

## DWES 01

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

Utilizaría una pagina estática para la pagina de presentación, ya que el contenido no a variar a menudo y dinámicas para las otras dos. En las cuales vamos a utilizar código javascript para validar los datos de entrada del usuario y vamos a obtener la información almacenada en nuestra base de datos.

**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?**

Vamos a utilizar código en javascript ejecutado en el navegador web para validar el contenido de los datos tecleados por el usuario. No es necesario recargar la página ya que este código que vamos a implementar nos mostrará un mensaje en caso de introducir los datos de manera no valida y no mandaremos los datos al servidor hasta que estén de manera correcta.

**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.**

Para comprobar que el mail introducido esta o no en la lista deberemos comprobarlo en nuestra base de datos. Para ello, según la arquitectura utilizada, nos conectaremos a dicha base de datos y comprobaremos si el registro esta o no. Ejemplos:

En la arquitectura ASP.Net tendremos nuestro código en .NET y nos conectaremos a Microsoft SQL Server para comprobar si existe o no, nuestro email.

En una arquitectura AMP, nuestro código puede estar en PHP y nos conectaremos a través de MySQL a la base de datos para comprobar que nuestro registro existe o no.

- 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.**

El almacenamiento de datos lo vamos a gestionar con los gestores de bases de datos. Como hemos visto anteriormente, según la arquitectura de nuestra aplicación tendremos disponibles unos u otros: MySQL (Open Source), PostgreSQL (Open Source), Microsoft SQL Server (Software propietario), ...

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

Tenemos diferentes opciones:

Java EE / AMP / CGI-Perl / ASP.Net

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

Son muchas las preguntas que nos podemos hacer a la hora de elegir una arquitectura. Un buen análisis previo siempre nos ayudara a la creación de nuestra aplicación.

Unos parámetros a tener en cuenta son: tamaño de la aplicación, lenguajes que conocemos, si vamos a utilizar Open Source o no, el tipo de licencia que vamos a dar a nuestra aplicación, si el trabajo lo hacemos solo o en equipo, si ya disponemos de un servidor web o sistema gestor de base de datos o podemos elegir libremente, etc

- 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.**

Necesitaríamos el servidor web: Apache y el servidor de bases de datos: MySQL o PostgreSQL.

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

Si seguimos con la arquitectura AMP, a parte del sistema operativo, necesitaremos el servidor web: Apache, el gestor de bases de datos: MySQL o PostgreSQL y un IDE, como NetBeans, que nos permita trabajar mas fácilmente con nuestro lenguaje de programación: PHP, Perl o Python.

- 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?**

El tipo de datos utilizado sería tipo cadena (string), que es un conjunto de caracteres delimitado por comillas simples o dobles.

**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.).

¿Que es Node.js?

Bajo el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web, fue creado [Node.js](#), un [entorno](#) en tiempo de ejecución multiplataforma de código abierto para la capa del [servidor](#) (pero no limitándose a ello), basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Por el contrario de la mayoría del código [JavaScript](#), no se ejecuta en un navegador, sino en el servidor. Implementa algunas especificaciones de [CommonJS](#). [Node.js](#) incluye un entorno REPL para depuración interactiva.

De hecho, es conocido como una tecnología que permite trabajar con [Javascript](#) del lado del servidor, pero poco a poco [Node](#) se está ganando un lugar importante en la comunidad.

\*\*\* Info: <https://www.facilcloud.com/noticias/5-ventajas-de-usar-node-js/>

¿Ventajas?

Rendimiento y escalabilidad.

Rapidez.

Manipulación de datos sencilla y sin una gran cantidad de cálculos.

Solo Javascript.

¿Desventajas?

No es tan fácil de implementar en aplicaciones existentes.

Documentación escasa.