UT4: EL SERVICIO WEB

1. **Funcionamiento del servicio HTTP.**
2. **Parámetros de configuración del servicio: Parámetros en el servidor y en el cliente.**
3. **Servidores web seguros. Protocolo HTTPS. Utilización de certificados.**
4. **Alojamiento web.**

1. Funcionamiento del servicio HTTP.

El protocolo **HTTP** ***(HyperText Transfer Protocolo*** Protocolo de Transferencia de HiperTexto) surgió para facilitar a los usuarios el acceso a información remota de una forma sencilla e intuitiva, dando lugar a lo que conocemos como WWW *(World Wide Web* o Telaraña Mundial). Hoy, el servicio HTTP es uno de los más utilizados y goza de gran popularidad en Internet. El servicio HTTP está asociado al puerto **80**.

Los usuarios que utilizan este servicio, acceden a documentos denominados páginas webs. Además de texto, estas páginas pueden incluir otros elementos como imágenes, sonido o vídeo. A veces los creadores de estas páginas, dan un carácter especial a algunos de sus elementos, permitiendo que a través de ellos se acceda a otras páginas o servicios, los cuales conocemos con el nombre de hiperenlaces. Al formato de estas páginas que permite incluir diversos tipos de información y referencias a otras páginas, se conoce como hipertexto o **HTML** ***(HyperText Markup Language*** o Lenguaje de Marcas de Hipertexto).

A priori, las páginas estáticas tienen un contenido que no admiten interacción por parte del usuario, simplemente se muestran y a lo sumo permiten el acceso a otras páginas mediante los mencionados hiperenlaces. Para conseguir que los usuarios interactúen con ellas, como por ejemplo, rellenar un formulario recibiendo avisos si no se rellenan todos los datos o por si son incorrectos, se requiere que las páginas sean programadas (páginas dinámicas). Algunos de los lenguajes y tecnologías utilizados para la creación de páginas dinámicas son:

PHP: lenguaje cuyas instrucciones forman parte del documento HTML, se interpretan y procesan en el servidor HTTP y, posteriormente, se envía al cliente el documento HTML resultante.

ASP: *(Active Server Pages* o Páginas Activas de Servidor): su código se implementa mediante *scripts* y se combina con documentos HTML. Al igual que PHP, se interpreta en el servidor y el documento HTML resultante se envía al cliente. Derivado de ASP y aprovechando la tecnología .NET, surgió el lenguaje orientado a objetos ASP.NET, que permite la utilización de lenguajes soportados por el marco de trabajo .NET, como por ejemplo C#, Visual Basic etc.

Java: lenguaje de programación multiplataforma basado en C++, permite incorporar animación e interacción en páginas mediante lo que se conoce como applets. Un **applet** es un pequeño programa que se obtiene y es ejecutado por el software cliente (navegador), como parte de la página solicitada. Un ***servlet*** también es un programa Java que se ejecuta en el servidor, cuya función es la de generar páginas en respuesta a las peticiones cliente.

JSP: *(Java Server Pages* o Páginas de Servidor Java): las páginas en JSP se escriben en HTML o XML, y utilizan etiquetas especiales para incluir contenido dinámico mediante código Java. El servidor interpreta o compila la página y, mediante *servlets,* genera una página en respuesta a la petición

generada por el cliente. Se puede decir que JSP es utilizado como interfaz con el cliente, apoyándose en Java para aportar dinamismo a las páginas.

Java-Script: lenguaje basado en Java. Sus instrucciones forman parte de una página o documento HTML y son interpretadas por el software del cliente o navegador.

CGI: *Common Gateway Interfaz* o Interfaz de Pasarela Común): no es propiamente un lenguaje, sino un interfaz que permite a documentos HTML intercambiar datos, por ejemplo con programas C o Perl.

AJAX: *(Asinchronous JavaScript And XML* o JavaScript y XML Asíncrono): utiliza varias tecnologías como TIIL y JavaScript. La comunicación del cliente con el servidor la efectúa en segundo plano, facilitando la interacción con el usuario y evitándole recargas de página.

Para una visualización adecuada de las páginas, el cliente utiliza software específico denominado **navegador**. El acceso a una página requiere que el usuario proporcione diversa información al navegador, como la página web que desea y dónde localizarla, especificando lo que se conoce como **URL** (Uniform Resource Locators o Localizador Uniforme de Recursos). Por tanto podríamos decir que una URL expresa la manera de acceder a un recurso utilizando un determinado servicio. Un ejemplo de URL: *http://fedoraproject.org/es/ndex.html.* Siguiendo el ejemplo:

htttp:// Indica el servicio o protocolo a utilizar (otros: HTTP, HTTPS, FTP, telnet o news).

fedoraproject.org Indica la dirección IP o el nombre del servidor que contiene el recurso, en este caso es el nombre del servidor web.

/es/ es la ruta al recurso, es decir, el directorio y subdirectorios del sitio web donde reside el recurso.

Index.html Indica el recurso al que se quiere acceder. Si se omite el servidor buscará generalmente un fichero de inicio como: index.html, index.htm, etc.

Cuando un cliente indica una URL en su navegador, una vez resuelta por el servidor DNS que tenga configurado, se establece una conexión TCP con el puerto 80 del servidor, que permanece a la escucha de solicitudes HTTP. El navegador utiliza esta conexión para solicitar la página o recurso deseado al servidor. El servidor atiende la petición, y envía la página o el recurso solicitado, o bien un mensaje de error en forma de página web si no existe o no está disponible, mostrándose al cliente en su navegador.

Siguiendo el ejemplo: al indicar http://fedoraproject.org/es/index.html en el navegador, el servidor DNS resuelve la dirección IP del servidor HTTP (fedoraproject.org) y se establece una conexión HTTP con el servidor, el servidor proporciona, al cliente, la página solicitada (index.html), ubicada en el directorio /es/ del sitio web

Cuando se lleva a cabo la transferencia de información entre el cliente y servidor mediante el protocolo HTTP, no existe una conexión de control explícita para

gestionar dicha transferencia, tal y como ocurre con el protocolo FTP. Para ello se utilizan las **líneas de encabezado**, que contienen información sobre la propia transferencia. El encabezado se transmite antes del contenido en sí de la página web. Por ejemplo, cuando un cliente se autentica en el servidor web, el usuario y contraseña que ha introducido viaja en dichos encabezados.

Sitio web. Las páginas que se ofrecen a los usuarios se almacenan en los servidores HTTP, localizándose en un directorio específico denominado sitio web (raíz del servidor o raíz de documentos). En este directorio, se suele establecer una jerarquía de subdirectorios para organizar las distintas páginas, así como los distintos elementos que las integran (fondos, imágenes, webs, etc). La página de inicio: *index.html, index.htm, …* se sitúa en el directorio raíz de la jerarquía del sitio y se utiliza a modo de índice para dar acceso al resto de páginas y elementos del sitio. Esta página también se muestra por defecto cuando en la URL sólo se indica el servidor al que se quiere acceder.

Tipos MIME. Entre el cliente y el servidor se puede transferir cualquier tipo de contenido, aunque debería estar definido por el estándar **MIME** ***(Multipurpose Internet Mail Extension*** o Extensión de Correo Multipropósito de Internet). Este estándar se definió para el envío de mensajes de correo electrónico, aunque posteriormente también se ha utilizado para transferencias de información mediante el protocolo HTTP. Este estándar define los formatos, tipo de letra y características de una página para que pueda ser visible por distintos navegadores.

WebDAV. La extensión al protocolo HTTP 1.1 WebDAV ***(Web-based Distributed Authoring and Versioning*** o Edición y Versionado Distribuidos sobre la Web), nos proporciona un entorno de colaboración para la elaboración y administración de los elementos del sitio de una forma remota y descentralizada. Es decir, aquellas personas encargadas de la elaboración de los diferentes elementos del sitio como webs, imágenes, vídeos o sonidos, y teniendo acceso al servidor, podrían, por ejemplo, editar los contenidos *online* o mover los ficheros de directorio dentro de la jerarquía del sitio. Para un mejor control de los cambios efectuados sobre los elementos del sitio, se lleva a cabo mediante software de control de versiones, como es el caso de **CVS** *(Concurrent Versioning System* o Sistema de Versiones Concurrentes). Paquetes ofimáticos como Open Office o Microsoft Office, ya incorporan soporte para WebDAV.

2. Parámetros de configuración del servicio http. Parámetros en el servidor y en el cliente.

Para que el servidor HTTP funcione habrá que configurarlo correctamente y si queremos que sea accesible desde Internet necesitamos una dirección IP pública, tendremos que habilitar el acceso al puerto HTTP (por defecto el 80) de nuestro router y además, redirigir las solicitudes cliente al equipo configurado como servidor HTTP.

Si el servidor dispone de cortafuegos, tendremos que permitir el acceso a dicho puerto. Si queremos que los clientes puedan acceder al servicio HTTP, mediante un nombre FQDN, tendrá que poder ser resuelto por algún servidor DNS (si nuestra IP no es fija, puede utilizarse servidores DNS dinámicos gratuitos como dyndns.org o no-ip.com).

Parámetros del servidor. Antes de proceder a la instalación y configuración de este servicio en cualquiera de las plataformas, se debe crear o determinar el directorio donde ubicará la raíz del sitio HTTP. Opcionalmente puede establecer una jerarquía de subdirectorios a efectos de organización. Establecer los permisos adecuados sobre directorios y subdirectorios. Es habitual delegar sobre uno o varios usuarios la responsabilidad de la gestión del sitio, conocidos como **webmaster.**

Antes de iniciar el servicio, hay que alojar las páginas y demás elementos que queremos ofrecer a los usuarios, en el directorio correspondiente del sitio. Asegurarnos también de haber establecido los hiperenlaces de manera adecuada para que hagan referencia a la página o elemento deseado. Utilizar una página índice (index) del sitio para permitir el acceso a los diferentes elementos que lo conforman.

Una vez establecido el directorio raíz del sitio, podremos configurar parámetros adicionales, como el soporte para conexiones seguras mediante HTTPS, requerir la autenticación de los usuarios para acceder al sitio, o la utilización de sitios adicionales sobre el mismo servidor.

La configuración de un acceso seguro mediante HTTPS, requerirá la instalación en el servidor de un certificado, autogenerado por el sistema operativo o firmado por una CA.

Parámetros en el cliente. La utilización del software cliente de este servicio, básicamente consiste en indicar la URL deseada en un navegador. Ciertos contenidos requieren la instalación de elementos adicionales en el navegador, activando o desactivando las opciones pertinentes. Si el equipo pertenece a una organización con **Proxy** (caché y filtro de páginas web), debemos configurar el navegador indicando el equipo o dispositivo que ejerce de proxy.

El acceso a sitios web mediante HTTPS, requiere acceder con ***https://***, en vez de ***http://***. Al acceder el navegador mostrará una advertencia en caso de que el certificado no haya sido firmado por una CA de nuestra confianza, permitiéndonos decidir su instalación o no. La mayoría de navegadores muestran en su barra de estado el símbolo de un candado para distinguir el acceso a un sitio web seguro.

3. Acceso seguro y utilización de certificados

En ocasiones, el acceso a determinadas páginas puede requerir determinada información confidencial, como la referente a transacciones financieras o la autenticación en servidores de correo. Debido a que la información viaja por la red sin cifrar, puede ser interceptada por un usuario malintencionado para hacer mal uso de ella, comprometiendo la seguridad del

cliente. Para paliar este problema surgieron algunos protocolos:

HTTPS ***(HyperText Transfer Protocol Secure*** o Protocolo Seguro de Transferencia de HiperTexto): se apoya sobre una conexión segura previamente establecida en la capa de transporte mediante la utilización de **SSL** ***(Secure Socket Layer*** o Capa de Conexión Segura) o **TLS** ***(Transport Layer Security*** o Seguridad de la Capa de Transporte).

Esta conexión segura, encripta la información susceptible de comprometer la seguridad del cliente, como por ejemplo su nombre de usuario y contraseña de sesión. El empleo de este tipo de conexiones seguras, requiere mecanismos de cifrado como los basados en clave pública y la utilización de certificados. El puerto que se suele utilizar para este tipo de conexiones es el **443** en vez del 80. El cliente hará referencia a este servicio seguro mediante el empleo de ***https*** en vez de *http* en la URL.

S·HTTP ***(Secure HyperText Transfer Protocolo*** Protocolo de Transferencia de HiperTexto Seguro): utiliza extensiones de las cabeceras HTTP que son intercambiadas entre cliente y servidor, tienen una sintaxis concreta donde incluyen información de seguridad, como los algoritmos utilizados para encriptar la información, por ejemplo PGP, muy utilizado por este protocolo. Por esta razón, no requiere la utilización de certificados digitales ni claves públicas. Este protocolo solo se puede utilizar, en la capa de aplicación, y no es válido para soportar y asegurar la utilización de otros protocolos como ocurre con SSL. La extensión de los documentos utilizados con este protocolo es ***".shttp".*** Este protocolo no ha tenido tanta aceptación como el anterior, de hecho existen pocos navegadores que soporten dicho protocolo.

Los **certificados** son utilizados para permitir que las comunicaciones sean seguras. Se basan en técnicas de encriptación, como los algoritmos de cifrado asimétrico, que utilizan una clave pública conocida por el emisor y el receptor, y una clave privada solo conocida por el destinatario de la información. La información a transmitir se cifra con la clave pública y solo el que posee la clave privada será capaz de descifrar la información. Algunos de los algoritmos utilizados son el **RSA, Triple DES (o TDES) y AES**, así como el protocolo **X.509**, que se ha establecido como estándar para los certificados.

La utilización de un certificado permite que su titular se identifique y autentique a la hora de acceder a un sitio cuya información puede verse comprometida, son certificados de usuario, por ejemplo cuando queremos acceder a la Agencia Tributaria para hacer la declaración de la renta online podemos autenticarnos con un certificado válido de usuario.

Un certificado de usuario es firmado por una **entidad certificadora** o **CA** ***(CertificationAuthority)*** como por ejemplo la Fábrica Nacional de Moneda y Timbre (**FNMT**) o la empresa ***Verisign,*** de forma que garantiza su autenticidad. Siguiendo con el ejemplo de la Agencia Tributaria, cuando accedemos mediante nuestro certificado de usuario, dicha entidad confía en la entidad certificadora, la FNMT, y por tanto nos permitirá el acceso. Cuando confiamos en una entidad certificadora, debemos instalar su certificado denominado raíz (autofirmado por la CA), de forma que los certificados que dicha entidad haya firmado también serán de nuestra confianza.

Cuando accedemos a un sitio web mediante el protocolo HTTPS, necesitamos conocer previamente que el servidor es realmente de la entidad a la que queremos acceder. Para demostrado deberá poseer un certificado que debe estar firmado por una CA que sea de nuestra confianza. En caso contrario, cuando accedamos al sitio web mediante nuestro navegador, nos pedirá confirmación para proseguir con nuestros trámites.

La tramitación de la firma de certificados por entidades certificadores como *Verisign,* suelen conllevar gastos, aunque como alternativa se pueden utilizar certificados autofirmados, que se pueden generar con herramientas software como **OpenSSL**. Para probar la utilización de certificados, se puede acceder a la página web de *Verisign* para la descarga de un certificado de carácter temporal.

4. Alojamiento web.

Un proveedor de alojamiento web es una empresa que alquila espacio web y ancho de banda para la publicación de sitios web. Normalmente se trata de una cuenta en un sistema Linux o UNIX que está permanentemente encendido donde alojamos los archivos de nuestro sitio web a través de la herramienta FTP (o SSH), y nuestra web es servida mediante un servidor web (como Apache).

Tipos de alojamiento web

Los requerimientos de una página personal no son los mismos que los de una gran empresa. En este último caso, el volumen de información y la cantidad de accesos al servidor será muchísimo mayor y, por tanto, el servidor que maneje dicha información deberá disponer de más recursos. En una web con miles de visitas al día, la no disponibilidad del servicio durante unas horas suponen pérdidas que no deben ser admisibles, por lo cual el hosting a contratar deberá ofrecer más calidad que una web personal.

Por esto mismo existen diferentes tipos de alojamiento web:

1. Alojamiento compartido gratuito: normalmente dispone de poca funcionalidad y estabilidad.
2. Alojamiento compartido: el más frecuente. En este caso el sitio web funciona en un servidor que aloja muchos otros sitios web.
3. Servidor dedicado: el proveedor se encarga de la administración del servidor.

1.- Alojamiento compartido gratuito

Elegir un alojamiento gratuito puede ser una buena solución para los que están empezando en el diseño web. Sin embargo, para proyectos

medianamente serios a la larga salen caros: inaccesibilidad, tiempo de espera, pérdida de información, cierre del servicio, limitaciones de funcionalidad, etc. Aunque podemos encontrar servidores gratuitos que funcionan bien, normalmente el servicio degenera con el tiempo.

2.- Alojamiento compartido

La solución más común en el mercado. Por un precio bastante razonable podemos disponer de varios miles de GB de transferencia al mes, alojar ilimitados dominios, direcciones de correo ilimitadas, soporte para ejecutar aplicaciones basadas en MySQL, PHP o ASP, etc. La seguridad del servidor suele ser menor al haber más clientes en él. Hay que prestar atención al número de usuarios/webs que aloja el servidor, es una práctica bastante común llenar los servidores de usuarios hasta el punto de que interfiera con el servicio prestado.

A no ser que se trate de un sitio web con un tráfico excesivo y que consuma demasiada CPU, el alojamiento web compartido es la solución con mejor relación calidad/precio (siempre que se trate de un buen proveedor).

3.- Servidor dedicado

Esta puede ser la solución cuando la cuenta en el servidor compartido se esté quedando pequeña para nuestro proyecto, aunque es necesario tener conocimientos de administración de sistemas, dependiendo del nivel de soporte que se contrate, se puede disponer de un experto administrador de sistemas que de soporte para hacer funcionar y personalizar nuestro servidor instalando y configurando los servicios que necesitemos.