

SOBRE ARQUITECTURAS

Entendemos arquitectura web a la estructura, servicios, herramientas y tecnologías usadas para desplegar una aplicación que use un entorno web para su ejecución. Fundamentalmente, para el funcionamiento de este tipo de aplicaciones es necesario disponer de un servidor web, para recibir las peticiones de los clientes, un servidor de bases de datos, para guardar los datos necesarios para el servicio que ofrezca la aplicación y un módulo de ejecución de código, para que el programa pueda realizar las operaciones necesarias antes de devolver al cliente los resultados. Además, tanto para la parte del cliente, como para la del servidor, la arquitectura de la aplicación usará varios lenguajes de programación para su desarrollo y tendrá un diseño por capas y organización concreta del código.

Ejemplo de arquitectura web: LAMP

LAMP es una arquitectura web que hace referencia a Linux, Apache, MySQL y PHP. Es decir, se trata de una arquitectura que usa distintos elementos para el despliegue de aplicaciones web, como por ejemplo Apache, como servidor web, MySQL como sistema gestor de bases de datos, PHP como lenguaje de programación principal de parte del servidor y todo corriendo bajo un sistema operativo Linux. En el caso del lenguaje de programación, PHP es el más común, pero en esta arquitectura también se puede encontrar PERL y PYTHON.

Existe una ligera modificación en esta arquitectura, ya que si en vez de Linux como sistema operativo del servidor se usa Windows, se denominaría a la arquitectura WAMP, coincidiendo, eso sí, en el resto de componentes. Incluso, aunque menos utilizada, existe la arquitectura llamada MAMP, que usa los mismos componentes que las dos anteriores excepto que utiliza sistemas operativos Apple Macintosh, MAC OS X.

Ejemplo de arquitectura web: MEAN

MEAN es una arquitectura web, framework o conjunto de software para desarrollar aplicaciones web, que usa las tecnologías MongoDB, ExpressJS, AngularJS, NodeJS, que conforman el acrónimo del nombre. Estas cuatro tecnologías están basadas en el lenguaje de programación Javascript, a diferencia de LAMP no describe el sistema operativo usado.

MongoDB, es la base de datos NoSQL que nos permitirá almacenar información, NodeJS nos permite ejecutar Javascript del lado del servidor, es la plataforma encargada del funcionamiento del servidor. ExpressJS es el framework JavaScript Web del Servidor, es un módulo de NodeJS que maneja las solicitudes o peticiones que se hacen por medio de los métodos del protocolo HTTP (GET, POST, etc.). Y finalmente AngularJS, es el framework JavaScript del Cliente, es la plataforma que se usa para trabajar en el front end.

Otra arquitectura diferente: JAVA WEB (JVAEE)

JavaEE realmente es una plataforma derivada de Java SE, preparada para programar aplicaciones que hagan uso de módulos ejecutados en un servidor de aplicaciones, y que está organizada en varias capas para facilitar su comprensión, mantenimiento y desarrollo. Dentro de las distintas capas, la que nos interesa es la Capa Web, donde la aplicación hace uso de páginas JPS (JavaServer Page), que son páginas web creadas por el servidor web, en las que parte de su código se ejecuta como si fuera un servlet y otras partes pueden ser texto estático para la visualización de los datos. Para desplegar y correr JavaServer Pages, se requiere un servidor web compatible con contenedores servlet como Apache Tomcat o Jetty. Los servlet son a pequeños programas que se ejecutan en el servidor web y amplían sus funcionalidades.

SOBRE FRAMEWORK

Bootstrap, anteriormente conocido como Blueprint, es el framework que he elegido para comentar. Lo conocí gracias a cursar la asignatura Diseño de Interfaces del Ciclo de Grado Superior de Aplicaciones Web, dónde se explicó en el tema 9. Fue creado por la compañía Twitter para desarrollar interfaces para sus aplicaciones internas, pero vieron potencial de uso en su estandarización en la construcción web a base de patrones de diseño.

Bootstrap es un kit de herramientas de código abierto para desarrollar sitios y aplicaciones web con HTML, CSS y JS. Es importante destacar que a diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end, que es precisamente para lo que se usó en la asignatura de Diseño de Interfaces.

Gracias a las clases ya construidas y programadas, simplemente asociando las librerías de CSS y JS a nuestra código, podemos hacer uso de todas las plantillas de diseño que contienen tablas, botones, tipografías, formularios, iconos, galerías de imágenes, menús de navegación y otros muchos elementos y efectos. Bootstrap básicamente consiste en una serie de hojas de estilo LESS (también se vio en la asignatura de Diseño de Interfaces) que implementan la variedad de componentes descritos anteriormente.

Es un framework ideal para el diseño responsivo de aplicaciones web, ya que haciendo uso de sus librerías, se puede adaptar cualquier sitio a dispositivos de diferentes pantallas y resoluciones de una forma sencilla y rápida, mostrando u ocultando contenido según convenga. Es lo que más me llamó la atención de este framework, ya que configura cualquier sitio o aplicación web con una estructura horizontal (una fila) de 12 columnas, lo que llaman el modelo de rejilla, en la que cada una de las divisiones (elemento div) de la página web dónde habrá un contenido, podrá ocupar la totalidad de columnas o mostrarse en menos, para variar la forma en visualizarse en función del dispositivo. Y por supuesto, también de una forma sencilla, se podrá indicar en cada división si se ocultará en alguna resolución.

Ejemplo de código: `<div class="col-lg-12 col-md-12 col-sm-12 col-xs-6"> nuestro bloque</div>`

Con esta simple línea de código, indicamos que el contenido de ese bloque se mostrará en todo el ancho de la pantalla ocupando las 12 columnas, en todas las resoluciones de dispositivo excepto en la más pequeña, que ocupará sólo la mitad.

Las sigas LG corresponden a resoluciones de más de 992px, MD a resoluciones de más de 768px, SM a resoluciones de más de 576px y XS, la más pequeña, a resoluciones de menos de 576px.

Tarea 01_2. Desarrollo Web en Entorno Servidor

Fernando Cebrián Carreras

CFGs DESARROLLO DE APLICACIONES WEB