

Alvarez_Valencia_Gianfranco_DWES06_ Tarea

Índice

1. Punto 1 3

2. Punto 2 3

3. Punto 3 3

4. Punto 4 4

5. Punto 5 4

6. Punto 6 4

7. Punto 7 4

8. Punto 8 5

9. Punto 9 5

10. Punto 10 5

Con el objetivo de gestionar una lista de direcciones de correo, se va a programar una aplicación web. En una primera aproximación, esta 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.

1. Punto 1

¿Utilizarás páginas estáticas o dinámicas para programar cada una de las páginas que componen tu aplicación? ¿Por qué?

La primera página sería estática ya que no hay que modificar nada.

La segunda página sería dinámica que, aunque la validación del formulario se puede hacer en el navegador estos datos se envían al servidor que genera la siguiente página en función de los datos introducidos.

La tercera dinámica ya que muestra la página generada por el servidor en función de los usuarios que se han registrado en la aplicación por lo tanto no siempre será la misma página.

2. Punto 2

Si en la página de introducción de datos quieres comprobar que el correo electrónico introducido cumple unas ciertas normas (por ejemplo, que tiene una @), ¿qué tecnología/lenguaje utilizarás? ¿será necesario recargar la página para explicarle al usuario que el formato del correo electrónico introducido no es correcto?

Usaría JavaScript, se le podía informar al usuario con un alert.

3. Punto 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 tecnologías o lenguajes utilizarás? Indica, al menos, dos opciones.

Habría que validar los datos introducidos con la base de datos en la que se almacenan los datos de los usuarios y generar que se cree la nueva página con el resultado mediante.

Se podrían usar PERL O PHP.

4. Punto 4

¿Qué tecnología podrías utilizar para almacenar las direcciones de correo de los distintos usuarios de forma permanente? Indica, al menos, tres ejemplos explicando si son o no open source.

Abría que usar gestores de bases de datos:

MySQL – open source

Microsoft SQL Server. – no open source

PostgreSQL – open source

5. Punto 5

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

Java EE (Enterprise Edition)

CGI/Perl.

AMP

ASP.Net

6. Punto 6

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

- El tamaño del proyecto.
- Lenguaje de programación que se va a usar, tener en cuenta si es conocido por el equipo de desarrollo.
- Uso de herramientas open source o no.
- Características del servidor web y gestor de base de datos disponible para el proyecto en caso de disponer de uno o las características de los nuevos servidores.
- Licencia para la aplicación a desarrollar, si es open source o propietaria.

7. Punto 7

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.

Servidor web: Apache

Gestor de bases de datos: MySQL

Programación: Modulo PHP para implantar la programación.

8. Punto 8

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

Un IDE para llevar a cabo el desarrollo de la aplicación, no es imprescindible pero facilita mucho la tarea.

9. Punto 9

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?

Será **cadena** (string).

10. Punto 10

Para profundizar: hasta ahora, hemos visto que JavaScript es una tecnología para el desarrollo web en cliente y así ha sido tradicionalmente. En los últimos años han surgido soluciones que llevan JavaScript al lado servidor. Busca información sobre Node.JS, explica en qué consiste esta tecnología y cuáles son sus ventajas y desventajas respecto a las tecnologías tradicionales de desarrollo web en lado servidor vistas en el tema (Java EE, AMP, ASP.Net, etc.).

Node.js es un entorno en tiempo de ejecución multiplataforma para crear y ejecutar aplicaciones JavaScript la capa del servidor (en el lado del servidor) basado en JavaScript.

Ventajas:

- Velocidad
- Sin búfer. Las aplicaciones de Node.js generan los datos en trozos (chunks), nunca los almacenan en búfer.
- Un subproceso escalable. utiliza un modelo de un solo subproceso con bucle de eventos. Gracias al mecanismo de eventos, el servidor responde sin bloqueos, como hemos dicho. Esto hace que el servidor sea altamente escalable comparando con los servidores tradicionales como el Servidor HTTP de Apache.
- Las aplicaciones de Node.js están escritas en JavaScript - en lenguaje de la moda - y pueden ejecutarse en OS X, Microsoft Windows y Linux.

Desventajas:

- Tiene una API inestable, quiere decir que entre versiones se tiende a romper la compatibilidad y se deben hacer cambios en tus aplicaciones para mantener todo funcionando.
- Cualquier operación con uso intensivo de la CPU anula todos los beneficios de rendimiento que Node.js ofrece con su modelo de E / S sin bloqueo controlado por eventos porque todas las solicitudes entrantes se bloquearán mientras el hilo esté ocupado con su procesamiento de números.

Fuentes:

- <https://sospnt.com/blog/57-hola-mundo-con-node-js-2#:~:text=Dentro%20de%20las%20desventajas%20node,beneficios%20de%20rendimiento%20que%20Node.>
- <https://www.itdo.com/blog/que-es-node-js-y-para-que-sirve/>