

INFORMACION COMPLEMENTARIA WIFI

REST API

Rest API es la api [ver descripcion](#) que se utiliza para las comunicaciones en internet entre servidores y clientes . Sin embargo antes de describir esta api , vamos a repasar cual es la base de las comunicaciones de internet , lo que utilizaremos en nuestra practica de wifi.

Concepto IP protocolos (UDP y TCP) y protocolo HTTP

Que es una IP

Una dirección IP es una dirección única que identifica a un dispositivo en Internet o en una red local. IP significa “protocolo de Internet”, que es el conjunto de reglas que rigen el formato de los datos enviados a través de Internet o la red local.

En esencia, las direcciones IP son el identificador que permite el envío de información entre dispositivos en una red. Contienen información de la ubicación y brindan a los dispositivos acceso de comunicación. Internet necesita una forma de diferenciar entre distintas computadoras, enrutadores y sitios web. Las direcciones IP proporcionan una forma de hacerlo y forman una parte esencial de cómo funciona Internet.

Una dirección IP es una cadena de números separados por puntos. Las direcciones IP se expresan como un conjunto de cuatro números, por ejemplo, 192.158.1.38. Cada número del conjunto puede variar de 0 a 255. Por lo tanto, el rango completo de direcciones IP va desde 0.0.0.0 hasta 255.255.255.255.

Las direcciones IP no son aleatorias. La Autoridad de números asignados de Internet (Internet Assigned Numbers Authority, IANA), una división de Internet Corporation para números y nombres asignados (Internet Corporation for Assigned Names and Numbers, ICANN), genera y asigna matemáticamente las direcciones IP. ICANN es una organización sin fines de lucro que se estableció en los Estados Unidos en 1998 para ayudar a mantener la seguridad de Internet y permitir que todos puedan utilizarla. Cada vez que alguien registra un dominio en Internet, debe dirigirse a un registrador del nombre de dominio, quien paga una pequeña tarifa a ICANN para registrarlo.

Que es un puerto asociado a una IP

Para un ordenador conectado a una red con una dirección IP, un puerto es un canal de comunicacion dentro de la misma direccion IP . Los puertos están designados por números, por debajo de 1024 cada puerto está asociado por defecto con un protocolo específico.

Por ejemplo, el puerto por defecto para el protocolo HTTP es 80 y el puerto por defecto para el protocolo HTTPS es 443, por lo tanto un servidor HTTP espera peticiones en esos puertos. Cada protocolo de internet está asociado con un puerto por defecto: SMTP (25), POP3 (110), IMAP (en-US) (143), IRC (194), y así sucesivamente.

Los números de puerto se indican mediante una palabra de un procesador de 16 bits (2 bytes), por lo que existen 65536 puertos, numerados del 0 al 65535. Aunque podemos usar cualquiera de ellos para cualquier protocolo, existe una entidad, la IANA, encargada de su asignación, la cual creó tres categorías:

Puertos bien conocidos: Los puertos inferiores al 1024 son puertos reservados para el sistema operativo y usados por "protocolos bien conocidos" como por ejemplo HTTP (servidor Web), POP3/SMTP (servidor de e-mail) y Telnet. Si queremos usar uno de estos puertos tendremos que arrancar el servicio que los use teniendo permisos de administrador. **Puertos registrados:** Los comprendidos entre 1024 (0400 en hexadecimal) y 49151 (BFFF en hexadecimal) son denominados "registrados" y pueden ser usados por cualquier aplicación. Existe una lista pública en la web del IANA donde se puede ver qué protocolo que usa cada uno de ellos. **Puertos dinámicos o privados:** Los comprendidos entre los números 49152 (C000 en hexadecimal) y 65535 (FFFF en hexadecimal) son denominados dinámicos o privados, normalmente se asignan en forma dinámica a las aplicaciones de clientes al iniciarse la conexión. Se usan en conexiones peer to peer (P2P).

Protocolos UDP/IP TCP/IP

La comunicacion entre elementos de una red sigue la norma del modelo OSI

Modelo OSI El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), conocido como "modelo OSI", (en inglés, Open Systems Interconnection) es un modelo de referencia para los protocolos de la red https://es.wikipedia.org/wiki/Modelo_OSI

UDP // TCP

El Protocolo de datagrama de usuarios, o UDP, es un protocolo de comunicación que se utiliza en Internet para transmisiones sujetas a limitaciones temporales, como la reproducción de vídeo o las búsquedas en DNS. Acelera las comunicaciones al no establecer formalmente una conexión antes de transferir los datos. Esto permite que los datos se transfieran muy rápido, pero también puede hacer que los paquetes se pierdan en tránsito, y crear oportunidades para vulnerabilidades en forma de ataques DDoS.

Como todos los protocolos de redes, el UDP es un método estandarizado para transferir datos entre dos ordenadores en una red. Comparado con otros protocolos, el UDP realiza este proceso de forma sencilla: envía paquetes (unidades de transmisión de datos) directamente a un ordenador de destino, sin establecer primero una conexión, ni indicar el orden de dichos paquetes, ni comprobar si han llegado como estaba previsto. (Los paquetes UDP se llaman "datagramas").

El Protocolo TCP/IP o Transfer Control Protocol consiste en un acuerdo estandarizado sobre el que se realiza la transmisión de datos entre los participantes de una red informática.

Los programas que forman redes de datos en una red de ordenadores emplean el protocolo TCP para crear conexiones entre sí, de forma que se pueda garantizar el flujo de datos entre las partes. A través de este protocolo se asegura que los datos lleguen a su destino en el mismo orden que se transfirieron y sin errores.

La inmensa mayoría de comunicaciones que se realizan en internet utilizan el protocolo TCP IP, como es el caso de navegadores, programas de intercambio de ficheros, servicios FTP y un largo etcétera.

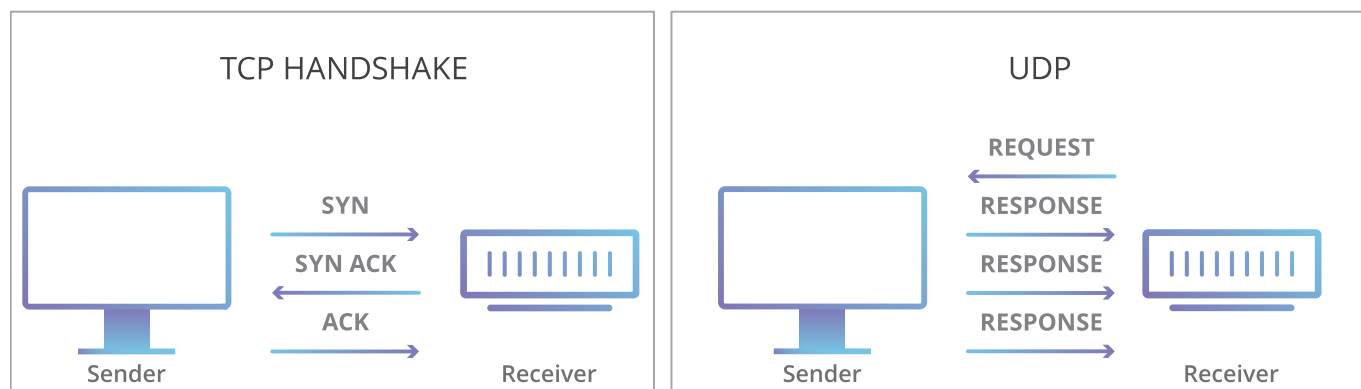
Las principales características del TCP Protocol son:

Es un protocolo que funciona mediante la conexión mutua entre cliente y servidor. Ordena los segmentos provenientes del protocolo IP. Monitorea el flujo de los datos y permite evitar la saturación de la red. Entrega los datos al protocolo IP en forma de segmentos de longitud variable. Permite circular de forma simultánea a la información proveniente de diferentes fuentes.

El UDP es más rápido pero menos fiable que el TCP, otro protocolo de transporte habitual. En una comunicación TCP, los dos ordenadores comienzan estableciendo una conexión mediante un proceso automatizado llamado "protocolo de enlace". Un ordenador no transferirá realmente paquetes de datos al otro hasta que no se haya completado el protocolo de enlace.

Las comunicaciones UDP no pasan por este proceso. En su lugar, un ordenador simplemente puede empezar a enviar datos al otro:

TCP vs UDP Communication

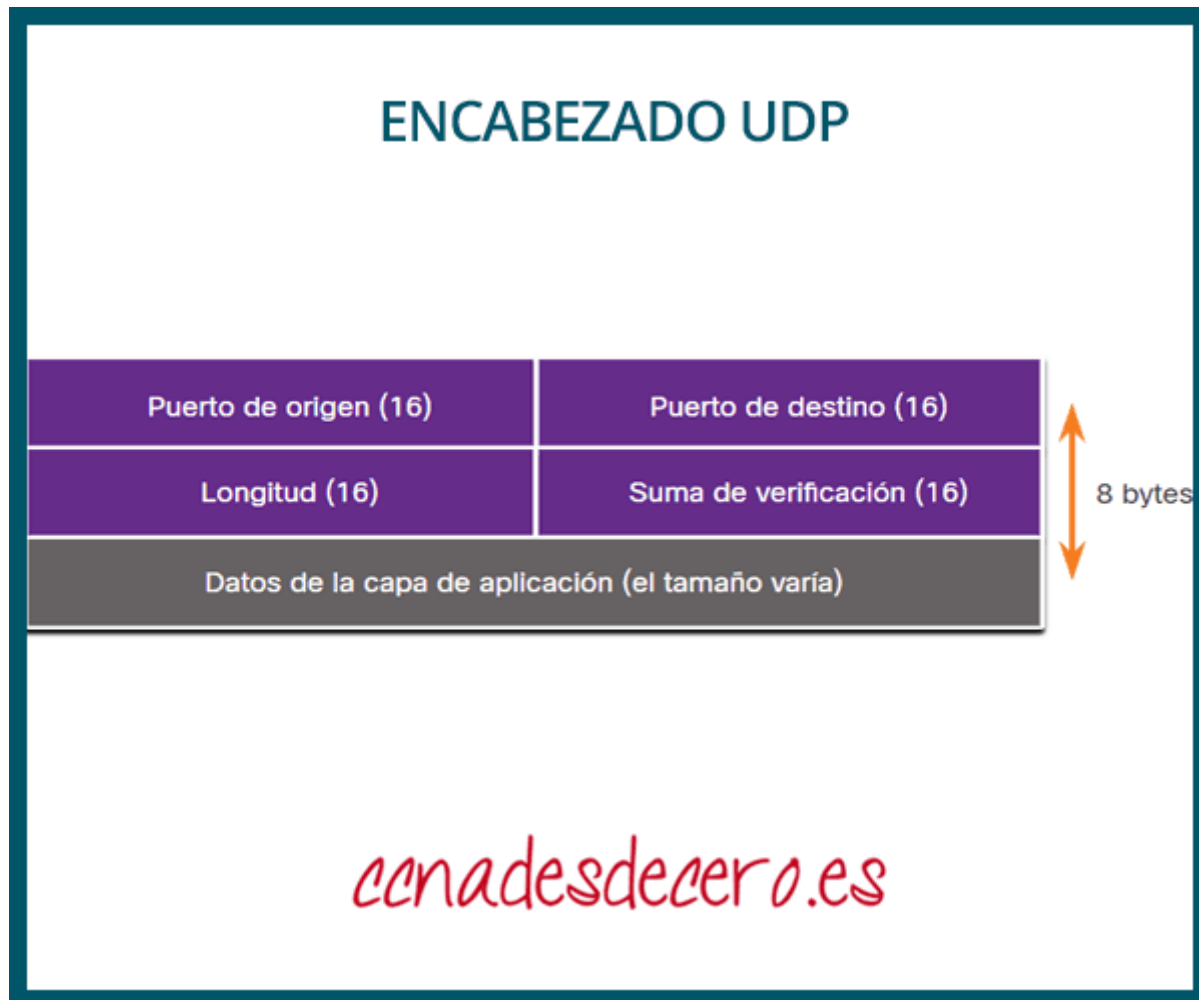


Además, las comunicaciones TCP indican el orden en que deben recibirse los paquetes de datos, y confirman que los paquetes están llegando según lo previsto. Si un paquete no llega, por la congestión de las redes intermedias, por ejemplo, el TCP exige que se vuelva a enviar. Las comunicaciones UDP no incluyen ninguna de estas funciones.

Estas diferencias generan algunas ventajas. Ya que el UDP no necesita un "protocolo de enlace" ni comprobar si los datos llegan correctamente, puede transferir datos mucho más rápido que el TCP.

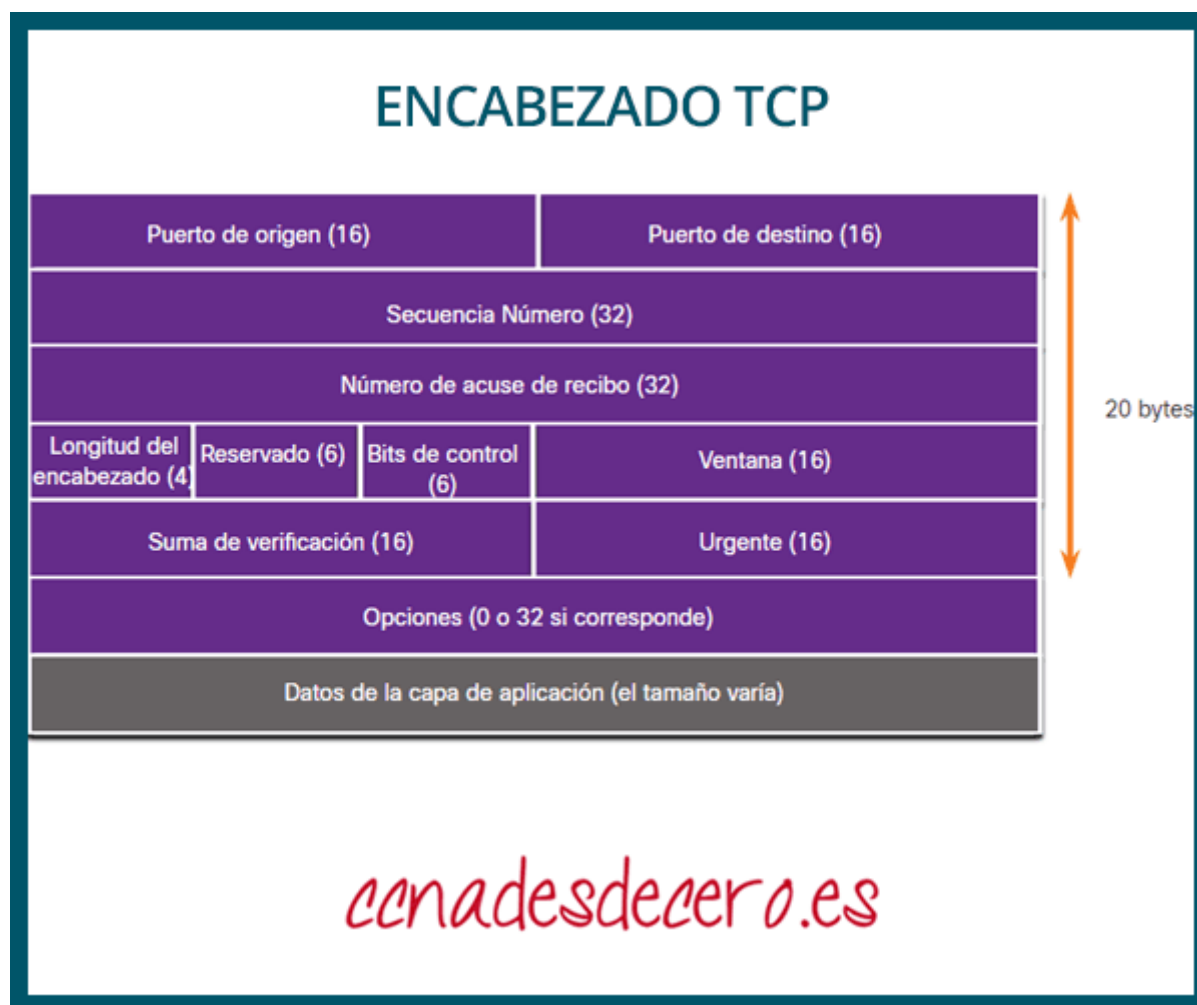
Sin embargo, esta velocidad genera desventajas. Si un datagrama del UDP se pierde en tránsito, no se volverá a enviar. Como resultado, las aplicaciones que utilizan UDP deben ser capaces de tolerar errores, pérdidas y duplicaciones.

(Técnicamente, esta pérdida de paquetes no es tanto un defecto del UDP como una consecuencia de cómo se ha diseñado Internet. La mayoría de los enrutadores de red, por su propio diseño, no realizan el ordenamiento de los paquetes y la confirmación de su llegada, porque hacerlo requeriría una cantidad inviable de memoria adicional. El TCP es una forma de llenar este vacío cuando una aplicación lo requiere).



La tabla identifica y describe los cuatro campos en un encabezado UDP.

Campo de encabezado UDP Descripción Puerto de origen Un campo de 16 bits utilizado para identificar la aplicación de origen por número de puerto. Puerto de destino Un campo de 16 bits utilizado para identificar la aplicación de destino por número de puerto. Longitud Un campo de 16 bits que indica la longitud del encabezado del datagrama UDP. Suma de comprobación Un campo de 16 bits utilizado para la verificación de errores del encabezado y los datos del datagrama.

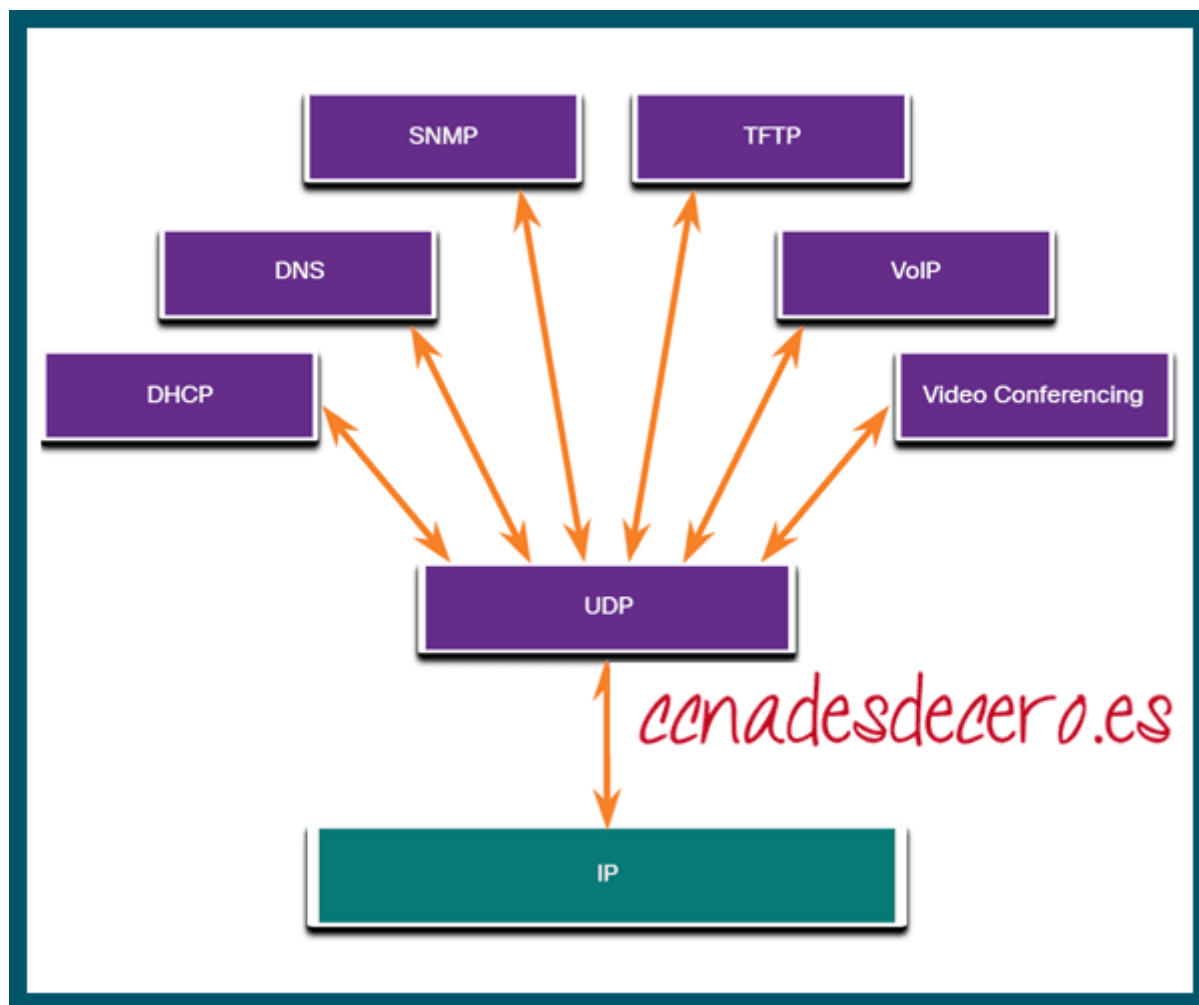


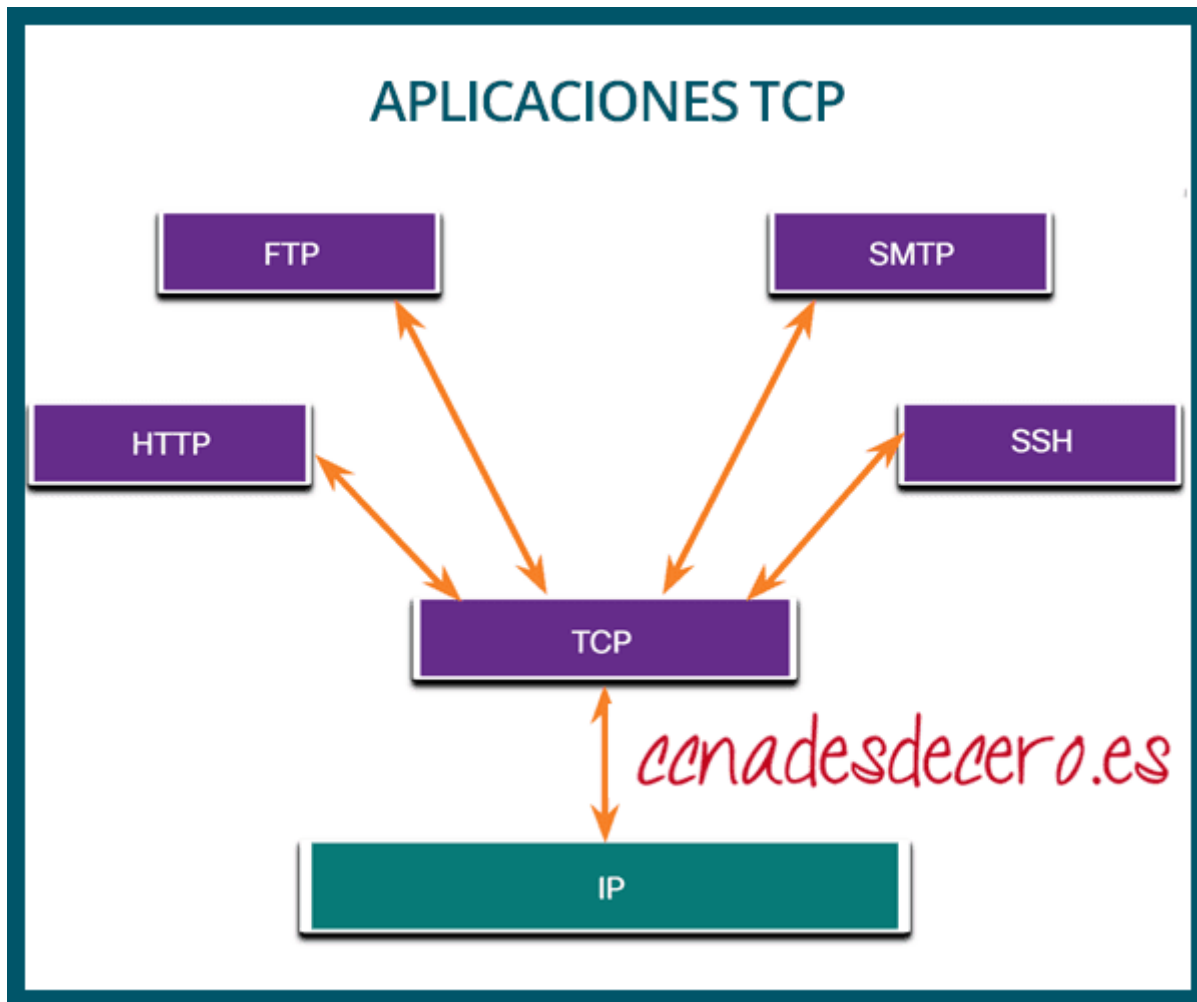
Componentes del encabezado TCP El encabezado del protocolo TCP está formado por una serie de componentes. Veamos lo que significan las diferentes secuencias de bits en un segmento TCP:

- Puerto de origen (16 bits): se refiere la aplicación que está en curso en el dispositivo de origen.
- Puerto de destino (16 bits): relativo la aplicación que está en curso en el dispositivo de destino.
- Número de secuencia (32 bits): son las secuencias de números que cliente y servidor se envían para que se pueda establecer la comunicación.
- Número de acuse de recibo (32 bits): indica el número de secuencia del siguiente segmento que se espera recibir.
- Margen de datos (4 bits): determina el lugar del segmento donde comienzan los datos.
- Reservado (6 bits): es un campo reservado para futuros usos. Su valor siempre es igual a cero.
- Indicadores (6 bit): Ofrecen información adicional::
 - URG: cuando su valor es 1, se trata de un segmento que se ha de procesar de manera urgente.
 - ACK: cuando su valor es 1, el segmento consiste en un acuse de recibo.
 - PSH: cuando su valor es 1, el paquete procesa los datos sin esperar a completar el segmento
 - RST: cuando su valor es 1, se reanuda la conexión.
 - SYN: cuando su valor es 1, se solicita un pedido para establecer una conexión.
 - FIN: cuando su valor es 1, determina el fin de la conexión..
- Ventana (16 bits): determina la cantidad de bytes que desea recibir el receptor sin acuse de recibo.
- Suma de control (24 bit): verifica la integridad del paquete a través de la comprobación de errores.
- Puntero urgente (16 bits): se utiliza para enviar datos urgentes e indica la ubicación de los datos que siguen a los urgentes.
- Opciones (bits variables): para definir funciones TCP que no forman parte del encabezado general.

- Relleno: su función es asegurar que la longitud de la cabecera sea múltiplo de 32 bits.

Aplicaciones UDP Y TCP





Protocolo HTTP

¿Qué es el HTTP? Todas las direcciones en Internet comienzan por `http://` (o `https://`). Estas siglas hacen referencia al protocolo HTTP, que es el que utiliza el navegador para acceder a una página web. ¿Qué es HTTP, en qué se diferencian las distintas versiones y qué otros conceptos están relacionados con el protocolo?

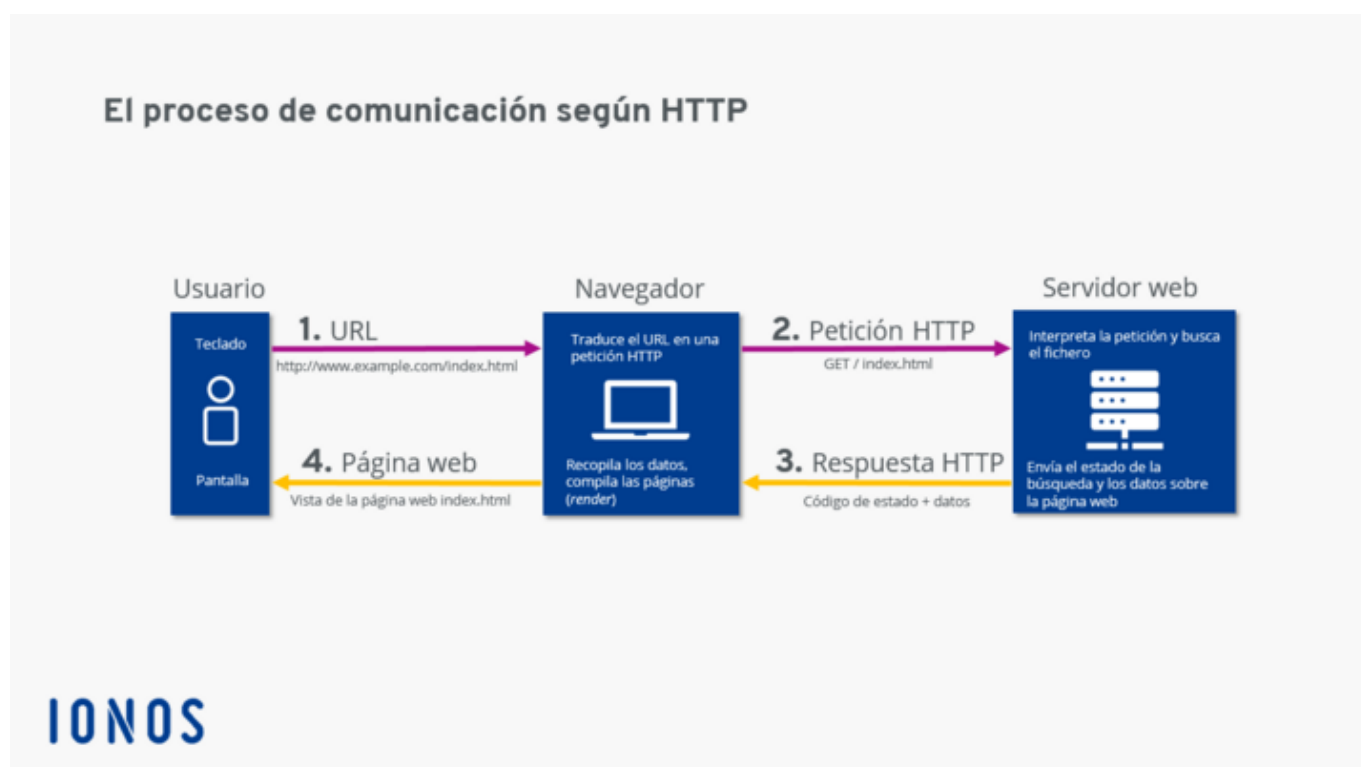
¿Qué significa HTTP? HTTP son las siglas de Hypertext Transfer Protocol, es decir, Protocolo de Transferencia de Hipertexto. Este concepto es uno de los que Tim Berners-Lee desarrolló en el CERN (Suiza) y formaron la base de la World Wide Web: los otros dos son HTML y URI. Mientras que HTML (Hypertext Markup Language) define la estructura de las páginas web, la dirección URL (Uniform Resource Locator), una forma derivada del URI, define cómo se localiza a un recurso (p. ej., una página web) en Internet. El protocolo HTTP, en cambio, regula cómo el servidor envía este recurso al cliente.

Pero ¿qué significa hipertexto, ese término que forma parte de las siglas HTTP y HTML? Se trata de un concepto que en realidad todos conocemos: el enlace a otros archivos, como los hiperenlaces que se usan en las páginas web para redirigir a otras páginas.

¿Qué función cumple HTTP? Cuando escribes una dirección web en tu navegador y se abre la página que desees, es porque tu navegador se ha comunicado con el servidor web por HTTP. Dicho de otra manera, el protocolo HTTP es el código o lenguaje en el que el navegador le comunica al servidor qué página quiere visualizar.

¿Cómo funciona HTTP? La manera más fácil de explicar cómo funciona HTTP es describiendo cómo se abre una página web:

En la barra de direcciones del navegador, el usuario teclea `example.com`. El navegador envía esa solicitud, es decir, la petición HTTP, al servidor web que administre el dominio `example.com`. Normalmente, la solicitud del cliente dice algo así como “Envíame este archivo”, pero también puede ser simplemente “¿Tienes este archivo?”. El servidor web recibe la solicitud HTTP, busca el archivo en cuestión (en nuestro ejemplo, la página de inicio de `example.com`, que corresponde al archivo `index.html`) y envía en primer lugar una cabecera o header. Esta cabecera le comunica al cliente, mediante un código de estado, el resultado de la búsqueda. Para conocer más detalles acerca de los códigos de estado, no te pierdas nuestro artículo al respecto. Si se ha encontrado el archivo solicitado y el cliente ha solicitado recibirlo (y no solo saber si existe), el servidor envía, tras el header, el message bodyo cuerpo del mensaje, es decir, el contenido solicitado: en nuestro ejemplo, el archivo `index.html`. El navegador recibe el archivo y lo abre en forma de página web.



¿Para qué se usa HTTP? Cuando se creó, el protocolo HTTP solo servía para solicitar documentos HTML a un servidor web. Hoy en día, en cambio, se usa con gran variedad de fines:

Los navegadores usan HTTP para solicitar cualquier tipo de archivo habitual en las webs modernas: de texto, de vídeo, de código de programación, etc. Los programas de aplicación utilizan HTTP para cargar archivos y actualizaciones de servidores lejanos. La API basada en REST es una solución que utiliza HTTP para controlar servicios web. Otra tecnología que se basa en HTTP es WebDAV. En la comunicación de máquina a máquina se utiliza HTTP como protocolo de comunicación entre servicios web. Los reproductores multimedia también utilizan HTTP. Las operaciones de acceso a bases de datos en la web y, por lo tanto, también las operaciones CRUD, pueden realizarse también mediante HTTP.

API REST

API es el acrónimo de interfaz de programación de aplicaciones (application programming interface en inglés). Es un conjunto de reglas bien definidas que se utilizan para especificar formalmente la comunicación entre dos componentes de software.

En la actualidad existen distintas clases de API, hablaremos acerca de las API REST

Una API REST es una interfaz de comunicación entre sistemas de información que usa el protocolo de transferencia de hipertexto (hypertext transfer protocol o HTTP, por su siglas en inglés) para obtener datos o ejecutar operaciones sobre dichos datos en diversos formatos, como pueden ser XML o JSON. [JSON](#) [XML](#) [HTML](#)

Se basa en el modelo cliente-servidor donde el cliente es el que solicita obtener los recursos o realizar alguna operación sobre dichos datos, mientras que el servidor es aquel ente que entrega o procesa dichos datos a solicitud del cliente.

Criterios de API REST

Existen diversos criterios para identificar si una API es REST o no. Algunos de ellos son que:

- Debe usar una arquitectura cliente-servidor.
- Las ejecuciones de la API no deben considerar el estado del cliente, el estado de peticiones anteriores o algún indicador almacenado que haga variar su comportamiento. La comunicación debe ser sin estado (stateless).
- Ha de estar orientada a recursos, usando las operaciones estándar de los verbos HTTP.
- Hace uso de la URL como identificador único de los recursos.
- Debe ser hipermedia: cuando se consulte un recurso, este debe contener links o hipervínculos de acciones o recursos que lo complementen.

Diferencia entre RESTful y RESTless A menudo escuchamos estos términos al momento de diseñar, construir o interactuar con API. La diferencia es sencilla. Llamamos RESTful a todas aquellas API que cumplen completamente los criterios REST; mientras que llamamos RESTless a aquellas API que no cumplen del todo con los criterios REST.

Por ejemplo, una API que utiliza el verbo POST para todas sus operaciones no es una API RESTful, si no una API RESTless.

Los componentes primordiales que se describen en un API Spec son los siguientes:

Accion (Verbo) HTTP Son aquellos verbos propios del protocolo HTTP que fueron tomados para definir operaciones muy puntuales y específicas sobre los recursos de la API.

Los más utilizados son:

- GET: listado de recursos. Detalle de un solo recurso.
- POST: creación de un recurso.
- PUT: modificación total de un recurso.
- PATCH: modificación parcial de un recurso.

- **DELETE:** eliminación de un recurso. En muchas ocasiones es un soft delete, es decir, no se elimina definitivamente un recurso sino que únicamente es marcado como eliminado o desactivado.
- **URL orientada a recursos:** la definición de las URL de los endpoint de la API están orientadas a recursos, es decir, a entidades que tienen coherencia dentro del contexto de la API. Por ejemplo, en una API para un sistema que administra libros sería fácil encontrar entidades como libros, autores, editoriales, colecciones, etc. Estas entidades las veremos reflejadas en URL orientadas a recursos que las representen, por ejemplo:
 - `/api-libros/v0/autores`: identifica los recursos autores
 - `/api-libros/v0/autores/{id-autor}`: identifica un recurso autor
 - `/api-libros/v0/autores/{id-autor}/libros`: identifica los libros de un autor en específico
 - `/api-libros/v0/libros`: identifica los recursos libros
 - `/api-libros/v0/editoriales`: identifica los recursos editoriales
 - `/api-libros/v0/editoriales/{id-editorial}/libros`: identifica los libros de una editorial

HTTP Status Son aquellos estatus de respuesta propios del protocolo HTTP que fueron tomados para informar sobre el resultado de la operación solicitada. Los más comunes en API REST son:

200 - OK 201 - Created 204 - No Content 400 - Bad Request 401 - Unauthorized 403 - Forbidden 404 - Not Found 500 - Internal Server Error

A continuación, adjuntamos una tabla de ejemplo para ilustrar estos conceptos:

VERBO HTTP	URL DEL RECURSO	ACCIÓN	HTTP STATUS
GET	/libros	Obtener una lista de libros	200 - OK
			204 - Not Content: cuando no hay libros a listar
GET	/libros/{id-libro}	Obtener detalle de un libro	200 - OK
			404 - Not Found: cuando no existe el libro buscado
POST	/libros	Crear un recurso nuevo libro	201 - Created
PUT	/libros/{id-libro}	Modificar un recurso libro completamente	200 - OK
			400 - Bad request: cuando algún parámetro enviado no cumple con las reglas
PATCH	/libros/{id-libro}	Modificar un recurso libro parcialmente	200 - OK
			201 - Created: si el recurso a modificar no existe se crea en automático

VERBO HTTP	URL DEL RECURSO	ACCIÓN	HTTP STATUS
			400 - Bad request: cuando algún parámetro enviado no cumple con las reglas
DELETE	/libros/{id-libro}	Eliminar un recurso libro	201 - Not Content: es el status standard a regresar en un verbo DELETE

Para realizar practicas de ordenes GET,POST.... recomendamos el siguiente software
<https://insomnia.rest/download> (la version gratuita)