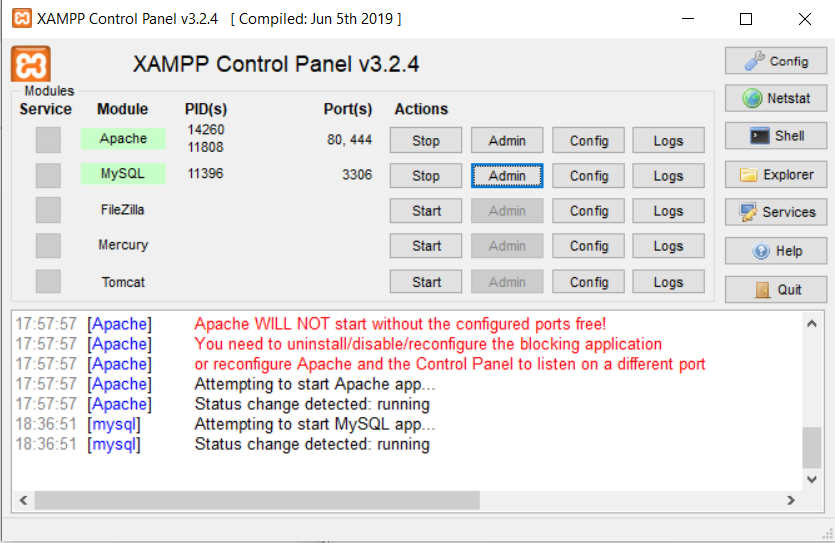
# Bases de Datos en XAMPP

Como sabes Xampp viene con MariaDB. Vamos a poner el servicio MySQL en marcha para poder utilizarlo.



Podremos gestionarlo desde <http://localhost/phpmyadmin/>

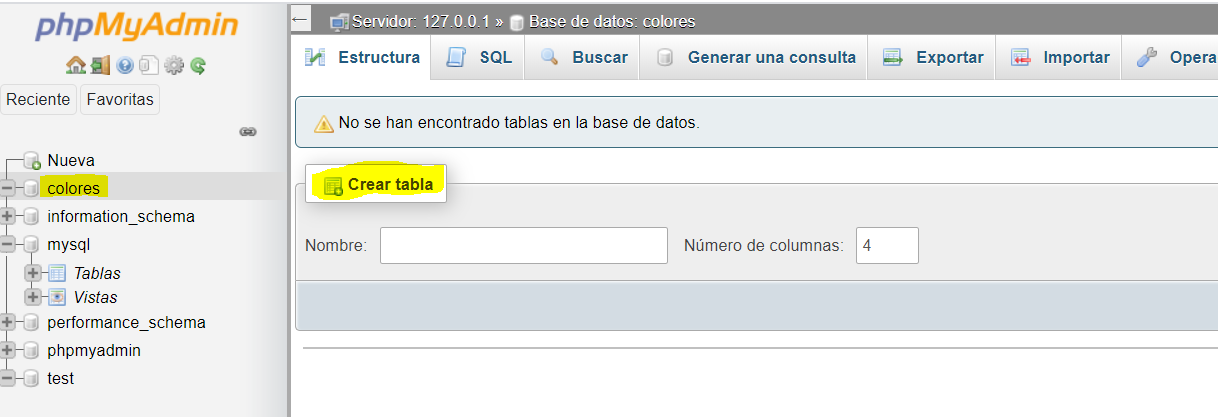


Tenemos que tener en cuenta que por defecto el usuario root de MariaDB en Xampp no tiene contraseña.

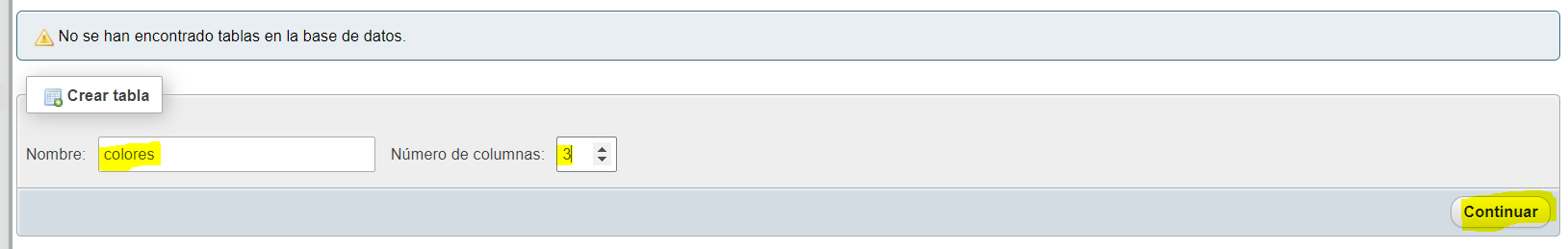
# Crear base de datos desde interfaz e introducir datos



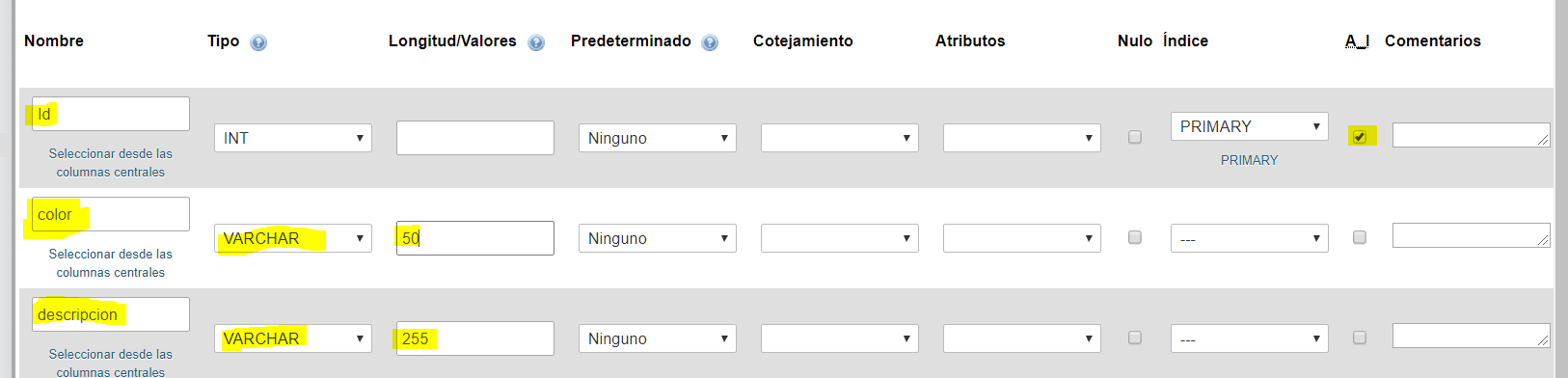
En la pestaña Bases de datos escribiremos el nombre de la base de datos que queremos crear, en nuestro caso se llamará **colores**. Cuando presionemos crear nos aparecerá la base de datos colores en el lado izquierdo de la pantalla.



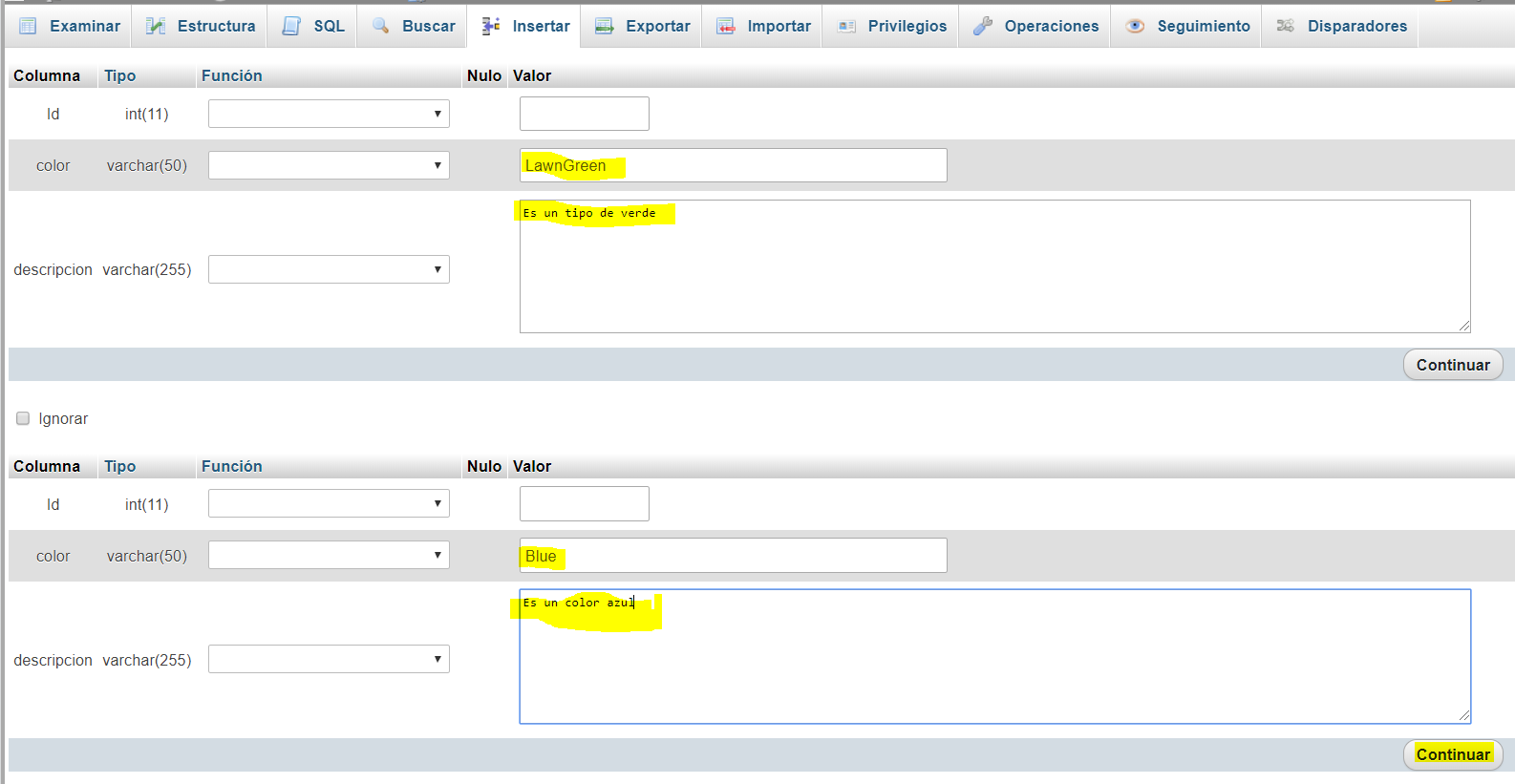
Si seleccionamos la base de datos que acabamos de crear nos da la opción de crear una tabla. Crearemos una tabla llamada colores, de tres columnas:



Cuando le demos a continuar nos pedirá que definamos las columnas:



Cuando le demos a guardar se nos creará la primera tabla. Para insertar valores clicaremos en Insertar e introduciremos un color. Si le damos a continuar creará una sentencia SQL, vuelve a darle a continuar.



Prueba a introducir otros dos colores.

# Conexión.php: Conectarse desde PHP

Crea una nueva carpeta llamada colores en tu carpeta htdocs de xampp. En esta carpeta crea un archivo llamado conexion.php y abre la carpeta en Visual Studio Code.

Hay varias formas de conectarse a una base de datos desde PHP, nosotros [usaremos PDO](https://www.php.net/manual/es/pdo.connections.php) porque se puede utilizar para varios tipos de bases de datos, no sólo mysql.

<?php

$link = 'mysql:host=localhost;dbname=colores';

$usuario="root";

$pwd = "";

try { //Intenta hacer algo

    //La variable mbd guardará la conexión a la base de datos

    $gbd = new PDO($link, $usuario, $pwd);

    print "Conectado!";

} catch (PDOException $e) {//Si salta un error en el try vendrá aquí

    print "¡Error!: " . $e->getMessage() . "<br/>";

    die();

}

Si cambias el valor de la variable usuario o contraseña verás que va al catch y te muestra un error de conexión de la base de datos.

# Sentencia SELECT

Creamos un archivo index.php y desde allí llamaremos a conexión.php para que se cree la conexión. Prueba este ejemplo:

<?php

    include\_once 'conexion.php';

    print " desde archivo index";

?>

En el resultado vemos que se ha ejecutado conexion.php y luego nos ha hecho el print.

Ahora leeremos el contenido de nuestra tabla. Para ello guardaremos la sentencia SQL para el SELECT en una variable y la prepararemos para ejecutar.

<?php

    include\_once 'conexion.php';

    /\*\* SELECT DE LA BASE DE DATOS \*/

    $sql\_lectura = "SELECT \* FROM colores"; //Sentencia SQL que queremos ejecutar

    $sentencia\_select = $gbd->prepare($sql\_lectura);

        //$gbd guarda la conexión creada en el archivo conexión.php

        //$sentencia\_select  será usada para ejecutar la sentencia

    $sentencia\_select->execute(); //para que se ejecute la sentencia que hemos preparado

    $resultado = $sentencia\_select->fetchAll(); //fetchall nos devuelte una matriz con el resultado de la sentencia

?>

Dejaremos este código al inicio del archivo y después del cierre de PHP añadiremos la estructura de HTML de una página web. Puedes copiar lo siguiente:

<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="utf-8">

  <title>

    Conexión con Base de Datos

  </title>

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="css/miestilo.css" title="Color">

</head>

<body>

  <h1>Ejercicio con base de datos</h1>

    <!-- Aquí añadiremos el contenido partiendo del resultado de la sentencia SQL-->

  <footer>

    <p>Escriba aquí su nombre</p>

  </footer>

</body>

</html>

Haremos que con cada tupla de la base de datos nos muestre una línea:

    <table>

        <?php

            foreach($resultado as &$dato) //de la matriz del resultado toma un dato (una tupla)

            {

                print "<tr><td>$dato[color]</td></tr>";

            }

        ?>

    </table>

Ahora cambiaremos el estilo para que el color de fondo de la celda sea el color que escribimos en ella. Para ello jugaremos con el estilo de la fila. Añadiremos también otra celda para la descripci La parte de php nos queda así:

<?php

     foreach($resultado as &$dato) //de la matriz del resultado toma un dato (una tupla)

     {

          print "<tr style=\"background-color:$dato[color]\">

                 <td>$dato[color]</td>

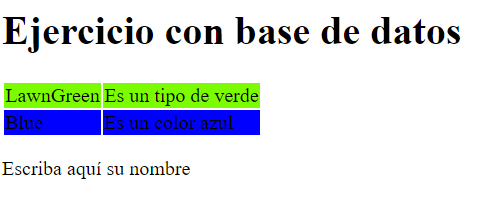
                 <td> $dato[descripcion]</td>

                 </tr>";

    }

?>

Este es el resultado:



# Sentencia INSERT

Para poder insertar los datos vamos a poner un formulario debajo de la tabla. Este formulario tendrá un hueco para introducir el color y otro para la descripción.

    <form action="index.php" method="POST">

            <p><label for="color">Color</label><input type="text" name="color"></p>

            <p><label for="descripcion">Descripción</label><input type="text" name="descripcion"></p>

            <p><button>Añadir</button></p>

    </form>

Con esto nos aparecerá un formulario debajo de la tabla de colores. Ahora trataremos la respuesta a este formulario en la misma página en la que estamos. Debajo del código para la lectura de la tabla de la base de datos añadiremos el código para insertar datos.

    /\*\*\*INSERT en la base de datos\*/

    if($\_POST){//Si le llegan datos de un formulario

        $color = $\_POST["color"];//busca el valor del input llamado color

        $descripcion = $\_POST["descripcion"];//busca el valor del input llamado descripcion

        if($color != "" and $descripcion!=""){ //Para que no guarde nada si uno de los valores está vacío

            $sql\_escritura = "INSERT INTO colores(color,descripcion) VALUES (?,?)";

            //poniendo ? evitamos que nos hagan SQL injection

            $sentencia\_insert = $gbd->prepare($sql\_escritura);

            $sentencia\_insert->execute(array($color,$descripcion)); //le damos un valor a los ?

            header("location:index.php");//para que una vez que se ejecute esto redirija a esa página, para que se cargue

        }

    }

# Sentencia EDIT

Para editar un archivo tenemos que tener en cuenta que si mandamos los valores como si se tratara de un formulario con el método GET los añade a la dirección de la página. Si cambiamos por un momento el método en el formulario para insertar datos veremos que se mandan los datos en este formato:

http://localhost/colores/index.php?id=1&color=Green&descripcion=este+color+es+verde

Crea un archivo que se llama editar.php, tendremos en cuenta cómo se mandan los datos de un formulario.

Para leer los datos que contiene el método GET haremos lo siguiente:

//Leemos los elementos que nos llegan por medio del método GET

$id = $\_GET['id'];

$color = $\_GET['color'];

$descripcion = $\_GET['descripcion'];

//Si todavía no se ha hecho la conexión la hacemos

include\_once 'conexion.php';

//UPDATE

$sql\_editar = "UPDATE colores SET color=?, descripcion=? WHERE id=?";

// Como antes, los ? evitan el sql injection

$sentencia\_update = $gbd->prepare($sql\_editar);

$sentencia\_update->execute(array($color, $descripcion,$id));

header("location:index.php");//para que una vez que se ejecute esto redirija a esa página, para que se cargue

Tenemos creada la opción para editar los datos de un color, pero por ahora no la llamamos en ningún sitio. En la tabla en la que cargamos los colores vamos a añadir una celda con el icono Unicode de una libreta, cuyo código es &#128221.

Y le añadiremos un hipervínculo, en el que se envía la id de color a la página index.php

 <?php

    foreach($resultado as &$dato) //de la matriz del resultado toma un dato (una tupla)

     {

        print "<tr style=\"background-color:$dato[color]\">

                <td> $dato[color]</td>

                <td> $dato[descripcion]</td>

                <td> <a href=index.php?id=$dato[id]>&#128221</a></td>

              </tr>";

     }

?>

Si probamos a pinchar el icono, veremos que en la url aparecerá la id del color elegido.

En este caso utilizaremos para editar ese color. Para ello la opción de AGREGAR ELEMENTOS no aparecerá en este caso, pero aparecerá la opción de EDITAR ELEMENTOS.

El código del formulario quedará así:

    <?php if(!$\_GET): //Si no hay metodo get aparecerá lo siguiente?>

    <h2>AGREGAR COLOR</h2>

    <form action="index.php" method="POST">

            <p><label for="color">Color</label><input type="text" name="color"></p>

            <p><label for="descripcion">Descripción</label><input type="text" name="descripcion"></p>

            <p><button>Añadir</button></p>

    </form>

    <?php endif ?>

    <?php if($\_GET): //Si hay método get, queremos editar, aparecerá lo siguiente?>

    <h2>EDITAR COLOR</h2>

        <form action="index.php" method="POST">

                <p><label for="color">Color</label><input type="text" name="color"></p>

                <p><label for="descripcion">Descripción</label><input type="text" name="descripcion"></p>

                <p><button>Editar</button></p>

        </form>

    <?php endif ?>

Pero tendremos que cargar los datos correspondientes a la id que nos ha llegado. Añadiremos un nuevo SELECT en la parte de arriba de index.php.

    /\*\* SELECT cuando recibimos una ID \*/

    if($\_GET){

        $id = $\_GET['id'];

        $sql\_lectura\_id = "SELECT \* FROM colores WHERE id=?"; //Sentencia SQL que queremos ejecutar

        $sentencia\_select\_id = $gbd->prepare($sql\_lectura\_id);

        $sentencia\_select\_id->execute(array($id));

        $resultado\_id = $sentencia\_select\_id->fetch(); //fetch porque viene sólo un resultado

    }

Volveremos ahora al formulario de editar y añadiremos los valores que hemos recibido en $resultado\_id

<?php if($\_GET): //Si hay método get, queremos editar, aparecerá lo siguiente?>

    <h2>EDITAR COLOR</h2>

        <form action="index.php" method="POST">

                <p><label for="color">Color</label><input type="text" name="color" value="<?php print $resultado\_id['color']?>"></p>

                <p><label for="descripcion">Descripción</label><input type="text" name="descripcion" value="<?php print $resultado\_id["descripcion"]?>"></p>

                <p><button>Editar</button></p>

        </form>

    <?php endif ?>

Ahora veremos que nos muestra la información que hemos escogido en este formulario. Pero el botón por ahora nos vuelve a añadir el mismo color. Vamos a cambiar la acción del formulario y hacer que también envíe la ID mediante in input de tipo hidden.

    <?php if($\_GET): //Si hay método get, queremos editar, aparecerá lo siguiente?>

    <h2>EDITAR COLOR</h2>

        <form action="editar.php" method="GET">

                <p><label for="color">Color</label><input type="text" name="color" value="<?php print $resultado\_id['color']?>"></p>

                <p><label for="descripcion">Descripción</label><input type="text" name="descripcion" value="<?php print $resultado\_id["descripcion"]?>"></p>

                <input type="hidden" name="id" value="<?php print $resultado\_id['id']?>">

                <p><button>Editar</button></p>

        </form>

    <?php endif ?>

# Sentencia DELETE

Añadiremos una nueva opción a la tabla con los colores con el icono de una papelera: &#128465. Este icono nos redireccionará a una nueva página php a la que se pasará el valor del ID.

La tabla de los colores quedará así:

<?php

  foreach($resultado as &$dato) //de la matriz del resultado toma un dato (una tupla)

  {

     print "<tr style=\"background-color:$dato[color]\">

              <td> $dato[color]</td>

              <td> $dato[descripcion]</td>

              <td> <a href=index.php?id=$dato[id]>&#128221</a></td>

              <td> <a href=eliminar.php?id=$dato[id]>&#128465</a></td>

            </tr>";

   }

?>

Si probamos clicando el icono veremos crea una url con la IP. Crearemos un nuevo archivo llamado eliminar.php que se encargará de eliminar la tupla elegida.

<?php

//Si todavía no se ha hecho la conexión la hacemos

include\_once 'conexion.php';

$id = $\_GET['id'];

$sql\_eliminar = "DELETE FROM colores WHERE id=?";

$sentencia\_eliminar = $gbd->prepare($sql\_eliminar);

$sentencia\_eliminar->execute(array($id));

header("location:index.php");//para que una vez que se ejecute esto redirija a esa página, para que se cargue