Formación Desarrollo Web

1. Cómo funcionan los sitios web

Desde el momento en que el usuario abre un navegador web, introduce una URL y presiona ENTER, se cumple el siguiente proceso:

a. La URL se "resuelve"
b. Una petición es enviada al servidor del sitio web
c. La respuesta del servidor es analizada
d. La página se renderiza y se muestra en el navegador

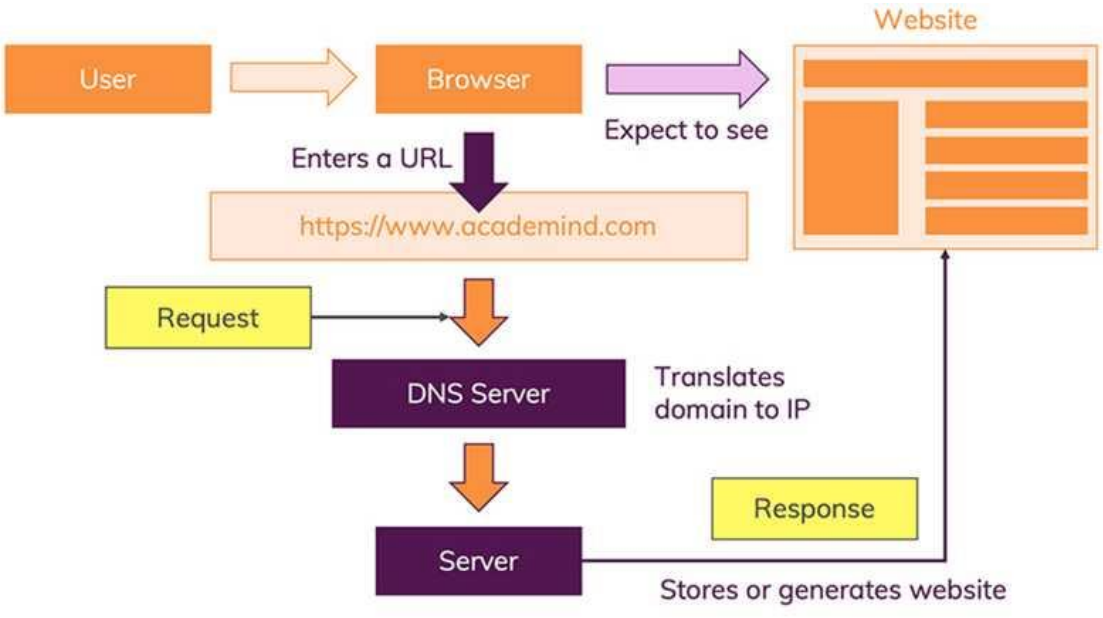

Ilustración 1 - Cómo funcionan los sitios web

2. *Server-side vs Browser-side*

Diferencia entre las dos "partes" cuando se habla de la web:

a. *Server-side*: lenguajes de programación del lado del servidor que no se ejecutan en navegadores, pero se pueden ejecutar en un ordenador normal (un servidor es, a fin de cuentas, un ordenador normal). Algunos ejemplos de lenguajes de programación del lado del servidor son Node.js, PHP y Python
b. *Browser-side*: lenguajes que se ejecutan en el navegador (HTML, CSS y JavaScript)

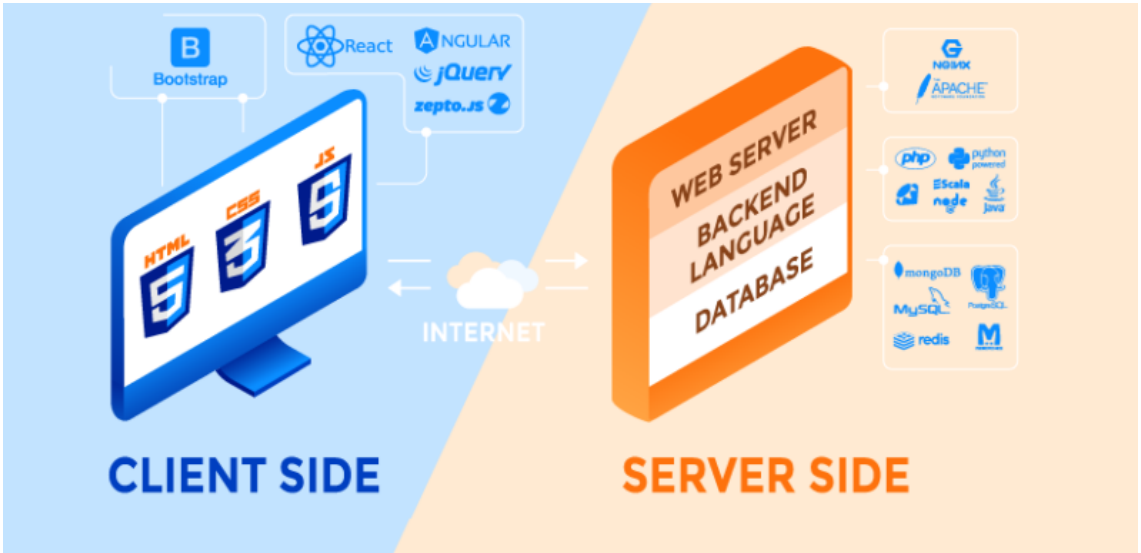

Ilustración 2 - Server-side vs Browser-side

3. Tecnologías del lado del cliente
    a. HTML: lenguaje de marcado (*HyperText Markup Language*) estándar usado para construir páginas web
    b. CSS: lenguaje de hojas de estilos en cascada (*Cascading Style Sheets*). Describe cómo serán mostrados los elementos HTML
    c. JavaScript: lenguaje de programación que permite realizar actividades complejas es una página web
4. HTML
    - Editores
    - Elementos
    - Atributos
    - Encabezados
    - Párrafos
    - Formato de texto
    - Comentarios
    - CSS (abreboca)
    - *Links*
    - Imágenes
    - Tablas
    - Listas
    - Elementos *block* e *inline*
    - Clases e identificadores únicos
    - Iframes
    - Rutas de archivo
    - Etiqueta *head*
    - Entidades
    - Guía de estilos
5. CSS
    - Sintaxis
    - Selectores
    - Cómo incluir CSS
    - Comentarios
    - Colores
    - *Backgrounds*
    - Bordes
    - Márgenes
    - *Paddings*
    - Alto y ancho
    - *Box model*
    - Texto
    - Fuentes
    - Iconos
    - *Links*
    - Listas
    - Tablas
    - *Display*
    - *Max-width*
    - Posición
    - *Overflow*
    - *Float*
    - *Inline-block*
    - Alinear
    - Combinadores

- Pseudo clases
- Pseudo elementos
- Opacidad y transparencia
- Selectores de atributos
- Unidades

6. JavaScript
    - Introducción
    - *Output*
    - *Statementents*
    - Sintaxis
    - Comentarios
    - Variables
    - Operadores
    - Aritmética
    - Asignación
    - Tipos de datos
    - Funciones
    - Objetos
    - Eventos
    - Cadenas
    - *Arrays*
    - Comparaciones
    - Condicionales
    - *Switch*
    - Conversiones de tipo
    - Errores
    - *Scope*
    - Palabra clave *this*
    - *Let*
    - *Const*
    - Función flecha
    - Clases
    - *Debugging*
    - Guía de estilos
    - Mejores prácticas
    - Errores comunes
    - Rendimiento
    - *Closures*
    - *Callbacks*
    - AJAX
    - Promesas

7. Tipos de aplicaciones web
    a. Estática

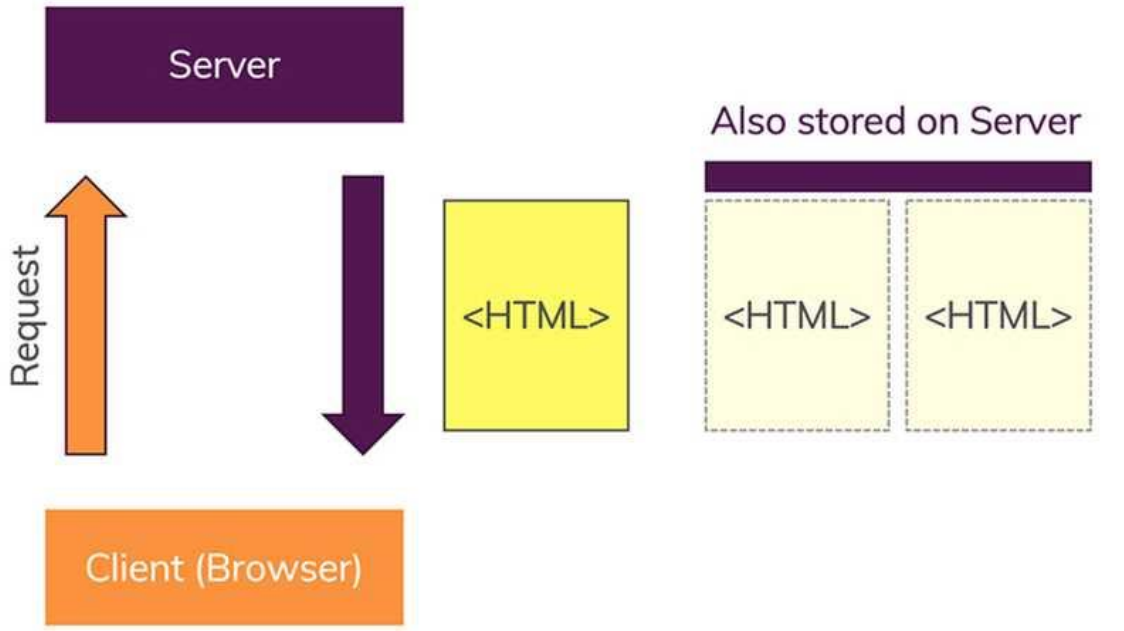

Ilustración 3 – Aplicación web estática

    b. Dinámica

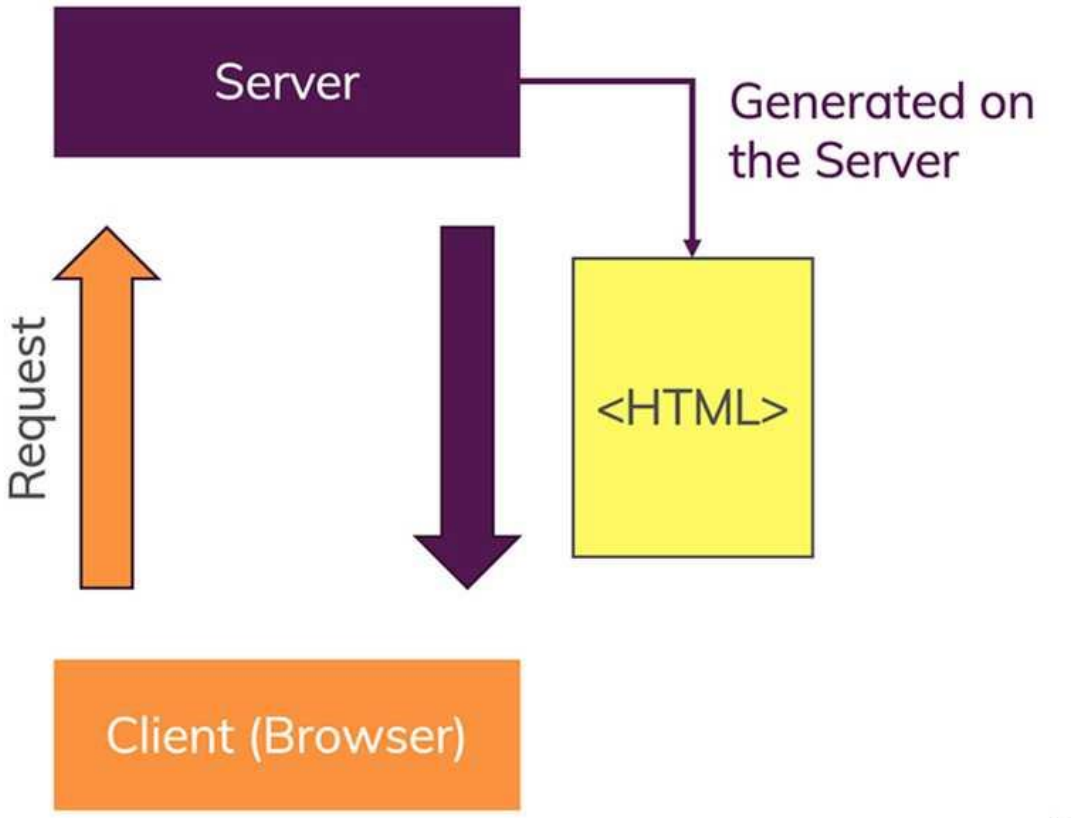

Ilustración 4 - Aplicación web dinámica

c. *Single Page Application* (SPA)

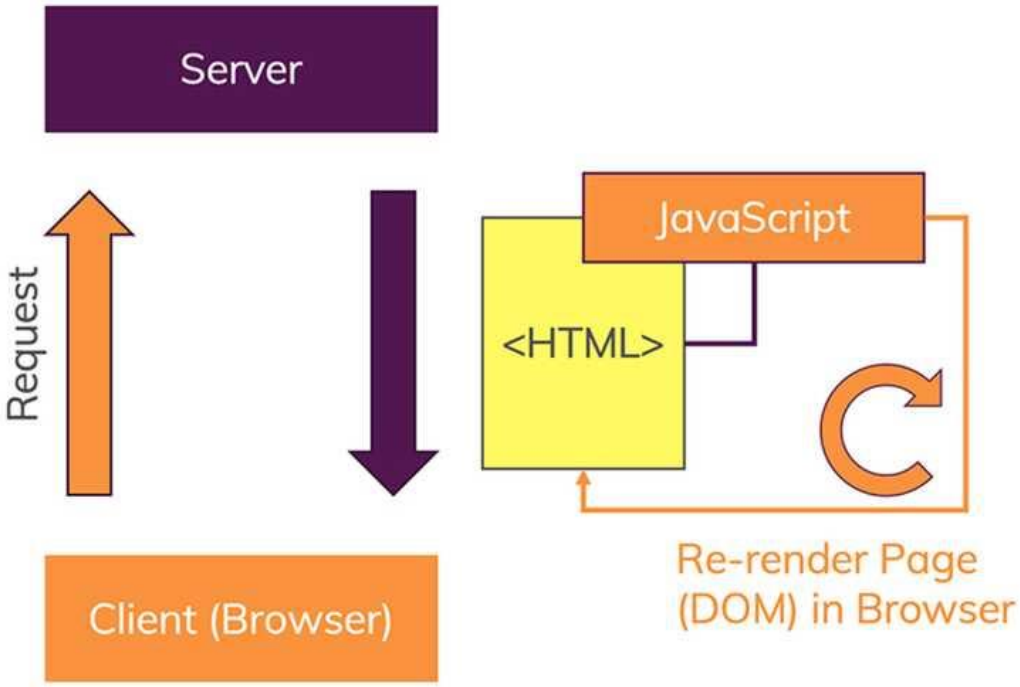

Ilustración 5 - Single Page Application (SPA)

8. Angular (*NPM* y *Visual Studio Code*)
    a. Conceptos
        - Conceptos básicos
        - Módulos
        - Componentes
        - Servicios
    b. Componentes y *Templates*
        - *Displaying data*
        - Sintaxis de *Templates*
        - *User Input*
        - Ciclo de vida de los *Hooks*
        - Interacción entre componentes
        - Estilos de componentes
        - Componentes dinámicos
        - Directivas
        - *Pipes*
    c. Formularios
        - Introducción
        - Formularios reactivos
        - Validación
        - Formularios dinámicos
    d. Observables & RxJS
        - Observables
        - La librería RxJS
        - Observables en Angular
        - Uso práctico
        - Comparación con otras técnicas
    e. *NgModules*
        - Introducción

- Módulos usados frecuentemente
- Componentes de entrada
- *Providers*
- Servicios *Singleton*
- Módulos *Lazy loading*
- Compartiendo módulos
- *ngModules API*
- *Routing & Navegation*

Tópico	**Estimación en horas**	
	Preparación	**Impartición**
HTML	8	3
CSS	8	3
JavaScript	8	5
Angular (*NPM y Visual Studio Code*)	24	8
Total	**48**	**19**