

# **Lucas Vázquez Entorno Servidor**

WWW



## **Teoria (Resumen)**

Conjunto de recursos interconectados que conforman el conocimiento humano actual

(Hubs, repetidores, puentes)

(TCP,IP,HTTPS,FTP,SMTP)

(DNS)

Cliente servidor, considerada la arquitectura mas habitual:

Cliente: es un componente consumidor de servicios

Servidor: es un proceso proveedor de servicios

CAPAS:

Capa de Presentación: La que visualiza el usuario

Capa de Negocio: Gestiona las funcionalidades que esperamos del sistema o aplicacion web

Capa de Datos: Donde se guardan los datos y encargada de acceder dentro

Classificacion de modelos de programacion web:

Segun la medida de los componentes:

FAT Client: El mayor peso se ejecuta en el cliente

FAT Server: Funcionalidad asociada al cliente limitada a la presentacion de la informacion enviada para el servidor

Segun la naturaleza del servicio ofrecido: Servidores de ficheros, servidores de BDD, servidores web

Clasificacion de las aplicaciones web:

Aplicaciones web estáticas: Usuario recibe pagina y la interaccion de la pagina no comporta ninguna acción.

Aplicaciones web dinámicas: interaccion del cliente con el recurso recibido por parte del servidor (gmail)

Aplicaciones web interactivas: Dialogo entre cliente servidor

Lenguajes de Programación:

Es el codigo del bien sea como un objeto procompilado o bien como un codigo interpretado, es ejecutado por un programa especifico en el componente que actua como servidor.

Tipos:

- Lenguajes de Scripting: La interacción del usuario hace que se genere un dialogo entre el cliente y el servidor, EJ:

**PHP, ASP, Perl, Python, JSP**

- Aplicaciones CGI: Delegan la creación de paginas dinamicas a un programa externo (RFC: Unifican criterios en muchas de los ambitos de telecomunicaciones)// Estandar CGI// El programa externo no depende del codigo a generar//

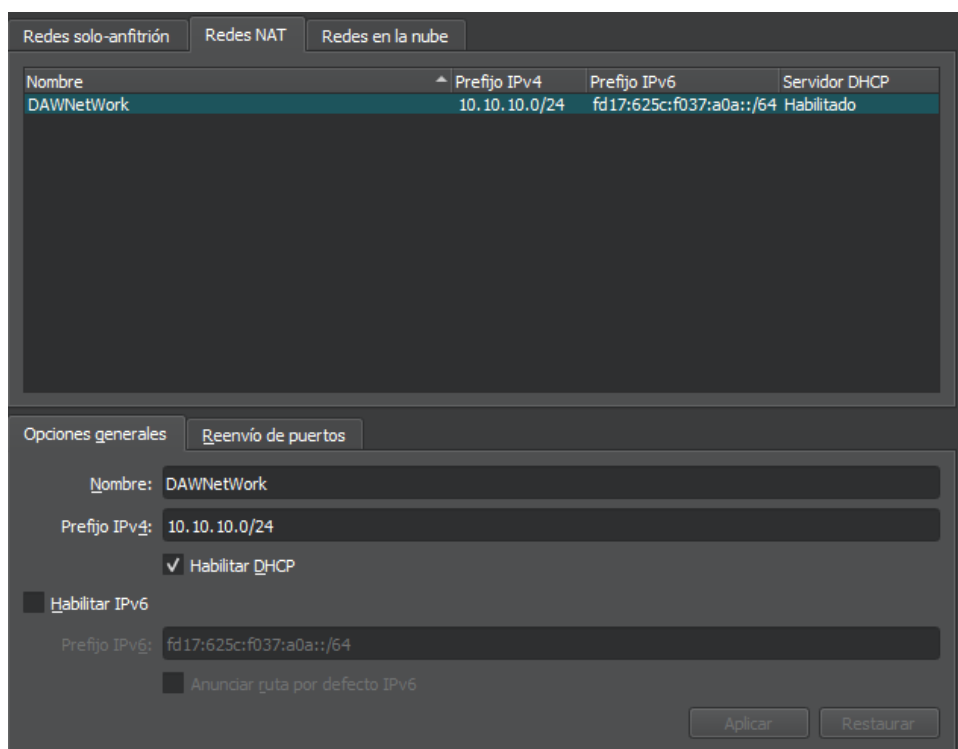
Desventajas: Escaso rendimiento a la hora de responder a multiples peticiones CGI simultaneamente

- Aplicaciones Híbridas de código repartido: Tecnología intermedia alternativa a los lenguajes de scripting y a las aplicaciones CGI y derivados

El proceso de desenvolvamiento no tiene porque realizarse en el mismo equipo en el qual finalmente se despliegue y ejecute la aplicacion web que se esta desenvolvando// Instrumentos: Navegadores, Editores de Documentos, Entornos de programación, BDD

## **Instalacion y Configuracion del Servidor**

Usamos la IP 10.10.10.0/24 para que el servidor y el cliente tengan visibilidad entre ellas, la ip del servidor es 10.10.10.4/24.



Una vez teniendo todo el servidor preparado, procederíamos a instalar PHP.

Usando este comando **sudo apt install php libapache2-mod-php**, aparte de instalar el lenguaje PHP, tambien instala el modulo que permite a apache, que en este caso seria nuestro servidor web, poder procesar lenguaje PHP.

Después instalariamos paquetes opcionales, pero que utilizaremos mas adelante, asi como:

```
sudo apt install php-cli
sudo apt install php-cgi
sudo apt install php-mysql
sudo apt install php-pgsql
```

Ahora tocaria configurar PHP

Primero tendriamos que crear un fichero php en esta direccion /var/www/html:

```
alumedaw@ubuntuserverdawcesur:/var/www/html$ ls
holamundo.php  index.html  phpinfo.php
alumedaw@ubuntuserverdawcesur:/var/www/html$
```

Aqui cree el fichero phpinfo.php, que dentro contiene

```
GNU nano 7.2
<?php
phpinfo();
?>
```

Tambien creamos un fichero php donde pudimos ver que podemos utilizar lenguaje html, usando prints en cada linea de codigo.

```
GNU nano 7.2
<?php
print("<!DOCTYPE html>");
print("<html lang=en>");
print("<script>>window.alert('Hola Mundo')</script>");
    print("<h1>Hola Mundo</h1>");
print("</html>");
?>
```

Puesto en practica ya en el entorno cliente, usando la ip del servidor, en este caso 10.10.10.4 y escribiendo el nombre del fichero que queramos (10.10.10.4/phpinfo.php), comprobariamos que esta todo funcionando


Oct 1 09:09

¡Gracias por descargar We x PHP 8.3.6 - phpinfo()

10.10.10.4/phpinfo.php


80%

PHP Version 8.3.6

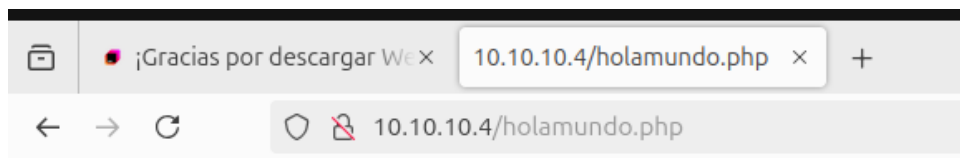


System	Linux ubuntu:server:24.04-generic 6.8.0-45-generic #45-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 30 12:02:04 UTC 2024 x86_64
Build Date	Jun 13 2024 15:23:20
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqld.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-mysqli.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.3/apache2/conf.d/20-pdo_pgsql.ini, /etc/php/8.3/apache2/conf.d/20-pgsql.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831.NTS
PHP Extension Build	API20230831.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v4.3.6, Copyright (c) Zend Technologies with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies



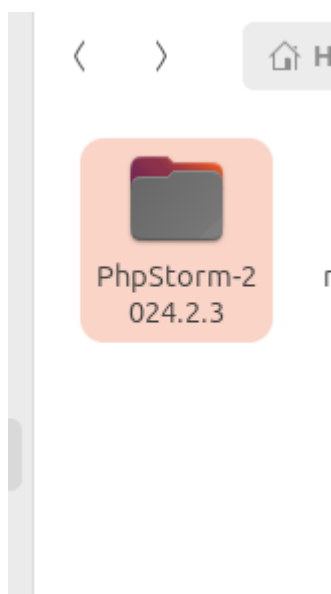
El otro fichero.



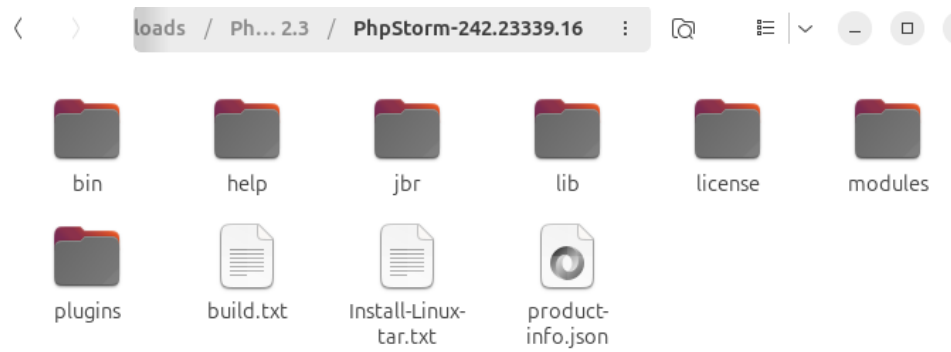
**Hola Mundo**

Instalacion de PhpStorm en el cliente.

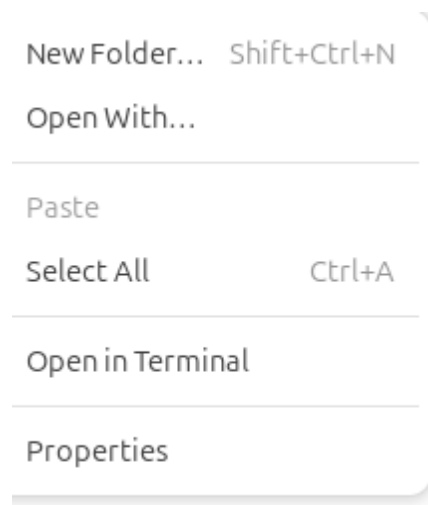
Primero simplemente nos bajamos el instalador desde el navegador



Entramos en bin



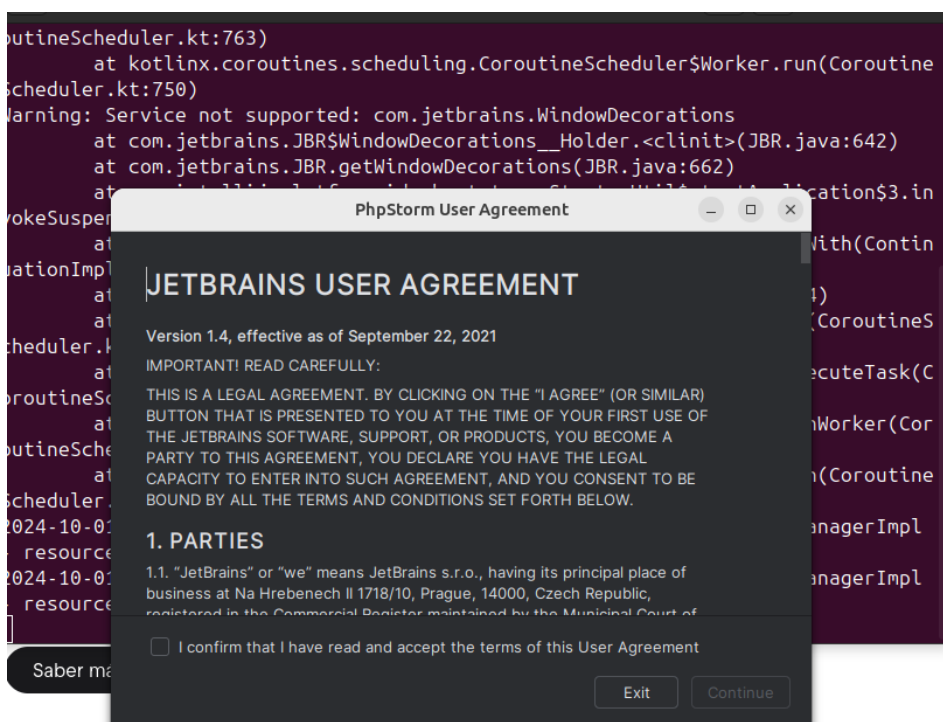
Hacemos click derecho y abrimos la terminal, para que estemos dentro ya del fichero de PhpStorm



Escribimos `./phpstorm.sh`, para que empiece la instalacion

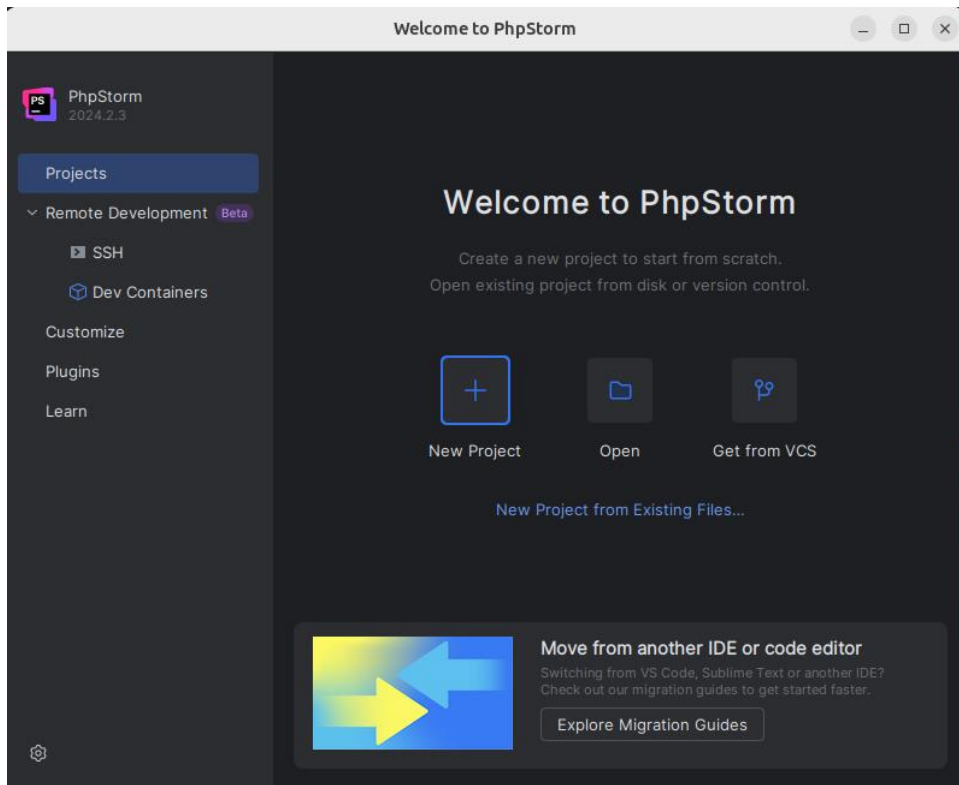
```
alumnedaw@alumnedaw-VirtualBox: ~/Downloads/PhpStorm-2024.2.3/PhpStorm-242.23339
6/bin$ ./phpstorm.sh
```

Aquí ya esta preparado para instalarse



Lo instalamos sin problema y activamos nuestra licencia





Instalar GIT

Usamos **sudo apt install GIT**

```
alumedaw@alumedaw-VirtualBox:~/Downloads/PhpStorm-2024.2.3/PhpStorm-242.2333  
16/bin$ git -v  
git version 2.43.0
```

## Act Clase Analizar Códigos:

Analitzau aquests codis. A continuació comproveu el seu funcionament I expliqueu (a la vostra CA) el funcionament de tot plegat amb només de 5 línies (doc 2,5 cms marge esq I dreta, lletra arial 12).

```

1  <form action='login.php' method='post'>
2  <fieldset>
3      <legend>Login</legend>
4      <div><span class='error'><?php echo $error; ?></span></div>
5      <div class='fila'>
6          <label for='usuario'>Usuario:</label><br />
7          <input type='text' name='inputUsuario' id='usuario' maxlength="50" /><br />
8      </div>
9      <div class='fila'>
10         <label for='password'>Contraseña:</label><br />
11         <input type='password' name='inputPassword' id='password' maxlength="50" /><br />
12     </div>
13     <div class='fila'>
14         <input type='submit' name='enviar' value='Enviar' />
15     </div>
16 </fieldset>
17 </form>

```

Este código es un HTML que representa un formulario que está conectado con login.php, usando fieldset para agrupar los datos del formulario. La línea php representa por si hay algún error. Luego usamos div para ir agrupando lo que nos interesa dándole clases. El label es para pintar por máquina Usuario:, y el input de modo text con un máximo de 50 caracteres para que escriba su usuario, luego el otro input usamos submit para que le llegue al php.

```

1  <?php
2  // Comprobamos si ya se ha enviado el formulario
3  if (isset($_POST['enviar'])) {
4      $usuario = $_POST['inputUsuario'];
5      $password = $_POST['inputPassword'];
6
7      // validamos que recibimos ambos parámetros
8      if (empty($usuario) || empty($password)) {
9          $error = "Debes introducir un usuario y contraseña";
10         include "index.php";
11     } else {
12         if ($usuario == "admin" && $password == "admin") {
13             // almacenamos el usuario en la sesión
14             session_start();
15             $_SESSION['usuario'] = $usuario;
16             // cargamos la página principal
17             include "main.php";
18         } else {
19             // Si las credenciales no son válidas, se vuelven a pedir
20             $error = "Usuario o contraseña no válidos!";
21             include "index.php";
22         }
23     }
24 }

```

Lo que hace la primera línea es comprobar si desde el form se ha enviado, creamos 2 variables para usuario y contraseña, que las cogemos usando POST. Si Usuario o Contraseña están vacíos damos un error, si no, si el usuario y contraseña tienen el valor admin, usamos session\_start para almacenar la sesión, y devolvemos la página principal que sería main.php, si no fuera el caso damos un error enviándolo al índice.

```

1  <?php
2      // Recuperamos la información de la sesión
3      if(!isset($_SESSION)) {
4          session_start();
5      }
6
7      // Y comprobamos que el usuario se haya autenticado
8      if (!isset($_SESSION['usuario'])) {
9          die("Error - debe <a href='index.php'>identificarse</a>.<br />");
10     }
11  ?>
12  <!DOCTYPE html>
13  <html lang="es">
14  <head>
15      <meta charset="UTF-8">
16      <meta name="viewport" content="width=device-width, initial-scale=1.0">
17      <title>Listado de productos</title>
18  </head>
19  <body>
20      <h1>Bienvenido <?= $_SESSION['usuario'] ?></h1>
21      <p>Pulse <a href="logout.php">aquí</a> para salir</p>
22      <p>Volver al <a href="main.php">inicio</a></p>
23      <h2>Listado de productos</h2>
24      <ul>
25          <li>Producto 1</li>
26          <li>Producto 2</li>
27          <li>Producto 3</li>
28      </ul>
29  </body>
30  </html>

```

Lo primero miramos para devolver la información de la sesión, si el usuario no está registrado, devolvemos error devolviéndole a index.php. Creamos un html dándole la bienvenida al usuario , y un a href por si quiere volver a logout.php o al inicio. Luego creamos una lista de los productos.

```

1  <?php
2      // Recuperamos la información de la sesión
3      session_start();
4
5      // Y la destruimos
6      session_destroy();
7      header("Location: index.php");
8  ?>

```

Este Código sirve para borrar las sesiones.