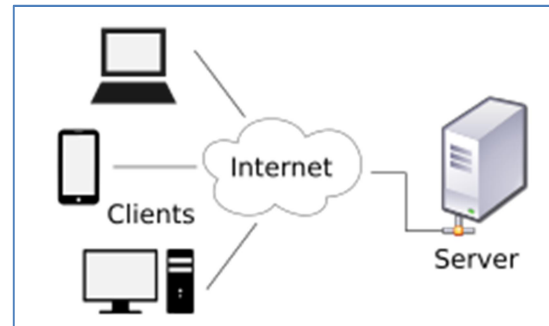

Tema 1.1 Evolución de las aplicaciones web

Objetivos:

Definir que es una aplicación web.

Cómo ha evolucionado a través de la historia



Una Aplicación Web o también llamada web app, es una aplicación **Cliente-Servidor** en la cual el cliente o la interfaz de usuario la proporciona un navegador web.

Dentro de las aplicaciones web podemos incluir el webmail, wikis, mensajería instantánea y prácticamente todo lo que funciona a través de un navegador.

En un principio, cada página web no era sino un documento estático en donde el usuario podía navegar entre documentos gracias a las bondades del hiper texto. El precio a pagar es que con cada clic se tenía que visitar al servidor para acceder a un nuevo recurso.

En 1995, Netscape introduce un lenguaje de script del lado del cliente llamado JavaScript con el cual el documento podía hacerse dinámico sin necesidad de acceder al servidor. Es decir, el usuario podía interactuar con el documento localmente.

En 1996, Macromedia introduce Flash, el cual permitió agregar a los navegadores animaciones embebidas en las páginas web.

Para 1999 aparece el concepto de “web application”, utilizando lenguaje Java.

En 2005 se potencia la programación del lado del cliente utilizando la tecnología Ajax, esto es, se pueden intercambiar datos con el servidor sin descargar una página web.

En 2011 se libera HTML5 el cual ya provee de forma nativa capacidades gráficas y multimedia sin necesidad de plug-ins. Incluye también otras características que permiten la creación de aplicaciones web enriquecidas.

Construir una tabla de fechas, personajes, aportaciones de la evolución de la web

Tema 1.2 Arquitectura de las aplicaciones web

Objetivos.

Cuál es la arquitectura que caracteriza las aplicaciones web.

Arquitectura web tiene que ver con la disposición y estructura de los contenidos en una página web. En otras palabras, todo lo que tiene que ver con cómo llegar a la experiencia final del usuario con los contenidos web y lo que se necesita para llegar ahí.

¿Qué debe suceder para que una aplicación web funcione?

Una aplicación web es una Aplicación Cliente-Servidor, dicho de otra manera, es una Aplicación Distribuida que divide las tareas entre los proveedores de recursos y servicios llamados Servidores y los que requieren de los servicios y recursos llamados Clientes.

El servidor de aplicaciones literalmente ejecuta varias aplicaciones en el Servidor y genera el recurso solicitado por el cliente. Un Cliente no comparte recursos, pero si inicializa la sesión con los servidores.

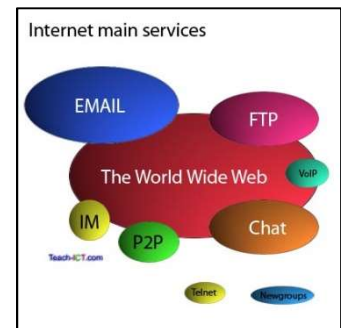
Algunas aplicaciones del tipo Cliente-Servidor son el Outlook, Netflix, Whatsapp, Google Earth, etc. Un servicio de internet llamado **World Wide Web** también es una aplicación cliente servidor con la característica especial que utiliza un navegador como cliente. Las aplicaciones que funcionan en base a este servicio se

les conoce como aplicaciones web. Existe una versión web de Netflix, de Whatsapp, de Microsoft Word, etc.

Al contrario de sus inicios, las aplicaciones web utilizan documentos web escritos en un formato estándar en su mayoría HTML y JavaScript los cuales están soportados por la mayoría de los navegadores web. Los documentos son descargados por los clientes utilizando procedimientos ya definidos en su mayoría a través del protocolo HTTP. Las actualizaciones del cliente se pueden dar cada vez que el cliente accede al servidor.

La World Wide Web, o simplemente la Web, es un medio en donde los usuarios pueden leer y escribir a través de computadores conectadas al **internet**. Es decir, la Web es uno de los tantos servicios que tiene Internet.

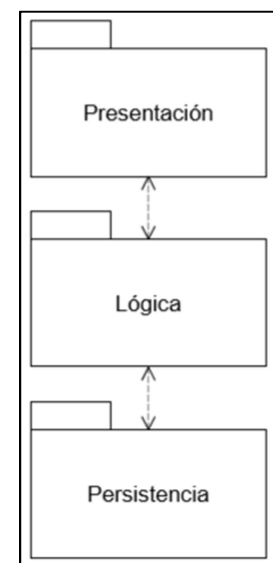
Internet es una red global de computadoras que utilizan el conjunto de protocolos **TCP/IP** y es a través de internet que se mueven diferentes servicios tales como el WWW, mail, telefonía IP, etc.



El navegador Web nos permite realizar peticiones a una capa “intermedia” la cual oferta servicios para cubrir dichas peticiones. Una tercera capa permite hacer consultas y actualizaciones a la base de datos.

Aunque la arquitectura de una aplicación web se puede entender solamente con dos capas, es decir, la capa del lado del cliente (frontend) y la capa del lado del servidor (backend), actualmente muchas aplicaciones web se dividen lógicamente en más capas tanto del lado del cliente como del lado del servidor.

Dicho lo anterior podemos distinguir claramente tres capas para toda aplicación web:



Capa de presentación (frontend)

- Es responsable de la presentación de los datos, recibiendo los eventos de los usuarios y controlando la interfaz de usuario.

Capa de lógica de negocios o de control (middleware)

- Los objetos de negocios que implementan las reglas de negocios “viven” aquí, y están disponibles para la capa de presentación
- Protege del acceso directo a la información desde la capa de presentación

Capa de persistencia (backend)

- Es la responsable del acceso y almacenamiento de los datos.

Tema 1.3 Tecnologías para el desarrollo de aplicaciones web

Objetivo. Conocer las principales tecnologías y su papel en el desarrollo de aplicaciones web.

Cada capa utiliza diferentes tecnologías las cuales han ido cambiando a través de los años.

Así para el Frontend encontramos tecnologías como: JavaScript, Flash o Silverlight, CSS, HTML.

Para el backend encontramos otras como: PHP, ASP, Java, Ruby, Python, etc.

Sin olvidarnos de los frameworks que implementan librerías con las tecnologías ya citadas, los más destacados JQuery, Bootstrap, Modernizr para el cliente y para el servidor Symfony, Laravel, Codeigniter, Node.js, Ruby on Rails [2].

También es importante considerar los gestores de contenidos los cuales son aplicaciones web prediseñadas y muy fáciles de implantar.

Los CMS (Content Management System) permiten crear, administrar y dar mantenimiento a un sitio web, generando soluciones sin necesidad de tener que tocar una línea de código [3].

Los CMS más representativos de la web nacieron entre el 2001 y el 2005 (Drupal, Wordpress y Joomla).

Los CMS por su propósito los podemos clasificar en:

- eCommerce para crear y administrar tiendas en línea.
- Foros para crear y administrar foros de discusión
- eLearning para crear y administrar sitios web de formación
- Wikis, CMS para generar wikis de contenidos

Algunos gestores de contenidos más utilizados son:

- Wordpress
- Joomla
- Drupal
- Squarespace
- Shopify
- Magento
- Wix
- Blogger
- PrestaShop

Tema 1.4 Planificación de aplicaciones web

Objetivo. Conocer las etapas para el desarrollo de aplicaciones web

Diferenciar la planificación de las aplicaciones web de la planificación para aplicaciones de escritorio y aplicaciones móviles.

Las etapas para llevar a cabo aplicaciones web suele variar dependiendo de varios factores:

- El tipo de proyecto
- Los recursos disponibles
- Las políticas y condiciones del cliente
- Las políticas y capacidades del equipo de desarrollo

Considerando lo anterior, es buena idea dividir el proyecto en varias etapas, planificarlas, evaluarlas y darles seguimiento:

- Análisis, es importante definir el proyecto, es decir qué es lo que vamos a resolver, que necesidades vamos a cubrir con nuestra hipotética solución, si la hay. Modelar el problema nos dará una mejor perspectiva del mismo. Algunas herramientas nos pueden apoyar en este propósito:
 - Diagrama conceptual
 - Casos de Uso (UML)
 - Diagramas de actividad (UML)
 - Diagrama BPMN
- Diseño. Definimos el esqueleto de manera precisa y clara. Se definen los elementos que el usuario final apreciará, los elementos de la lógica de negocio y del modelo donde estarán los datos. Algunas herramientas que nos pueden ser útiles son las siguientes:
 - Frontend(Photoshop, Gimp, Power point)
 - Middleware (Diagrama de clases UML, Diagrama de Flujo)
 - Backend (Diagrama EER)
- Maquetación. Podemos armar toda la parte estructural del frontend ajustándolo a las necesidades y gustos del usuario final. Herramientas importantes para esta etapa son las siguientes:
 - HTML/CSS
 - Dreamweaver
 - Aptana
 - Amaya
- Implementación cliente. Validaciones y comportamiento de la aplicación desde la perspectiva del cliente.
 - Javascript
 - Flash
 - Silverlight
 - Frameworks como JQuery, Bootstrap, AngularJS, etc.
- Implementación servidor. Respetando el diseño, desarrollamos la lógica de negocios.
 - PHP
 - Python
 - ASP
 - Java
 - Ruby
 - Frameworks: Codeingiter, Symfony, Laravel, NodeJS

- Workbench, SqlYog, PHPMYAdmin, MySQL (MariaDB), etc.
- Implantación local: Realizar los cambios necesarios en un ambiente controlado para realizar todo tipo de pruebas a nuestra aplicación.
 - Apache
 - IIS
 - GlassFish
 - Suites (Wamp, Lamp, Mamp, Xamp)
 - Software para control de versiones
 - Subversion
 - Git
- Implantación hosting. Extender las pruebas de la aplicación al usuario final.
 - FTP, SSH
 - Nginx
 - Google Web Server
 - Microsoft IIS
 - Apache
- Soporte (ticketing software). Administrar las fallas de la aplicación y de los usuarios y resolverlas.
 - Bugzilla
 - DotProject
 - Zendesk
 - SysAid
 - Helpdesk
- Mantenimiento. Actualizar la aplicación repitiendo las etapas necesarias.