## **Tarea para DWES01**

# **Tarea online**

**Título de la tarea: Gestionar Direcciones de Correo**

**Unidad 1: Plataformas de programación web en entorno servidor. Aplicaciones LAMP.**

**Ciclo formativo y módulo: Ciclo superior de DAW. Módulo de DWES**

**Curso académico:**...

#### ¿Qué contenidos o resultados de aprendizaje trabajaremos?

#### Contenidos

1. Características de la programación web.
2. Páginas web estáticas y dinámicas (I).
3. Páginas web estáticas y dinámicas (II).
4. Aplicaciones web.
5. Ejecución de código en el servidor y en el cliente.
6. Tecnologías para programación web del lado del servidor.
7. Arquitecturas y plataformas.
8. Selección de una arquitectura de programación web.
9. Integración con el servidor web.
10. Lenguajes.
11. Entorno de programación PHP.
12. XAMPP.
13. Apache Netbeans IDE.
14. Conceptos básicos de programación web con PHP.
15. Crear proyecto PHP "Hola mundo".
16. Literales, constantes, variables y tipos de datos.
17. Operadores y funciones de librería.
18. Evaluación de expresiones.

#### Resultados de Aprendizaje

* **RA 1:** Selecciona las arquitecturas y tecnologías de programación Web en entorno servidor, analizando sus capacidades y características propias.
* **RA 2:** Escribe sentencias ejecutables por un servidor Web reconociendo y aplicando procedimientos de integración del código en lenguajes de marcas.
* **RA 3**: Escribe expresiones en PHP.

# **1.- Descripción de la tarea**

## **Caso práctico**

**Juan** debe explicar a **Carlos** las necesidades que tendrá un nuevo proyecto para controlar direcciones de correo, para ello le pide que piense detenidamente que tecnología serían las óptimas y cuáles no. **Carlos** con los conceptos que le ha explicado se pone manos a la obra,

#### ¿Qué te pedimos que hagas?

Quieres programar una aplicación web para gestionar una lista de direcciones de correo. En una primera aproximación, tu aplicación se compondrá de tres páginas:

* Una página de presentación, donde explicas el cometido de la aplicación y su funcionamiento. Contiene un enlace que te lleva a la siguiente página.
* Una página de introducción de datos, donde cualquiera puede darse de alta introduciendo su nombre y su dirección de correo. Contiene un formulario que, una vez rellenado, envía los datos y te lleva a la siguiente página.
* Una página de visualización de datos, en la que se muestra la lista, nombres y direcciones de correo, de todos los que se han anotado en la aplicación.

Tienes que crear un documento en el que respondas a las siguientes preguntas sobre la aplicación que vas a desarrollar:

1. **¿Qué tipo de páginas, estáticas o dinámicas, utilizarás para programar cada una de las páginas que componen tu aplicación? ¿Por qué?**

* Para la página de presentación: utilizaríamos páginas estáticas ya que no necesitamos obtener datos de ninguna fuente y el código de programación no variara.
* Para la página de introducción de datos: necesitaríamos utilizar páginas dinámicas ya que se encargarán de comprobar los datos que se vaya a enviar como el correo y luego enviarlo al servidor para ser almacenado en una base de datos.
* Y para la página de visualización de datos: también tiene que ser una página dinámica ya que se encargará de obtener la información de la base de datos que el cliente web ha solicitado en su petición.

1. **Si en la página de introducción de datos quieres comprobar, antes de enviar los datos, que el correo electrónico introducido cumple unas ciertas normas (por ejemplo, que tiene una @), ¿qué tecnología/lenguaje utilizarás?**

* Podría utilizar el lenguaje de marcas HTML del lado cliente, y en la etiqueta input de tipo email y en su atributo “pattern” codificaríamos una expresión regular que evalué el formato de la dirección de correo.
* También podríamos utilizar el lenguaje JavaScript del lado del cliente para comprobar el patrón de la dirección de correo que se vaya a enviar, utilizando de la misma manera expresiones regulares.
* Podríamos utilizar PHP en el lado servidor, pero lo que haríamos es sobrecargar el servidor con un proceso más cuando solo lo que queremos es hacer una validación.
* Y utilizando AJAX (JavaScript asíncrono y XML) una tecnología Frontend un poco más avanzada que permite validar la información antes de ser enviada, es como comprobaríamos si el correo cumple con ciertas normas o patrones de una dirección de correo antes de enviarlo.

1. **Si en esa misma página, ahora quieres comprobar que el correo electrónico introducido no se haya introducido anteriormente y ya figure en la lista, ¿qué tecnología/lenguaje utilizarás?**

* Para la comprobación del email desde una lista o este dado de alta lo que utilizaremos será PHP, un lenguaje de servidor junto con una base de datos como MySQL, pero también podemos utilizar AJAX para la comprobación o validación de datos.

1. **¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación?**

Para la ejecución de aplicación podrías utilizar arquitecturas Open Source o de código propietario.

**¿Cómo es el o los lenguajes que se usa en cada una de esas arquitecturas: de guiones, compilado a código nativo o compilado a código intermedio?**

* Al tratarse de una pequeña aplicación con pocos requisitos y que no exige demasiada rapidez podemos optar por una arquitectura AMP de código abierto que apenas se necesita hacer una inversión. El lenguaje de PHP es un lenguaje de programación de guiones, por tanto, se interpretará cada vez que se ejecute teniendo la ventaja que el código fuente se puede modificar ya que no se compila, sin embargo, tenemos la desventaja de que su rendimiento y rapidez es peor.
* Si la aplicación fuese más exigente podríamos recurrir a una arquitectura JavaEE con un servidor de aplicaciones Tomcat que es de código abierto, el lenguaje de programación sería JSP de código compilado a código intermedio, pero con la ventaja de tener una mayor rapidez.
* Si optamos por una arquitectura propietaria podríamos utilizar ASP.net con un servidor web IIS de Microsoft que integra ASP y ASP.net como lenguaje de programación compilado, y una base de datos propietaria, SQL Server. El rendimiento y velocidad sería superior que a la de PHP.

1. **¿Qué parámetros debes tener en cuenta para decidirte por usar una arquitectura u otra?**

* Para decidirnos por una arquitectura debemos tener en cuenta los siguientes puntos:
* Tamaño del proyecto.
* Lenguaje de programación que se adapte al proyecto o con el que más experiencia contamos.
* Código abierto o propietario, para no originar más gastos de los necesarios.
* Programaré sólo o en equipo, dependiendo del nivel del proyecto.
* Alojamiento de nuestro proyecto web tanto para producción como para desarrollo.
* Licenciamiento de la aplicación, que para reducir coste puede ser GPL.

1. **Si te decides por utilizar una arquitectura AMP para la aplicación ¿qué componentes necesitas instalar en tu servidor para ejecutar la aplicación? Indica algún producto concreto para cada componente.**

* Un servidor web como Apache en su versión 2.4.6 donde se almacenarán todos los ficheros que formen nuestro sitio web.
* Un servidor de bases de datos como MySQL 9 que servirá para almacenar de forma persistente la información que haya recibir la página web.
* Un lenguaje de programación como el lenguaje de PHP en su versión 8.3 con el que comprobaremos en el lado servidor la validez de los datos introducidos que se envía por el formulario.
* También podríamos instalar un panel de administración para la base de datos como el phpMyAdmin 3.5 con lo que conseguiremos administrar las distintas bases de datos, creando nuevas bases de datos, alterando las existentes o borrando las innecesarias.

1. **¿Qué necesitas instalar en tu ordenador para poder desarrollar la aplicación?**

* Si no disponemos de un servidor web podríamos optar por virtualizarlo o en el mejor de los casos descargarnos e instalar el paquete XAMPP que es gratuito y nos permitirá en una sola instalación tener todo lo del punto 6 además de un servidor FTP, un servidor de aplicaciones Tomcat y el lenguaje Perl.
* Una vez instalado el paquete XAMPP se necesitará realizar una configuración mínima de seguridad ya que por defecto no es tan estricta.

1. **Si utilizas el lenguaje PHP para programar la aplicación, ¿cuál será el tipo de datos se utilizará para manipular cada una de las direcciones de correo?**

* Las direcciones de correo electrónico pueden estar compuestas por caracteres alfabéticos, numéricos y especiales, por lo que el tipo de datos será el de cadena (string), aunque en PHP no es necesario declararla.

1. **Escribe una expresión PHP que calcule la edad de las personas de la agenda. La expresión utiliza las variable $diaNac, $mesNac, $anyoNac con los datos de la fecha de nacimiento de la persona y $diaHoy, $mesHoy y $anyoHoy con clos datos de la fecha de hoy. Debes programar, sin sentencias de control, una expresión que calcule la edad de la persona. Fíjate que la edad no se calcula restando los valores de los años ya que puede ser que la persona no haya cumplido todavía en el año en curso.**

|  |
| --- |
| 1. $diaHoy = date("d"); 2. echo $diaHoy; 3. $mesHoy = date("m"); 4. echo $mesHoy; 5. $anyoHoy = date("Y"); 6. echo $anyoHoy; 7. $diaNac = 10; 8. $mesNac = 11; 9. $anyoNac = 1985; 10. function CalculaEdad($diaNac, $mesNac, $anyoNac, $diaHoy, $mesHoy, $anyoHoy) { 11. return(($mesHoy < $mesNac) || ($mesHoy == $mesNac && $diaHoy < $diaNac) ? ($anyoHoy - $anyoNac) - 1 : $anyoHoy - $anyoNac ); 12. } 13. echo "<br>"; 14. echo 'La edad en años es: ' . CalculaEdad($diaNac, $mesNac, $anyoNac, $diaHoy, $mesHoy, $anyoHoy); |

# **2.- Información de interés**

#### Recursos necesarios y recomendaciones

##### Recursos:

No se necesitan recursos específicos para esta tarea.

##### Recomendaciones:

Se recomienda leer detenidamente cada una de las cuestiones planteadas.

## **Indicaciones de entrega**

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**apellido1\_apellido2\_nombre\_DWES01\_Tarea**

# **3.- Evaluación de la tarea**

#### Criterios de evaluación implicados

* Se han caracterizado y diferenciado los modelos de ejecución de código en el servidor y en el cliente Web.
* Se han reconocido las ventajas que proporciona la generación dinámica de páginas Web y sus diferencias con la inclusión de sentencias de guiones en el interior de las páginas Web.
* Se han identificado los mecanismos de ejecución de código en los servidores Web.
* Se han reconocido las funcionalidades que aportan los servidores de aplicaciones y su integración con los servidores Web.
* Se han identificado y caracterizado los principales lenguajes y tecnologías relacionados con la programación Web en entorno servidor.
* Se han verificado los mecanismos de integración de los lenguajes de marcas con los lenguajes de programación en entorno servidor.
* Se han reconocido y evaluado las herramientas de programación en entorno servidor.
* Se han caracterizado y diferenciado los modelos de ejecución de código en el servidor y en el cliente Web.
* Se han reconocido las ventajas que proporciona la generación dinámica de páginas Web y sus diferencias con la inclusión de sentencias de guiones en el interior de las páginas Web.
* Se han identificado los mecanismos de ejecución de código en los servidores Web.
* Se han reconocido las funcionalidades que aportan los servidores de aplicaciones y su integración con los servidores Web.
* Se han identificado y caracterizado los principales lenguajes y tecnologías relacionados con la programación Web en entorno servidor.
* Se han verificado los mecanismos de integración de los lenguajes de marcas con los lenguajes de programación en entorno servidor.
* Se han reconocido y evaluado las herramientas de programación en entorno servidor.
* Se han programado expresiones en PHP.

#### ¿Cómo valoramos y puntuamos tu tarea?

|  |  |
| --- | --- |
| **Rúbrica de la tarea (Total 10 puntos.)** | |
| Apartados 1, 2, 3, 5, 7, 8. Responder correcta y justificadamente a las cuestiones planteadas. | 0,5 Punto por pregunta. |
| Apartados 4 y 6 .Responder correcta y justificadamente a las cuestiones planteadas. | 2 Puntos por pregunta. |
| Apartado 9. Expresión correcta | 3 Puntos |