

UNIDAD 1 Tarea 1.- Investigar los temas y conceptos claves sobre aplicaciones web, como su evolución, arquitectura y planificación y realiza un resumen.

Evolución de las aplicaciones Web

La web 1.0

La Web 1.0 (1991-2003) es la forma más básica que existe, con navegadores de sólo texto bastante rápidos ya que es de sólo lectura. El usuario no puede interactuar con el contenido de la página (nada de comentarios, respuestas, citas, etc), estando totalmente limitado a lo que el Webmaster sube a ésta.

Algunos elementos de diseño típicos de un sitio Web 1.0 incluyen:

- Páginas estáticas en vez de dinámicas por el usuario que la visita.
- El uso de framesets o Marcos.
- Extensiones propias del HTML como <blink> y <marquee>, etiquetas introducidas durante la guerra de navegadores web.
- Botones GIF, casi siempre a una resolución típica de 88x31 pixels en tamaño promocionando navegadores web u otros productos.
- Formularios HTML enviados vía email.
- No se podían adherir comentarios ni nada parecido
- Todas sus páginas se creaban de forma fija y muy pocas veces se actualizaban.

La web 2.0

El término Web 2.0 comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web estáticos donde los usuarios se limitan a la observación pasiva de los contenidos que se han creado para ellos.

Servicios asociados:

- Blogs: Un blog es un espacio web personal en el que su autor puede escribir cronológicamente artículos, noticias... (con imágenes y enlaces).
- Wikis: Una wiki es un espacio web corporativo, organizado mediante una estructura hipertextual de páginas donde varias personas elaboran contenidos de manera asíncrona.
- Redes sociales: Sitios web donde cada usuario tiene una página donde publica contenidos y se comunica con otros usuarios. Ejemplos: Facebook, Twitter, etc.
- Documentos: Google Drive y Office Web Apps (SkyDrive), en los cuales podemos subir nuestros documentos, compartirlos y modificarlos.
- Videos: Youtube, Vimeo, Dailymotion, Dalealplay... Contienen miles de vídeos subidos y compartidos por los usuarios.

La web 3.0

Web 3.0 es una expresión que se utiliza para describir la evolución del uso y la interacción de las personas en internet a través de diferentes formas entre las que se incluyen la transformación de la red en una base de datos, un movimiento social hacia crear contenidos accesibles por múltiples aplicaciones non-browser, el empuje de las tecnologías de inteligencia artificial, la web semántica, la Web Geoespacial o la Web 3D.

Se basa en la idea de añadir metadatos semánticos y ontológicos a la World Wide Web. Esas informaciones adicionales (que describen el contenido, el significado y la relación de los datos) se deben proporcionar de manera formal, para que así sea posible evaluarlas automáticamente por máquinas de procesamiento. El objetivo es mejorar Internet ampliando la interoperabilidad entre los sistemas informáticos usando "agentes inteligentes". Agentes inteligentes son programas en las computadoras que buscan información sin operadores humanos. Con la web 3.0 se busca que los usuarios puedan conectarse desde cualquier lugar, cualquier dispositivo y a cualquier momento.

Entre sus innovaciones destacan:

- Bases de datos
- Inteligencia artificial
- Web semántica y SOA
- Evolución al 3D

La web 4.0

Las aplicaciones ya no estarán en nuestras PC's, estarán en la Internet y por ende en todos lados. Pasaremos de una red "tonta" a una red "inteligente" donde el objetivo primordial será el de unir las inteligencias donde tanto las personas como las cosas se comuniquen entre sí para generar la toma de decisiones.

Arquitectura de las Aplicaciones Web

La arquitectura de aplicaciones web es un esquema de cómo interactúan entre sí los distintos componentes de tu aplicación web.

Puede ser tan simple como definir la relación entre el cliente y el servidor. También puede ser tan compleja como definir las interrelaciones entre un enjambre de servidores backend en contenedores, equilibradores de carga, pasarelas API y frontends de una sola página orientados al usuario.

La forma de diseñar tu aplicación web desempeña un papel clave tanto en su usabilidad como en la optimización de sus costes.

La arquitectura de un Sitio Web tiene tres componentes principales:

- Un servidor Web
- Una conexión de red
- Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó.

Las aplicaciones Web están basadas en el modelo Cliente/Servidor que gestionan servidores web, y que utilizan como interfaz páginas web.

Las páginas Web son el componente principal de una aplicación o sitio Web. Los browsers piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web. En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página.

Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe, es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente). Cuando el browser ejecuta un script en el cliente, éste no tiene acceso directo a los recursos del servidor. Hay otros componentes que no son scripts, como los applets (una aplicación especial que se ejecuta dentro de un navegador) o los componentes ActiveX. Los scripts del cliente son por lo general código JavaScript o VBScript, mezclados con código HTML.

Planificación de Aplicaciones Web

Planificar una aplicación web implica pensar y definir muchas cuestiones de importancia:

1. Planificar el futuro

Puede sonar un poco duro que nos tengamos planear el futuro de algo que aún no existe pero es así, de una buena planificación depende la dificultad en un futuro de realizar cambios en la plataforma que sostiene nuestra aplicación.

2. Documentación

Documenta todo lo que puedas tu aplicación en especial aquellas partes de la misma que hayan requerido un mayor tiempo de desarrollo, en un futuro cercano te librarás de más de un quebradero de cabeza.

3. No empieces por los pequeños detalles

A menudo es más sencillo plantearse el desarrollo de pequeños servicios cómo pueden ser los RSS o una galería de imágenes, pero es fundamental comenzar por lo que vendrá a ser la base de nuestra aplicación, de este modo podremos ir completando nuestra aplicación en un orden lógico, primero los cimientos.

4. Diseño de la interfaz

Es importante conseguir una interfaz que se atractiva y útil para la gran mayoría de los usuarios, lo más importante es llegar a conectar con el usuario y la mejor manera es ofrecerle la mayor cantidad de información posible para que se llegue a sentir parte del proyecto pero todo esto muchas veces no lo llegamos a conseguir por una mala elección en la combinación de los colores que se integran en nuestra interfaz.

5. Evita construirte librerías o frameworks innecesarios

Muchas personas siempre sostienen que es mejor que cada desarrollador utilice sus propias librerías, pero la realidad es que en muchas ocasiones esto supone una gran pérdida de tiempo, es mejor buscar una librería o framework que se parezca lo más posible a lo deseado que desarrollarlo por ti mismo.

6. Ten un plan de proyecto

Asegúrate de que cada desarrollador que trabaje en el proyecto sepa qué es lo que debe de hacer y cuándo tiene que hacerlo, teniendo todo esto redactado en un documento te ayudará a identificar problemas.

7. Elige el lenguaje adecuado

Ya sabes que hay numerosas discusiones sobre ¿Cuál es el mejor lenguaje de programación?, la respuesta a menudo es depende de para que quieras utilizarlo, por eso asegúrate de escoger el lenguaje adecuado.

8. Evita las distracciones

Optimiza tu tiempo, cuando estés trabajando que nada te distraiga, ni llamadas telefónicas, ni clientes de mensajerías instantáneas, proponte un buen número de horas al día de trabajo aislado dedicado al proyecto.

9. Sigue un buen control de calidad

Muchas veces nos equivocamos al planificar los tiempos de proyecto y esto lleva a menudo a los desarrolladores a tener que escribir “código sucio”, esta es una buena opción si lo que quieres es una demo de la aplicación funcionando, pero la actitud te traerá problemas en el futuro.

10. Plantéate los pasos a seguir después del desarrollo

Ten en cuenta que esta fase de desarrollo algún día deberá acabar, estudia con qué frecuencia se deberá actualizar y que se necesita para que este funcione.