

Universidad Tecnológica de Tijuana



Tecnologías de la Información: Desarrollo de Software Multiplataforma

Aplicaciones Web

Portafolio de actividades

Juan Andrés Cuevas Zapata

3D

Ray Brunett Parra Galaviz

Tijuana, Baja California a 31 de octubre de 2024

Índice

Introducción a Linux	3
Homebrew en Linux	3
PHP	5
Conceptos básicos	5
Comandos básicos	6
Imprimir	6
Declarar variables	6
Imprimir variables	6
Condicional IF	6
Operador Ternario	6
Ciclo DO-WHILE	6
Ciclo WHILE	6
Ciclo FOR	6
Ciclo FOR-EACH	7
Funciones	7
Funciones con valores predeterminados	7
Funciones con argumentos variables	7
PHP aplicado a un proyecto HTML	8
MySQL	8
Conexión MySQL en PHP	10
Realizar INSERT en la base de datos desde PHP	11
Conclusión	14

Introducción a Linux

En el primer parcial trabajamos en la parte visual de una página web (Front End) usando HTML y CSS. A partir de ahora trabajaremos en la parte lógica (Back end) con el lenguaje de programación PHP. Para poder trabajar con un archivo PHP, debe ser cargado en un servidor; para esto, instalaremos PHP y MySQL (DBMS) con ayuda de la aplicación Homebrew.



Homebrew en Linux

Homebrew es un gestor de aplicaciones que nos ayuda a instalar y configurar de manera eficaz aplicaciones que no vienen por defecto en nuestro sistema Linux, la página de Homebrew es la siguiente: <https://docs.brew.sh/Homebrew-on-Linux>.



Para instalar Brew se deben correr los siguientes comandos:

```
sudo apt-get install build-essential procps curl file git
```

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

La instalación puede demorar unos minutos, así que debemos tener paciencia.

Una vez terminada la instalación, ahora tendremos Brew instalado en nuestro equipo y podemos proceder a instalar PHP y MySQL con los siguientes comandos:

```
brew install php
```

```
brew install mysql
```

Para verificar que se haya instalado correctamente en tu equipo se puede usar el comando:

php --version

mysql --version

```
sks-kronos@sk-s-kronos-HP-ZBook-15u-G3:~$ php --version
PHP 8.3.12 (cli) (built: Sep 24 2024 18:08:04) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.12, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.12, Copyright (c), by Zend Technologies
sks-kronos@sk-s-kronos-HP-ZBook-15u-G3:~$ mysql --version
mysql Ver 9.0.1 for Linux on x86_64 (Homebrew)
sks-kronos@sk-s-kronos-HP-ZBook-15u-G3:~$
```

PHP

Como mencioné previamente, este parcial estará enfocado al desarrollo Back End con el lenguaje de programación PHP.

Es considerado uno de los lenguajes más comunes, sencillos y fáciles de utilizar para el desarrollo web. Además, cuenta con una muy buena implementación de sentencias para realizar una conexión a base de datos MySQL.



Conceptos básicos

Un archivo PHP debe ser creado con terminación `.php` para que el navegador interprete que se trata de un archivo PHP.

Para escribir comandos PHP en el archivo, debe ir dentro de las etiquetas:

```
<?php
    //aquí va el código...
?>
```

Un archivo `.php` no va a ejecutar su código adecuadamente si no está almacenado en un servidor (puede ser local), por lo que ahora debemos prender el servidor localhost con ayuda de las siguientes instrucciones:

- Abrir la terminal.
- Posicionarse en la ruta de tu archivo `.php`
- Ejecutar el siguiente comando:
`php -S localhost:9000`
El número `9000` indica el puerto que se va a utilizar para lanzar el servidor y se puede cambiar por otro número si así lo desea.
- Abrir el navegador y escribir la ruta `localhost:9000/nombreArchivo.php`.
Si no se escribe el nombre del archivo, el navegador busca por defecto si existe un archivo llamado `index.php` y lo ejecuta por defecto.

Ahora todo está listo para empezar a escribir código y verlo reflejado en nuestra página web.

Comandos básicos

En PHP se pueden hacer las operaciones más comunes de los lenguajes de programación como imprimir en pantalla, declarar y asignar valores a variables, usar condicionales, ciclos, crear funciones...

A continuación, una demostración de cómo usar cada una de estas sentencias:

Imprimir

```
print "Hola Mundo PHP";  
print "<br>";
```

Declarar variables

```
$x = 90;  
$f = 78.23;  
$s = "foo bar";  
$b = true;  
$arr = [1,2,3,4];
```

Imprimir variables

```
print $x.$f.$s.$b.$arr[1];
```

Condicional IF

```
if ($x>$f) {  
    print "x es mayor que f";  
} else {  
    print "f es mayor que x";  
}
```

Operador Ternario

```
print ($x>$f) ? "X es mayor que F" : "F es mayor que X";
```

Ciclo DO-WHILE

```
$a = 0;  
do {  
    print $a;  
    $a++;  
} while ($a <= 10);
```

Ciclo WHILE

```
$i=0;  
while ($i <= 10) {  
    print $i;  
    $i++;  
}
```

Ciclo FOR

```
for ($i=0; $i < 10; $i++) {
```

```
    print $i;
}
```

Ciclo FOR-EACH

```
foreach ($arr as $value) {
    print $value;
}
```

Funciones

```
function foo($name){
    return $name;
}
print foo("Bar");
```

Funciones con valores predeterminados

```
function bar($name,$d="default"){
    print "Hola ".$name." d: ".$d;
    print "<br>";
}
bar("Foo");
bar($d="Foo Bar", $name="Joe Doe");
```

Funciones con argumentos variables

```
function foobar($np,$np2, ...$p){
    //print $p;
    print "Normal Param: ".$np;
    print "<br>";
    print "Normal Param 2: ".$np2;
    print "<br>";
    foreach ($p as $arg) {
        print "Array: ".$arg." ";
    }
    print "<br>";
}
foobar(1,2,3,4,5,6,7);
foobar("a","b","c");
```

PHP aplicado a un proyecto HTML

En este punto, utilizaremos PHP para crear una página web con información dinámica recolectada de una base de datos.

La página que vamos a crear es de bienes raíces en la que habrán varias pantallas. La página principal (index.php) tendrá información de la empresa y de las propiedades. También, tendrá un apartado para ingresar a los vendedores y las propiedades. Además, tendrá una pantalla donde se muestran las propiedades que están en venta y la posibilidad de cambiarlas al estado "vendido". Por último, una pantalla donde se muestran todas las propiedades vendidas con más información.

Todos los datos debe ser actualizados dinámicamente y se deben guardar en alguna parte, para esto utilizaremos una conexión con base de datos en MySQL.



MySQL

La base de datos va a estar almacenada en el localhost, por lo que debemos cargar la base de datos en nuestro servidor local. Gracias a que instalamos previamente MySQL con ayuda de Homebrew, podremos avanzar en este punto.

A continuación, vamos a crear la base de datos para bienes raíces y sus respectivas tablas:

- Abrimos una terminal e ingresamos el siguiente comando:

```
mysql -u root
```

Esto debe cambiar el prompt de la terminal de la siguiente manera:

```
sks-kronos@sks-kronos-HP-ZBook-15u-G3:~$ mysql -u root debe cambiar el prompt de la
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 9.0.1 Homebrew

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

- Ahora vamos a crear la base de datos con el siguiente comando:
`CREATE DATABASE bienesraices2;`
- Una vez creada, ahora tenemos que seleccionarla para ejecutar comandos en ella:
`USE DATABASE bienesraices2;`
- Primero vamos a crear la tabla de vendedores con el siguiente comando:

CREATE TABLE sellers(id INT PRIMARY KEY NOT NULL, name VARCHAR(32) NOT NULL, email VARCHAR(32) NOT NULL, phone VARCHAR(10));

Y debe tener la siguiente estructura:

```
mysql> describe sellers;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    | auto_increment |
| name  | varchar(32)   | NO   |     | NULL    |                |
| email | varchar(32)   | NO   |     | NULL    |                |
| phone | varchar(10)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0,00 sec)

mysql>
```

- Después, vamos a crear la tabla de estados de una propiedad con el siguiente comando:
CREATE TABLE prop_status(code VARCHAR(8) PRIMARY KEY NOT NULL, description VARCHAR(16) NOT NULL);

Y debe tener la siguiente estructura:

```
mysql> describe prop_status;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| code           | varchar(8)    | NO   | PRI | NULL    |                |
| description    | varchar(16)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0,01 sec)

mysql>
```

- Por último, vamos a crear la tabla de propiedades con el siguiente comando:
CREATE TABLE propiedades(id INT PRIMARY KEY NOT NULL, title VARCHAR(32) NOT NULL, price decimal(10,2) NOT NULL, image VARCHAR(256), description LONGTEXT, rooms INT, wc INT, garage INT, timestamp date, id_seller INT NOT NULL, status VARCHAR(8) NOT NULL, FOREIGN KEY(id_seller) REFERENCES sellers(id), FOREIGN KEY(status) REFERENCES prop_status(code));

Y debe tener la siguiente estructura:

```
mysql> describe propiedades;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id         | int           | NO   | PRI | NULL    | auto_increment |
| title      | varchar(32)   | NO   |     | NULL    |               |
| price      | decimal(10,2) | NO   |     | NULL    |               |
| image      | varchar(256)  | YES  |     | NULL    |               |
| description | longtext      | YES  |     | NULL    |               |
| rooms      | int           | YES  |     | NULL    |               |
| wc         | int           | YES  |     | NULL    |               |
| garage     | int           | YES  |     | NULL    |               |
| timestamp  | date          | YES  |     | NULL    |               |
| id_seller  | int           | NO   | MUL | NULL    |               |
| status     | varchar(8)    | YES  | MUL | NULL    |               |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0,00 sec)

mysql>
```

Ahora ya tenemos lo necesario para guardar la información que mandemos en el formulario para crear propiedades y vendedores.

Conexión MySQL en PHP

Para realizar una conexión MySQL en nuestra página web, vamos a crear un archivo específico que haga la conexión. En este caso le pondré de nombre “*connectdb.php*”. Este archivo tendrá el siguiente código:

```
<?php
function connectdb() : mysqli{
    $db = mysqli_connect("localhost", "root", "", "bienesraices2");

    if(!$db){
        echo "Connection failed";
    }
    return $db;
}
?>
```

Cuando mandemos a llamar a este archivo, vamos a poder hacer referencia a la función “*connectdb()*” y nos debe realizar la conexión con la base de datos. Si llega a haber algún error, entonces va a imprimir en pantalla el mensaje “*Connection failed*”.

Para poder usar la función, se debe hacer de la siguiente manera desde cualquier archivo *.php*:

```
require "includes/config/connectdb.php";
$db = connectdb();
```

La palabra “*require*” busca el archivo especificado en la ruta y lo toma en cuenta como una extensión para el código del archivo actual, pero no lo ejecuta. Esto quiere decir que en el archivo de *require* pueden haber tantas líneas de código como queramos, pero sólo se van a ejecutar

hasta que nosotros le indiquemos en el archivo actual. Es por esto que le podemos asignar a la variable `$db` el valor que retorna la función `connectdb()`.

Una vez teniendo la conexión hecha, se pueden ejecutar sentencias `mysql` para interactuar con la base de datos.

Por ejemplo, si queremos realizar una consulta, entonces debemos usar la sentencia `mysql_query($conexión, $consulta)`. Si queremos leer los datos de una consulta de selección, entonces utilizaremos la sentencia `mysql_fetch_assoc($respuesta)`.

Realizar INSERT en la base de datos desde PHP

Vamos a tomar de ejemplo la pantalla de crear vendedores para la inserción en base de datos.



The screenshot shows a web page with a teal header and footer. The header contains the text 'BIENES RAICES' and navigation links: 'Create Propierty', 'Create Seller', 'Sell Propierty', 'Sales', 'Home', 'Us', 'Contact', and 'Blog'. The main content area has the title 'Venta de Casas y Departamentos Exclusivos de Lujo' and a section titled 'Sellers Form'. Inside this section, there is a legend that says 'Fill all Fields to Create a New Seller'. Below the legend are three input fields: 'Name' with placeholder 'Your Name', 'E-Mail' with placeholder 'your@email.com', and 'Phone' with placeholder '6645555555'. A 'Create a New Seller' button is located below the phone field. The footer contains the same navigation links and the text 'Todos los Derechos Reservados by: Yo'.

Este es el formulario para registrar un nuevo vendedor. En él, se pide el nombre, e-mail y el teléfono del vendedor. Este formulario está declarado de la siguiente manera en HTML:

```
<form action="crearSellers.php" method="POST">
  <fieldset>
    <legend>Fill all Fields to Create a New Seller</legend>
    <div>
      <label for="name">Name</label>
      <input type="text" id="name" name="name" placeholder="Your Name" minlength="3"
maxlength="32" required>
    </div>
    <div>
      <label for="email">E-Mail</label>
      <input type="email" id="email" name="email" placeholder="your@email.com"
minlength="8" maxlength="32" required>
    </div>
    <div>
```

```

        <label for="phone">Phone</label>
        <input type="text" id="phone" name="phone" placeholder="6645555555"
minlength="10" maxlength="10">
    </div>
</div>
    <button type="submit">Create a New Seller</button>
</div>
</fieldset>
</form>

```

Es importante resaltar las partes que están en **negritas**.

El código dice que el formulario va a enviar los valores que tengan los input por el método **POST**.

La página **destino** es ella misma.

Son 3 los datos que serán enviados: **name**, **email** y **phone**.

Este envío va a ser realizado después de que se presione el botón de tipo **submit**. Gracias a esto, ahora tenemos una forma de recolectar la información que se quiere insertar en la base de datos, así que la guardaremos en variables.

Vamos a colocar el siguiente bloque de código *PHP* hasta arriba de nuestro archivo *.php* para que sea lo primero que se ejecute al entrar en la página.

```

<?php
    require "includes/config/connectdb.php";
    $db = connectdb();

    if ($_POST) {
        $name = $_POST["name"];
        $email = $_POST["email"];
        $phone = $_POST["phone"];

        $query = "INSERT INTO sellers(name,email,phone) VALUES('$name','$email','$phone');";

        $response = mysqli_query($db, $query);

        if ($response) {
            echo "Seller created";
        }
        else{
            echo "Seller no created";
        }
    }
?>

```

En **\$db** guardaremos la conexión con la base de datos gracias al **require**.

Como este bloque es lo primero que se ejecuta al abrir la página, queremos que lo siguiente sólo ocurra en caso de que el usuario ya haya mandado datos, por lo que validamos la variable superglobal **\$_POST**. Si el usuario mandó datos del formulario con el botón **submit**, entonces **\$_POST** será verdadero y podremos continuar con el inserto.

Se guardan los datos de **\$_POST** en variables individuales descriptivas.

Se arma la consulta (**\$query**) que se quiere ejecutar en la base de datos concatenando las variables previamente creadas.

Se le asigna a la variable de respuesta (**\$response**) el método **mysqli_query(\$db, \$query)**. Este método requiere de primer parámetro la conexión y después la consulta que se quiere realizar.

Después se evalúa **\$response**. Si es verdadero, significa que la consulta se realizó con éxito y se puede notificar al usuario que se creó el vendedor. De caso contrario, no se habrá podido insertar el nuevo registro.

Antes de darle al botón "Create a New Seller":



Después de darle al botón "Create a New Seller":



Registro en la base de datos:

```
mysql> SELECT * FROM sellers;
+----+-----+-----+-----+
| id  | name  | email      | phone |
+----+-----+-----+-----+
| 1001 | Foo Bar | foo@bar.com | 1234567890 |
| 1002 | Bar Foo | bar@foo.com | 1234567890 |
| 1005 | Andres Cuevas | andres@cuevas.com | 6641234567 |
| 1006 | Gustavo Tello | zorthon@gmail.com | 2223334444 |
| 1008 | Kronos | kronos@gmail.com | 6644766977 |
| 1009 | Luis Arzola | luis@gmail.com | 6648754235 |
+----+-----+-----+-----+
6 rows in set (0,00 sec)
```

Conclusión

Este parcial fue dedicado a desarrollar una página web por el lado de Back End con el lenguaje PHP. Utilizar Linux facilitó mucho las cosas porque tienes total control de las herramientas que estás utilizando. Habilitar y deshabilitar el servidor con el comando `php -S localhost:` hacía que esta tarea fuera rápida y sencilla. De igual forma, poder administrar la base de datos desde la terminal hacía que las consultas de información fueran más rápidas que haber utilizado XAMPP o WAMP.

PHP es un lenguaje recomendable para aquellos que van iniciando en la programación de páginas web porque no tiene muchas complicaciones y existe mucha documentación en internet.