



NOMBRE DE LA MATERIA
Programación Web

NOMBRE DEL DOCENTE
Prado López Efrén Emmanuel

NOMBRE DEL TRABAJO
Actividad 1

NOMBRE DEL ALUMNO
Alejandro Guevara de Luna

UNIDAD
1

FECHA Y LUGAR
26 de enero del 2023





Actividad 1

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

Evolución:

Web 1.0:

Esta fue creada entre 1991 y 2003 fue la primera forma más básica que existe, con navegadores de solo textos bastante rápidos, 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.

Algunas características más normales en la Web 1.0 eran:

- Páginas dinámicas
- Uso de Framesets o Marcos
- Extensiones de HTML como 'Blink' y 'Marquee'
- Libros de visitas Online
- Formularios HTML vía email.
- Páginas Fijas

Web 2.0:

Se desconoce el año de dicha web pero dentro de sus hazañas estaba compartir información fácil, la interoperabilidad, diseño para el usuario, y la famosa World Wide Web.

Servicios asociados de la web 2.0:

- Blogs
- Wikis
- Redes sociales
- Videos
- Fotos
- Almacenamiento





Web 3.0:

Esta web se utiliza más como una expresión para denotar la evolución y la interacción de las personas en internet a través de diferentes formas entre las que se incluye la transformación de la red en una base de datos.

Innovaciones de la web 3.0:

- Base de datos
- Inteligencia Artificial
- Web semántica y SOA
- Evolución 3D

Web 4.0:

Es la era que conocemos actualmente, las aplicaciones ya no están en nuestra PC, están en internet, lo que significa acceder por todos lados. La red se volvió inteligente donde el objetivo es unir las inteligencias donde tanto las personas se comunican entre sí para generar decisiones.

Arquitectura:

Una aplicación Web es proporcionada por un servidor Web y utilizada por usuarios que se conectan desde cualquier punto vía clientes Web (browsers o navegadores). La arquitectura de un Sitio Web tiene tres componentes principales:

- Un Servidor Web
- 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.

La colección de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario.

Planificación:

A menudo antes de empezar un proyecto no tenemos claro cuál es el objetivo del mismo o qué pretendemos alcanzar por eso esta pequeña lista te puede ayudar a la hora de planificar un proyecto.

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, que pasa si en un futuro tuviéramos problemas de escalabilidad, y sí tuviéramos que migrar las bases de datos de MySQL a Oracle, 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

Esta entrada puede que te ayude en esta parte, es importante conseguir un buen diseño para nuestra interfaz, es importante conseguir un interfaz que se atractivo 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 esa actitud te traerá problemas en el futuro.

10. Plántate 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.

1.1 Evolución de las aplicaciones Web - Programación WEB. (n.d.).

<https://sites.google.com/site/programacionwebhegm/unidad-1-arquitectura/1-1-evolucion-de-las-aplicaciones-web>

