**Tarea para DWES01.**

* **¿Qué tipo de páginas, estáticas o dinámicas, utilizarás para programar cada una de las páginas que componen tu aplicación? ¿Por qué?**
* Primera pagina: página estática creada en HTML con CSS para el aspecto visual.
* Segunda página: página estática creada en HTML con un formulario en PHP donde se puede introducir la información y enviarla al servidor.
* Tercera página: página dinámica escrita en PHP que crea automáticamente una tabla de HTML con la información de los usuarios que han introducido la información. Dicha información será guardada en una base de datos de MySQL.
* **¿Qué generación de página web será? Razona tu respuesta**

Sería una página web de segunda generación, al ser parte de ella (la tercera página) una web dinámica.

* **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?**

Se puede comprobar con PHP utilizando el siguiente código:

// comprobamos que el campo no esté vacío

if (empty($\_POST["email"])) {  
    $emailErr = "Campo email obligatorio";  
  } else {  
    $email = test\_input($\_POST["email"]);  
    // comprobamos que el formato sea válido  
    if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {  
      $emailErr = "Formato de dirección inválido";   
    }  
  }

* **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?**

PHP puede acceder a la base de datos y realizar una consulta para comprobar que dicha dirección de correo no se encuentre ya en la base de datos. También se le podría asignar al campo “email” de la base de datos la propiedad “UNIQUE” para que no permita duplicidades.

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

Dada la simplicidad de la aplicación da un poco igual qué arquitectura se utilice. No tendría mucho sentido WAMP o ASP.Net al incluir software propietario y no darnos ninguna ventaja sobre el software libre, así que la solución lógica sería utilizar LAMP.

No solo es software libre, utilizar un lenguaje de guiones (scripting) como PHP o Python nos da ciertas ventajas sobre lenguajes interpretados como C. El hecho de que se ejecute directamente sobre el código fuente nos permite hacer cambios de una forma más rápida que teniendo que recompilar el código, y aunque en principio tengan un rendimiento peor, el hecho de que vaya a hacer múltiples consultas cortas a la larga hace que sea más eficiente que ir creando un proceso nuevo para cada consulta.

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

Lenguajes de programación que se conozcan, o si vale la pena aprender un lenguaje nuevo para crear una aplicación concreta, sobre qué sistema operativo se van a ejecutar las aplicaciones, si el software libre es la mejor opción o, si no lo es, si merece la pena el coste del software propietario.

* **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.**
* Sistema operativo: puede ser Linux (LAMP), Windows (WAMP), MAC (MAMP).

Sería recomendable instalar las versiones de servidor de dichos sistemas operativos, como Ubuntu Server, Windows Server o macOS Server.

* Apache como servidor HTTP.
* Servidor de MySQL para la base de datos.
* Runtime de PHP/Perl/Python.

También, en lugar de instalar cada componente por separado, se podría instalar directamente el paquete XAMPP, que incluye todo lo necesario en un único instalador y se puede ejecutar en cualquier sistema operativo.

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

Por necesitar, no se necesita nada más que un procesador de texto plano, pero es recomendable instalar un IDE que no sólo nos facilite la escritura del código, también nos permita ir viendo a tiempo real los cambios que vamos realizando a la página. Un buen ejemplo sería utilizar Visual Studio Code con la extensión Live Server.

* **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?**

Cadena, el único tipo de datos simple que trabaja con texto.