

INTRODUCCIÓN AL DESARROLLO DE APLICACIONES WEB

Prof. Antonio Gabriel González Casado
2ºDAW
Desarrollo Web en Entorno Cliente

ÍNDICE

1. CREACIÓN DE APLICACIONES WEB
 - 1.1. QUÉ ES UNA APLICACIÓN WEB
 - 1.2. PROCESO DE DESARROLLO DE APLICACIONES WEB
 - 1.3. FRONTEND, BACKEND Y FULLSTACK

2. TECNOLOGÍAS DEL LADO CLIENTE
 - 2.1. BREVE RECORRIDO POR LAS TECNOLOGÍAS FRONT-END
 - 2.2. EL TRIUNFO DE JAVASCRIPT

3. VERSIONES DE JAVASCRIPT
 - 3.1. LENGUAJES PREPROCESADOS
 - 3.2. ESTÁNDARES A PARTIR DEL AÑO 2015
 - 3.3. SITUACIÓN ACTUAL DE JAVASCRIPT

4. DESARROLLADOR FRONT-END
 - 4.1. QUÉ ES UN DESARROLLADOR FRONT-END
 - 4.2. TECNOLOGÍAS QUE DEBE USAR
 - 4.3. DESARROLLADOR POR TIPOS DE APLICACIONES

CREACIÓN DE APLICACIONES WEB

QUÉ ES UNA APLICACIÓN WEB

Una aplicación web es una aplicación que se ha creado para ser ejecutada en **cualquier dispositivo** a través de un **navegador**.

Los navegadores web más famosos son: Google Chrome, Mozilla Firefox, Safari, Ópera, Microsoft Edge,...



Algunos ejemplos de aplicaciones web que más usamos en nuestro día a día son google, gmail, facebook,...

PROCESO DE DESARROLLO DE APLICACIONES WEB

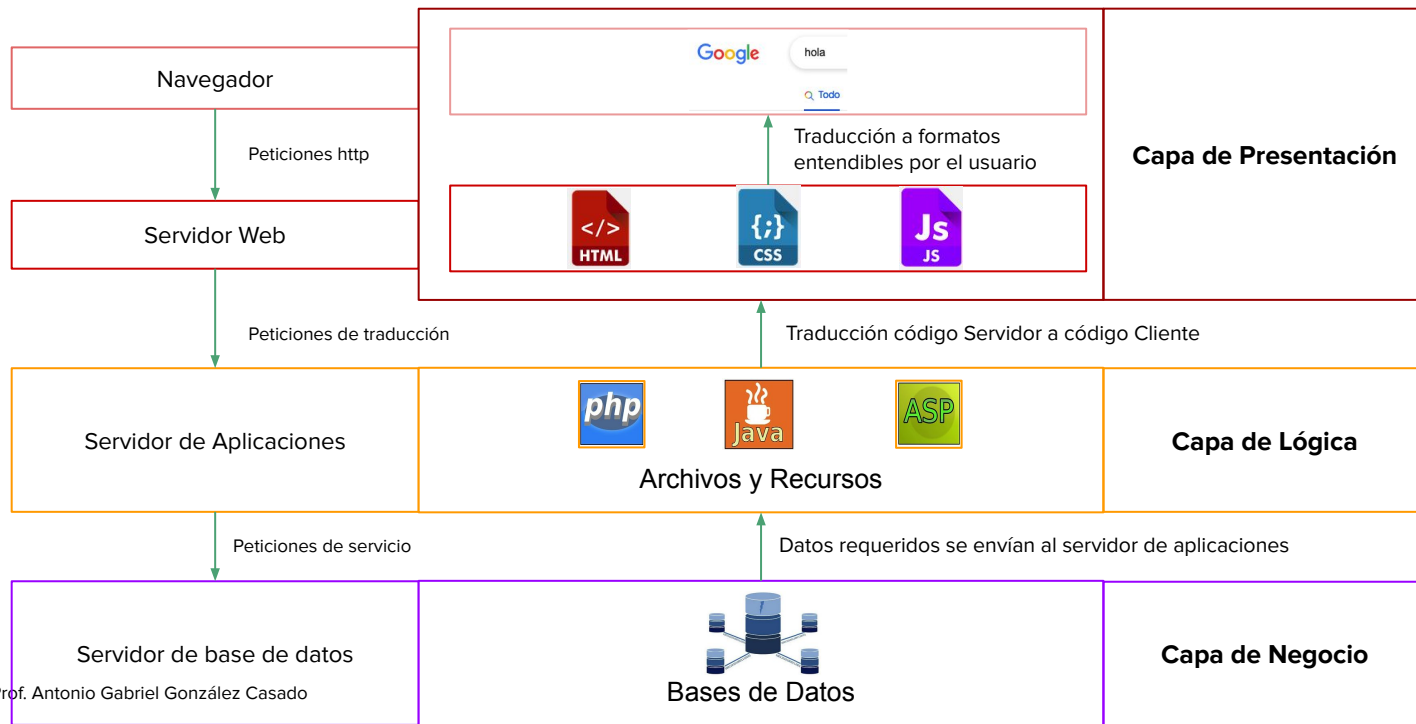
Las aplicaciones web pueden estar construidas en diversos lenguajes y tecnologías, pero hay dos tipos de clasificación para la mayoría de ellas:

- **Lenguajes y tecnologías en el lado cliente:** Acompañan al HTML y son interpretadas por el navegador. El código fuente queda **visible** al usuario.
- **Lenguajes y tecnologías en el lado servidor:** Están alojados en el servidor web que se encarga de ejecutarlos y servir al navegador los datos resultantes. El código fuente queda **oculto** al usuario.

Una aplicación web suele usar ambas tecnologías.

PROCESO DE DESARROLLO DE APLICACIONES WEB

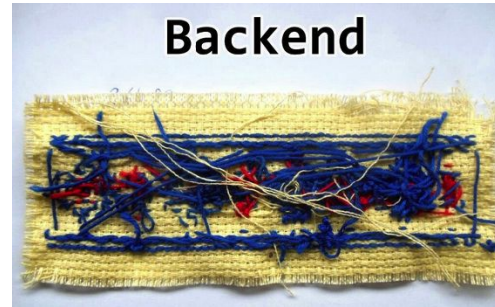
Las aplicaciones web actuales están construidas en lo que se conoce como arquitectura de **tres capas**.



FRONTEND, BACKEND Y FULL STACK

Frontend: Interfaz de usuario (UI). Se tiene que tener en cuenta los principios de usabilidad y experiencia de usuario (UX).

Backend: Es donde está toda la lógica de negocio de la aplicación que no tiene que ver con la usabilidad de la interfaz.



Full Stack: Utiliza el lenguaje de programación Javascript en las tres capas:

Presentación, Negocio y Datos

TECNOLOGÍAS DEL LADO CLIENTE

BREVE RECORRIDO POR LAS TECNOLOGÍAS FRONTEND

Existen tres **tecnologías principales**:

- **HTML**: Formato de documentos que incluye etiquetas para dar significado al contenido.
- **CSS**: Lenguaje de diseño encargado de determinar el formato y la maquetación de los elementos HTML.
- **JavaScript**: Lenguaje de programación que le da dinamismo a la página ayudando a la mejora de la UI.

Tecnologías secundarias:

- **XML**: Lenguaje de marcas que se utiliza actualmente principalmente en el intercambio de datos entre aplicaciones y servicios.
- **JSON**: Formato documental procedente del lenguaje JavaScript y que está reemplazando al XML como formato de intercambio de datos.
- **SVG**: Lenguaje basado en XML que permite mostrar imágenes vectoriales en los navegadores.
- **Flash y Applets** fueron tecnologías que se usaron en desarrollos de aplicaciones web del lado del cliente pero hoy están desfasados y **está contraindicado su uso**.

EL TRIUNFO DE JAVASCRIPT

- Es el único lenguaje de programación que los navegadores son capaces de traducir sin necesidad de extensiones o plugins.
- Lo creó la empresa Netscape para poder dar dinamismo a las páginas webs.
- El estándar actual de Javascript se llama ECMAScript.
- A partir del 2012 la 5ª versión de ECMAScript fue totalmente adoptada por todos los navegadores.
- El crecimiento continuo de JavaScript hizo que aparecieran aplicaciones webs semejantes a las de escritorio llamadas **RIA (Rich Internet Applications)**
- Ryan Dahl desarrolló Node.js que permitió a JavaScript salir del navegador.
- Actualmente con Javascript se pueden crear aplicaciones de escritorio, programar dispositivos de hardware, generar aplicaciones nativas de móvil y ser un lenguaje de manipulación de sistemas de bases de datos (mongodb)
- JSON es el acrónimo de JavaScript Object Notation, es un formato de documentos originario de la forma en la que se escriben los objetos en JavaScript. Actualmente se considera un formato de intercambio de datos independiente a JavaScript.

VERSIONES DE JAVASCRIPT

LENGUAJES PREPROCESADOS

Un lenguaje preprocesado es aquel que se escribe en un determinado lenguaje y se convierte con ayuda de un software especial llamado **preprocesador**, en un lenguaje estándar.

Las necesidades de los programadores han llevado que se creen nuevos lenguajes que mejoran el actual:

- CoffeScript. Dota a JavaScript sintaxis inspirada en el lenguaje Python. **Ningún navegador de forma nativa es capaz de soportarlo.**
- **TypeScript.** Creado por Microsoft en 2012 para poder usar tipos de datos avanzados. Es muy utilizado ya que **es el lenguaje base de Angular** (de Google)

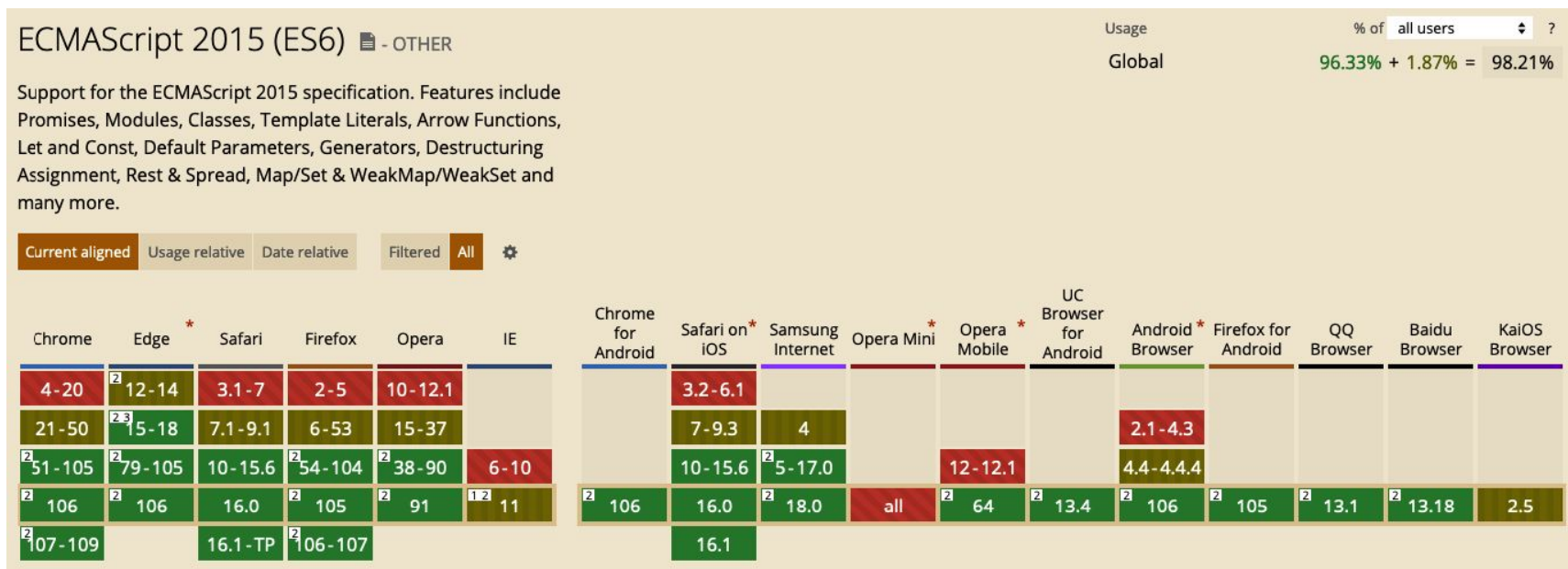
ESTÁNDARES A PARTIR DEL AÑO 2015

En el año 2015 apareció la norma ECMAScript 6 (ES6) que ha mejorado mucho el lenguaje JavaScript al dotarlo de elementos avanzados de otros lenguajes.

Los estándares son entes vivos que siguen evolucionando e incorporando mejoras, de ahí que ya existe la versión ES9 del estándar ECMAScript.

SITUACIÓN ACTUAL DE JAVASCRIPT

Ante tanta variedad de lenguajes y normas existe una web que se encarga de aclarar dónde y cómo se puede usar todo lo relacionado con el lenguaje (<https://caniuse.com>)



SITUACIÓN ACTUAL DE JAVASCRIPT

¿Es JavaScript un lenguaje de programación interpretado o compilado?

SITUACIÓN ACTUAL DE JAVASCRIPT

En sus inicios JavaScript fue diseñado para funcionalidades muy simples por lo que era interpretado.

Actualmente JavaScript tiene la potencia de cualquier lenguaje de programación. Entonces los navegadores usan la compilación **JIT (Just in Time)**, es decir, que los navegadores tienen una **JavaScript Virtual Machine (JVM)** que es la que se encarga de compilar el código para que sea ejecutado de la manera más óptima y eficiente.

Por lo anterior, hoy en día se puede decir que JavaScript es un lenguaje **compilado**.

DESARROLLADOR FRONT-END

QUÉ ES UN DESARROLLADOR FRONT-END

Un desarrollador Frontend se encarga de programar la interfaz web de una aplicación que se ejecuta en los navegadores. Estas son algunas de sus principales tareas:

- Traducir a HTML y CSS el diseño de una aplicación
- Ser creativo para proporcionar accesibilidad y usabilidad mediante su programación.
- Conocer cómo se comunicará el frontend con el backend, es decir, saber usar **APIS**.
- Adaptar el diseño a cualquier navegador y dispositivo, es decir, debe saber de **diseño adaptativo (Responsive)**.
- Tener en cuenta los aspectos de rendimiento que puede ocasionar la ejecución de su código en el lado cliente.
- Conocer las herramientas disponibles para construir eficientemente las aplicaciones.

TECNOLOGÍAS QUE DEBE USAR

Este es un resumen de las tecnologías que un desarrollador frontend debe conocer:



Este es un resumen de las tecnologías que un desarrollador de aplicaciones web debería conocer:



DESARROLLADOR POR TIPOS DE APLICACIONES FRONTEND

Aplicaciones monolíticas que usan JavaScript o alguna librería (jQuery, ReactJS,...) en su sistema de vistas:

- Desarrollador Full Stack

Aplicaciones basadas en APIS de servicios SOAP o REST con Frontend y Backend separado en aplicaciones distintas:

- Desarrollador Backend
- Desarrollador Frontend
- Desarrollador Full Stack

CONCLUSIÓN

- Existen dos tipos de tecnologías en el desarrollo de aplicaciones web: Frontend y backend
 - Los lenguajes frontend están en la capa de presentación y son visibles al usuario.
 - HTML (semántica), CSS (formato) y JavaScript (interactividad)
 - Su éxito ha permitido que se use para otros fines que no son puramente el desarrollo web
 - Actualmente, los estándares de la ECMA son ampliamente aceptados.
-