



Ficha de ayuda: Cómo funciona una aplicación web

Arquitectura cliente-servidor

Una aplicación web se diseña siguiendo ciertos principios. Estos definen los elementos que conforman una aplicación y las relaciones entre ellos.

Estos principios se denominan **arquitectura**, que es la forma en que se crea una aplicación.

El tipo de arquitectura de aplicaciones web más común es el de **arquitectura cliente-servidor**.

Un cliente, un servidor e Internet son los elementos que conforman la arquitectura cliente-servidor. El cliente y el servidor operan de forma independiente. El cliente se encarga de la interacción con el usuario. El servidor realiza las operaciones lógicas, los cálculos y el almacenamiento de datos. Internet conecta al cliente y al servidor.

Un cliente es un sistema que se conecta a un servidor y transforma las acciones del usuario o usuaria en solicitudes.

Un servidor es un sistema que procesa las solicitudes de un cliente y genera una respuesta. Por ejemplo, guarda un pedido o envía información sobre su precio.

El cliente y el servidor se comunican a través de una red.

Internet, o la **red**, es un sistema de dispositivos interconectados que ayuda al cliente y al servidor a intercambiar datos.

Aplicación web

Un navegador es un software que envía solicitudes a un servidor, recibe respuestas, procesa la información y muestra los datos en forma de página web.

El usuario o usuaria puede realizar acciones en una aplicación web mediante el uso de un navegador.

Una aplicación web es:

- Un conjunto de páginas web que un usuario o usuaria ve en un navegador. Podemos mencionar como ejemplos la página principal de TripleTen y la página de inicio del bootcamp de QA engineer.
- Un conjunto de operaciones y cálculos lógicos que procesan las solicitudes de un usuario o usuaria. Por ejemplo, al hacer clic en el botón "Elegir un bootcamp", el navegador envía tu solicitud a un servidor, obtiene una respuesta y muestra una lista de todos los bootcamps de TripleTen en la página correspondiente.

Las páginas web las desarrollan profesionales de **desarrollo de front-end**. El código de las operaciones y cálculos que procesan las solicitudes y generan las respuestas lo escriben profesionales de **desarrollo de back-end**.

Front-end y back-end

El front end es la interfaz con la que el usuario o usuaria interactúa directamente. Incluye todo lo que se puede ver, tocar y experimentar en su navegador: desde los botones y formularios hasta las animaciones y transiciones. En el contexto del modelo cliente-servidor, el front-end se refiere al código que se ejecuta en el dispositivo del usuario o usuaria.

El back-end es la parte no visible al usuario o usuaria, pero esencial para el funcionamiento de la aplicación. Maneja procesos como cálculos, lógica de negocio y gestión de bases de datos. Si se trata de un modelo cliente-servidor, el back-end es el código que se ejecuta en un servidor remoto.

URL

Una URL (Uniform Resource Locator) es la dirección de un recurso web. Una URL muestra la ubicación de una aplicación web, una página web o un fragmento suyo, e indica cómo solicitarlos.

Un nombre de dominio es la dirección de una aplicación web que se utiliza para registrar esta aplicación en Internet. Los nombres de dominio están compuestos por

caracteres asignados a una aplicación por el sistema de nombres de dominio (Domain Name System o DNS, por sus siglas en inglés).

Al ingresar una URL en la barra de direcciones, el navegador envía una solicitud a un servidor **DNS**.

Una dirección IP es el identificador único de un servidor que almacena la información necesaria.

HTTP y HTTPS

Un protocolo de transmisión de datos es un conjunto de reglas que regulan el intercambio de datos entre dispositivos.

Uno de estos protocolos es el **Protocolo de transferencia de hipertexto**, o **HTTP** (de sus siglas en inglés, HyperText Transfer Protocol). Hoy en día, se puede utilizar HTTP para la transferencia tanto de hipertexto como de otros tipos de datos, por ejemplo, imágenes, audio y video.

Para garantizar la seguridad, los sitios web emplean un HTTP con protección ampliada que se denomina **HTTPS** (por sus siglas en inglés, HyperText Transfer Protocol Secure). Encripta una conexión mediante protocolos criptográficos. De este modo, un cliente y un servidor pueden intercambiar mensajes entre sí de forma segura.

Estructura de una solicitud HTTP

El front-end usa HTTP para enviar solicitudes, mientras que el back-end lo utiliza para enviar respuestas.

Una solicitud es un mensaje especial. El front-end lo emplea para pedir al back-end que realice acciones específicas con los datos, por ejemplo, que los envíe, almacene o modifique.

Una solicitud está estructurada según las reglas de transmisión de datos de HTTP. Consta de cuatro partes:

- una línea de inicio;
- encabezados;
- cuerpo del mensaje;
- una línea divisoria vacía.

Línea de inicio

La **línea de inicio** tiene tres elementos, que son: el método, la ruta de acceso a un recurso y la versión del protocolo.

- **El método** indica una acción que procesa el back-end. Los métodos más comunes son GET, POST, PUT, DELETE.
- **La ruta de acceso a un recurso** es una dirección que el front-end utiliza para enviar una solicitud al back-end.
- **La versión del protocolo** es el número de una versión HTTP. Actualmente, es HTTP/1.1.

Encabezados de solicitud

Los **encabezados de solicitud** representan información adicional que el front-end envía al back-end.

Hay:

- encabezados generales;
- encabezados de solicitud;
- encabezados de entidad.

Los **encabezados de solicitud** contienen información adicional sobre el cliente.

Los **encabezados generales** contienen parámetros que se transfieren tanto en solicitudes como en respuestas.

Los **encabezados de entidad** establecen las características del contenido dentro del cuerpo del mensaje.

Cuerpo del mensaje

El **cuerpo del mensaje** es la información que envía el front-end.

Estructura de una respuesta HTTP

Una **respuesta HTTP** incluye:

- una línea de estado;
- encabezados;

- cuerpo del mensaje.

Línea de estado

La **línea de estado** incluye la versión del protocolo, un código de estado y un mensaje de estado.

Por ejemplo: `HTTP/1.1 200 OK`.

- **La versión del protocolo** corresponde al número de la versión HTTP respecto a la cual se genera una respuesta. Por ejemplo, "HTTP/1.1".
- **El código de estado** es un número de tres dígitos que indica el resultado de la solicitud. Ejemplo: 200.
- **El mensaje de estado** es un texto breve explicativo que va junto con el código de estado. Por ejemplo, "OK".

Los desarrolladores utilizan una lista estandarizada de códigos y mensajes de estado: [Hypertext Transfer Protocol \(protocolo de transferencia de hipertexto\) — HTTP/1.1](#).

Encabezados de respuesta

Los encabezados de respuesta contienen más información sobre el back-end y el tipo de datos para el front-end. La estructura de los encabezados es similar a la del front-end.

- **Los encabezados generales** contienen parámetros que se transmiten tanto en una solicitud como en una respuesta. Ejemplo: "Connection" es el estado de la conexión entre un cliente y un servidor, y "Date" es la fecha de creación de la respuesta del servidor.
- **Los encabezados de respuesta** permiten a un servidor transmitir información adicional que no cabe en la línea de estado. Ejemplo: "Server" es el software que utiliza un servidor para gestionar la solicitud y generar la respuesta, y "Location" indica la ubicación exacta de un recurso.
- **Los encabezados de entidad** establecen las características del contenido dentro del cuerpo del mensaje. Ejemplo: "Content-Type" es el tipo de datos que contiene la respuesta.

Cuerpo del mensaje

El **cuerpo del mensaje** es utilizado por el back-end para comunicar el resultado obtenido del procesamiento de una solicitud.

Revisión de solicitudes del front-end y respuestas del back-end

Por lo general, las solicitudes y las respuestas se visualizan en las DevTools (herramientas de desarrollo web). **DevTools** está disponible en los navegadores modernos.

- Para abrir DevTools en Google Chrome, haz clic con el botón derecho en cualquier lugar de la página web y elige la opción "Inspeccionar", o usa el atajo `Ctrl+Shift+i` para Windows, `Cmd+Opt+J` para macOS.
- En la parte superior de la página verás las siguientes pestañas: Elementos, Consola, Fuentes, Red, Rendimiento, Memoria, Aplicación y Lighthouse. Estas son las herramientas de desarrollo. Cada pestaña tiene sus propias capacidades.

La pestaña "Red" en las herramientas de desarrollo despliega información detallada acerca de los recursos intercambiados entre el front-end y el back-end:

- Los nombres de los recursos transmitidos a través de la red por el front-end y el back-end (la columna "Name").
- El tiempo de carga (columna "Time").
- El tamaño de los archivos (columna "Size").
- La información adicional sobre las solicitudes al servidor, por ejemplo, el volumen de tráfico.

Cookies

Las cookies son los datos que se almacenan en el dispositivo del usuario o usuaria. El servidor genera los datos, que se almacenan en su computadora o smartphone. Cuando el usuario o usuaria vuelve a abrir la aplicación, el servidor recibe de nuevo los datos.

Imagina que te vas de vacaciones y eliges un hotel mediante la aplicación web de una plataforma de búsqueda de hoteles. Al abrir la página principal de la plataforma, el navegador no solo envía una solicitud al servidor, sino que también busca las cookies

de la aplicación en tu dispositivo. Si habías utilizado esta plataforma de búsqueda antes, el navegador enviará las cookies guardadas al servidor. Por ejemplo, las cookies pueden contener información sobre si el usuario está registrado.

El servidor procesará la solicitud y generará una respuesta basada en estos datos, por ejemplo, añadirá una lista de tus reservas a la respuesta. Sin las cookies, el servidor no enviará los datos, porque te identificará como un nuevo usuario.

¿Qué información almacenan las cookies?

Las cookies pueden almacenar diferentes datos:

1. Datos para la autenticación

Por ejemplo, inicias sesión en una aplicación web, luego cierras la pestaña y, después de un tiempo, vuelves a abrir la aplicación. No tendrás que introducir tu nombre de usuario o usuaria y contraseña una vez más, porque los datos de la cuenta se almacenan en cookies.

2. Datos de usuario o usuaria

Por ejemplo, información sobre hoteles, geolocalización, idioma y moneda que el usuario o usuaria ha utilizado para sus consultas.

3. Datos para recopilar estadísticas

Por ejemplo, estadísticas sobre el dispositivo o las páginas visitadas por el usuario o usuaria. Basándose en las estadísticas, el equipo de desarrollo de la aplicación saca conclusiones y toma decisiones sobre cómo funciona una funcionalidad concreta.

Cómo ver las cookies

Las cookies son pequeños archivos de texto que puedes ver directamente en el navegador.

Almacenamiento local

El **almacenamiento local** consiste de datos almacenados en el navegador. Esto es diferente de las cookies, porque los datos del Almacenamiento Local no se transfieren al servidor: los utiliza solo el lado del cliente de la aplicación.

Por ejemplo, rellenas un formulario de reserva de hotel e introduces tu nombre completo, correo electrónico y número de teléfono. Si el front-end de la aplicación

almacena datos en el almacenamiento local, no se perderán incluso después de actualizar la página. Si decides reservar otro hotel y volver al formulario de reserva, tu nombre y apellido, correo electrónico y número de teléfono se cargarán desde el almacenamiento local. Entonces no tendrás que introducirlos de nuevo.

Cómo ver datos del almacenamiento local

Data from Local Storage can be viewed in DevTools.

Selecciona la pestaña Aplicación → busca la sección Almacenamiento (Storage) → abre la lista desplegable Almacenamiento local (Local Storage) → haz clic en la URL de la aplicación.

Cómo borrar datos del almacenamiento local

Ahora imagina que tienes que probar la aplicación en un nuevo usuario. El usuario no tiene la tarifa y los requisitos seleccionados, por lo que debes eliminar todos los datos de almacenamiento local:

1. Abre DevTools → busca la lista desplegable "Local Storage" (Almacenamiento local) → haz clic con el botón derecho en la URL de la aplicación.
2. Haz clic en "Eliminar".

Pruebas de caché

Caché (almacenamiento en caché) son los datos de una página web. Cuando abres imágenes, archivos de audio, video, CSS, HTML y JS, se almacenan en tu dispositivo.

Las cookies y el almacenamiento local facilitan el uso de la aplicación. El usuario o usuaria no necesita ingresar los mismos datos cada vez, por ejemplo, el nombre de usuario o usuaria, la contraseña y el número de teléfono.

La memoria caché del navegador tiene otro propósito. Cuando abres una página web de nuevo, se cargará más rápido porque el navegador tomará los archivos del almacenamiento en caché y no del servidor.

A veces, la causa de los errores está en la memoria caché. Imagina que encuentras un error y creas un ticket en el rastreador de errores. El área de desarrollo corrige el error y devuelve el problema para probarlo. Ahora debes asegurarte de que todo funcione bien. Pruebas la aplicación de nuevo, pero nada cambia. El error aún ocurre, pero el área de desarrollo afirma haberlo corregido.

Lo más probable es que los datos simplemente se almacenaron en la memoria caché de tu navegador. Es por eso que ves la versión anterior de la página desde tu computadora y no la nueva versión desde el servidor. Para solucionar el problema, tienes que borrar la memoria caché. Sigue estos pasos para practicar:

1. Abre Google Chrome.
2. En la esquina superior derecha, abre el menú del navegador y selecciona "Historial".
3. Haz clic en "Borrar datos de navegación".
4. Marca solo una casilla: "Archivos e imágenes en caché".
5. Establece el intervalo de tiempo: "Desde siempre".
6. Haz clic en "Borrar datos".

Puedes encontrar la guía para otros navegadores en Internet.

Después de la actualización de la aplicación, es posible que una parte de los archivos se descargue del servidor y otra del almacenamiento en caché. Esto puede causar algunos problemas, por ejemplo, con el diseño de la aplicación.

Si notas este error en producción, no te apresures a borrar la memoria caché. Las personas que usarán la aplicación podrán encontrarse con el mismo problema. Informa al equipo de desarrollo y te ayudarán a descubrir qué hacer.