UD2: Selección de arquitecturas y herramientas de programación.

Sitio:	Aula Virtual do IES de Teis	Impreso por:	Abel Mahón Cortés
Curso:	Desarrollo web en entorno servidor 2023-24 (DAW-DUAL-A)	Data:	Luns, 15 de Xaneiro de 2024, 09:02
Libro:	UD2: Selección de arquitecturas y herramientas de programación.		

Táboa de contidos

1. Fundamentos de las aplicaciones web

Fundamentos de las aplicaciones web

Como aproximación inicial a las arquitecturas de las aplicacione web podemos tomar como base la presentación disponible online de Juan Pavón Mestras: **Protocolos y arquitecturas de apliación en internet**

Archivo PDF

Transparencias: 1, 2, 3, 5, 8, 11 con énfasis en HTML

Arquitecturas: 19, 20, 21

2. Web estática vs web dinámica

Web estática

El cliente genera una petición HTTP mediante una URL a un servidor web que almacena en su sistema de ficheros documentos HTML y los envía en la respueta HTTP El navegador interpreta su contenido y luego se muestra al usuario. Esto se conoce como *procesamiento del lado del cliente* .

Web dinámica

El cliente genera una petición HTTP mediante una URL a un servidor web que envía la solicitud al **servidor** de aplicaciones . El servidor de aplicaciones puede **solicitar datos de una base de datos** primero y luego **construye la respuesta HTTP** basada en los datos recuperados de la base de datos. Esta respuesta se devuelve al **servidor web** , que devuelve el archivo HTML, construido por el servidor de aplicaciones, al cliente, a través de **la respuesta HTTP** . Esto se llama *procesamiento del lado del servidor* .

Este proceso se ilustra en la imagen disponible aquí. [Fuente]

3. Lado cliente

Se encarga de iniciar una comunicación vía HTTP a través de un navegador (Chrome, Firefox, Safari, etc.)

Tecnologías involucradas:

- JavaScript
- HTML
- CSS
- HTTP

4. Lado servidor

Componentes

Servidor web: Un proceso software que **atiende peticiones HTTP y envía respuestas HTTP.** Pueden ofrecer contenido web estático —por ejemplo, páginas HTML, archivos, imágenes, vídeo— y/o relacionarse con un servidor de aplicaciones. Puede estar alojado en un equipo hardware independiente o en una máquina virtual.

Servidor de aplicaciones: Infraestructura software que facilita la interacción entre los clientes de usuario final y el código de aplicación del lado del servidor(la *lógica empresarial*) para generar y entregar contenido dinámico. Esa infraestructura puede facilitar el uso de transacciones, la gestión de sesiones de usuario o el acceso a bases de datos remotas. Teóricamente, el cliente de un servidor de aplicaciones puede ser la propia interfaz de usuario final de la aplicación, un navegador web o una aplicación móvil, y la interacción cliente-servidor puede producirse a través de diversos protocolos de comunicación.
En la práctica, se ha desdibujado la línea entre los servidores web y los servidores de aplicaciones puesto que los clientes más habituales son los navegadores web y, a menudo, el mismo software es capaz de realizar ambas funciones.

4.1. Servidores web y servidores de aplicaciones

Servidores web

- Nginx es un servidor web de código abierto que incluye funciones de equilibrio de carga, proxy de correo y memoria caché HTTP. Es uno de los más utilizados en la actualidad.
- Apache HTTP Server (también conocido simplemente como "Apache"), es otro servidor web gratuito de código abierto.
- IIS (Internet Information Services): Conocido como servidor web que ofrece un conjunto de servicios que transforman un sistema Microsoft Windows en un servidor capaz de ofrecer servicios Web, FTP y SMTP entre otros. A diferencia de Apache y Nginx que dividen la carga de trabajo en diversos subprocesos, dependiendo de la configuración incluso en subprocesos por cada petición HTTP, IIS opta por el modelo de proceso único, es decir, que un solo proceso maneja todas las peticiones.

Servidores de aplicaciones

- Apache Tomcat es un servidor de aplicaciones de código abierto que ejecuta *servlets* Java (unas clases capaces de atender peticiones HTTP y generar respuestas HTTP) , representa y entrega páginas web que incluyen código JavaServer Page y actúa como servidor para aplicaciones Java Enterprise Edition (Java EE).
- Glassfish es un servidor de aplicaciones Java EE de código abierto alojado en la actualidad por Eclipse Foundation (enlace externo a IBM). Glassfish admite servlets Java, Enterprise JavaBeans (EJB) y más, pero también puede funcionar como servidor web y dar servicio a contenido web en respuesta a solicitudes HTTP.

[Fuente]

4.2. Lenguajes del programación en el entorno servidor

Se trata del lenguaje de programación cuyo código, bien sea como precompilado o bien interpretado, es ejecutado por un software específico en el servidor.

Existen múltiples alternativas a la hora de ejecutar código en el servidor.

- Uso de **lenguajes de scripting (PHP, ASP, Perl, Python, etc.),** donde el código es interpretado y se intercala con una plantilla de código HTML con la estructura básica de la página que se envía al cliente. Un lenguaje de scripting (o embebido) aquel que se intercala con el código HTML que forma una aplicación web. Para que estas porciones de código sean ejecutadas, el agente que actúa de servidor (servidor web) debe tener instalado un módulo (scripting engine) que sea capaz de reconocer e interpretar el código.
- Uso de **enlaces a programas y componentes ejecutables (CGI, JSP**, etc.), de tal forma que el código ejecutado por el servidor está almacenado en unidades precompiladas y ejecutadas de forma independiente, que generan las páginas enviadas al cliente.
- Uso de **estrategias "híbridas"** basadas en técnicas de respaldo (denominadas code-behind), como **ASP.NET** de Microsoft, en el que algunos elementos de código se intercalan con la lógica de presentación mientras que se mantiene la funcionalidad de dichos elementos en ficheros o librerías independientes en el servidor.

4.3. Tecnologías

- **Java EE** (Enterprise Edition), anteriormente también conocido como J2EE. Es una plataforma orientada a la programación de aplicaciones en lenguaje Java. Pueden trabajar con diferentes administradores de bases de datos, e incluye varias bibliotecas y especificaciones para el desarrollo de aplicaciones de forma modular. Es aplicable para el desarrollo de aplicaciones medianas o grandes. Una de sus principales ventajas es la multitud de librerías existentes en ese lenguaje y la gran base de programadores que lo conocen. Dentro de esta arquitectura existen diferentes tecnologías como las páginas JSP y la servlets, ambos orientados a la generación dinámica de páginas web, o EJBs, componentes que suelen proporcionar la lógica de la aplicación web. La última versión data del 2017 y a partir de ahí, ha dado el salto a JAKARTA EE8, cuando Oracle cedió todas las especificaciones a la fundación Eclipse. Se puede encontrar información sobre JAKARTA EE en su página web. Las novedades más significativas se pueden consultar aquí

- **AMP**: Responde a las siglas de Apache, MySQL y PHP/Perl/Python. El último componente corresponde al lenguaje de programación utilizado, que puede ser PHP, Perl o Python. Dependiendo del sistema operativo utilizado para el servidor, el significa LAMP (para Linux), WAMP (para Windows) o MAMP (para Mac). Tambien es posible utilizar otros componentes, como el administrador de base de datos PostgreSQL en en lugar de MySQL. Existen paquetes de software que incluyen un paquete de todos ellos en una sola instalación . Son tecnologías de código abierto y cuentan con una amplia comunidad de desarrolladores. Algunos ni siquiera necesitan ser instalados, e incluso tienen versiones para diferentes sistemas operativos como Linux, Windows o Mac. Uno de los mas conocido es XAMPP.

- **CGI/Perl.** Es la combinación del lenguaje de código libre Perl creado originalmente para la administración del servidor y CGI, un estándar para permitir que el servidor web ejecute programas genéricos, escritos en cualquier lenguaje (por ejemplo, C o Python también se puede usar), que devuelven páginas web (HTML) como resultado de su ejecución. Es el más antiguo de las arquitecturas que comparamos aquí. El principal inconveniente de esta combinación es que el CGI es lento.

- **ASP.Net** es la arquitectura comercial propuesta por Microsoft para el desarrollo de aplicaciones. Es la parte de la plataforma .Net destinada a la generación de páginas web dinámicas. Proviene de la evolución de la tecnología anterior de Microsoft, ASP. El lenguaje de programación puede ser Visual Basic.Net o C#. La arquitectura utiliza el servidor web de Microsoft, IIS, y puede obtener información de varios administradores de bases de datos, incluido, por supuesto, Microsoft SQL Server. La arquitectura .Net incluye todo lo que se necesita para el desarrollo y despliegue de aplicaciones. Por ejemplo, tiene su entorno de desarrollo propio, Visual Studio, aunque existen otras opciones disponibles. La mayor desventaja es que es una plataforma comercial de código propietario .

4.4. Ejemplos

Ejemplo de CGI

En una aplicación CGI, el servidor web pasa las solicitudes del cliente a un programa externo. Este programa puede estar escrito en cualquier lenguaje que soporte el servidor. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.

```
#!/usr/local/bin/perl
use CGI;
$query = new CGI;
$nome= $query->param('nome');
if ($nomee eq " ") {$nomee="Mundo";}
print "Content-type: text/html\n\n";
print <<"EOF";
<html>
  <head>
    <title>Ola Mundo CGI (Perl)</title>
    <script type="text/javascript" src="/js/jquery-1.6.1.min.js"></script>
    <script type="text/javascript" src="/js/jquery-1.3.2.min.js"></script>
   <script type="text/javascript" src="/js/jquery.easing.1.3.js"></script>
   <script type="text/javascript" src="/js/matxintrad.js"></script>
   <script type="text/javascript" src="/js/opentrad.js"></script>
   <link rel="stylesheet" type="text/css" media="screen" href="/css/estilo.css" />
   <link rel="stylesheet" type="text/css" media="screen" href="/css/estilo_imaxin.css" />
</head>
  <body>
    <h1>Ola, $nome</h1>
  </body>
</html>
```

Ejemplo de ASP.NET

Podéis consultar un ejemplo de implementación de una página con ASP.NET aquí

Ejemplo de uso de Servlets

Los Servlets son clases Java que deben implementar la clase abstracta **HttpServlet**, en especial el método **doGet()** o **doPost()** y deben ser previamente compilados.
Podéis consultar un ejemplo de uso de Servlets aquí en la sección Creación de Servlets con NetBeans

Ejemplo de uso de Java Server Pages (JSP)

Los archivos JSP pueden contener código Java entre código HTML utilizando los símbolos **<%** y **%>**. Por esto un archivo JSP debe ser interpretado por el servidor al momento de la petición por parte del usuario
Podéis consultar ejemplos aquí : Introducción a JSP a través de ejemplos

Ejemplo de uso de PHP

Se puede consultar aquí: https://www.php.net/manual/es/tutorial.firstpage.php

5. Herramientas de desarrollo

Para trabajar en el desarrollo de aplicaciones web es habitual:

Uso de navegadores web: Chrome, Firefox, Safari, Edge, etc.

- El uso de un IDE como Visual Studio Code, Eclipse, IntelliJ IDEA, PHPStorm, Visual Studio, etc.
- Herramientas que permitan crear y administrar bases de datos.
- Un servidor web y/o servidor de aplicaciones

Nosotros trabajaremos con XAMPP y Visual Studio Code