

IMPLANTACIÓN DE ARQUITECTURAS WEB	1
La arquitectura web y algunos modelos	1
Un modelo simple para el despliegue de aplicaciones Web	3
Qué es una aplicación web.....	4
Fases de un proyecto de aplicación web	4
Aplicaciones Web Vs Escritorio	6

IMPLANTACIÓN DE ARQUITECTURAS WEB

La arquitectura web y algunos modelos

Una aplicación web, o web en general necesita de una estructura que permita su acceso desde diferentes lugares (máquinas). Esta estructura es lo que se denomina Arquitectura Web (realmente este nombre se da también al diseño de toda la estructura).



¿Diferencia entre página web y aplicación web?

La gran mayoría de las arquitecturas web en la actualidad se basan en un modelo **cliente/servidor**: una comunicación asimétrica en la que uno de los extremos ofrece uno o más servicios y el otro hace uso de él. No hay que olvidar otros modelos como **P2P (peer-to-peer)**, **B2B (business to business)**, etc.

El término **servicio** es muy amplio y muchas veces confuso. Por ejemplo se puede considerar una web a la que acudimos a comprar productos un servicio en sí misma, pero a la vez dicho servicio está compuesto de servicios de seguridad, de sesión, de transacciones, etcétera.

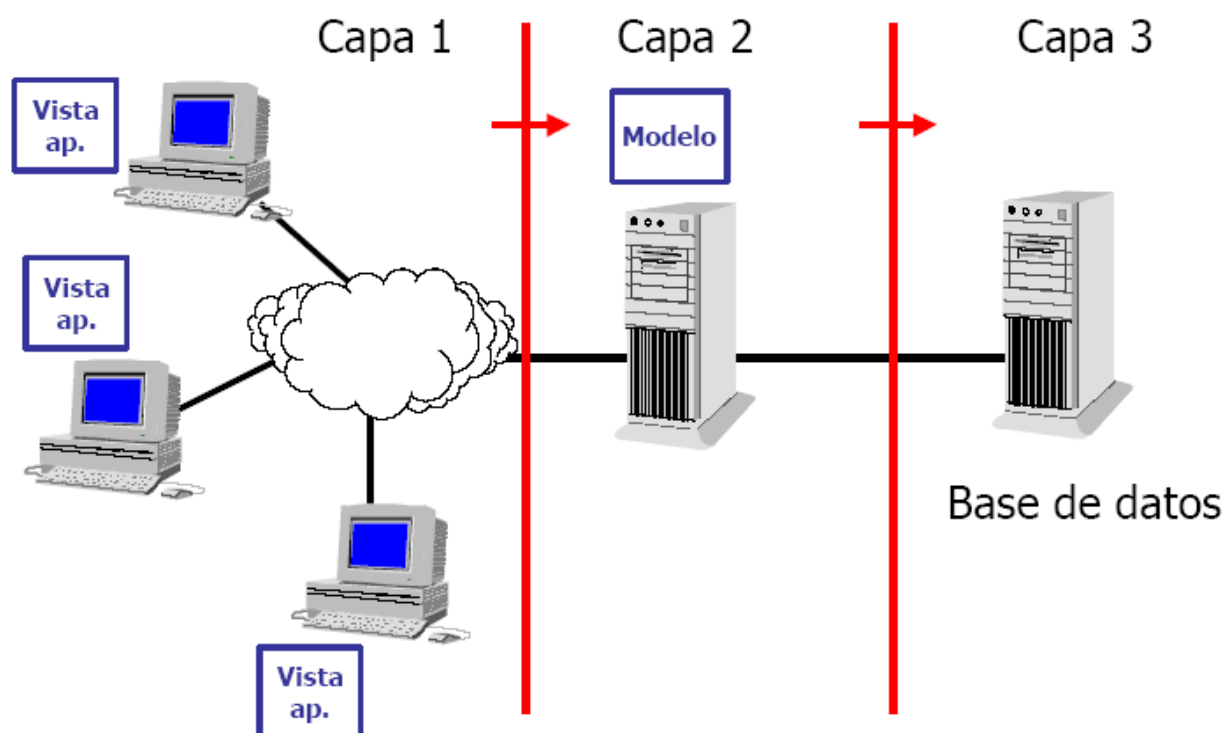
La estructura de una Arquitectura Web actual sigue el siguiente **modelo de tres capas**:

1. Una **capa cliente**: es generalmente el navegador Web ejecutándose en el ordenador del usuario final. Existen otras opciones más básicas pero en la actualidad la potencia y diversidad de los navegadores existentes (así como su gratuidad) han relegado las demás opciones a la práctica desaparición.
2. Un servidor Web (**capa de negocio**): La capa cliente puede acceder a diferente lógica y procedimientos que existen en la capa de negocio. Aquí la lógica puede ser mucho más compleja que en la capa anterior. Los componentes de esta capa pueden ser desde simples archivos HTML hasta Servlets de Java. Existen muchas tecnologías que pueden usarse en este nivel: por ejemplo scripting web como PHP, ASP o JSP a lenguajes de programación como TCL, CORBA y PERL.
3. Una **capa de datos**: Se compone de un sistema de almacenamiento acceso a datos que se utilizan para confeccionar la página Web. Generalmente es un gestor de bases de datos relacionales (SGDB) pero pueden ser ficheros de texto plano, ficheros XML, etc. Una opción

cada vez más usada es la creación de ficheros XML a partir de datos almacenados en una base de datos y su presentación mediante por ejemplo XSLT.

La capa de negocio puede estar a su vez dividida en dos partes si el sistema es suficientemente grande o complejo. Puede dividirse en una capa de **presentación** y una capa de **lógica de negocio**.

- La **capa de presentación** se encarga de componer las páginas integrando la parte dinámica en la estática. Además también procesa las páginas que envía el cliente (por ejemplo datos en formularios). Algunas soluciones para esta subcapa son los ASP de Microsoft o los JSP de Java. Esta parte la realiza generalmente un servidor web.
- La **capa de lógica de negocio** lleva a cabo operaciones más complejas. Se corresponde con la implantación de un servidor de aplicaciones. Realiza muchos tipos de operaciones entre los que destacan:
 - Realizar todas las operaciones y validaciones. o Gestionar el flujo de trabajo (workflow) incluyendo el control y gestión de las sesiones y los datos que se necesitan.
 - Gestionar todas las operaciones de accesos a datos desde la capa de presentación.



En el caso de estar usando páginas **web estáticas** (no cambian en función de diversas variables) no existiría la capa de datos ya que estos van incorporados en los propios archivos de marcas que serán las conforman las páginas web.

¿Diferencia entre página web estática y dinámica?

Este supuesto es cada vez menos común. Debido a la introducción de dinamismo en las páginas, la estructura vista anteriormente se ha visto alterada sensiblemente:

- Los navegadores Web son capaces de interpretar diferentes elementos dinámicos autónomamente o mediante plugins (javascript, flash, etc.)
- Los servidores Web también pueden interpretar código para generar las páginas web. Así se pueden introducir pequeños programas que alteren el contenido o aspecto final de una página web dependiendo de diferentes elementos como el usuario que accede o la información solicitada en cada momento. El servidor web necesita de algún módulo adicional para poder interpretar este código. Generalmente se empotra en el propio servidor web para lenguajes de script o se incorpora en un servidor aparte (de aplicaciones) para los lenguajes más potentes. Algunos lenguajes que típicamente se usan en las páginas dinámicas en el servidor son PHP, Python, Ruby o Java. Estos lenguajes también permiten el acceso a la capa de datos y la intercalación de estos datos entre los elementos de la página final.

Un ejemplo del **modelo completo** estaría compuesto por un servidor Apache y un Tomcat que se conecte a una base de datos. Un ejemplo del simplificado sería un servidor **LAMP**.

¿LAMP?



A pesar de que el modelo Cliente/Servidor es el más extendido, el W3C describe cuatro modelos de arquitectura de servicios web. Estos conceptos los veremos por encima por no tener una gran influencia en el resto del curso:

Un modelo simple para el despliegue de aplicaciones Web

En la actualidad la mayor parte de la información y lógica de un negocio debe ser accesible desde diferentes lugares. Aquí entran en juego las aplicaciones web.

Todos pensamos en tiendas online como modelo de una aplicación web, pero hay otros muchos como por ejemplo una aplicación de compra venta de activos entre dos bancos en el sector entre negocios (B2B).



Se puede imaginar inmediatamente que la **seguridad** es un aspecto clave en este tipo de aplicaciones, pero no el único. Muchas veces la **velocidad** y **estabilidad** de la comunicación y del servicio en sí pueden ser tanto o más críticas.

Cuando uno va a desarrollar e implantar una aplicación web debe tener en cuenta varios factores. Lo primero debe ser hacerse una **idea general de la aplicación** y de las diferentes **soluciones** que podemos utilizar. Se deben tener en cuenta las tres capas. Un error muy común es el uso de un único conjunto de tecnologías constantemente. Por supuesto conocer una tecnología es un punto a favor de su uso, pero muchas veces vamos a llevar a cabo una solución manifiestamente mejor únicamente por no haber considerado usar otras y afrontar su aprendizaje.

El siguiente aspecto a considerar sería el **coste**. Cuánto nos va a costar y qué presupuesto tenemos.

Estos factores hay que considerarlos **antes de firmar** ningún contrato e incluso antes de dar un presupuesto aunque sea orientativo.

Por ejemplo una compañía de venta de vehículos industriales quiere una aplicación web para publicar sus datos de ventas y que los comerciales puedan acceder a ella remotamente. Necesitaremos una base de datos en la que se almacenen los diferentes vehículos y sus ventas. También hará falta una lógica que mantenga todo el sistema actualizado y permita modificaciones. Además necesitaremos una capa de cliente con autenticación para que los diferentes vendedores puedan acceder al sistema, consultar y actualizar los datos.

Después de la evaluación se puede decidir no afrontar el proyecto por muchos motivos. Además de los costes ya mencionados podría darse el caso de que no tengamos los conocimientos o la infraestructura para llevar a cabo el proyecto.

Qué es una aplicación web

Es una aplicación que se va a ejecutar a través de internet. Constará de dos partes (al menos) una en el lado servidor y otra que se ejecutará en la máquina del cliente en un navegador web. Las aplicaciones web se engloban en el concepto superior de aplicaciones distribuidas. El servidor pone a disposición del cliente diferentes recursos. Ejemplos de aplicaciones web son el correo electrónico web, las tiendas online, las redes sociales, etc.

Fases de un proyecto de aplicación web

Se pueden considerar cuatro fases en el proyecto:

1. **Concepto:** Durante esta fase se debe obtener una idea clara y concreta de qué quiere el cliente. Además hay que obtener una idea general de cómo se llevará a cabo y de si es viable o no. Hay que determinar las limitaciones reales con que nos podemos encontrar. Por ejemplo la conexión a internet existente en la zona puede no ser suficiente para obtener los resultados previstos. Otro ejemplo de problema pueda ser que la tecnología necesaria sea demasiado cara. Es vital que al terminar esta fase se tenga una documentación que defina claramente los límites y objetivos del proyecto.
2. **Diseño:** Esta fase se centra en responder a cómo haremos la aplicación. Hay que concretar las tecnologías (tanto software como hardware) que usaremos y cómo se van a comunicar entre ellas. También hay que determinar los distintos módulos que usaremos y sus interfaces. Es muy importante realizar un plan de proyecto realista en el que se dividan las tareas y responsabilidades y se calculen los tiempos para cada elemento así como su secuencia y dependencias. También hay que obtener una especificación funcional en la que se detallen tanto el funcionamiento como el flujo de la aplicación.
3. **Desarrollo:** En esta fase se debe desarrollar el proyecto en sí. Es muy importante llevar a cabo pruebas tanto unitarias como de integración así como gestionar una documentación del desarrollo y un control de versiones.
4. **Pruebas e implantación:** Cuando el proyecto está totalmente terminado es necesario probarlo intensivamente antes de ponerlo en producción. Es necesario tener en cuenta tanto nuestra aplicación como su comunicación con otros sistemas informáticos. Cuanto más se parezca el sistema de pruebas al real mejor. El último paso es la instalación y puesta en marcha del sistema. Es un momento crítico.

Una fase común a todos los proyectos informáticos y que no se incluye aquí es el **mantenimiento**. Este concepto incluye dos partes. El mantenimiento del servicio y corrección de errores y las mejoras. La primera consiste en asegurarse de que todo sigue funcionando y solucionar los posibles errores y “caídas” del servicio.

El **orden** correcto para el desarrollo es empezar de abajo a arriba. Es decir, primero la capa de datos, luego la de negocio y por último la presentación al cliente. Muchas veces se tiene la tentación de hacerlo al revés. Esto es debido a que no se han identificado bien las necesidades y objetivos de proyecto o no se ha realizado un diseño concreto. Es un error que nos conducirá a muchas más modificaciones y errores en nuestra aplicación.

Aplicaciones Web Vs Escritorio

Las diferencias entre una aplicación web y una de escritorio tiene ventajas e inconvenientes.

Ventajas de las aplicaciones web:

- No es necesario instalarlas en aquellos equipos en que se vayan a utilizar.
- Se instalan y se ejecutan solamente en un equipo, en el servidor.
- Esto permite uso simultáneo en diferentes equipos.
- Centralización de la gestión: hacer copias de seguridad de los datos, corrección de errores, actualización.
- Independencia del SO y de la arquitectura de la máquina (procesador, potencia, ...)
- Generalmente el cuello de botella lo presenta el equipo, no la máquina.
- Se pueden utilizar desde cualquier lugar en el que dispongamos de conexión con el servidor.

Inconvenientes de las aplicaciones Web:

- El interface de usuario de las aplicaciones web es la página que se muestra en el navegador.
- Esto restringe las características del interface a aquellas de una página web.
- Dependemos de una conexión con el servidor para poder utilizarlas.
- Si nos falla la conexión, no podremos acceder a la aplicación web.
- La información que se muestra en el navegador debe transmitirse desde el servidor.
- Esto hace que cierto tipo de aplicaciones no sean adecuadas para su implementación.
- Por ejemplo, las aplicaciones que manejan contenido multimedia, como las de edición de vídeo.
- Al viajar la información por la red, existe un problema extra de seguridad.

Investigar el significado y poner ejemplos de los siguientes conceptos:

- Plataforma web
- Plataforma de desarrollo web
- Entornos de desarrollo web.

- <https://sites.google.com/site/sergiocuesta>
- http://es.wikieducator.org/Usuario:ManuelRomero/micursos/DWES/Tema1/Aplicaciones_Web#Aplicaciones_Web_Vs_Escritorio