;

}

.regresar .fa {

    margin-right: 15px;

}

/\* --- Footer --- \*/

footer .copyright {

    text-align: center;

    color:rgba(255,255,255,.5);

    font-size: 16px;

    padding: 15px 0;

}

/\* --- Paginacion --- \*/

.paginacion {

    width: 100%;

    margin-bottom: 30px;

    overflow: hidden;

}

.paginacion a{

    display: inline-block;

    padding: 15px 20px;

    border-radius: 50px;

    border:1px solid rgba(255,255,255,.2);

    -webkit-transition: all .3s ease;

    -o-transition: all .3s ease;

    transition: all .3s ease;

}

.paginacion a:hover {

    border:1px solid #fff;

    text-decoration: none;

}

.paginacion .izquierda .fa {

    margin-right: 10px;

}

.paginacion .derecha .fa {

    margin-left: 10px;

}

/\*Galeia\*/

#bg

{

background: #333333;

color: white;

}

#bg>p

{

padding-bottom: 30px;

}

**Php:**

<?php

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Debemos de hacer un filtro para cualquier posible accion de l "USUARIO"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

session\_start();//Accedemos a la session si es que hay una

//Si existe una session y esta vacia entonces debemos asumir que no esta logeado.

if(isset($\_SESSION))

{

if(empty($\_SESSION))

{

header("refresh: 0; url = Login.php");

}

}

//Si existe alguna variable en el logout entonces debemos de asumir que el usuario quiere salir de la web

if(isset($\_GET["logout"]))

{

//Cerramos la session por motivos de seguridad

session\_destroy();

//Lo mandamos a la pagina de login para indicarle que su session fue cerrada

header("refresh: 2; url = Login.php");

}

//Obtenemos el numero de la pagina

if(isset($\_GET['page'])&&!empty($\_SESSION))

{

//verificamos que la variable no este vacia

if(!empty($\_GET['page']))

{

// solo verificamos page sea algo valido

if($\_GET['page']<0||$\_GET['page']==0)

{

//Si es <=0 entonces asumimos que esta drogrado el usuario e imponemos nuestra autoridad mandandolo a la pagina de home de todos modos.

$page=1;

}

// si page es cualquier otra cosa pues se lo asignamos a la variable $page;

else

{

$page=$\_GET['page'];

}

}

else

{

//sabemos que esta vacia la variable de page debemos de darle un index hacia la pagina prinicipal

//$page=1;

}

}

else

{

//Como sabemos el usuario termino haciendo algo que no preveimos ... debemos de arreglarlo

// $page=1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Ya que sabemos que el usuario hasta el momento se porto bien o mas bien paso los filtros XD podemos continuar.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Requerimos parte del archivo 'Funciones.php' para poder hacer uso de ellas

require 'funciones.php';

//Realizamos la conexion a la base de datos

$conexion = conexion('u720179037\_3exam', 'root', '');

//Solo si algo salio mal hay que cortar la conexion;

if (!$conexion)

{

  die();

}

//Variables que ocupamos en el flujo de datos de la primera Query

$pagetitle="";

$Menu=array();

/\*

//-->Primera Query!

\*/

//Preguntamos al la base de datos sobre las paginas existentes

$statement=$conexion->prepare('SELECT \* FROM page\_data'); //Limitamos el numero de cosas por el especio en el css

$statement->execute();

$result=$statement->fetchAll();

//Comprobamos y efectuamos cambios

if($result>0)

{

foreach ($result as $key => $value)

{

//obtenemos todos los titulos del la base de datos y los ponemos en un arreglo

array\_push($Menu,$value['title']);

//comparamos si el id de la base de datos es la misma del id que el usuario quiere ver

if($value['id']==$page)

{

//Si es asi pues el title de la pesta;a debe de cambiar

$pagetitle=$value['title'];

}

}

}

else

{

//Mensaje de debug despues se debe de cambiar por el error 404; (Literalmente no se encontro la pagina)

echo"Algo fallo al tratar de traer todos los menus posibles y ligas";

}

/\*

//-->Segunda Query!

\*/

//preguntamos a la base de datos sobre el contenido de la pagina ya validada

$statement=$conexion->prepare('SELECT \* FROM content\_data where page\_id=:page '); // Modificacion para que las paginas puedan cambiarse de lugar... #Quiero entregar para ser libre

$statement->execute( array(':page'=>$page));

$result=$statement->fetch();

if($result>0)

{

/\*

Agurpamos los resultados de el Query en Arreglos:

el lugar numero 0 sea la classe

el lugar numero 1 sea el contenido

\*/

//Titulo

$title=array();

array\_push($title,$result['titleclass']);

array\_push($title,$result['title']);

//Description

$Description=array();

array\_push($Description,$result['descclass']);

array\_push($Description,$result['description']);

// Subbaner

$Subbaner=array();

array\_push($Subbaner,$result['subbanerclass']);

array\_push($Subbaner,$result['subbaner']);

/\*problema de extra\*/

/\*

Extra es un campo que debendiendo de lo que tenga adentro es lo que va a mostrar

\*/

/\*problema de extra\*/

}else

{

//Mensaje de debug despues se debe de cambiar por el error 404; (Literalmente no existe contenido en esa pagina y por consecuente no debemos de mostrar la pagina)

echo" No Existe Contenido En este Index";

}

/\* Debemos de ver si realmente exixte una galeria en la pagina\*/

//Si la casilla de extra tiene algo

$fotos\_por\_pagina = 12;

$pagina\_actual = (isset($\_GET['p']) ? (int)$\_GET['p'] : 1);

$inicio = ($pagina\_actual > 1) ? $pagina\_actual \* $fotos\_por\_pagina - $fotos\_por\_pagina : 0;

$statement=$conexion->prepare('SELECT SQL\_CALC\_FOUND\_ROWS \* FROM galery\_data WHERE page\_id=:page LIMIT '.$inicio.', '.$fotos\_por\_pagina.'');

$statement->execute( array(':page'=>$page) );

$fotos = $statement->fetchAll();

if (!$fotos) {

 header('refresh: 0; url = index.php');

}

$statement = $conexion->prepare("SELECT FOUND\_ROWS() as total\_filas");

$statement->execute();

$total\_post = $statement->fetch()['total\_filas'];

$total\_paginas = ceil($total\_post / $fotos\_por\_pagina);

/\*Debemos de ver si relmente existe una galeria en la pagina\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

Ya que tenemos tanto el contenido de la pagina como el titulo y otras cosas hacemos el requerimiento de el template o la

maqueta de la pagina.

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

require "index.base.php";

//Fin del flujo de datos

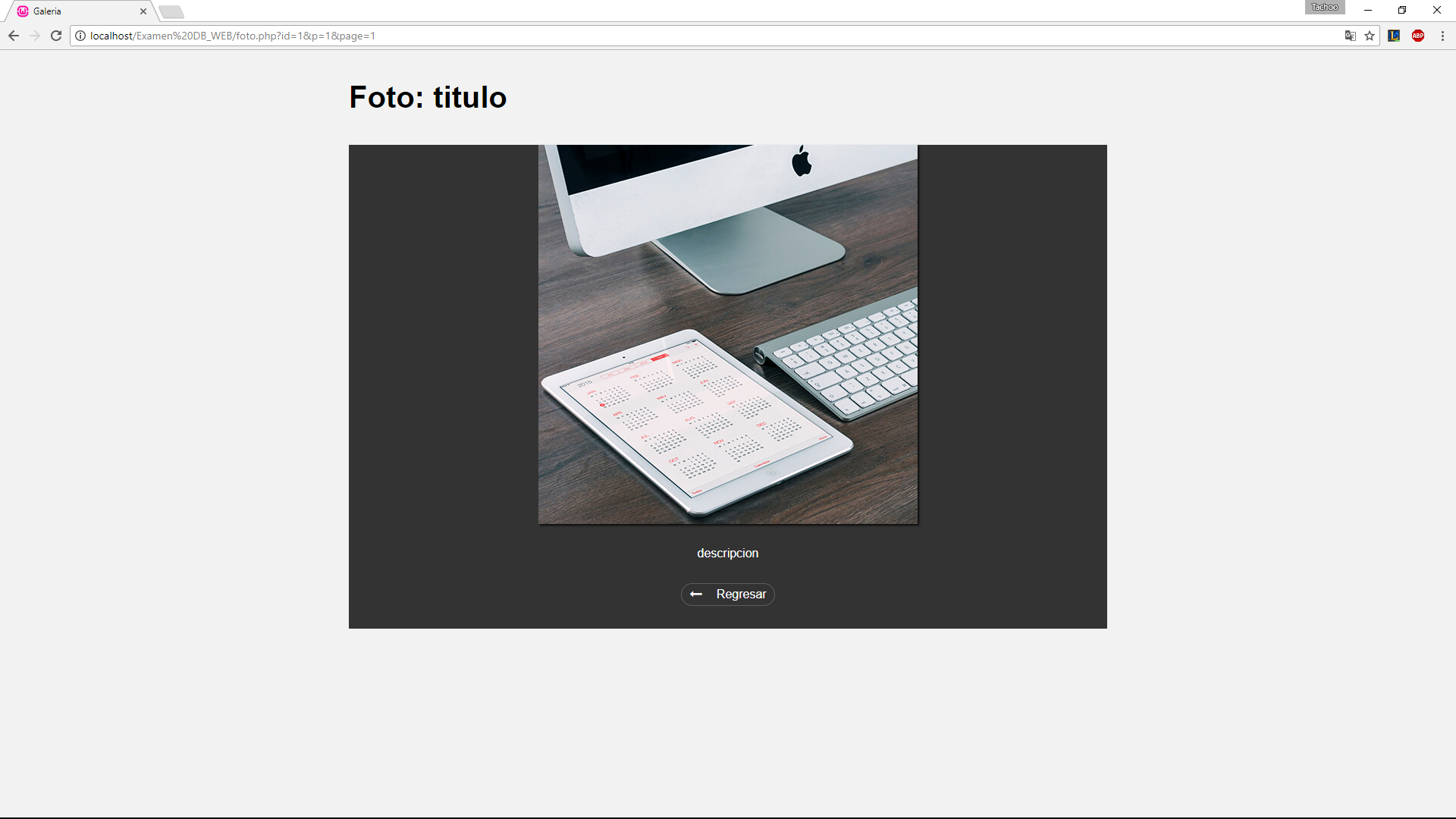
?>

La idea es tan simple como ir a la base de datos preguntar si hay cosas o no. Evaluar guardarlas en variables y despues en el docuemnto donde tiene etiquetas ponerlas en el espacio destinado con el acomodo que viene de default desde la base de datos.

Manipulacion de strings para ir y venir de una pagina a otra , consistencia de datos … etc…

Nota : Puede de que existan otras maneras diferentes de hacerlo pero esta es la forma en la que decidi hacerlo por convicion propia.

**Foto.php:**



**Html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, user-scalable=no,

     initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

     <link href='https://fonts.googleapis.com/css?family=Slabo+27px' rel='stylesheet' type='text/css'>

     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css">

     <link rel="stylesheet" href="CSS/dinamic\_presets.css">

     <link rel="stylesheet" href="CSS/estilos.css">

    <title>Galeria</title>

</head>

<body>

    <header>

        <div class="contenedor">

            <h1 class="titulo">Foto: <?php if (!empty($foto['titulo'])) {

                echo $foto['titulo'];

            } else {

                echo $foto['imagen'];

            } ?></h1>

        </div>

    </header>

    <div class="contenedor" id="bg">

        <div class="foto">

            <img src="img/fotos/<?php echo $foto['imagen']; ?>" alt="">

            <p class="texto"><?php echo $foto['descripcion'].'</p>';?>

            <a href="index.php?page=<?php echo"$Back&p=".$page; ?>" class="regresar"><i class="fa fa-long-arrow-left"></i> Regresar</a>

        </div>

    </div>

</body>

</html>

**PHP:**

<?php

require "funciones.php";

$Back=$\_GET['page'];

$RedirecionAutomatica="Location: index.php?page=.$Back";

$conexion = conexion('u720179037\_3exam', 'root', '');

if (!$conexion) {

    die();

}

$id = isset($\_GET['id']) ? (int)$\_GET['id'] : false;

$page=isset($\_GET['p']) ? (int)$\_GET['p']:false;

if (!$id) {

    header($RedirecionAutomatica);

}

$statement = $conexion->prepare('SELECT \* FROM galery\_data WHERE id = :id');

$statement->execute(array(':id' => $id));

$foto = $statement->fetch();

if (!$foto) {

    header($RedirecionAutomatica);

}

require "foto.base.php";

?>

**NOTAS:** Se uso estilos externas para el uso de estilos con la finalidad de que no sea solo estilos basicos y poder ver como se maneja la industria en la actualidad.

Concluimos:

El proyecto como tal es interesante a tal grado que se puede llegar a crear cosas de mucha calidad. Puede que por tiempos , disponibilidad , otros factores no se dieron las cosas para el mayor aprendizaje posible en web 1 pero!! Se cumplio el objetivo tener la capacidad de hacer una web desde 0 para “n” propocitos y por su puesto diseñar , gestionar e implementar una base de datos en tecnologias web.

No es que me sienta satisfecho por el desempeño que logre en esta materia… por algun motivo me gustaria seguir aprendiendo el mundo de las tecnologias web y poder ser capaz de solucionar problemas cotidianos que me pueda topar en este mi camino.