

Desarrollo web en entorno servidor

Tarea 1 RA1

Alejandro Rico Soler

16 de octubre del 2024

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:**

Esta será una página estática, ya que su contenido no cambiará de acuerdo a los usuarios o interacciones. Simplemente ofrecerá información sobre la aplicación y el enlace a la siguiente página.

- **Página de introducción de datos:**

Esta será una página dinámica, ya que incluye un formulario para que los usuarios puedan enviar información. Necesita interactuar con el servidor para validar y procesar los datos introducidos.

- **Página de visualización de datos:**

También será una página dinámica porque mostrará la lista de personas que se han registrado, y esa lista se actualizará constantemente a medida que más personas se inscriban.

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?

Para hacer una validación en el lado del cliente antes de enviar el formulario, usaría JavaScript. Es una tecnología común para validar datos en el navegador del usuario sin necesidad de recargar la página o enviar la información al servidor. En este caso, puedes usar una expresión regular (regex) para verificar que el correo electrónico contenga una "@" y otros aspectos básicos de la sintaxis.

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

Para esta verificación, se requiere validar contra los datos almacenados en el servidor. Usaría un lenguaje de servidor como PHP o Node.js junto con una base de datos (por ejemplo, MySQL). El flujo sería que el servidor reciba el correo, lo compare con los registros en la base de datos y, si ya existe, devuelva un mensaje de error al usuario.

4. ¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación? ¿Cómo es el o los lenguajes que se usan en cada una de esas arquitecturas: de guiones, compilado a código nativo o compilado a código intermedio?

- **Arquitectura LAMP (Linux, Apache, MySQL, PHP):**

PHP es un lenguaje de guiones (scripts) que se ejecuta en el servidor interpretado directamente por el intérprete PHP.

- **Arquitectura MEAN (MongoDB, Express.js, Angular, Node.js):**

Node.js utiliza JavaScript, que es interpretado (similar a PHP) y se ejecuta en un motor como V8.

- **Arquitectura con .NET (Windows, IIS, SQL Server, ASP.NET):**

ASP.NET puede compilarse a código intermedio (IL) y luego ser ejecutado por la máquina virtual .NET (CLR).

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

- **Escalabilidad:**

Algunas arquitecturas son más fáciles de escalar horizontalmente (por ejemplo, MEAN con Node.js).

- **Facilidad de mantenimiento:**

Si ya tienes experiencia en un lenguaje o arquitectura (como PHP o .NET), eso puede ser un factor importante.

- **Rendimiento:**

Algunas arquitecturas como .NET pueden ofrecer mejor rendimiento en aplicaciones más grandes debido a su compilación a código intermedio.

- **Coste:**

Algunas arquitecturas (LAMP) usan software de código abierto, lo que reduce los costes frente a soluciones propietarias como .NET.

- **Compatibilidad con otros sistemas:**

Dependiendo del ecosistema de la organización, podrías elegir una u otra arquitectura (Windows/SQL Server vs Linux/MySQL).

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

Si optamos por una arquitectura LAMP:


- Sistema operativo: Linux (por ejemplo, Ubuntu).
- Servidor web: Apache.
- Lenguaje de servidor: PHP.
- Base de datos: MySQL o MariaDB.

En el caso de usar MEAN:

- Sistema operativo: Linux o Windows.
- Servidor web: Express.js (que se ejecuta sobre Node.js).
- Lenguaje de servidor: JavaScript (Node.js).
- Base de datos: MongoDB.

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

- Un entorno de desarrollo como Visual Studio Code o Sublime Text.
- Servidor local como XAMPP (para PHP) o instalar Node.js (para un entorno MEAN).
- Base de datos local como MySQL o MongoDB, según la arquitectura.
- Un navegador para probar la aplicación.
- Opcionalmente, alguna herramienta de control de versiones, como Git, para gestionar el código.



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?

El tipo de datos para manipular las direcciones de correo en PHP será una cadena de texto (string), ya que las direcciones de correo son secuencias de caracteres. Además, podrías utilizar algunas funciones de PHP para validar el formato del correo, como `filter_var($email, FILTER_VALIDATE_EMAIL)`.