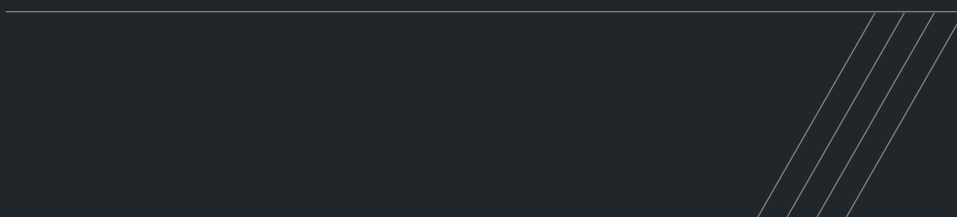


INNOVARE SUITE

FUNDAMENTOS WEB

Parte 1



FULL STACK DEVELOPER JUNIOR

La Web – Fundamentos

¿Cómo funciona la web?

Si bien el podemos crear una página html estática, guardarla en nuestra computadora y visualizarla en el navegador web, el concepto de web requiere un componente esencial que oficia como alojamiento del contenido en una web, y sirve este contenido de forma remota a través de las direcciones URL (Universal Resource Locator): el servidor web. Si bien podemos ver contenido html sin necesidad de un servidor web, el no tener un servidor web que canalice el contenido de un sitio web a través de la red nos impide, entre otras cosas:

- Mostrar contenido dinámico, obtenido desde una base de datos o peticiones a webservices.
- Ejecutar código específico realizado en tecnologías PHP, .NET, Python, entre otras.
- Establecer sistemas de seguridad contra ataques remotos.
- Integrar bases de datos remotas para alojar contenido que alimenta nuestra web.
- Redireccionar el contenido a un dominio.

Servidor Web:

Un servidor web o web server es una aplicación que procesa el contenido, usualmente páginas web, del lado del servidor. Realiza conexiones uni/bidireccionales de forma sincrónica o asincrónica con la(s) máquina(s) cliente, entregando o recibiendo respuestas. Comúnmente un servidor web utiliza el protocolo HTTP para enviar el contenido html a cada cliente que lo solicita. Son instalados y utilizados en servidores dedicados para tal fin, y se interconectan con la red internet, a través de redireccionamiento IP, entre IP públicas y privadas.

Para el caso de servidores web que se ejecutan en empresas, a modo de Intranet, se suele utilizar lo que conocemos como Personal Web Server. Dentro de los servidores más comunes en el mercado, podemos destacar a Microsoft IIS (Internet Information Services), Apache web Server, y NGinx (Engine-ex), entre otros.

URL:

Proviene de la sigla Universal Resource Locator. Su traducción al español es básicamente literal (Localizador de recursos uniformes), y es la tecnología que se ocupa de enlazar recursos, usualmente a través de la web o intranets, que se mueven a través de direcciones IP o nombres de dominios. La URL se ocupa de “volver amigable” a través de una dirección web, el acceso a un recurso específico, como

puede ser una página html, un documento PDF, una imagen, un video, o una aplicación para ser descargada de Internet.

Por ejemplo, cuando accedemos a un sitio web, simplemente tipiamos en el navegador web la dirección www.midominio.com, y visualizamos la página principal de dicho sitio. Esta página es un archivo html que contiene por ejemplo la información institucional de una empresa. La página html está alojada en un disco rígido físico de una computadora remota, bajo una ruta local que puede ser d:\webs\midominio\www\index.html. El web server, a través de una URL, se ocupa de rutear este contenido de manera transparente para el usuario que lo solicitó, para que éste simplemente escriba el dominio al que desea acceder, y acceda al contenido principal, sin tener que escribir por completo toda la URL.

Páginas html:

Las páginas html son las que alojan el contenido general que compone cada texto. Estas páginas están armadas con contenido HTML, contenido CSS, JavaScript, imágenes, videos, etc. Todo este contenido es armado de forma óptima y estática, para que pueda ser visualizado a través de navegadores. En los casos donde los contenidos requieran interacción con base de datos, se denomina a las páginas web como dinámicas en lugar de páginas estáticas.

Estas solo se ocupan de agrupar el contenido html básico, que conforma la estructura de la página, el contenido CSS que la formatea de manera gráfica, y el contenido en un lenguaje de programación específico como puede ser PHP o .NET, entre otros, que consulta la base datos, lee los registros necesarios, y transforma estos datos en contenido html, para que finalmente este sea el que llegue en modo estático a nuestro navegador. Este contenido dinámico es procesado en el servidor web, de forma transparente para el usuario. Este último recibirá en su navegador web todo el contenido en formato plano, como si fuese estático, para que el navegador web lo renderice y muestre de la manera correcta. Las páginas web usualmente tienen la extensión HTML, pero en los casos de tecnologías de páginas dinámicas, estas pueden contener extensiones como PHP, .ASPX, .PY, entre otras. Esto será transparente para el usuario final, pero clave para que nuestro servidor web pueda procesar contenido de forma óptima.

Hipervínculos:

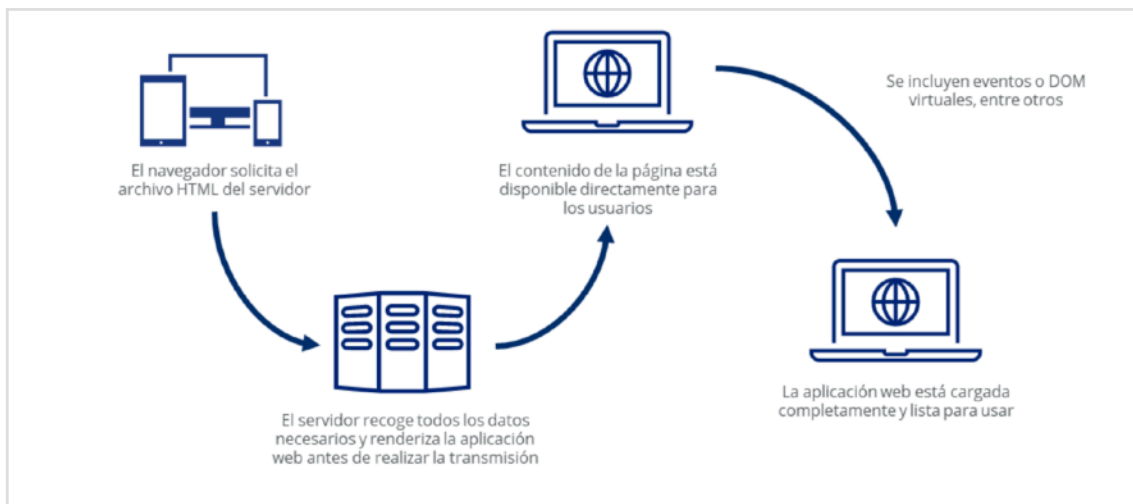
Todo el contenido web de páginas, emails, archivos, direcciones web, etc. se interrelaciona a través de hipervínculos. Es básicamente la dirección física de un archivo o recurso, dentro del mismo servidor web o hacia un servidor web externo.

Recursos adicionales:

Todo contenido adicional que complementa un sitio web, como videos, archivos de audio, imágenes, documentos, PDF, aplicaciones instalables, etc. son tratados y conocidos como recursos adicionales.

Generalidades del protocolo HTTP

HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Así, una página web completa resulta de la unión de distintos sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc...



Clientes y servidores se comunican intercambiando mensajes individuales (en contraposición a las comunicaciones que utilizan flujos continuos de datos). Los mensajes que envía el cliente, normalmente un navegador Web, se llaman peticiones, y los mensajes enviados por el servidor se llaman respuestas.

Diseñado a principios de la década de 1990, HTTP es un protocolo ampliable, que ha ido evolucionando con el tiempo. Es lo que se conoce como un protocolo de la capa de aplicación, y se transmite sobre el protocolo TCP, o el protocolo encriptado TLS (en-US), aunque teóricamente podría usarse cualquier otro protocolo fiable. Gracias a que es un protocolo capaz de ampliarse, se usa no solo para transmitir documentos de hipertexto (HTML), si no que además, se usa para transmitir imágenes o vídeos, o enviar datos o contenido a los servidores, como en el caso de los formularios de datos. HTTP puede incluso ser utilizado para transmitir partes de documentos, y actualizar páginas Web en el acto.

Métodos de petición HTTP

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos: ej. un request method puede ser safe, idempotent (en-US), o cacheable.

GET

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

HEAD (en-US)

El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

POST

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE (en-US)

El método DELETE borra un recurso en específico.

CONNECT

El método CONNECT establece un túnel hacia el servidor identificado por el recurso.

OPTIONS (en-US)

El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino.

TRACE

El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

PATCH

El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

A continuación los estados más utilizados y una breve descripción:

200 (OK): La solicitud ha sido recibida, entendida y procesada correctamente.

201 (Created): La solicitud se procesó correctamente y generó un nuevo recurso en el proceso.

400 (Bad Request): Solicitud no apropiada, faltan requisitos para ser válida.

401 (Unauthorized): No autorizado credenciales inválidas.

403 (Forbidden): Prohibido credenciales insuficientes, no tiene permisos para acceder.

404 (Not found): Recurso no encontrado.

500 (Internal server error): Hubo un error en el servidor y la solicitud no pudo ser completada.

