

1.- Descripción de la tarea

¿QUÉ TE PEDIMOS QUE HAGAS?

Supón que 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 desarrollarías:

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

- **Página de presentación:** página estática ya que su contenido no cambia frecuentemente y no requiere interacción con el usuario, utilizaría HTML para la estructura y CSS para los estilos.
- **Página de introducción de datos:** Dinámica, ya que necesita procesar la información del formulario y mostrar resultados específicos según la interacción del usuario. Utilizaría HTML para la estructura del formulario, estilos con CSS y validaría la información del formulario con PHP donde para introducir la información, validarla y enviarla al servidor.
- **Una página de visualización de datos:** Dinámica, para mostrar la lista actualizada de direcciones de correo, utilizaríamos PHP para actualizar la visualización de datos almacenados en una base de datos como MySQL.

2. *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?*

Utilizaría **JavaScript** para la validación del correo electrónico en el lado del cliente antes de enviar los datos al servidor.

3. *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?*

Utilizaría un lenguaje del lado del servidor como **PHP** para realizar la verificación y manipulación de datos en la base de datos.

4. *¿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?*

Para una aplicación pequeña sin requisitos específicos y sin necesidad de alta velocidad, se elige una arquitectura **AMP**, que consiste en un servidor web **Apache**, un servidor de base de datos **MySQL** y el lenguaje de programación **PHP**. Estos componentes son **de código abierto**, lo que elimina la necesidad de inversión. PHP, siendo un lenguaje de guiones, se interpreta cada vez que se ejecuta, lo que permite modificar el código sin necesidad de compiladores. Aunque la interpretación puede ralentizar el proceso, esto es generalmente insignificante para volúmenes pequeños o medianos de datos.

Para una aplicación más compleja, se consideraría una arquitectura **Java EE** con un servidor de aplicaciones **JOnAS**, que incluye un contenedor web **Tomcat**. Se utilizarían páginas JSP, un código compilado a código intermedio, lo que mejoraría la velocidad en comparación con PHP en la arquitectura mencionada anteriormente.

Si se opta por una arquitectura propietaria, se podría usar **ASP.Net** con el servidor web IIS de **Microsoft**, que admite páginas dinámicas **ASP** y **ASP.NET**. es un lenguaje compilado a código nativo, lo que aumenta la velocidad de procesamiento, pero también aumenta los **costos**. Se utilizaría el servidor de bases de datos **SQL Server de Microsoft**, que también es propietario.

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

Para determinar la arquitectura más adecuada, es crucial considerar:

- **Dimensión del proyecto:** En este escenario, se presenta como relativamente reducido.
- **Lenguajes familiares:** PHP, cierta cantidad de Java y ASP conforman nuestro conjunto de habilidades. Dadas las proporciones del proyecto, no se aconseja destinar recursos significativos a aprender nuevos lenguajes ni a perfeccionar los ya conocidos. La imperiosa necesidad consiste en llevar a cabo la programación de manera eficiente, precisa y con una inversión justa en la actualización de habilidades.
- **Software de código abierto o propietario:** Al igual que en los puntos anteriores, la escala limitada del proyecto sugiere la preferencia por soluciones de código abierto y gratuito, evitando así gastos en software.
- **Programación individual o en colaboración:** La magnitud del proyecto me permite abordarlo de manera individual.
- **Poseo un servidor web o decido emplear el que estime conveniente:** Dispongo de un servidor web con arquitectura AMP para albergar integralmente el proyecto.
- **Tipo de licencia de la aplicación:** Optaré por una licencia GPL de código abierto, manteniéndome fiel al enfoque minimalista de la aplicación.

6. *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.*

La estructura AMP constará de:

- **Un servidor web**, como por ejemplo Apache donde se guardarán todos los archivos que configuren el sitio completo.
- **Un servidor de bases de datos**, como MySQL, que se utilizará para almacenar de manera constante la información que el sitio deba enviar o recibir.
- **Un lenguaje de programación**, como el lenguaje de guiones PHP, mediante el cual verificaremos **en el lado del servidor** la validez de los datos ingresados. Además, se encargará de enviar los datos a la base de datos y solicitar información de la misma.
- También podríamos implementar **un panel de administración para la base de datos**, como phpMyAdmin para administrar diversas bases de datos sin necesidad de programación, facilitando la creación de nuevas bases, la modificación de existentes o la eliminación de las innecesarias.

Para probar el proyecto, podemos instalar en nuestro equipo un servidor web del tipo AMP mediante el uso de un paquete como **XAMPP(X-operating system, Apache, Mysql, Php, Perl)**, que integra todos los componentes mencionados anteriormente.

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

Para llevar a cabo el proyecto, necesitaremos descargar e instalar los siguientes componentes:

XAMPP: Descarga e instala este paquete gratuito que incluye:

- **Un servidor web** (por ejemplo, Apache) para almacenar todos los archivos del sitio.
- **Un servidor de bases de datos (MySQL)**, para gestionar la información del sitio.
- **Un lenguaje de programación (PHP)**, para verificar la validez de los datos y trabajar con la base de datos.
- **Servicios adicionales** como FTP (Filezilla), un servidor de aplicaciones (Tomcat), y el lenguaje Perl.

Reforzar la Seguridad de XAMPP: Dado que XAMPP carece de una seguridad estricta por defecto, es esencial tomar medidas para fortalecerla. Utilizando las herramientas incluidas en XAMPP destinadas a mejorar la seguridad y aplicando configuraciones adicionales para proteger las partes críticas del sistema.

8. *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?*

Utilizando PHP para programar la aplicación, el tipo de datos que utilizaré para manipular las direcciones de correo electrónico será un tipo de dato **string**. En PHP, las direcciones de correo electrónico se manejan como cadenas de texto y se pueden realizar operaciones y validaciones específicas para asegurar su formato correcto y su manipulación adecuada en la aplicación.