



Universidad de
Oviedo



APLICACIONES EN EL CLIENTE JS

Enol García González
Universidad de Oviedo
10 de noviembre de 2025

CONTENIDOS

- 1 Lenguajes de Script
 - 2 JavaScript y ECMAScript
 - 3 APIs
 - 4 JQuery
-

SCRIPT

Repasemos acontecimientos:

- En 1989. Tim Berners-Lee crea la World Wide Web para intercambio de documentos científicos.
 - Inicialmente se utilizaba XML y HTML para intercambiar documentación en la WWW.
 - El siguiente paso fue darle aspecto con las hojas de estilo CSS. El lenguaje CSS fue propuesto en 1994.
 - Siguiente paso en la evolución: Dinamizar ese contenido que se intercambia.
 - Y para eso surgen los lenguajes de Script
-

VENTAJAS DE LOS LENGUAJES DE SCRIPT

- Interfaces de usuario interactivas:
 - Cálculos en el cliente.
 - Validación de formularios.
 - Modificación del árbol DOM para vistas interactivas o personalización.
- Inclusión de datos del sistema del cliente.
- Consumir servicios web.

JAVASCRIPT

- Publicado en 1995 es el lenguaje de Script más famoso.
- No es el único. Existen otros lenguajes de Script menos conocidos como VBScript o JScript.
- La mayoría de esos otros lenguajes a penas se usan y muchos ya no se mantienen, pero que existan significa que JavaScript no es el único lenguaje que pueden ejecutar los navegadores.
- Que ejecutan los navegadores modernos: ECMAScript

ECMAScript

- ECMAScript es el nombre del lenguaje de Script estandarizado.
- ECMAScript esta definido por un estándar que se comenzó a escribir en 1996 para unificar las diferencias entre los diferentes lenguajes de Script que estaban apareciendo.
- ECMAScript es un estándar internacional mantenido por la W3C, que es el mismo organismo que mantiene HTML y CSS.
- “Actualmente” hay dos lenguajes que implementan este estándar: JavaScript (de Oracle) y JScript (de Microsoft).

APIs

- Estamos acostumbrados a llamar API a un servicio web que consumimos a través de REST o SOAP.
 - En su definición más pura, una API hace referencia a un conjunto de propiedades y métodos agrupados para un fin concreto.
 - En ECMAScript se toma esta definición más pura de una API para hacer referencia a propiedades y métodos del lenguaje destinados a tareas concretas, por ejemplo:
 - Ficheros
 - Dibujo
 - Almacenamiento
 - Encriptación
 - En <https://developer.mozilla.org/es/docs/Web/API> podemos ver la lista de todas las APIs con su documentación.
-

EJEMPLO

En los materiales tenemos un ejemplo que utiliza las APIs de Fetch y de geolocalización.

JQUERY

- Es un librería para JavaScript que pretende facilitar el desarrollo e integrar Ajax también de la forma más cómoda posible.
 - JQuery se publicó en 2006 y siempre recibió críticas porque lo único que aportaba era un poco de ligereza al código y el soporte de Ajax.
 - Desde que el estandar ECMAScript incluye la API Fetch en 2015, JQuery está perdiendo mucha popularidad.
-

FUNCIONALIDADES

- Una de las principales funcionalidades es la manipulación del árbol DOM.
 - `$(“selector”)` selecciona el primer elemento que haga match con el selector que tiene dentro. El selector puede tener formato CSS o XPath.
 - En Javascript existe el método `document.querySelector(“selector”)` desde 1998 que cubre esa funcionalidad con selectores CSS y el método `document.evaluate(“selector”)` que cubre lo mismo con XPath desde 2004.
 - Una vez obtenido el elemento del arbol DOM se puede modificar el contenido del html con `.html(“nuevo valor”)` en JQuery y con `.innerHTML = “nuevo valor”` sin utilizar JQuery.
-

FUNCIONALIDADES

- La segunda gran funcionalidad de JQuery es el registro de eventos
- Con JQuery, una vez obtenido un elemento, se puede registrar un evento para ese elemento con el método `on()`. Por ejemplo:
`$("#btnEjemplo").on('click', function() {...})`
- Sin JQuery, tenemos el método `addEventListener` desde 2003. Sería:
`document.getElementById("btnEjemplo").addEventListener('click', function() {
... })`

FUNCIONALIDADES

- La última gran funcionalidad que presenta JQuery es que viene integrado con Ajax. Para hacer una petición con Ajax se llamaría a `$.ajax(datos)`.
- El parámetro `datos` es un diccionario que tiene todo lo necesario para la petición: URL, método, cabeceras, qué función se llamará a la vuelta, etc.
- En ECMAScript se incluyó la librería Fetch en 2015