Control semana 4: Base de datos con PHP

Jorge Peñaloza

Programación Avanzada I

Instituto IACC

5 de julio de 2021

**Desarrollo**

**Respuestas del problema**

***Contexto***

1. Según los conocimientos previos adquiridos, identifique la importancia del uso de las bases de datos en las organizaciones. Fundamente.
2. Se requiere que realice una base de datos con PHP para una tienda de venta de artículos de limpieza, que contenga una tabla con los siguientes datos: código del producto, descripción, precio y cantidad en stock.

* Realizar la conexión respectiva con el servidor.
* Crear una base de datos llamada “artículos\_de\_limpieza”.
* Ingresar mínimo 10 productos.
* El precio de los productos debe estar en un rango entre 1000 y 5500 pesos.
* Mostrar todos los datos de la tabla.
* Mostar solo la descripción y código de los productos que cumplan con un precio > 5000.
* Modificar la estructura de la tabla para añadir un nuevo campo “categoría”.
* Ingresar 2 productos nuevos, para un total de 12.
* Seleccione el producto de su preferencia y modifique la cantidad del stock a ”20”.
* Selecciones 5 productos de su preferencia, una vez mostrado en pantalla elimine 2 de ellos.

***Respuesta a pregunta 1***

Desde el nacimiento de la computación hasta nuestros días, el manejo y uso de la información que hoy en día conocemos como *informática* paso por varias evoluciones desde “La pascalina” hasta las maquinas actuales, pero no fue, sino hasta 1963, cuando nació el termino ***base de datos*** en un simposio en California según (Estrada, s.f.).

Desde el nacimiento del termino *base de datos,* la informática evoluciono, hasta el punto de que hoy en día todo es almacenado en discos duros, SSD, NAS, hasta llegar a los de mayor envergadura, como los Data Center y los servicios en la Nube que incluyen almacenamiento de datos. Todos estos dispositivos electrónicos han evolucionado por la necesidad de almacenar información que tienen todas y cada una de las empresas, organizaciones y países existentes.

Las bases de datos actualmente son parte de la vida cotidiana, porque aparte de la información logística, administrativa o financiera. Hoy en día se utiliza la información de historial de navegación, historial de compras, redes sociales que son un tipo de base de datos que aunque cumplen con las restricciones de un modelo de bases de datos relacional, se consideran bases de datos muy relevantes para funcionamiento del mundo actual, ya que se requieren para analizar y predecir como cambiara algún, producto, servicio, tendencia política o social inclusive.

En resumen, en la actualidad todo requiere el uso de bases de datos.

***Diseño de prototipo***

Lo primero es crear la base de datos *artículos\_de\_limpieza* desde phpMyAdmin, ta como se aprecia en la figura 1, para esta acción se debe ingresar el nombre y presionar el botón *crear.*

Luego se inicia la creación de tablas, en donde la primera acción es dar el nombre. En este caso el nombre la tabla será *artículos.* Se puede ver este paso en la figura 2.

Figura 1

*Creación de base de datos en phpMyadmin*

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 2

*Indicar nombre y cantidad de columnas de la tabla en creación*

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Fuente: Elaboración propia

En la figura 3 se puede ver como se dan las propiedades de campo de la tabla. Se dará clic en Previsualizar SQL para poder obtener y almacenar la estructura de la tabla creada tal como se ve en la figura 4..

Figura 3

*Seleccionar propiedades de cada campo de la tabla*

Una captura de pantalla de una computadora

Descripción generada automáticamente con confianza media

Fuente: Elaboración propia

Figura 4

*Código SQL creado por phpMyAdmin para crear la tabla*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

En la figura 5 se puede apreciar la estructura de la tabla creada.

Figura 5

*Tabla artículos de base de datos artículos\_de\_limpieza*

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto, Correo electrónico

Descripción generada automáticamente

Fuente: Elaboración propia

Para insertar los 10 productos y comprobar la conexión en simultaneo, se utilizó el *código de programación 1* que permite insertar los 10 elementos desde un array *$valores.*

En la figura 1 se puede ver como se valida que fue efectivo el ingreso de los datos, y se pueden apreciar por phpMyAdmin en la figura 2.

Código de programación 1

*Archivo: insertar.php*

<?php

spl\_autoload\_register( function ($nombre\_clase)

{

    include $nombre\_clase . '.php';

});

$campos = array('codigo', 'descripcion', 'precio', 'stock');

$valores = array(

                array(1, 'Calcetines rojos', 1000, 6),

                array(2, 'Calcetines verdes de invierno', 5000, 7),

                array(3, 'Calcetines negros de polar', 5500, 20),

                array(4, 'Pack calzonsillos negros', 4500, 3),

                array(5, 'Cacelines blancos cortos', 1200, 1),

                array(6, 'Cacelines blancos largos', 1800, 12),

                array(7, 'Cacelines negros cortos', 1500, 9),

                array(8, 'Cacelines amarillos cortos', 1120, 9),

                array(9, 'Cacelines amarillos largos', 4000, 3),

                array(10, 'Cacelines azules cortos', 3900, 6)

                );

$conexion = new dB();

foreach ($valores as $key => $value)

{

    if($conexion->insertar('articulos',$campos,$value))

        echo "Insertado";

    else

        echo "No insertado";

    echo "<br>";

}

?>

Fuente: elaboración propia

Figura 1

*Comprobación por navegador el resultado de la inserción de datos*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 2

*Tabla vista en phpMyAdmin*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

Para mostrar todos los productos y comprobar la conexión en simultaneo, se utilizó el *código de programación 2* que permite mostrar en formato de *tabla html* los valores de la consulta.

En la figura 3 se puede ver la visualización de todos los productos de la consulta.

Código de programación 2

*Archivo: mostrarTodo.php*

<?php

spl\_autoload\_register( function ($nombre\_clase)

{

    include $nombre\_clase . '.php';

});

$campos = array('codigo', 'descripcion', 'precio', 'stock');

$conexion = new dB();

if($resultado=$conexion->buscarTodo("articulos"))

{

    echo "<table>";

    echo "<tr>";

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

    {

        echo "<th>".$value."</th>";

    }

    echo "</tr>";

    echo "<tr>";

    foreach ($resultado as $fila)

    {

        $cadena = "<tr>";

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

        {

            $cadena .= "<td>".$fila[$value]."</td>";

        }

        echo $cadena;

        echo "</tr>";

    }

    echo "</tr>";

    echo "</table>";

}

else

    echo  "No hay registros";

?>

Fuente: Elaboración propia

Figura 3

*Vista de mostrarTodo.php*

Texto

Descripción generada automáticamente con confianza media

Fuente: Elaboración propia

Para mostrar los productos con código y descripción cuyo precio sea mayor a 5000 y comprobar la conexión en simultaneo, se utilizó el *código de programación 3* que permite mostrar en formato de *tabla html* los valores de la consulta.

En la figura 3 se puede ver la visualización de todos los productos de la consulta.

Código de programación 3

*Archivo: mostrarMayorque5000.php*

<?php

spl\_autoload\_register( function ($nombre\_clase)

{

    include $nombre\_clase . '.php';

});

$campos = array('codigo', 'descripcion');

$conexion = new dB();

if($resultado=$conexion->buscar("articulos",$campos,"precio > 5000"))

{

    echo "<table>";

    echo "<tr>";

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

    {

        echo "<th>".$value."</th>";

    }

    echo "</tr>";

    echo "<tr>";

    foreach ($resultado as $fila)

    {

        $cadena = "<tr>";

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

        {

            $cadena .= "<td>".$fila[$value]."</td>";

        }

        echo $cadena;

        echo "</tr>";

    }

    echo "</tr>";

    echo "</table>";

}

else

    echo  "No hay registros";

?>

Fuente: Elaboración propia

Figura 4

*Vista de mostrarMayorque5000.php*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

Fuente: Elaboración propia

Para agregar la columna categoría se debe ir a *estructura* en *phpMyAdmin* y luego dar clic en *continuar* para agregar 1 columna después de stock, como se parecía en la figura 5.

Figura 5

*Agregando columna a la estructura*

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Fuente: Elaboración propia

Luego se definen los atributos del campo *categoría* tal como se ve en la figura 6. Luego se da clic en visualizar SQL para ver la consulta de modificación de estructura tal como se aprecia en la figura 7.

Finalmente se puede ver la modificación de la estructura en la figura 8.

Figura 6

*Configurando parámetros de la columna*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 7

*SQL de modificación de estructura*

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 8

*Estructura vista en phpMyAdmin*

Captura de pantalla de un celular

Descripción generada automáticamente

Fuente: Elaboración propia

Para insertar los 2 elementos solicitados se debe ver *código de programación 4*, y para ver como resulto la ejecución, se debe observar las figuras 9 y 10. En la figura 10 se observa el nuevo campo insertado con la categoría reflejada en la consulta.

Código de programación 4

*Archivo: insertar2.php*

<?php

spl\_autoload\_register( function ($nombre\_clase)

{

    include $nombre\_clase . '.php';

});

$campos = array('codigo', 'descripcion', 'precio', 'stock','categoria');

$valores = array(

                array(11, 'Calcetines morados largos', 1000, 26,'Ropa de invierno'),

                array(12, 'Calcetines morados cortos', 5000, 27,'Ropa de verano'),

                );

$conexion = new dB();

foreach ($valores as $key => $value)

{

    if($conexion->insertar('articulos',$campos,$value))

        echo "Insertado";

    else

        echo "No insertado";

    echo "<br>";

}

?>

Fuente: Elaboración propia

Figura 9

*Vista en navegador de insertar2.php*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 10

*Vista de datos luego de insertar los 2 nuevos elementos en phpMyAdmin*

Tabla

Descripción generada automáticamente

Fuente: Elaboración propia

Para la modificación de stock se utilizará el código 2 que actualmente tiene un stock de 7, para esta modificación se utilizara *código de programación 5.*

Código de programación 5

*Archivo: modificar.php*

<?php

spl\_autoload\_register( function ($nombre\_clase)

{

    include $nombre\_clase . '.php';

});

$campos = array(  'stock');

$valores = array(  20 );

$conexion = new dB();

if($conexion->actualizar('articulos',$campos,$valores,"codigo = 5"))

    echo "modificado";

else

    echo "No modificado";

echo "<br>";

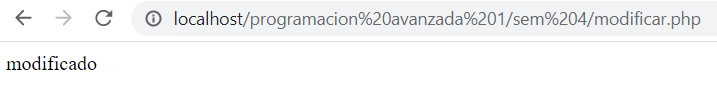
?>

Fuente: Elaboración propia

Una vez ejecuta el script de *código de programación 5* se puede apreciar en la figura 11 la vista de ese script y en figura 12 la vista de datos de phpMyAdmin

Figura 11

*Vista en navegador de modificar.php*



Fuente: Elaboración propia

Figura 12

*Vista de datos en phpMyAdmin*

Tabla

Descripción generada automáticamente

Fuente: Elaboración propia

Para la solicitud final que pide mostrar 5 productos a elección se seleccionan los productos con códigos 1,2,3,4,5 y de ellos de eliminan los productos con códigos 1 y 2. Para ellos se utilizó el código de programación 6.

En la figura 13 se puede observar la ejecución del script y en la figura 14 se puede observar la tabla en phpMyAdmin.

Código de programación 6

*Archivo: mostrarYeliminar.php*

<?php

spl\_autoload\_register( function ($nombre\_clase)

{

    include $nombre\_clase . '.php';

});

$campos = array('codigo', 'descripcion','precio','stock','categoria');

$conexion = new dB();

$condicion = "codigo = 1 or codigo = 2 or codigo = 3 or codigo = 4 or codigo = 5";

if($resultado=$conexion->buscar("articulos",$campos,$condicion))

{

    echo "<table>";

    echo "<tr>";

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

    {

        echo "<th>".$value."</th>";

    }

    echo "</tr>";

    echo "<tr>";

    foreach ($resultado as $fila)

    {

        $cadena = "<tr>";

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

        {

            $cadena .= "<td>".$fila[$value]."</td>";

        }

        echo $cadena;

        echo "</tr>";

    }

    echo "</tr>";

    echo "</table>";

}

else

    echo  "No hay registros";

$condicion = "codigo = 1 or codigo = 2";

if($conexion->borrar("articulos",$condicion))

    echo "Borrado";

else

   echo "No borrado";

?>

Fuente: Elaboración propia

Figura 13

*Vista en navegador de mostrarYeliminar.php*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 14

*Vista de datos en phpMyAdmin*

Tabla

Descripción generada automáticamente

Fuente: Elaboración propia

La clase que tienes los métodos de inserción, modificación, eliminación y consulta de la base de datos utilizado se puede ver implementados en *código de programación 7.*

Código de programación 7

*Archivo: dB.php*

<?php

class dB

{

    private $host   = "localhost";

    private $usuario = "root";

    private $pass  = "";

    private $db     = "artículos\_de\_limpieza";

    public $conexion;

    public function \_\_construct()

    {

        $this->conexion = new mysqli($this->host, $this->usuario, $this->pass,$this->db) or die(mysql\_error());

        $this->conexion->set\_charset("utf8");

    }

    public function insertar($tabla, $campos, $datos)

    {

        # Ejemplo

        # INSERT INTO `articulos` (`codigo`, `descripcion`, `precio`, `stock`) VALUES ('1', '1', '1', '1')

        $query = "INSERT INTO ";

        $query .= $tabla;

        $query .= " (`";

        $query .= implode("`,`", $campos);

        $query .= "`) VALUES ('";

        $query .= implode("','", $datos);

        $query .= "');";

        $resultado =    $this->conexion->query($query) or die($this->conexion->error);

        if($resultado)

            return true;

        return false;

    }

    public function buscar($tabla, $campos = null,$condicion = null)

    {

        $query = "SELECT ";

        if($campos == null)

        {

            $query .= " \* ";

        }

        else

        {

            $query .= " `";

            $query .= implode("`,`", $campos);

            $query .= "`";

        }

        $query .= " FROM ";

        $query .= $tabla;

        $query .= " WHERE ";

        if($condicion == null)

        {

            $query .= " 1";

        }

        else

        {

            $query .= $condicion;

        }

        #echo $query;

        $resultado = $this->conexion->query($query) or die($this->conexion->error);

        if($resultado)

            return $resultado->fetch\_all(MYSQLI\_ASSOC);

        return false;

    }

    public function buscarTodo($tabla)

    {

        $query = "SELECT \* FROM ".$tabla." WHERE 1";

        $resultado = $this->conexion->query($query) or die($this->conexion->error);

        if($resultado)

            return $resultado->fetch\_all(MYSQLI\_ASSOC);

        return false;

    }

    public function actualizar($tabla, $campos, $valores, $condicion)

    {

        #"UPDATE $tabla SET $campos WHERE $condicion"

        $query = "UPDATE ";

        $query .= $tabla;

        $query .= " SET ";

        if(count($campos) == count($valores))

        {

            $sw = false;

            foreach ( array\_combine( $campos, $valores) as $nombre => $valor)

            {

                if ($sw)

                    $query .= ",";

                $query .= "`".$nombre."` = '".$valor."' " ;

                $sw = true;

            }

        }

        $query .= " WHERE ";

        $query .= $condicion;

        #echo $query;

        $resultado  =   $this->conexion->query($query) or die($this->conexion->error);

        if($resultado)

            return true;

        return false;

    }

    public function borrar($tabla, $condicion)

    {

        $query = "DELETE FROM ".$tabla." WHERE ".$condicion;

        #echo $query;

        $resultado  =   $this->conexion->query($query) or die($this->conexion->error);

        if($resultado)

            return true;

        return false;

    }

}

?>

Fuente: Elaboración propia

El código fuente del proyecto se encuentra en el archivo Jorge\_Peñaloza\_Control4.rar adjunto.

Bibliografía

Develop your code. (s.f.). *Conexión y métodos de consulta a una BD MYSQL*. Obtenido de https://codea.app/php/conexion-y-metodos-de-consulta-a-una-bd-mysql

Estrada, L. (s.f.). *Base de datos. Apuntes.* Obtenido de http://docencia.fca.unam.mx/~lestrada/

IACC. (2020). Base de datos con PHP. En *Programación avanzada 1. Semana 4.*

IACC. (2021). Tarea semana 4.

The PHP Group. (s.f.). *Manual de PHP*. Obtenido de https://www.php.net/manual/es/index.php

*w3schools.com*. (s.f.). Obtenido de https://www.w3schools.com/