

Fundamentos de la web

¿Qué es la World Wide Web?

WWW o World Wide Web constituye un sistema global de información donde los documentos y otros recursos son accesibles a través de esta red interconectada entre nodos (internet), aquí estos recursos son interconectados generalmente por enlaces hipertextuales que se identifican por medio de localizadores uniformes de recursos (URL). Esta misma se compone de varios elementos como:

- **Lenguaje de estructuración:** De esto se encarga HTML, como tal no es un lenguaje de programación sino más bien es un lenguaje de marcado que permite crear el esqueleto de lo que va a ser un sitio web, tomando en cuenta la creación de cada una de sus páginas.
 - **Protocolos de comunicación:** Actúa principalmente en los protocolos HTTP y HTTPS, permiten la carga y descarga de páginas web o sitios web completos que operan un modelo de solicitud-respuesta entre un cliente y un servidor.
 - **URL:** Actúan como direcciones que permiten localizar cada recurso disponible en la red.
 - **Navegadores:** Son aplicaciones normalmente gráficas por las que los usuarios pueden interactuar con los recursos dentro de la red, su principal función es la interpretación del código HTML, CSS o meramente lenguajes de programación del lado del cliente como JavaScript, de esta forma pueden interpretar el código de la página web que el cliente solicita para que de esa manera puedan mostrárselo de una forma simple y visualmente atractiva.
- Algo importante de mencionar, internet y la web no es lo mismo, pero si son infraestructuras que dependen una de la otra actualmente y funcionan bajo el mismo paradigma.

¿Qué es un dominio y sus tipos?

Un dominio es un identificador único que no se puede repetir en internet, este mismo se utiliza como dirección para facilitar el acceso a un contenido en específico reemplazando la necesidad de memorizar secuencias numéricas o direcciones IP, en su lugar se puede acceder al mismo recurso pero en este caso teniendo que saber únicamente cadenas de caracteres.

Cada servidor expuesto a la internet tiene una dirección IP única asignada a él y para introducir a él de forma remota se debe introducir su dirección IP, en este punto entra el juego el DNS, el cual permite que traduzca las direcciones.

Los dominios poseen varias categorías:

- **TLD o Dominio Primario:** Los Top Level Domain son el nivel más alto de dominios dentro de la jerarquía de internet, básicamente son los últimos segmentos de un nombre de dominio localizados después del último punto.
- **SLD o Dominio Secundario:** Los Second Level Domain está por debajo en nivel de jerarquía del dominio primario. En otras palabras, este es el dominio que generalmente se registra por individuos o entidades, por consecuencia es la parte de nombre de dominio que no se puede repetir.

- **Subdominio**: A nivel de jerarquía están en un nivel más bajo que los dominios anteriores. Estos subdominios se pueden repetir.

Componentes de una URL

Ejemplo de URL:

<https://www.ejemplo.com:8080/blog/articulo?id=123#comentarios>

Son las siglas de Uniform Resource Locator, es esa dirección que apunta a un recurso en específico de forma única y está compuesta por lo siguiente:

- **Esquema/Protocolo**: Indica el método de acceso que se va a utilizar para obtener un recurso.
- **Segmento delimitador**: básicamente se representa por **://**, sirve para indicarle al navegador como debe ser interpretada la información que sigue.
- **Dominio/IP**: Identifica al host de destino donde se encuentra el recurso a consultar.
- **Puerto**: Es un componente opcional en una URL, únicamente se utiliza para forzar la conexión se realice por un puerto en específico, esto último utilizando el delimitador y el puerto que se desea usar (**:8080**).
- **Path/Ruta**: Indica la ubicación específica del recurso que se desea consultar. Esta misma se puede componer de varios segmentos separado por barra (**/**).
- **Parámetros**: Se utilizan para dar parámetros adicionales al servidor de la petición dada, estos parámetros están precedidos por un signo de interrogación (**?**).
- **Fragmento/Ancla**: Se representa con un numeral (**#**) y generalmente se utilizan para enlazar secciones.

Diferencias entre un sitio web y una página web

Una página web es un documento singular que es parte de la World Wide Web y es accesible a través de un navegador, cada página web es identificada por una URL, esto significa que se puede encontrar y acceder a través de su enlace específico. Por otro lado, un sitio web es un conjunto de páginas web que están relacionadas y agrupadas bajo un mismo dominio.

Hosting, VPS y Servidor dedicado

- **Hosting**: Es la opción más utilizada para regir sitio web, por lo general, son sitios pequeños. Su arquitectura consta de que hay sitio web alojados bajo el mismo servidor físico, compartiendo recursos de hardware.
- **Servidor dedicado**: Este proporciona todos sus recursos a un único cliente.
- **VPS o Virtual Private Server**: En este cada sitio opera de manera independiente y con recursos asignados, pero dentro de un solo dispositivo.

Frontend y Backend y sus vulnerabilidades

Frontend y sus vulnerabilidades

Este se centra en la parte de la aplicación con la que los usuarios interactúan de forma directa, las tecnologías más fundamentales dentro de este son lenguajes como HTML, CSS o JavaScript. En el frontend las vulnerabilidades más comunes son: el Cross-Site Scripting (XSS), Clickjacking o Cross-Site Request Forgery (CSRF)

Backend y sus vulnerabilidades

Es la parte trasera de cualquier aplicación y la que nos usuario no ven. Esta es la capa donde se procesan los datos, se gestionan las interacciones del usuario y se ejecutan los procesos lógicos de la aplicación, adicionalmente es el responsable de la comunicación entre la base de datos y el frontend. Las principales tecnologías con las que se trabajan son Java, Python, Ruby o PHP y la implementación de bases de datos relacionales o no relacionales, pueden incluir APIs con sus diferentes protocolos.

Las principales vulnerabilidades en el backend son: Inyecciones SQL (SQLi), Remote Code Execution (RCE), Acceso y abuso de APIs o Fuga de información por malas configuraciones.

Principales servidores web

Cuando se habla de servidores web se hace referencia a un software que acepta solicitudes del protocolo HTTP y HTTPS, cuyas funciones principales son procesar dichas solicitudes y entregar el contenido; esto último dependiendo de la solicitud, manejar la seguridad entre las conexiones o solamente mantener un registro de la actividad. Existen muchos de este tipo de servidores pero los principales son:

Apache

Es uno de los más utilizados y utiliza el lenguaje de programación C, además, Apache se basa en una estructura modularizada; en otras palabras, permite únicamente cargar los módulos necesario para las aplicaciones web específicas que se quieran servir por medio de él. Un módulo sería mod_security que actúa como una especie de “firewall”.

Nginx

Está enfocado a entornos de alto tráfico, balanceo de carga, servidores proxy inverso y entrega eficiente de contenido estático.

IIS

Este es creado por Microsoft, se encuentra principalmente corriendo en Windows server. Se utiliza para aplicaciones que requieran tecnologías específicas propias de Microsoft, como por ejemplo ASP.net.

Apache Tomcat

Implementa las especificaciones de Java, en concreto las de Java Servlet y JavaServer Pages (JSP). Es principalmente usado para correr aplicaciones en Java.

Jetty

Es un servidor ligero que se caracteriza por consumir pocos recursos

Principales Content Management System (CMS)

Los sistemas de gestión de contenidos se han hecho populares últimamente ya que permiten a las personas sin mucho conocimiento técnico crear un sitio web, los principales tipos son:

WordPress

Fue creado originalmente para crear blogs, pero con el paso del tiempo fue evolucionado hasta el punto en el que cerca del 40% de los sitios web del mundo utilizan WordPress. Como tal WordPress utiliza PHP como lenguaje de programación y se apoya en MySQL y MariaDB para la gestión de las bases de datos.

Joomla

Se podría decir que es como un WordPress ya que también usa PHP y MySQL. La diferencia radica en que Joomla no es tan simple de utilizar a nivel usuario.

Drupal

Está ideado para sitios web que buscan crear una estructura escalable.

Métodos de envío de información del protocolo HTTP

Estos definen la acción que se desea realizar sobre un recurso identificado por una URL, entre los métodos están:

- **GET**: Solicita la representación de un recurso específico. Solo deben ser de recuperación y no deben cambiar el estado del recurso. Envía los datos solicitados a través de la URL
- **POST**: Envía datos al servidor para crear un nuevo recurso. Los datos enviados se procesan en el cuerpo de la solicitud.
- **PUT**: Envía datos al servidor para actualizar un recurso existente o crear un nuevo recurso.
- **DELETE**: Elimina el recurso especificado.
- **PATCH**: Se utiliza para hacer modificaciones parciales a un recurso.
- **HEAD**: Solicita solo los metadatos asociados con el recurso sin el cuerpo de la respuesta.
- **TRACE**: Permite al cliente ver lo que está recibiendo un servidor intermedio en una cadena de solicitudes/proxys.
- **CONNECT**: Convierte la conexión de solicitud/respuesta en un canal de comunicación transparente, útil para túneles a través de un servidor proxy. Se utiliza principalmente para cifrar las comunicaciones del protocolo HTTPS a través de un proxy.
- **OPTIONS**: Utilizado para describir las opciones de comunicación permitidas para el recurso objetivo, proporcionando una lista de los métodos HTTP soportados.

Principales cabeceras del protocolo HTTP

Son componentes esenciales tanto al nivel de las solicitudes y respuestas en servicios que empleen este protocolo; básicamente transmiten información adicional entre el cliente y el servidor, las principales cabeceras son:

- **Host**: Identifica el dominio y puerto que el cliente quiere acceder en un servidor.
- **User-Agent**: Identifica el navegador y el sistema operativo del cliente que hace la solicitud.
- **Accept**: Especifica los tipos de contenido que el cliente puede procesar.
- **Accept-Language**: Indica el idioma preferido por el usuario, permitiendo que el servidor responda con un recurso en el idioma apropiado.
- **Accept-Encoding**: Especifica los formatos de codificación que el cliente acepta para la comprensión de datos.
- **Content-Type**: Especifica el tipo de medio del cuerpo de la solicitud o respuesta.
- **Content-Length**: Especifica el tamaño en byte del cuerpo de la solicitud o respuesta.
- **Cookie**: Contiene las cookies enviadas por el cliente al servidor para mantener el estado o la sesión.
- **Connection**: Especifica si la conexión con el servidor debe mantenerse abierta o cerrarse después de completar la solicitud.
- **Referer**: Indica la URL de origen desde la cual se accedió al recurso solicitado.
- **Authorization**: Especifica las credenciales necesarias para autenticar al cliente con el servidor.
- **Cache-Control**: Especifica directivas para la caché tanto para las solicitudes como para respuestas controlando cómo se almacenan y recuperan los datos.
- **X-Forwarded-For**: Indica la dirección IP original del cliente, utilizado por servidores o proxies intermedios.
- **Set-Cookie**: Indica al cliente que almacene una cookie y especifica su valor junto con atributos como la caducidad, ruta y dominio.
- **X-XSS-Protection**: Especifica configuraciones para el filtro de protección contra ataques de tipo (XXS) en navegadores compatibles.

Códigos de estado HTTP

Proporcionan una rápida indicación sobre el resultado de la solicitud realizada por un cliente, en base al código del estado que se reciba se puede interpretar en qué estado se encuentra la solicitud que se envió, estos códigos de estado se pueden dividir en 5 categorías:

1XX o Respuestas informativas

Son código de estado que indican que la solicitud ha sido recibida y entendida por el servidor, de la misma manera, indica que el proceso está en curso. El código de estado más relevante es el 100 indica que el cliente puede seguir enviando el resto de la solicitud.

2XX o Respuesta satisfactoria

Indica que la solicitud fue recibida, entendida y aceptada por el servidor de destino. Los códigos más relevantes son:

- 200 indica que la solicitud fue exitosa.
- 201 indica que la solicitud fue exitosa y resultó en la creación de un nuevo recurso.
- 204 indica que la solicitud fue exitosa pero no hay contenido para enviar.

3XX o Redirecciones

Indica que se deben de tomar más acciones por parte del cliente para completar la solicitud. Las más relevantes son:

- 301 indica que el recurso solicitado ha sido movido a una nueva URL de forma permanente.
- 302 indica que el recurso solicitado ha sido movido temporalmente a otra URL.

4XX o Errores del cliente

Estos códigos señalan la existencia de un error. Los más comunes son:

- 400 indica que la solicitud no se pudo entender o estaba mal formada.
- 401 indica que la solicitud requiere autenticación.
- 403 indica que el servidor entendió la solicitud pero se niega a autorizarla.
- 404 indica que el recurso solicitado no fue encontrado en el servidor.

5XX o Errores del servidor

Son los errores que se producen del lado del servidor, los más comunes son:

- 500 indica que el servidor encontró una condición inesperada que le impidió completar la solicitud.
- 502 indica que el servidor, actuando como una puerta de enlace o proxy, recibió una respuesta inválida.
- 503 indica que el servidor no está disponible temporalmente.