

## EJERCICIO TEÓRICO

Desarrollo de Aplicaciones Web  
(2WET)

### Desarrollo Web Entorno-Servidor

---

#### DOCENTE

César Martín Alcolea

#### ALUMNO

Hugo Ruiz Sánchez

---

#### TEMA

1

#### NUMERACIÓN DE EJERCICIO

1

---

#### ENUNCIADO

*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.

---

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

- **Página de presentación:** Puesto que la página preliminar únicamente se encargará de proyectar información no susceptible a modificaciones frecuentes, y tampoco contendrá elementos interactivos, utilizaría una página estática.
- **Página de introducción de datos:** Si el usuario prosigue a la siguiente página, necesitará interactuar con un formulario para introducir sus datos; dicha información habría de ser remitida al servidor, por lo que inevitablemente, la página debería implementarse de manera dinámica.
- **Página de visualización de datos:** La visualización de los datos está sujeta al estado inmediato de la información almacenada en el servidor, por lo que es susceptible de cambio. Utilizaría una página dinámica.

**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?**

En el caso concreto de que quisiera un tipo de dato determinado – sea, por ejemplo, un correo electrónico que contuviera una arroba –, usaría las etiquetas especializadas para tales fines en *HTML*. Adjunto de un formulario para correo electrónico en *HTML*, habilitado para restringir campos de texto que no se adecúan a los patrones no correspondientes a un *e-mail*:

```
<input type="email" id="email" required />
```

O, si bien quisiera acotar patrones más precisos, usaría expresiones regulares, también denominadas *REGEX* (Regular Expressions). Adjunto de una expresión regular que valida un correo electrónico:

```
^[a-zA-Z._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$
```

**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?**

Usaría un lenguaje de consulta estructurado o *Structured Query Language* (SQL) en conjunto con un lenguaje de programación del lado servidor, a saber, *PHP*, *Python*, *Node.js*, entre otros . Mediante ambos, efectuaría una consulta en la base de datos comparando la información contenida en el formulario con los registros presentes en las tablas de los servidores.

**¿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?**

- **Arquitectura AMP:** Esto es, las siglas de *Apache*, *MySQL* y *PHP/Perl/Python*, para el servidor web, la base de datos y el lenguaje de programación; los lenguajes son interpretados (de guión o *scripts*), lo que significa que son interpretados directamente por el servidor web.
- **Java EE:** Se trata de un lenguaje compilado a nivel intermedio, por lo que se ejecuta en una máquina virtual o entorno de ejecución, en este caso, la JVM (Máquina virtual de Java o *Java Virtual Machine*).
- **CGI/Perl:** Combina lenguaje Perl con la tecnología CGI. Es un lenguaje interpretado o guionizado.
- **ASP.Net:** Basado en lenguajes como Visual Basic.Net o C#. Se compilan en código intermedio y se ejecuta en un entorno de ejecución denominado CLR (*Common Language Runtime*).

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

Deben tenerse presente la escala de la aplicación – si es simple o compleja -, el rendimiento requerido – quizá una arquitectura con lenguajes compilados es menos óptima que con lenguajes de interpretados -, la portabilidad deseada– si usamos lenguajes compilados mejoraremos la compatibilidad –, experiencia personal en un determinado lenguaje o tecnología, el desembolso que requerirá, su mantenimiento o si no lo tuviere, su comunidad, su seguridad e integración.

## **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.**

AMP son las siglas de *Apache*, *MySQL* y *PHP/Perl/Python*, tecnologías empleadas respectivamente para el servidor web, la base de datos y el lenguaje de programación.

Usaría MySQL para sostener las bases de datos de mi servidor o, en su defecto, su alternativa MariaDB, reconocido como un fork – bifurcación – del mismo, y totalmente compatible. Usaría Apache HTTP Server para el mantenimiento del servidor web, y concretamente PHP-

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

### **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?**

Para desarrollar una aplicación en PHP que administre direcciones de correo electrónico, es necesario un entorno de desarrollo que incluya lo antedicho:

1. Un servidor web (por ejemplo, Apache).
2. Un sistema de gestión de bases de datos (como MySQL).
3. PHP instalado.

Para manipular direcciones de correo electrónico, utilizarse utilizar variables de tipo "string" (cadena de caracteres) en PHP para almacenar y procesar las direcciones de correo electrónico.