

## Actividad Evaluable 1.

Álex Soriano Rueda

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. su contenido no varía en ningún momento y es la misma para todos los usuarios. Solo contiene texto y un enlace que te lleva a la siguiente página.
- **Página de introducción de datos:** Página estática. El contenido de esta es la misma para todos los usuarios. Contiene un formulario y un botón que confirma los datos y te lleva a la página de visualización de datos. La confirmación conecta con la base de datos y guarda los datos recogidos por el formulario.
- **Página de visualización de datos:** Página dinámica. Se actualiza cuando se envía una petición y muestra contenido diferente dependiendo de la información almacenada en la base de datos. Es decir, que la página HTML se genera de forma dinámica.

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?

Se puede hacer utilizando una función que, pasándole unas propiedades, compruebe si cumplen ciertas condiciones y devuelva una respuesta. Se puede hacer incrustando un script directamente en HTML o llamando a una función de un archivo externo. Yo personalmente utilizaría JavaScript, pero también se puede utilizar PHP.

He encontrado una función de validación de correo electrónico en JavaScript que además incorpora otras condiciones, cómo que el valor tenga que empezar con una letra:

```
function validarEmail(valor) {  
  if (/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3,4})+$/ .test(valor)){  
    alert("La dirección de email " + valor + " es correcta.");  
  } else {  
    alert("La dirección de email es incorrecta.");  
  }  
}
```

*Validar email en JavaScript*

Y para que la página funcione, tiene que estar alojada en un servidor web, como por ejemplo Apache.

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?

Se tiene que utilizar un lenguaje de programación capaz de establecer una conexión y enviar peticiones a la base de datos, y un sistema gestor de bases de datos.

Utilizaría un script o un archivo escrito en PHP que realice una conexión con la base de datos y haga una consulta para comprobar si el correo está repetido o no.

Como SGBD para almacenar la lista de correos electrónicos utilizaría MongoDB, ya que es una base de datos orientada a documentos, pero también se puede utilizar otros como MySQL.

4. ¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación? (Servidor, base de datos y lenguaje de programación).

Podemos usar diferentes arquitecturas web en el lado del servidor para ejecutar nuestras aplicaciones, y la forma de escogerla dependerá de las necesidades y características de la aplicación. Se necesita un servidor web, una base de datos y un lenguaje de script.

- Como **servidores web** para alojar la aplicación y servir las peticiones a los clientes podemos utilizar: Apache HTTP Server, Nginx o Internet Information Services. Estos son los más conocidos y utilizados, pero también se pueden escoger otros como Tomcat, NodeJS... etc.
- Los **lenguajes de programación** que se ejecutan en el lado del servidor pueden ser: PHP, ASP.NET, JSP, Python, Node.js, Go, Ruby...etc. Yo utilizaría Node.js, ya que cuenta con el respaldo de Google y me siento más familiarizado con él.
- Las **bases de datos** que podemos utilizar son MySQL, MariaDB, PostgreSQL, Microsoft SQL Server, MongoDB...etc.

Existen paquetes de software como **XAMPP** que crean una distribución fácil de instalar pueden ayudar en el desarrollo. XAMPP es una distribución de Apache que contiene MariaDB/MySQL, PHP y Perl.