**DESARROLLADOR FRONTEND:**

• Developer que se encarga de todo lo que pasa del lado del cliente.

• Implementa la estructura y la estética (bocetos y diseños del sitio).

• La adaptación a dispositivos (que el sitio sea visible en cualquier dispositivo).

• Y el trabajo en equipo (comunicación constante entre el desarrollador Frontend y Backend).

• Lenguajes: HTML5 (estructura), CSS3 (estilos visuales) y JavaScript5 (interacción dentro del sitio).

**ARQUITECTURA CLIENTE-SERVIDOR:**

• Modelo de comunicación que vincula a varios dispositivos con un servidor a través de una red, el internet.

• Objetivo: que fluya la comunicación bajo estrictos estándares de organización.

• Para que uno de estos actores responda, primero necesita escuchar atentamente lo que el otro está requiriendo.

**Elementos de la arquitectura:**

• Cliente: dispositivos que hacen peticiones de servicios o recursos a un servidor. Dentro de internet, suele acceder a estos servicios y recursos a través de un navegador web (browser). Ej.: computadora, teléfono celular, tablet, consola de videojuegos o cualquier implemento que pueda conectarse a una red.

• Servidor: equipo que brinda los servicios y recursos a los que acceden los clientes, es quien responde los pedidos del cliente.

• Webserver: programa que se encarga de procesar los pedidos y generar y enviar las respuestas.

• Pedidos (requests): solicitudes que hacemos a través del navegador (cliente) a un servidor. Ej.: la página de Facebook almacenada en sus servidores.

• Respuestas (responses): el servidor recibe nuestra solicitud, la procesa y envía como resultado una respuesta al cliente (navegador). Ej.: devolverá la página principal del sitio.

**Importancia Flujo Request-Response**: dentro del mundo del desarrollo web, la mayoría de las aplicaciones tienen dos claros frentes:

• Frontend: todo lo que pasa del lado del cliente (en el navegador). Se incluyen todos los elementos gráficos que conforman la interfaz del sitio y también parte de su funcionalidad.

• Backend: todo lo que pasa del lado del servidor. Se incluye todo el funcionamiento interno y lógica del sitio, es lo que permite que se carguen todas las peticiones solicitadas por el cliente. Se puede encontrar: bases de datos (MySql, MongoDB), lenguajes (PHP y JavaScript) y frameworks o marcos de trabajo (Express, Lavarel). Hoy existe la posibilidad de correr JS del lado del servidor, permitiéndonos programar en un mismo lenguaje tanto en Front como en Back, logrando que el proceso de desarrollo sea más fluido.

• Importante: se puede tener en cuenta que la misma computadora puede ser el cliente y el servidor al mismo tiempo, es lo más normal en el entorno de desarrollo de un sitio o aplicación web. Al instalar un webserver en la computadora, se generará una URL especial que permitirá simular el flujo de request – response, así se puede tener un entorno local donde se pueda desarrollar y hacer pruebas antes de subir el sitio a un servidor real.

**DIFERENCIAS ENTRE INTERNET Y WEB:**

|  |  |
| --- | --- |
| Web: | Internet: |
| Sistema de distribución que interconecta documentos de hipertexto. Es un modelo que esta construido sobre internet. | Red descentralizada que conecta computadores y servidores entre sí. Se construyó en principio para fines científicos conectando universidades con instituciones estatales (Proyecto ARPANET desarrollado por el Dpto. de Defensa de los EE.UU en 1969). |
| Utiliza protocolo HTTP. | Utiliza protocolo TPC/IP, entre otros. |
| Información y archivos. | Infraestructura. |
| Ambos términos están relacionados porque uno necesita del otro para poder existir, pero son cosas diferentes. | |

**HERRAMIENTAS DE UN DESARROLLADOR WEB:**

• Wireframe: del departamento de diseño recibiremos un boceto donde se representa visualmente, de forma muy sencilla y esquemática, la estructura de una página web.

• Escribir código: una vez identificados los componentes de ese boceto, escribimos el código necesario utilizando algún IDE (entorno de desarrollo integral). Ej.: VSCode.

• Navegadores: el navegador se encarga de interpretar nuestro código y podemos ir visualizando como va quedando nuestro proyecto. Es, además, el programa que utilizaran los usuarios para interactuar con las páginas web.

• Dispositivos: el sitio web que desarrollamos debe adaptarse a diferentes dispositivos. Como frontend developers debemos asegurarnos de su correcto funcionamiento en celulares, tablets y pantallas grandes.

**Wireframes:** planos de un sitio o aplicación, nos dicen cómo está conformada la estructura y qué es cada elemento. Permiten que todas las personas involucradas en el proyecto tengan una clara referencia de cómo se espera que quede armado el sitio o la app. Son representaciones de baja calidad y el objetivo es demostrar donde se encontrarán los elementos de la página y como el usuario interactúa con ella. Da una idea rápida de cómo va andar el sitio, podemos probarlo y mejorar la navegación, también permite ver la ubicación del contenido, modificarlo o corregirlo, rápidamente. Da un elemento sobre el cual debatir y plantear nuevas ideas.

**•** Figma: herramienta de prototipado con la cual podemos crear una interfaz de manera colaborativa. Esta principalmente creada para realizar interfaces de apps mobile o sitios web, pero no quita que podamos realizar cualquier tipo de diseño. Puede ser utilizada sin la necesidad de bajar una app, esto quiere decir que se puede utilizar mediante un navegador web. Como desarrolladores no tenemos que ser diseñadores pero es importante saber usarlo para conocer qué elementos necesitamos para el bocetado de nuestra app o sitio.

**Estructura de carpetas:**

• Root: carpeta raíz de nuestro proyecto.

* CSS: carpeta con las hojas de estilos .css.
* style.css: nombre más utilizado para nuestros archivos css.
* img: carpeta de imágenes, van todos los archivos de imagen (jpg, gif, png) que utilicemos en nuestro proyecto.
* index.html: página de inicio de nuestro proyecto.

• Importante: con dicha estructura, vamos a poder separar por responsabilidades cada uno de los elementos de nuestro proyecto y de esta manera, gestionar mejor nuestro trabajo y el del equipo. Mantener una estructura similar en todos nuestros trabajos nos facilita las tareas de planificación y mantenimiento.

**IDE**: conjunto de herramientas diseñadas para facilitarnos la creación y el desarrollo de nuestros programas o apps.

**Navegadores y dispositivos:**

• Navegadores: son intérpretes de código, a través de estos programas podemos acceder y navegar por internet.

• Compatibilidad: forma en la que el navegador interpreta y muestra las páginas web. Está establecida por el consorcio W3C (organización de estándares de internet). Conocer los distintos tipos de navegadores, así como sus incompatibilidades, es importante a la hora de desarrollar una página web. No todos los usuarios de un sitio web visitan la página con el mismo navegador que los creadores utilizan. Esto debe tenerse en cuenta para evitar que haya usuarios que abandonen nuestro sitio porque no se puede visualizar correctamente.

• Responsive design: se refiere a hacer que un sitio web sea accesible y adaptable en todos los dispositivos: tablets, celulares, pantallas grandes, etc.