Centro de Estudios de Postgrado

Máster en Profesorado de Enseñanza Secundaria Obligatoria, Bachillerato, Formación Profesional y Enseñanza de Idiomas

Universidad de Jaén

Centro de Estudios de Postgrado

Trabajo Fin de Máster

JAVASCRIPT Y DESARROLLO EN SITIOS WEB

Alumno: Hortelano Ruiz, Francisco Javier

Tutor: Prof. D. José Manuel Fuertes García
Tutor: Prof. D. Guillermo Garrido Luque

Dpto: Departamento de Informática

Junio, 2021

Resumen

El presente Trabajo Fin de Máster (TFM) tiene como objetivo elaborar una Unidad Didáctica dentro del módulo “Desarrollo web en entorno cliente” del segundo curso del Ciclo Formativo de Grado Superior “Desarrollo de Aplicaciones Web”. Esta unidad didáctica desarrolla los diferentes contenidos enmarcados dentro de los bloques y las directrices establecidos por la ley. Para el desarrollo de este trabajo se han puesto en práctica todos aquellos conocimientos adquiridos durante el curso del Máster, incluyendo los adquiridos en el periodo de prácticas en el IES Virgen del Carmen, centro en el que se enmarca esta unidad. Este trabajo comenzará con una fundamentación epistemológica sobre el tema desarrollado, JavaScript; posteriormente se contextualizará el centro y el grupo de alumnos y alumnas al que se dirige, para desarrollar los diferentes elementos de los que cuenta y finalizar con la exposición de la unidad didáctica, todo ello fundamentado en la ley educativa vigente.

Palabras clave: “Unidad didáctica, Informática, JavaScript, Ciclos Formativos, Formación Profesional”

Summary

The objective of this Master's Thesis (TFM) is to elaborate a Didactic Unit within the module "Web development in client environment" of the second course of the Higher Level Training Cycle "Web Applications Development". This didactic unit develops the different contents framed within the blocks and guidelines established by law. For the development of this work has been put into practice all the knowledge acquired during the course of the Master, including those acquired during the internship at the IES Virgen del Carmen, center in which this didactic unit is framed. This work will begin with an epistemological foundation on the subject to be developed, JavaScript; then the center and the group of students to which it is addressed will be contextualized, to then develop the different elements of which it counts and end with the exposition of the didactic unit, all based on the current educational law.

Keywords: "Didactic unit, Computer Science, JavaScript, Vocational Training, Vocational Training".

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. Introducción

1.1. Motivación y descripción del problema

Debido a la situación global provocada por la pandemia, se ha forzado un salto tecnológico en cuanto al desarrollo de sistemas inteligentes, Big Data, seguridad, banca digital, y servicios basados en Internet. Los distintos confinamientos en los países alrededor del globo, sumado a las restricciones de movilidad entre los mismos y dentro de ellos, ha incrementado el uso de servicios por Internet, provocando así una rápida (e inevitable) transición a los servicios en nube, servicios y aplicaciones web.

Asimismo, se añade la instalación del 5G, implantando una nueva generación de redes móviles y, por ende, un empuje al uso de este tipo de servicios por parte de la sociedad. Además de la velocidad de conexión, el 5G ofrece una disminución de los tiempos de respuesta entre dispositivos, subidas y descargas en un tiempo menor, un nuevo tipo de procesamiento de señal… (de Looper, Martinok, 2021) que incentiva aún más el uso de las tecnologías de red en la sociedad.

Para dar respuesta a esta creciente demanda, es necesario montar una infraestructura que pueda soportar y dar respuesta eficientemente a los requisitos planteados, por lo que el uso de lenguajes para servicios web se vuelve indispensable.

JavaScript es el lenguaje por excelencia que permite la interactividad de las páginas web (y por ende la facilitación de la incrustación de servicios web en las mismas) en el lado del cliente. Del mismo modo, contiene variaciones, como Node.js, que agregan una mayor funcionalidad al lado servidor, permitiendo establecer relaciones y conexiones con los objetos del entorno, proporcionando control sobre los mismos. Tal y como se menciona, además es un lenguaje orientado a objetos, lo cual lo hace ser un lenguaje completo y robusto.

Es por esto que JavaScript merece un hueco en todas las programaciones didácticas de informática, sobre todo en Formación Profesional, puesto que es una herramienta muy poderosa para realizar soluciones profesionales a una gran cantidad de problemas. Además de abrir a los/as alumnos/as más posibilidades en cuanto a la inserción laboral se refiere, les permitirá realizar programas complejos desde el primer momento, a la par que facilitará su aprendizaje de lenguajes similares orientados al mismo ámbito.

No obstante, antes de construir una Unidad Didáctica sobre esta temática, es necesario comprender en profundidad qué es JavaScript, las posibilidades que ofrece, su versatilidad, su historia y por qué le ha llevado a ser el lenguaje más utilizado actualmente para el desarrollo de sitios web.

2. Antecedentes y Estado del arte

2.1. Historia de JavaScript

Para hablar del surgimiento de JavaScript es necesario remontarse al comienzo de la World Wide Web o W3, o comúnmente conocida como la Web. Hay que aclarar que la WWW no es lo mismo que Internet; Internet es una red de computadores interconectados que pueden intercambiar datos, mientras que la W3 es el conjunto de páginas públicas interconectadas construida sobre dicha red de computadoras. A la W3 se accede mediante Internet (Mozilla Developer, 2021).

La World Wide Web fue creada por Tim Berners-Lee y Robert Caillau durante su estancia trabajando en el CERN, como una propuesta de un proyecto de Hipertexto. Se define Hipertexto como una manera de enlazar y acceder a información de distintos tipos (con los que trabajaban en el CERN: informes, datos y documentación de experimentos, datos de personal, listas de correo electrónico…) concebida como una red de nodos en la que el usuario puede navegar a voluntad (Berners-Lee, Cailliau 1990).

Del mismo modo, surgen los conceptos de navegador web, servidor web y página web, que serían fundamentales para la estructuración de la W3. Así comenzó a estructurarse la Web, formándose por los siguientes componentes:

- Protocolo HTTP, Protocolo de Transferencia de HiperTextos, el cual se encarga de transportar la información (textual, imágenes, o animaciones) entre computadoras (o más recientemente, entre servidor y cliente).
- URL: Enlace, como lo definieron sus creadores (Berners-Lee, Cailliau 1990), a un componente web, consistente en un identificador universal único.
- HTML (Lenguaje Marcado de HiperTexto): formato más común para la publicación de documentos y componentes web. Aunque inicialmente no lo crearon, apareció cuando el uso de la W3 comenzó a pasar a una herramienta más global y necesitó estandarización.

Así pues, en 1991 publicaron la World Wide Web, y en 1993 se hizo la presentación de forma pública (CERN, 2020), pasando a ser de dominio público.

Conforme fue creciendo y evolucionando la W3, fueron apareciendo distintas carencias que contenía el sistema para trabajar a una mayor escala. La principal era la velocidad de transmisión de datos. Al principio de los años 90, las conexiones se realizaban mediante módems con una velocidad muy reducida. Al comenzar a desarrollarse las primeras aplicaciones web y páginas web con formularios más complejos, quedó de manifiesto la necesidad de ejecutar código en el computador del usuario. De esta forma, los tiempos de respuesta podían reducirse, ofreciendo una navegación más ágil; si los datos estaban mal introducidos, no era necesario esperar la respuesta del servidor indicando los errores.

Así, Brendan Eich, programador de Netscape desarrolló una solución adaptando algunas tecnologías existentes, como EaseScript, para el navegador Netscape Navigator de la

versión del 2002. Esta herramienta cubría y solucionaba estos problemas, e inicialmente tomó el nombre de LiveScript, modificándose posteriormente a JavaScript.

Este lenguaje tuvo una buena acogida y una gran repercusión, por lo que Netscape decidió volver a incluirlo en su Netscape Navigator 3.0, con la última versión que había en el momento, la 1.1. Paralelamente, Microsoft desarrolló una variante de JavaScript, a la que denominó JScript (Chandra, Chandra & Chandra, 2003), para evitar problemas legales.

Debido a esto, en 1997 Netscape estandariza JavaScript en la versión 1.1, realizado por el organismo ECMA (European Computers Manufacturers Association) (Mozilla Developers, 2021). ECMA creó el comité TC39 para estandarizar un lenguaje de script multiplataforma manteniendo la independencia de cualquier empresa. Así, el primer estándar se denominó ECMA-262 (ECMA-262 - 1st edition, 1997), definiéndose por primera vez el lenguaje ECMAScript.

Netscape también desarrolló una implementación de JavaScript en el lado servidor en 1994, la cual siguió desarrollándose hasta que en los 2000 surgió una gran cantidad de implementaciones de JavaScript en el lado servidor, por lo que se menciona en el siguiente párrafo, destacando notablemente Node.js.

No obstante, no fue hasta la aparición de Ajax, que se construyeron muchos frameworks y librerías de ámbito general, aportando mejoras a JavaScript y popularizando su uso fuera de los navegadores web, surgiendo así Node.js. En 2009 se inauguró el proyecto CommonJS para especificar y estandarizar una librería común cuya aplicación principal fuera el desarrollo fuera del navegador web.

De esta forma, y tras su estandarización, JavaScript (o ECMAScript) ha continuado desarrollándose y evolucionando como un lenguaje versátil y convirtiéndose en uno de los lenguajes más utilizados en Internet.

2.2. Estado actual de JavaScript

JavaScript actualmente está disponible en la versión 9 de ECMA-262 (ECMA-262 - 9th edition, 2021), del año 2018. Desde entonces no ha sufrido ninguna revisión ni actualización a un nuevo estándar.

Se ha incluido en diversos frameworks y librerías que ofrecen extensiones de JavaScript para facilitar la manipulación del DOM (Document Object Model), la programación en el lado servidor y otras áreas de la programación. Algunas de las librerías de JavaScript más utilizadas actualmente son las siguientes (IONOS, 2019):

- **jQuery:** La más utilizada de todas las librerías de JavaScript debido a que puede escribirse este código para cualquier tipo de navegador y para el que existe una gran diversidad de plugins. Se utiliza principalmente como una interfaz del DOM ofreciendo numerosas funciones; la más popular, integrar consultas con Ajax (Chaffer & Swedberg, 2011).

- **jQuery UI:** Extensión gratuita para jQuery para el diseño y creación de interfaces de usuario. Permite la generación de efectos e interacciones de forma sencilla (Wellman, 2009).
- **Dojo Toolkit:** Diseñado para la creación de aplicaciones web y contenidos web dinámicos. Una de las primeras que surgieron que permite utilizarse con DOM y con Ajax (Holzner, 2008).
- **React:** Publicada originalmente como una librería de código abierto para la creación de interfaces de usuario, que puede ser utilizado tanto en el cliente como en el servidor y durante el desarrollo de aplicaciones. Destaca por el flujo de datos en una dirección: los cambios en el código inferior jerárquicamente no pueden afectar al superior (Fedosejev, 2015).
- **Zepto:** Librería de JavaScript muy ligera, cargando más rápido que el resto y ocupando menos que el resto, eliminando, entre otras cosas, la compatibilidad con los navegadores más antiguos. No obstante, para ser compatible con Ajax, requiere módulos adicionales (Pointer, 2013).
- **CreateJS:** Compuesta por cuatro librerías (EaselJS, TweenJS, SoundJS y PreloadJS) que no necesitan dependencias entre ellas, permitiendo incluir en el proyecto únicamente las necesarias. Ofrecen herramientas para simplificar el trabajo con librerías de JavaScript, y se centra en el desarrollo de aplicaciones HTML5 (Mehrabani, 2014).

En cuanto a los frameworks, debido a que se utilizan para construir aplicaciones más complejas, y suelen ser más modificados y especificados por los desarrolladores, son menos numerosos que las librerías. No obstante, destacan los siguientes:

- **AngularJS:** Propiedad de Google, uno de los más reconocidos y utilizados globalmente, se emplea principalmente para construir aplicaciones web de una única página. Basado en jQueryLite y se adapta a cualquier tipo de de plataforma, móvil o escritorio (Darwin & Kozlowski, 2013).
- **Ember.js:** Framework del lado del cliente que permite crear aplicaciones web de una sola página y aplicaciones de escritorio. Está respaldada por una gran comunidad que discute los cambios que se realizarán en el mismo, y está orientada a desarrolladores con un cierto bagaje en la creación de aplicaciones web (Kelonye, 2014).
- **Meteor:** También llamado MeteoRJ, es un framework de JavaScript orientada al desarrollo cross-platform permitiendo montar aplicaciones web y móviles utilizando el mismo código. Del mismo modo, permite enviar los cambios en el código a los clientes utilizando un Distributed Data Protocol propio. Utiliza JavaScript y Node.js y permite desarrollar tanto backend como frontend sin cambiar el código (Strack, 2015).

Como se ha descrito JavaScript contiene una gran cantidad de elementos que lo toman como base, o que amplían sus posibilidades, de forma que cubren un amplio porcentaje de todos los proyectos web (aplicaciones web, páginas web…).

Concretamente tiene un gran peso en el creciente Internet of Things (IoT). El estudio realizado por Oliveira y Mattos (Oliveira & Mattos, 2019), recoge datos de la aplicación de

JavaScript a aplicaciones de IoT, demostrando que JavaScript puede ser integrado perfectamente en este tipo de proyectos, aunque no se centre en trabajar con navegadores. Muestra también, cómo JavaScript ha ido ganando versatilidad con los años, integrándose en proyectos de IoT, y cómo la investigación en la combinación de ambas ha ido creciendo en los últimos años, publicándose más "papers" cada año.

No obstante, en el último año 2020 han comenzado a consolidarse más otras tecnologías como TypeScript, que sigue ganando popularidad. Typescript está siendo adoptado de forma generalizada en la situación actual, con el tipado estático que ofrece, aunque hasta el momento coexisten ofreciendo un ecosistema muy rico en la programación.

Debido a que no existen muchos estudios recientes que puedan arrojar luz sobre el estado actual de JavaScript, dando una imagen fidedigna del panorama, se han tomado los datos recopilados de diferentes encuestas a programadores de todo el mundo en la web (Greiff & Benitte, 2020).

A lo largo de los diferentes gráficos generados a partir de las respuestas podemos apreciar cómo, cada vez más, nuevas tecnologías empiezan a destacar más e incluso a escucharse a niveles similares que JavaScript. Sí que es cierto que, aunque muchas de ellas son frameworks basados en JavaScript, otras son variaciones de JavaScript que, han cambiado tanto, que se pueden considerar una tecnología independiente de esta.

Otros, por el contrario, como TypeScript, surgen como ampliación a JavaScript, (Bierman, Abadi & Torgersen, 2014), a modo de lenguaje de alto nivel, que simplifica la interacción con este y lo compila de forma nativa.

Así pues, podemos concluir que JavaScript sigue formando una pieza esencial en el desarrollo de aplicaciones web, aunque su uso de forma explícita y nativa se ha visto reducido por las tecnologías nuevas que surgen a modo de respuesta de alcanzar mayor complejidad con menos. Sin embargo, al usar la inmensa mayoría de ellas JavaScript como base, se puede tratar a JavaScript como el lenguaje y tecnología dominante aún.

3. Definición de Javascript

3.1. Características

Una vez contextualizada esta tecnología, se procede a definirla y hablar de ella más detenidamente. JavaScript es un lenguaje de programación de scripts, multiplataforma y orientado a objetos. Es el lenguaje que, en páginas y aplicaciones web, se encarga de la interactividad de las mismas, permitiendo añadir y controlar animaciones complejas, eventos de dispositivos de entrada, etc.

Tal y como se ha comentado en la historia de JavaScript (véase apartado 2.1. Historia de JavaScript), inicialmente fue concebida como una tecnología para ejecutarse en el lado del cliente, agilizando la interacción con las páginas web. Sin embargo, se ha ampliado su funcionalidad con el tiempo, permitiendo ejecutarse también en el lado servidor:

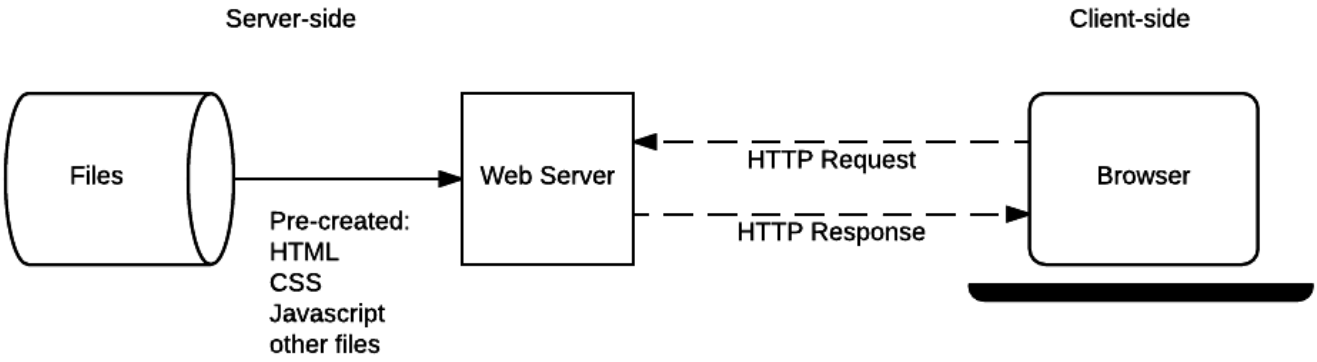

Figura 1 JavaScript en lado servidor. Fuente: https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Introduction

- Lado cliente: JavaScript proporciona objetos para controlar un navegador y su DOM mediante extensiones del lenguaje. Estas extensiones permiten a una aplicación ubicar elementos dentro del cuerpo del HTML y responder a eventos de la interacción con el usuario.
- Lado servidor: JavaScript proporciona objetos relevantes para poder ser ejecutado en el servidor, ampliando el núcleo del lenguaje. Estos objetos permiten la comunicación entre aplicación y base de datos, mantener la continuidad e integridad de la información entre invocaciones, etc. Se popularizó con la aparición del patrón MVC (Modelo-Vista-Controlador) y los frameworks de este tipo (Delcev & Draskovic, 2018).

Al ser un lenguaje orientado a objetos, como Python o C++, permite organizar dichos objetos de forma simple y ser reutilizados por todo el código. Estos objetos a su vez no necesariamente necesitan de estar definidos por una clase, sino que JavaScript proporciona los objetos generales, de forma que se agiliza la programación al poder definir las propiedades necesarias sin necesitar su previo planteamiento.

Esta característica de los objetos en JavaScript se debe a que es un lenguaje no tipado; dicho de otra forma, no necesita especificar el tipo de dato al declarar una variable. Esto permite utilizar el dato según las conveniencias del código, tratándolo como un tipo u otro, aunque puede provocar errores más fácilmente si no se maneja correctamente.

JavaScript es un lenguaje de alto nivel, lo que reduce la curva de aprendizaje drásticamente, y simplifica interacciones complejas del navegador con la página o aplicación web. Por esto mismo, JavaScript es un lenguaje interpretado, lo que le proporciona ventajas como ser multiplataforma o reducir el procesamiento en servidores web al poder ejecutarse en el lado cliente.

Sin embargo, JavaScript es un lenguaje compilado, aunque esté extendida la creencia de que no lo es. Al estar estandarizado JavaScript en ECMA, y por ende convertirse en un dialecto de ECMAScript, las características de compilado vienen especificadas en estándar. Según el estándar ECMA-262, "Una implementación conforme debe, antes de la primera evaluación de un guión o módulo, validar todas las reglas de error tempranas de las producciones utilizadas para analizar ese guión o módulo. Si se infringe alguna de las reglas de error tempranas, el script o módulo no es válido y no puede ser evaluado." (ECMA-262 - 9th edition, 2021). El compilado de JavaScript no es igual que el de C++ o de otro lenguaje que generaría un ejecutable; es un proceso realizado previo de precompilación y ejecución para evitar errores que puedan invalidar el programa.

Las aplicaciones de JavaScript ya han sido descritas en los puntos anteriores, remarcando la cantidad de herramientas construidas tomando este lenguaje como base y extendiendo sus funcionalidades y límites aún más. Por tanto, este lenguaje de programación destaca, no sólo por las operaciones que puede realizar para la interacción, sino también la multidisciplinariedad que presenta, permitiendo ser aplicada a casi cada tipo de proyecto web.

3.2. Sintaxis y semántica

La sintaxis de JavaScript utiliza el conjunto de caracteres Unicode. Unicode es un superconjunto (conjunto que contiene a un subconjunto) de ASCII y Latin-1, y soporta cada lenguaje escrito utilizado en el planeta (Korpela, 2006). Es un lenguaje sensible a mayúsculas y minúsculas (case-sensitive). Existen muchos nombres de etiquetas en HTML que tienen el mismo nombre que las propiedades y objetos correspondientes en JavaScript; así pues, mientras que en HTML se pueden escribir en minúsculas o mayúsculas independientemente, en JavaScript deben escribirse en minúsculas para acceder a dichas propiedades u objetos del DOM.

JavaScript ignora los espacios fuera de las delimitaciones del programa (por ejemplo, los corchetes de una función) y los saltos de línea, lo que permite indexar y formatear el código de forma que sea legible, sencillo de entender e intuitivo. Además de los espacios regulares,

JavaScript también reconoce los siguientes caracteres como espacios en blanco (Flanagan, 2011, p.22):

- Tabulación horizontal
- Tabulación vertical
- Salto de página
- Otros caracteres Unicode de la categoría Zs.

En lo referente a la semántica del código, JavaScript distingue entre dos tipos de variables: var y let. Ambas variables son un mismo tipo de variable no tipada, diferenciándose en el alcance (scope) que tienen dentro del programa. Una variable "let" está limitada al alcance local de la función o bloque en la que se ha declarado, mientras que una variable declarada con "var" es de ámbito global, o variable global (Mozilla Developers, 2021).

JavaScript no contiene funcionalidades para operaciones de entrada/salida, son proporcionadas por el entorno de ejecución. No obstante, la mayoría de estos entornos proporcionan el objeto "console" que permite el flujo de salida de información en la consola de depuración.

En cuanto al resto de la sintaxis y semántica, es muy similar a Java, siendo así la construcción de bucles, condicionales, declaración de clases y objetos, etc. Prácticamente igual a muchos lenguajes de programación orientados a objetos.

Proporciona también un tipo específico de función, llamada "función flecha", que es una alternativa a una expresión tradicional de función, aunque es limitada y su uso se restringe a ciertas situaciones (Mozilla Developers, 2021).

3.3. Integración con otras tecnologías web

Principalmente se emplea JavaScript incluido o embebido en páginas HTML, manipulando el DOM (objetos de documentos y elementos que contiene) y, por ende, permitiendo a JavaScript codificar HTML. Del mismo modo, puede modificar la presentación y visualización del contenido codificando estilos y clases CSS, y definir el comportamiento de ciertos componentes mediante el manejo de eventos (Flanagan, 2011, p.310). A la combinación de los comportamientos anteriores con HTML se le conoce como Dynamic HTML (DHTML).

De forma general, el código JavaScript se integra con documentos HTML en el lado del cliente, por lo explicado en el apartado anterior (véase apartado 3.1. Características), y se realiza de cuatro maneras (Flanagan, 2011, p.311):

- Incluido en el código entre etiquetas <script>.

- Desde un archivo externo al código HTML incluido en el mismo en la etiqueta <script> mediante la propiedad “src”.
- En el manejador de eventos de atributos HTML, como “onwheel” u “ondrag”.
- Incluido en una URL que incluya el protocolo especial “javascript:”.

Cabe destacar también el uso de JavaScript para las animaciones en la interacción con la página web, incluido en el manejo de eventos de elementos del documento HTML, permitiendo así una interacción, por parte del usuario, más fluida y atractiva.

A continuación, a modo de colofón de este apartado de sintaxis, se muestra un fragmento de código JavaScript y el resultado visual en web, como ejemplo de lo descrito en este apartado. El código (véase la Figura 2) muestra una función en JavaScript que obtiene el nombre del usuario desde la consola y lo añade en el texto de la cabecera de la página web.

```
function setUserName() {
  let myName = prompt('Please enter your name.');
  if(!myName) {
    setUserName();
  } else {
    localStorage.setItem('name', myName);
    myHeading.textContent = 'Mozilla is cool, ' + myName;
  }
}
```

Figura 2. Función de ejemplo de JavaScript. Fuente: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

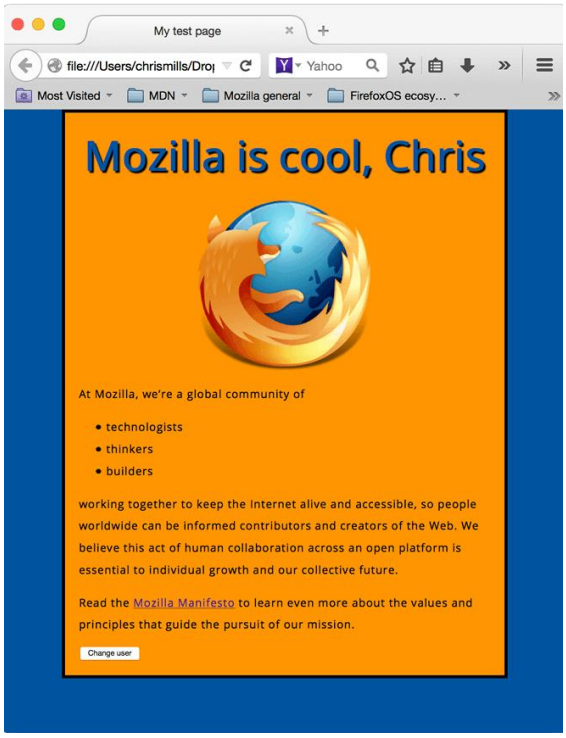

Figura 3. Resultado de la función descrita en la Figura 2. Fuente: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

3.4. Seguridad

Una de las principales desventajas de JavaScript es el uso malicioso de sus funciones: permite la ejecución de código malicioso sin el consentimiento del usuario, comprometiendo así su seguridad.

En los últimos años, debido su auge, se utiliza para el minado de criptomonedas en los navegadores de los usuarios (Matthieu Faou, 2017). En estos casos, la minería se ejecuta en el navegador del usuario al navegar en determinados sitios web, lo que elimina el paso de explotar posibles vulnerabilidades o infectar a la víctima. Aprovechando que JavaScript está activado en el navegador del usuario por defecto, pueden realizar este minado de criptomonedas, todo sin el consentimiento del usuario, y generalmente, sin ser conscientes.

Para evitar casos como el anterior, los navegadores ofrecen dos tipos de restricciones: permiten únicamente la ejecución de scripts relacionados con el funcionamiento de la web, evitando aquellos que sobrepasan estos límites, como la creación de archivos. Por otro lado, estos scripts también tienen restringido el acceso a información proporcionada por el usuario, como cookies o contraseñas.

Para solucionar estas vulnerabilidades surgen proyectos como Secure ECMAScript: un lenguaje de programación que implementa un mayor nivel de seguridad mediante funciones de evaluación que limitan la ejecución de scripts de terceros (Google Code, 2021).

Otro de los problemas más comunes de seguridad es el llamado cross-site scripting (XSS). Este es un tipo de vulnerabilidad surgido a partir de la presencia de errores de filtrado en las entradas del usuario en las aplicaciones web (ausencia de validación de campos). La vulnerabilidad permite al atacante la inyección de código JavaScript en páginas web visitadas por el usuario (Bisht & Venkatakrishnan, 2008). La principal contramedida a esta vulnerabilidad consiste en validar todos los campos del usuario y el uso de variables en lugar de SQL puro. En contrapartida, la vulnerabilidad depende de la implementación del sitio web por parte del equipo desarrollador, por lo que depende principalmente del “factor humano”.

3.5. Lenguajes derivados

A raíz de la aparición de JavaScript han ido surgiendo diferentes lenguajes de programación basados en este lenguaje o que compilan JavaScript. A continuación, se hablará de los dos lenguajes que más relevancia han tenido y que más se utilizan actualmente:

- TypeScript: Un lenguaje de programación de código abierto (open source) construido sobre JavaScript, proporcionando mejoras en la definición de los objetos, una documentación más detallada y la posibilidad de validar el código

escrito (Bierman, Abadi & Torgersen, 2014). El código TypeScript es convertido a código JavaScript mediante su propio compilador, llamado Babel, permitiendo ser ejecutado en navegadores, aplicaciones propias y que integren Node.js.

- PureScript: Lenguaje de programación que compila a JavaScript y facilita la reutilización de código JavaScript (Dupal, 2016). Consiste en una vasta colección de librerías para el desarrollo de aplicaciones web, servidores web, aplicaciones, etc. Además ofrece una amplia gama de herramientas y un buen soporte de edición con compilaciones instantáneas.

Aunque existen más lenguajes que comparten características con estos, los dos más relevantes en la actualidad; el resto de lenguajes siguen compitiendo por escalar en cuanto al uso y la aplicación en los desarrollos web.

4. Tendencias futuras

Tal y como se ha visto en el desarrollo del apartado del estado actual del arte de JavaScript (véase el apartado 2.2. Estado actual de JavaScript), y observando que se ha mantenido en los últimos años como tecnología de desarrollo web por excelencia, la tendencia en los años venideros es de mantener el puesto que conserva.

Es necesario destacar que, aunque va a seguir en la vanguardia del desarrollo web, no lo hará de forma nativa y explícita como otros lenguajes de programación, sino que estará presente en los frameworks de desarrollo, como elemento constructor, y en los nuevos lenguajes que aporten nuevas herramientas avanzadas a lo que ya ofrece JavaScript.

En la Figura 4, podemos observar la evolución de estos nuevos lenguajes que han ido surgiendo y cogiendo presencia en el panorama actual, en los últimos 5 años.

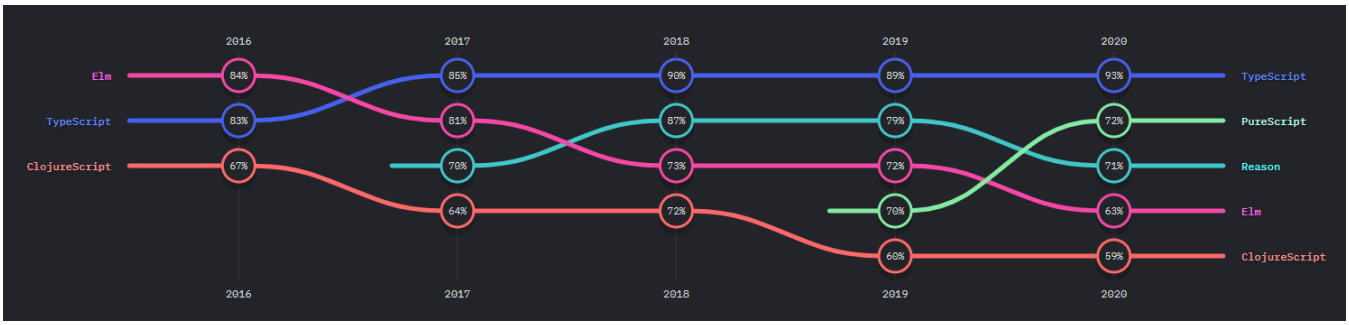

Figura 4. Clasificación de lenguajes basados en JavaScript según satisfacción, interés, uso y conocimiento. Fuente: https://2020.stateofjs.com/es-ES/technologies/front-end-frameworks/

Se puede apreciar cómo TypeScritp se posiciona en el lenguaje líder en los últimos cuatro años, con tendencia creciente. Cada vez son más los lenguajes "avanzados de JavaScript" que aparecen, ofreciendo algunos unas opciones y soluciones muy sugerentes, pero TypeScript es la tendencia futura para programar en un lenguaje de script en el desarrollo web. Sin embargo, es necesario hacer mención a PureScript, que en los últimos años ha aumentado su uso, y a Reason y Elm que mantienen su posición.

Una de las principales razones por las que no han remplazado a JavaScript es debido a lo explicado en el apartado anterior (véase apartado 3.5. Lenguajes derivados): la inmensa mayoría son lenguajes de programación que compilan JavaScript, están basados en él o traducen internamente su código a código JavaScript (como es el caso de TypeScript con Babel). La mayoría de estos lenguajes siguen utilizando JavaScript internamente, o incluso permiten la combinación del propio lenguaje con JavaScript.

Asimismo, a la hora de aprender un lenguaje de programación (a nivel básico) de desarrollo web, se suele enseñar JavaScript. De esta forma, el cambio de JavaScript a estos nuevos lenguajes implica un gran cambio en aplicaciones ya existentes (sobre todo aquellas de una gran envergadura) además del coste de la formación de los desarrolladores en estas tecnologías. Al ser nuevos estos lenguajes, muchas empresas no deciden dar el paso en sus

productos principales, y lo comienzan a incorporar en los nuevos proyectos que desarrollan. No obstante, la creciente tendencia al uso de estos lenguajes en los nuevos proyectos, prevé una futura sustitución de JavaScript por las nuevas alternativas.

Con respecto a los frameworks de JavaScript, la tendencia es a seguir utilizando los más potentes y extendidos hasta la fecha: jQuery, Angular y ReactJS. Los tres tienen una comunidad bastante amplia y cuentan con el respaldo de grandes empresas que las integran en sus proyectos, afrontando una variedad de temáticas inmensa y respondiendo bien a los retos y problemas que plantean.

Por tanto, la actualidad se presenta optimista con lo que al uso y desarrollo de JavaScript se refiere, manteniendo su relevancia y ofreciendo nuevos lavados de cara que impedirán que quede obsoleto.

5. Conclusiones

A modo de resumen de todo lo expuesto hasta ahora, JavaScript es una tecnología muy completa y versátil, siendo la primera de su campo y que ha sabido mantenerse como líder durante casi el mismo tiempo que lleva existiendo la World Wide Web.

A raíz de la estandarización con ECMAScript, ha seguido una evolución más constante, actualizándose a la par que iban surgiendo nuevas necesidades tecnológicas y cubriéndolas. Esto le ha permitido mantenerse a la vanguardia y servir de base para todas las aplicaciones web, e incluso innovando en nuevas técnicas.

Tal y como expone el artículo periodístico de Juan Cabrera (Juan Cabrera, 2021), en 2021 el perfil tecnológico más solicitado será el de full stack developer. Los puestos de trabajo relacionados con el ámbito del desarrollo web no dejan de aumentar y mantenerse en los puestos más altos entre los perfiles más demandados en los últimos años.

Sumado a que JavaScript es uno de los 5 lenguajes más demandados en el mercado tecnológico (Pavel Ramírez, 2020), convierten a este lenguaje en una opción esencial a la hora de enseñarse tanto en Formación Profesional como en cualquier estudio orientado al mundo tecnológico.

JavaScript ha sido y es un lenguaje que ha permitido evolucionar rápidamente el desarrollo de los sistemas web y continúa siendo una pieza básica en la construcción de cualquier servicio o aplicación web que pretenda ofrecer la mayor cantidad de servicios al cliente.

6. Unidad Didáctica: Programación en JavaScript (Desarrollo web en entorno cliente).

Se presenta la unidad didáctica denominada “Programación en JavaScript” situada en la programación didáctica del módulo “Desarrollo web en entorno cliente” del segundo curso del Ciclo Formativo de Grado Superior “Desarrollo de Aplicaciones Web”. Para definir esta unidad didáctica, se incluyen diferentes aspectos relacionados con la programación de la asignatura, incluyendo normativa, organización del aula y mecanismos e instrumentos de evaluación.

6.1 Introducción

La programación web es un área latente y con gran auge en el sector de la informática hoy día, pues la tendencia es de utilizar servicios web, por ejemplo, mediante aplicaciones. En este tipo de aplicaciones es esencial saber dominar lenguajes de marcas y lenguajes que permitan una interacción entre el cliente y el servidor. En esta asignatura, al basarse en la programación del lado del cliente, se impartirán nociones del lenguaje más fundamental y utilizado profesionalmente en este ámbito: JavaScript. Con los contenidos que se ofrecen en esta unidad didáctica se pretende que el alumnado sepa construir servicios básicos y realizar un servicio web funcional y completo que le permitirá desenvolverse en el ámbito profesional con una buena base.

En la Tabla 1 (véase en este capítulo la Tabla 1) se incluye un resumen de la Unidad Didáctica de “Programación en JavaScript”, diseñada para introducir al alumnado en los principales contenidos de la materia, tal y como se establece en el currículo descrito en la Orden de 16 de junio de 2011 por la que se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web, el Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones y en la normativa incluida en dicha tabla.

Dicha materia será impartida de la siguiente manera:

- Etapa: CFGS
- Módulo: Desarrollo web en entorno cliente
- Curso: 2º curso
- Duración del módulo: 126 horas, 6 horas semanales
- Duración de la Unidad Didáctica: 12 horas
- Número de sesiones: 6
- Días de clase: lunes, miércoles y viernes. 2
- Trimestre: 1er trimestre
- Fechas de impartición: 19 de octubre 2020 - 30 octubre 2020

En esta UD también se incluye de manera transversal la comunicación, el diálogo y la resolución de conflictos, que se tratarán implícitamente en las actividades que se realicen en grupos.

<table>
<tr><td colspan="4">Unidad Didáctica</td><td colspan="4">PROGRAMACIÓN EN JAVASCRIPT</td></tr>
<tr><td>Curso</td><td>2º</td><td>Trimestre</td><td>1</td><td>Nº Horas</td><td>12</td><td>Nº sesiones</td><td>6</td></tr>
<tr><td colspan="4">Real Decreto</td><td colspan="4">Orden</td></tr>
<tr><td colspan="4">• Ley Orgánica 2/2006, de 3 de mayo, de Educación (LOE)
• RD 21/2020, de 9 de junio, de medidas urgentes de prevención, contención y coordinación para hacer frente a la crisis sanitaria ocasionada por el COVID-19.
• Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas.
• Orden EDU/2887/2010, de 2 de noviembre, por la que se establece el currículo del ciclo formativo de Grado Superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web</td><td colspan="4">• Ley 17/2007, de 10 de diciembre, de Educación de Andalucía (LEA)
• Orden de 16 de junio de 2011, por la que se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web
• ORDEN de 29 de septiembre de 2010, por la que se regula la evaluación, certificación, acreditación y titulación académica del alumnado que cursa enseñanzas de formación profesional inicial que forma parte del sistema educativo en la Comunidad Autónoma de Andalucía.</td></tr>
</table>

Tabla 1. Introducción a la Unidad Didáctica. Elaboración propia

6.2. Contextualización del centro

Para el desarrollo de la Unidad Didáctica, en primer lugar se expone la contextualización del centro escolar en el que se impartirá, incluyendo el nivel socioeconómico y las características del alumnado. El centro en el que se impartirá será el Instituto de Educación Secundaria Virgen del Carmen, en Jaén.

6.2.1. Situación socio-económica del centro

El IES Virgen del Carmen se ubica en el centro de la ciudad de Jaén, localizado en una zona céntrica, en la intersección de dos de las vías más importantes de la ciudad. A nivel de Educación Secundaria y Bachillerato, la mayoría de alumnos inscritos se encuentran en el centro debido a la proximidad del mismo al sitio de trabajo de sus padres.

Sin embargo, a nivel de Ciclos Formativos, debido a la amplia variedad de los mismos que oferta el centro, siendo algunos los únicos que se imparten en la ciudad, y siendo el único que oferta turno de tarde para los Ciclos Formativos, el perfil del alumnado inscrito varía. En estos ciclos, el alumnado incluye al de toda la ciudad, independientemente de su proximidad al centro, y de pueblos cercanos.

6.2.2 Características del alumnado

El alumnado del módulo en el que se imparte esta Unidad Didáctica (véase apartado 6.1 Introducción) comprende a alumnos y alumnas procedentes de Ciclos Formativos de Grado Medio y de Bachillerato. En su totalidad son alumnos mayores de edad que buscan la obtención del título para su pronto acceso al mundo laboral que no se encuentran trabajando durante la duración del curso escolar.

Una parte de este alumnado también proviene de pueblos cercanos a la capital a cursar dichos estudios, por lo que, normalmente, suelen depender del horario de autobuses para poder asistir a clase. Esto suele provocar, mayoritariamente en las clases de última hora, que el/la alumno/a tenga que abandonar antes el aula para coger el transporte público para volver a su residencia. Del mismo modo, es factible que se produzcan faltas a ciertas clases si ocurre algún problema en el desplazamiento y no tienen quien los acerque al centro.

El alumnado suele provenir de familias trabajadoras del sector servicios, que pueden clasificarse dentro del estrato social medio. En su mayoría se refiere a familias estructuradas con la mayor parte de sus miembros convivientes en el domicilio (o la totalidad de miembros) con el espacio y material suficiente y disponible para realizar sus tareas de estudio.

Un sector más reducido proviene de familias trabajadoras del sector primario, aunque mantienen la estructura descrita en el párrafo anterior.

6.3. Contenidos curriculares

Se muestran a continuación los contenidos curriculares para el módulo “Desarrollo Web en Entorno Cliente” según lo dispuesto en la Orden de 16 de Junio de 2011, por la que se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de

Aplicaciones Web, aplicados a la Unidad Didáctica. En la Tabla 2 (véase en este apartado la Tabla 2) y en los epígrafes siguientes, podemos ver un resumen de los contenidos, objetivos, resultados de aprendizaje, líneas de actuación y competencias profesionales, que comprende y desarrolla la presente Unidad Didáctica.

- Objetivos generales:
    - g) Utilizar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones Web con acceso a bases de datos.
    - i) Utilizar lenguajes de marcas y estándares Web, asumiendo el manual de estilo, para desarrollar interfaces en aplicaciones Web.
- Competencia general:
    - La competencia general de este título consiste en desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.
- Orientaciones pedagógicas:
    - a) La integración de lenguajes de programación y lenguajes de marcas.
    - b) La incorporación de funcionalidades en documentos Web.
    - c) La utilización de características y objetos propios del lenguaje y de los entornos de programación y ejecución.
    - d) La utilización de mecanismos para la gestión de eventos y la interacción con el usuario.
- Líneas de actuación:
    - b) La utilización de las características específicas de lenguajes y entornos de programación en el desarrollo de aplicaciones para clientes Web.
    - d) La incorporación de mecanismos de actualización dinámica en aplicaciones Web.
- Competencias profesionales, personales y sociales:
    - k) Desarrollar servicios para integrar sus funciones en otras aplicaciones Web, asegurando su funcionalidad.
- Contenidos conceptuales:
    - b) Manejo de la sintaxis del lenguaje.
    - c) Utilización de los objetos predefinidos del lenguaje.
    - d) Programación con <<arrays>>, funciones y objetos definidos por el usuario.
    - e) Interacción con el usuario, eventos y formularios:
        - i. Modelo de gestión de eventos.
        - ii. Manejadores de eventos.

Elementos Curriculares					
Competencia general	La competencia general de este título consiste en desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.				
Objetivos didácticos	- Manejar y controlar la sintaxis del lenguaje de programación JavaScript. - Conocer y utilizar los objetos predefinidos del lenguaje. - Conocer y programar utilizando funciones, objetos definidos y <<arrays>>. - Gestionar la interacción con el usuario mediante un modelo de gestión de eventos y manejadores de eventos.				
Objetivos generales	**Orientaciones pedagógicas**	**Líneas de actuación**	**Competencias profesionales, personales y sociales**	**Contenidos conceptuales**	**Temas transversales**
G, I	A, B, C, D	B, D	K	B, C, D, E, E.1, E.2	TIC, idiomas, inclusión
Resultados de aprendizaje			**Instrumentos**		
2. Escribe sentencias simples, aplicando la sintaxis del lenguaje y verificando su ejecución sobre navegadores Web.			Rúbrica 1, Rúbrica 2, Cuaderno del profesor		
4. Programa código para clientes Web analizando y utilizando estructuras definidas por el usuario.			Rúbrica 1, Rúbrica 2, Cuaderno del profesor		
5. Desarrolla aplicaciones Web interactivas integrando mecanismos de manejo de eventos.			Rúbrica 1, Rúbrica 2, Cuaderno del profesor		

Tabla 2. Elementos curriculares

7. Contenidos transversales

Se llaman elementos transversales a los conocimientos que, sin ir implícitos en la unidad didáctica, se dan para ampliar el contexto y la significatividad de las sesiones. Los temas transversales que se van a tratar en esta unidad didáctica, tal y como se han mencionado en la Tabla 2 (véase la Tabla 2, del apartado 6.3. Elementos curriculares) son:

- Manejo de las TIC. Desarrollada implícitamente en la Unidad Didáctica por la naturaleza tecnológica del módulo.
- Idiomas. Parte de los enlaces de documentación opcionales para el desarrollo de los guiones de prácticas y ejercicios varios, se ofrecerán en inglés, puesto que es el idioma en el que más documentación existe de las tecnologías. De esta forma, se consigue desarrollar esta habilidad para que, en un contexto laboral en el que deban investigar o desarrollar una funcionalidad concreta, tengan la destreza para documentarse de las fuentes oficiales en inglés.
- Educación en valores: inclusión. Se trabajará la inclusión de las personas, daptación de las tecnologías para personas con discapacidades visuales: daltonismo.

8. Metodología

En el presente apartado se desarrollarán todos los aspectos relacionados con la metodología empleada en el desarrollo de la Unidad Didáctica, incluyendo la organización del aula, los agrupamientos realizados, descripción de los materiales y recursos didácticos, tipos de actividades y su temporalización a lo largo de la duración de la Unidad Didáctica.

8.1. Organización del aula

Las clases, a excepción de la primera que se realizará en el aula normal de clase, se impartirán en su totalidad en el aula de prácticas. La disposición de los ordenadores en la misma será de dos filas pegando las pantallas a la pared de forma que, desde cualquier lugar del aula, se puedan ver las pantallas, quedando los cables del lado de la pared. Esta distribución se puede apreciar más visualmente en la Figura 3 (véase en este mismo apartado 8.1, la Figura 3).

El material del alumno será sus cuadernos y, si lo necesitan, el libro de la asignatura. El resto de material necesario para la realización de las prácticas y el seguimiento de las clases,

será proporcionado por el centro. Este material está detallado en el apartado 8.3 de la memoria (véase apartado 8.3. Materiales y recursos didácticos).

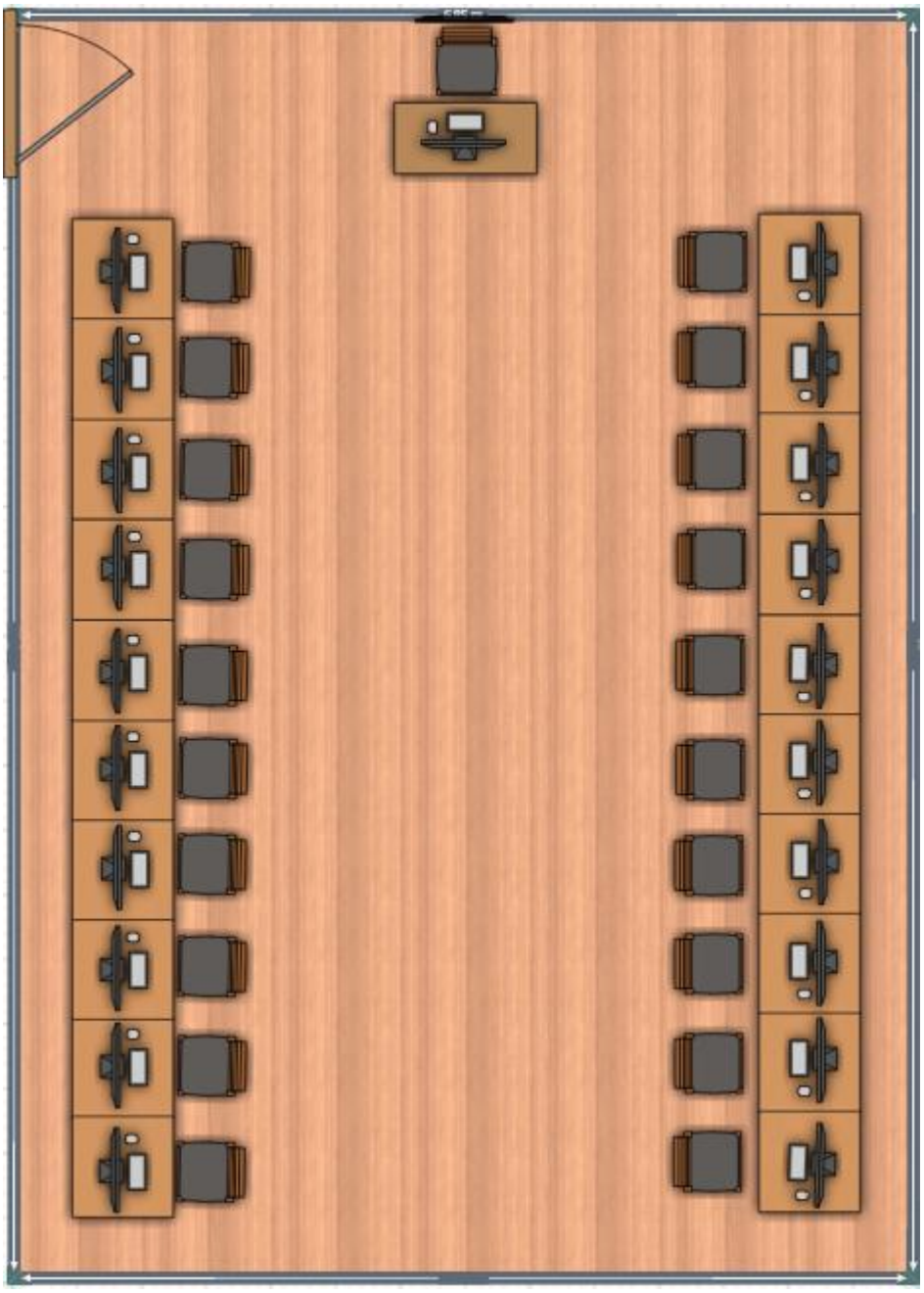

Figura 5. Representación del aula. Elaboración propia.

8.2. Tipos de agrupamiento

En esta metodología, se realizarán los siguientes agrupamientos del alumnado:

- Trabajo individual: Este es el que posibilita un mayor grado de personalización de la enseñanza, permitiendo la adaptación al ritmo y posibilidades del alumnado. Permite

afianzar conceptos más fácilmente y realizar un minucioso seguimiento por parte del profesorado del proceso de aprendizaje.

- Trabajo en grupos: En este tipo de agrupamiento se potencian al máximo las posibilidades de comunicar, compartir y realizar trabajos simultáneamente, contando en todo momento con la participación activa de todos los miembros del grupo. Este agrupamiento estará reservado para las presentaciones de teoría y resolución de dudas grupal.

8.3. Materiales y recursos didácticos

Cada miembro del alumnado dispondrá de:

- Hardware: Ordenadores del aula
- Pizarra
- Software
    - SO Ubuntu (u otra distribución basada en Debian).
- IDE de programación o editor de textos del sistema operativo
- Información y enlaces disponibles en la plataforma de Moodle.
- Diapositivas de clase y teoría disponibles en la plataforma Moodle.
- Actividades de ampliación y refuerzo, en formato digital, proporcionadas por el profesor.

8.4. Metodología y tipos de actividades

Las actividades de enseñanza-aprendizaje son la manera activa y ordenada de llevar a cabo las propuestas metodológicas o experiencias de aprendizaje. La metodología seguida para conseguir un aprendizaje significativo por parte del alumnado seguirá la siguiente estrategia:

- En la primera parte de cada sesión se realizará un repaso de la clase anterior, con el objetivo de reforzar los contenidos vistos, y aclarar cualquier duda surgida.
- A continuación, se introducirán los contenidos nuevos que van a tratarse en la sesión que introducen al alumnado en la problemática. En estas clases tendrán un carácter práctico, y se irá combinando la teoría dada con ejemplos prácticos que el profesor

realizará frente a los alumnos para ilustrar dicha teoría. Estos ejemplos podrán ser reproducidos a la vez por el alumnado.

- Después proseguirá con la explicación, y el profesorado indicará al alumnado el trabajo autónomo a realizar por cada alumno.
- Al final de cada UD, el alumnado presentará sus resultados al profesorado que evaluará su trabajo autónomo.

Con actividades se hace referencia a las tareas realizadas por el alumnado con la finalidad de conseguir distintos aprendizajes. Las actividades seguidas en esta unidad didáctica son las siguientes:

- **Actividades de desarrollo de los contenidos,** los cuales permiten al alumno/a la adquisición de nuevos contenidos. Se realizarán algunos supuestos prácticos de la relación de ejercicios propuestos para la unidad de trabajo y relacionados con los contenidos explicados en la sesión. El docente ayudará a resolver los primeros.
- **Actividades de consolidación**, con las que el alumnado contrasta las nuevas ideas con las previas y aplican los aprendizajes adquiridos y nuevos. Continuando con el supuesto anterior en que están resolviendo los ejercicios de una relación, una vez que el profesor ayuda a resolver algunas actividades, dejará al alumnado que resuelva el resto y, de esta forma, consolide sus conocimientos. Estas actividades tendrán que ser resueltas por el alumnado de manera autónoma en casa y serán entregadas antes del día de la evaluación.
- **Actividades de síntesis-resumen**, que permiten a los alumnos o alumnas establecer relación entre los distintos contenidos aprendidos, así como la contrastación con aquellos conocimientos que ya tenían. Se pueden realizar en dos momentos:
    - Al principio de las unidades de trabajo, donde se plantea un resumen de lo que se va a ver en clase.
    - Al finalizar una unidad de trabajo, se elaborará un resumen de los nuevos contenidos aprendidos de manera individual, el profesor elaborará junto con el alumnado un mapa conceptual en pizarra.

  Este tipo de actividades, aunque se pueden realizar al principio y al final, no siempre se realizarán en ambos momentos, dependiendo de la sesión y los contenidos, se darán ambos casos o solo uno.
- **Actividades de repaso y ampliación**, las cuales se afrontarán de forma síncrona. Una vez resueltas las actividades propuestas en la relación de ejercicios para cada unidad de trabajo, el profesorado entregará actividades de refuerzo a aquellos alumnos o alumnas que no hayan alcanzado los conocimientos trabajados, y el resto del

alumnado continuarán construyendo conocimientos a través de las actividades de ampliación.

- **Actividades de evaluación**, que pretenden proporcionar una justificación acerca de lo aprendido por los alumnos/as. Por tanto, se debe tomar en consideración lo establecido en los criterios de evaluación propuestos. Al finalizar cada unidad de trabajo, se realizará una prueba específica de carácter práctico que permitan evaluar al alumnado a nivel conceptual y procedimental. Este tipo de actividades no sólo evaluará el trabajo del alumnado, sino que también le permite poner en conocimiento del docente si la metodología empleada es la más adecuada a las características del grupo.

8.5. Temporalización de actividades

Proceso de Enseñanza-Aprendizaje				
Actividad	**Tiempo**	**Sesión**	**Agrupamiento**	**Recursos**
1.1 Actividad de Síntesis Resumen. Descripción breve de los contenidos básicos de programación en Javascript que se van a presentar en esta sesión y metodología de aprendizaje seguida en la misma. También se indicarán los mecanismos de evaluación de la misma.	15’	1	Grupal	- Ordenador - Software de presentaciones - Proyector o Pizarra digital
1.2 Actividad de desarrollo de contenidos. Se presentarán de forma teórica los contenidos teóricos que se tratarán en todas las sesiones de la Unidad Didáctica: 1. Introducción teórica a Javascript a) ¿Qué es Javascript? b) ¿Quién lo creó? c) ¿En qué año se creó? d) Diferencias entre Java y Javascript 2. Manejo básico de Javascript a) Hola mundo b) Modificar una parte del contenido de la página 3. Tipo de elementos a) Tipos de variables que existen b) Operadores c) Tipos de datos	10’	1	Grupal	- Ordenador - Proyector o Pizarra digital

d) If/else e) Bucles f) Funciones g) Clases h) Objetos 4. Interacciones con la página a) Eventos				
1.3 Actividad de desarrollo de contenidos Se introducen de forma teórica las bases de Javascript, centrándose en una introducción al lenguaje tratando la historia, forma, diferencias, etc. 1. Introducción teórica a Javascript a) ¿Qué es Javascript? b) ¿Quién lo creó? c) ¿En qué año se creó? d) Diferencias entre Java y Javascript	40’	1	Grupal	- Ordenador - Software de presentaciones - Proyector o Pizarra digital
1.4 Actividad de desarrollo de contenidos Se introducen de forma teórica las bases del manejo elemental de Javascript,. 2. Manejo básico de Javascript a) Hola mundo b) Modificar una parte del contenido de la página	30’	1	Grupal	- Ordenador - Software de presentaciones - Proyector o Pizarra digital
1.5 Actividad de consolidación Propuesta de realización de ejercicios de manera autónoma en clase trabajando con los conceptos de la actividad 1.4 y utilizando lo trabajado en clase. Se pueden completar en casa si no da tiempo a finalizarla en clase. Tiempo estimado: 25 minutos	25’	1	Individual	- Ordenador - IDE de programación
2.1 Actividad de Síntesis Resumen. Repaso de los principales conceptos vistos en la clase anterior y resolución de dudas.	5’	2	Grupal	- Ordenador - Proyector o Pizarra digital
2.2 Actividad de desarrollo de contenidos. Se verán los contenidos siguientes a lo largo de la sesión orientados a la	55’	2	Individual	- Ordenador

programación en Javascript, de un modo más avanzado: 3. Tipo de elementos a) Tipos de variables que existen b) Operadores c) Tipos de datos				- Proyector o Pizarra digital - IDE de programación
2.3 Actividad de consolidación Propuesta de realización de un ejercicio de manera autónoma en clase y casa trabajando con los conceptos de la actividad 2.2 y utilizando lo trabajado en clase. Tiempo estimado de trabajo autónomo: 1hora y 30 minutos	55’	2	Individual (autónoma)	- Ordenador - IDE de programación
2.4 Actividad de Síntesis Resumen. Repaso de los principales conceptos vistos durante la clase que deben afianzar para la siguiente sesión.	5’	2	Grupal	- Ordenador - Proyector o Pizarra digital
3.1 Actividad de Síntesis Resumen. Repaso de los principales conceptos vistos en la clase anterior y resolución de dudas.	5’	3	Grupal	- Ordenador - Proyector o Pizarra digital
3.2 Actividad de desarrollo de contenidos. Se verán los contenidos siguientes a lo largo de la sesión orientados a la programación en Javascript, de un modo más avanzado: 3. Tipo de elementos d) If/else e) Bucles f) Funciones	55’	3	Individual	- Ordenador - Proyector o Pizarra digital - IDE de programación
3.3 Actividad de consolidación Propuesta de realización de un ejercicio de manera autónoma en clase y casa trabajando con los conceptos de las actividades 2.2 y 3.2 y utilizando lo trabajado en clase. Tiempo estimado de trabajo autónomo: 2 horas	55’	3	Individual (autónoma)	- Ordenador - IDE de programación
3.4 Actividad de Síntesis Resumen. Repaso de los principales conceptos vistos durante la clase que deben afianzar para la siguiente sesión.	5’	3	Grupal	- Ordenador - Proyector o Pizarra digital
4.1 Actividad de Síntesis Resumen.	5’	4	Grupal	- Ordenador

Repaso de los principales conceptos vistos en la clase anterior y resolución de dudas.				- Proyector o Pizarra digital
4.2 Actividad de desarrollo de contenidos. Se verán los contenidos siguientes a lo largo de la sesión orientados a la programación en Javascript, de un modo más avanzado: 3. Tipo de elementos g) Clases h) Objetos	55’	4	Individual	- Ordenador - Proyector o Pizarra digital - IDE de programación
4.3 Actividad de consolidación Propuesta de realización de un ejercicio de manera autónoma en clase y casa trabajando con los conceptos de las actividades 2.2, 3.2 y 4.2 y utilizando lo trabajado en clase. Tiempo estimado de trabajo autónomo: 3 horas	55’	4	Individual (autónoma)	- Ordenador - IDE de programación
4.4 Actividad de Síntesis Resumen. Repaso de los principales conceptos vistos durante la clase que deben afianzar para la siguiente sesión.	5’	4	Grupal	- Ordenador - Proyector o Pizarra digital
5.1 Actividad de Síntesis Resumen. Repaso de los principales conceptos vistos en la clase anterior y resolución de dudas.	5’	5	Grupal	- Ordenador - Proyector o Pizarra digital
5.2 Actividad de desarrollo de contenidos. Se verán los contenidos siguientes a lo largo de la sesión orientados a la programación en Javascript, de un modo más avanzado: 4. Interacciones con la página a) Eventos	55’	5	Individual	- Ordenador - Proyector o Pizarra digital - IDE de programación
5.3 Actividad de consolidación Propuesta de realización de un ejercicio de manera autónoma en clase y casa trabajando con los conceptos de las actividades 2.2, 3.2, 4.2 y 5.2, y utilizando lo trabajado en clase. Tiempo estimado de trabajo autónomo: 3 horas	55’	5	Individual (autónoma)	- Ordenador - IDE de programación
5.4 Actividad de Síntesis Resumen.	5’	5	Grupal	- Ordenador - Proyector o Pizarra digital

Repaso de los principales conceptos vistos durante la clase que deben afianzar para la siguiente sesión.				
6.1 Actividad de evaluación Desarrollo del proyecto final de la asignatura. Finalizar el trabajo realizado en casa. Tema transversal: Gama de colores para personas daltónicas	60’	6	Individual	- Ordenador - Proyector o Pizarra digital - IDE de programación
6.2 Actividad de evaluación Exposición y demostración del proyecto realizado. Cada alumno deberá ejecutar su programa delante de toda la clase y explicar brevemente los elementos incluidos. Tema transversal: Inclusión -> Gama de colores para personas daltónicas	60’	6	individual	- Ordenador - Proyector o Pizarra digital - IDE de programación

Tabla 3. Temporalización de actividades.

Se describe con detalle cada una de las actividades de consolidación, repaso y ampliación en el ANEXO 1. (*)

9. Evaluación

9.1. Evaluación en el proceso de aprendizaje

Normativa: Según el Decreto 436/2008, de 2 de septiembre, donde se establece la ordenación y las enseñanzas de la Formación Profesional inicial que forma parte del sistema educativo, se dispone, que la evaluación deberá ser continua y se realizará por módulos profesionales. La evaluación se realizará por el profesorado tomando como referencia los objetivos y los criterios de evaluación de cada uno de los módulos profesionales y los objetivos generales del ciclo formativo. Asimismo, el profesorado tendrá la obligación de evaluar tanto el proceso de enseñanza-aprendizaje del alumnado como su propia práctica docente, de acuerdo con lo que establezca por Orden la Consejería competente en materia de educación.

9.2. Criterios de evaluación

Los criterios de evaluación y los estándares de aprendizaje asociados de esta UD son los indicados en la Tabla 4. Todos los criterios de evaluación y estándares de aprendizaje tienen el mismo peso en esta Unidad Didáctica.

Resultados de aprendizaje	Criterios de Evaluación
2. Escribe sentencias simples, aplicando la sintaxis del lenguaje y verificando su ejecución sobre navegadores Web.	a) Se ha seleccionado un lenguaje de programación de clientes Web en función de sus posibilidades. b) Se han utilizado los distintos tipos de variables y operadores disponibles en el lenguaje. c) Se han identificado los ámbitos de utilización de las variables. d) Se han reconocido y comprobado las peculiaridades del lenguaje respecto a las conversiones entre distintos tipos de datos. e) Se han añadido comentarios al código. f) Se han utilizado mecanismos de decisión en la creación de bloques de sentencias. g) Se han utilizado bucles y se ha verificado su funcionamiento.

	h) Se han utilizado herramientas y entornos para facilitar la programación, prueba y depuración del código.
4. Programa código para clientes Web analizando y utilizando estructuras definidas por el usuario.	a) Se han clasificado y utilizado las funciones predefinidas del lenguaje. b) Se han creado y utilizado funciones definidas por el usuario. c) Se han reconocido las características del lenguaje relativas a la creación y uso de arrays. d) Se han creado y utilizado arrays. e) Se han reconocido las características de orientación a objetos del lenguaje. f) Se ha creado código para definir la estructura de objetos. g) Se han creado métodos y propiedades. h) Se ha creado código que haga uso de objetos definidos por el usuario. i) Se ha depurado y documentado el código.
5. Desarrolla aplicaciones Web interactivas integrando mecanismos de manejo de eventos.	a) Se han reconocido las posibilidades del lenguaje de marcas relativas a la captura de los eventos producidos. b) Se han identificado las características del lenguaje de programación relativas a la gestión de los eventos. c) Se han diferenciado los tipos de eventos que se pueden manejar. d) Se ha creado un código que capture y utilice eventos.

Tabla 4. Resultados de aprendizaje y criterios de evaluación

9.3. Técnicas e Instrumentos

En esta UD se van a usar las siguientes técnicas e instrumentos de evaluación:

- **Observación**: A través de esta técnica percibiremos las habilidades conceptuales, procedimentales y actitudinales del estudiante de forma detallada y permanente, con el propósito de orientar al alumnado cuando lo requiera el proceso de enseñanza y/o aprendizaje. El profesor tomará información de la participación en clase del alumnado,

su actitud respecto al funcionamiento de la clase, cuidado del material, uso del material para lo que está dispuesto, etc. en el cuaderno del profesor.

- **Prácticas individuales**: Se emplean para evaluar el conocimiento de lo que el estudiante hace, además de lo que sabe en otro tipo de técnicas. Se realizarán una serie de prácticas en el ordenador, y cada una de las prácticas serán evaluadas, para ver su correcto funcionamiento y eficiencia.
- **Proyecto**: Se realizará un proyecto final que comprenda todos los elementos de la unidad didáctica, el cual tendrá un gran peso de la asignatura y será evaluado finalmente por el profesor. Dicho proyecto consistirá en aunar lo trabajado en prácticas individuales en una página web que de contexto global y coherencia a todo lo realizado.
- **Exposición**: El trabajo práctico será expuesto ante el profesor y la clase mediante una exposición para demostrar que el alumnado ha asimilado correctamente los contenidos que han sido resueltos en las diferentes actividades. Dicha exposición será evaluada por el docente a través de una rúbrica

Los instrumentos de evaluación por cada técnica se describen a continuación y en la Tabla 5 se encuentra detallada cada técnica:

Técnicas	Instrumentos
Observación	Rúbrica 1
	Cuaderno del profesor
Proyecto	Rúbrica 2
Actividades de casa	Rúbrica 2
Exposición	Rúbrica 2

Tabla 5. Relación instrumentos - técnicas de evaluación.

- **Cuaderno del profesor**: En este cuaderno el profesor anotará cualquier comportamiento positivo realizado por el alumnado. Las anotaciones serán usadas para aumentar en 0.1 puntos cada petición que el profesor haga en clase, hasta

alcanzar la máxima nota, siempre y cuando, estén contempladas en la rúbrica. Igualmente, los comportamientos no adecuados restarán sobre la misma calificación.

- **Rúbrica 1**: Rúbrica elaborada para la evaluación del comportamiento del alumnado en clase. Nota: 0-10 puntos. Esta rúbrica valorará de forma positiva, la actitud, el comportamiento, la puntualidad, la motivación del alumnado en clase. Dicha evaluación será llevada a cabo durante las sesiones de clase. Para realizar esta rúbrica, se utilizarán las anotaciones del cuaderno del profesor.
- **Rúbrica 2**: Rúbrica elaborada para la evaluación del desempeño en prácticas y su exposición por parte del alumnado en clase. Nota: 0-10 puntos. Esta rúbrica se aplicará a cada una de las prácticas y al proyecto final realizadas en la UD. Todas las prácticas tienen el mismo peso. Cada elemento de la rúbrica tendrá el peso indicado, en función de si el elemento se aplica para el caso (proyecto o trabajo individual) o no.

Indicador	**Alta (2 puntos)**	**Media (1,3 puntos)**	**Baja (0,6 puntos)**
PUNTUALIDAD Llegar puntual y justificar ausencias	Nunca se ha retrasado ni faltado a clase, o faltado con justificación	Tiene algún retraso o falta sin justificar	No asiste con asiduidad a clase teniendo más de diez retrasos y faltas sin justificar.
MATERIAL Propio y del aula y material necesario para la asignatura	Trae todo el material necesario para la asignatura y lo cuida y respeta el material del centro dejando todo en su lugar tras su uso (silla sobre la mesa, material deportivo, en el aula/taller)	En varias ocasiones se ha olvidado el material necesario y/o ha tenido una actitud impropia en lo referente al cuidado del material del centro	No trae ningún tipo de material necesario y/o no respeta el material del centro ni de los demás compañeros/as
MOTIVACION E INTERES Participación e iniciativa	Tiene iniciativa e interés aplicando lo aprendido y asociándolo a su entorno diario compartiendo información adicional y extracurricular y realizando trabajos voluntarios	Rara vez se ofrece voluntario. Contesta únicamente cuando se le pregunta	Actitud completamente pasiva y desmotivación total. Se echa a dormir.
RESPETO Al profesor, a los compañeros y al	Saluda al entrar en clase y se despide al marchar. Levanta la mano para hablar, respeta el turno de palabra, habla con un vocabulario correcto y respetuoso	Ha tenido algunas faltas de respeto puntuales.	Interrumpe constantemente el normal desarrollo de

ambiente general de trabajo	cuando se dirige tanto al profesorado como a sus compañeros y compañeras.		las clases con gestos y vocabulario inapropiado, elevando el tono de voz y sin respetar turnos
CUMPLIMIENTO DE LAS NORMAS	No come chicle ni ningún alimento durante las clases. No ensucia ni deteriora en modo el alguno el entorno. (pintar mesas/paredes, tirar basura). Entra de forma adecuada a las aulas y no grita ni corre por los pasillos. No usa su teléfono móvil.	En varias ocasiones no ha cumplido una o varias normas	Incumple varias normas de forma reiterativa y constante

Tabla 6. Rúbrica 1

Indicador	**Excelente (2 puntos)**	**Bien (1,5 puntos)**	**Regular (1 punto)**	**Mal (0,5 puntos)**
Asimilación o análisis del problema 25% para las actividades 25% para el proyecto	El alumno/a ha comprendido completamente el problema, relacionándolo con otros problemas vistos en clase.	Tiene algún retraso o falta sin justificar	No asiste con asiduidad a clase teniendo más de diez retrasos y faltas sin justificar.	No se ha comprendido el problema
Desarrollo de la solución 50% para las actividades 45% para el proyecto	El alumno/a desarrolla una solución al problema óptima, eficiente, y más allá de lo que se exige	Se resuelve el problema. La solución no es la mejor, pero es una solución funcional.	Se desarrolla una solución que no funciona pero relativa al problema planteado, o desarrolla una solución funcional pero del problema analizado incorrectamente	No se ha desarrollado ninguna solución funcional
Exposición del resultado 20% para las actividades 15% para el proyecto	El alumno/a es capaz de presentar el resultado de manera clara, y concisa. Además, es capaz de comentar y criticar posibles soluciones alternativas.	Presenta el resultado de manera confusa, aunque la solución sea clara.	Se presenta el resultado de manera confusa, y poco clara, con indicios de copia, sin analizar la solución.	No es capaz de transmitir una opinión sobre el problema. Es una copia.

Creatividad 5% para las actividades 5% para el proyecto	El alumno/a ha presentado una solución creativa fuera de la explicada por el profesor.	Hay algo de creatividad en la solución, pero hay un intento de resolución correcta. Plantea soluciones dentro de las esperadas.	No hay creatividad en la solución, pero hay un intento de solución que no es la adecuada.	No se ha dado solución al problema.
Ampliación 10% para el proyecto	El alumno ha realizado los ejercicios opcionales planteados para el proyecto correctamente.	El alumno ha realizado algunos ejercicios opcionales correctamente.	El alumno ha realizado los ejercicios opcionales con fallos.	No se ha realizado ningún ejercicio opcional.

Tabla 7. Rúbrica 2

9.4. Criterios de calificación

El mecanismo de evaluación de esta UD se realizará de manera clásica donde se identifica cada uno de los contenidos, criterios de evaluación e instrumentos de evaluación y peso de cada una de las partes. En la tabla 1 puede verse esta relación.

El detalle de este mecanismo de evaluación viene a continuación detallado y en la Tabla 8.

Tarea	**Proyecto**	**Actividades**	**Entrevista**	**Observación**
Criterios de calificación	50%	30% 20% -> Actividades de consolidación 10% -> Actividades de repaso y ampliación	10%	10%
Instrumento de evaluación	Rúbrica 2	Rúbrica 2	Rubrica 2	Rúbrica 1 Cuaderno del profesor
Momento	Durante la UD	Todo el periodo docente	Fin de la UD	Todo el periodo docente

Tabla 8. Desglose de la evaluación

Cada falta de asistencia, incumplimiento de las normas será regulada por el centro.

La **nota de la evaluación de la UD** se realizará realizando la media ponderada de cada una de las actividades realizadas de acuerdo con los criterios de calificación en base a tareas anteriormente descritas. La UD se dará por superada si se obtiene una puntuación >=5, sobre 10. Para acceder a la media ponderada, deberán tener un mínimo de un **4** tanto en el proyecto como en las prácticas.

Para la recuperación de la UD, en el caso de no superar los requisitos mínimos para aprobarla, el alumnado deberá realizar lo siguiente:

- Realización de todos los ejercicios de la asignatura con fecha de entrega, la fecha de la prueba escrita. Su peso será de un 20% de la nota.
- Prueba escrita sobre los contenidos de la asignatura, de tipo test, con 4 opciones, que tendrá un peso del 55% de la nota.
- Entrevista con el profesor, mediante la defensa de los ejercicios realizados, con un peso del 25%.

El alumnado será evaluado positivamente, si la media ponderada de cada una de las actividades es >= 5. Esta recuperación se realizará en 3 momentos a lo largo del curso:

- Al finalizar el trimestre.
- Al finalizar el curso, en la convocatoria ordinaria.
- En septiembre, en la convocatoria extraordinaria.

En el caso de no poder asistir al día de la evaluación o perder la evaluación, el alumnado deberá presentarse en la siguiente evaluación posible siendo su nota de actitud, la única de esta UD.

9.5. Atención a la diversidad

Se atribuye el término de Necesidades Educativas Especiales (NEE) a aquellos alumnos/as que presentan mayores dificultades que el resto de su mismo grupo y nivel educativo a la hora de alcanzar, por vía ordinaria, los objetivos y contenidos curriculares propuesto. Se engloba dentro del término de NEE a tres tipos de colectivos:

- Alumnado con incorporación tardía y con dificultades de integración.
- Alumnado superdotado intelectualmente.
- Alumnado con necesidades educativas que presentan discapacidades físicas, psíquicas, sensoriales o graves trastornos de personalidad o conducta.

Atendiendo a lo dispuesto en el RD 1538/2006, los principios de actuación con este alumnado son la no discriminación y la normalización educativa, a fin de lograr la igualdad de oportunidades para todos.

En esta unidad didáctica, la atención a la diversidad se aplicará conforme al ritmo de aprendizaje del alumnado, manteniendo en todo momento colaboración con el departamento de orientación del centro, que indicará el trato más adecuado a las necesidades del alumnado:

- Para los alumnos con un ritmo de aprendizaje más lento, se plantearán actividades adaptadas a su ritmo y con el consecuente nivel de dificultad adaptado, reforzando los objetivos mínimos para conseguirlos.
- Se utilizará con estos alumnos el trabajo cooperativo; de esta manera se regulan los diferentes ritmos de aprendizaje, ofreciendo más oportunidades de aprendizaje a unos, y reforzando lo aprendido en otros, al realizar explicaciones de contenido entre iguales.

De igual modo, para alumnos con un ritmo de aprendizaje más alto o acelerado, se planteará un mayor número de actividades con una aumentada dificultad.

10. Conclusión final

Cuando se habla de Formación Profesional, se habla de formar a alumnos para adquirir un título que les proporcione unas destrezas profesionales que les permita incorporarse al mundo laboral ofreciendo soluciones funcionales y de calidad. Todos los estudios de informática deben formar en tecnologías y contenidos actuales, que se sigan desarrollando en la actualidad y les sirvan de base para enfrentarse a otras que no conocen consiguiendo una curva de aprendizaje alta.

JavaScript es una tecnología que, como se ha mostrado, no sólo se sigue utilizando, sino que augura seguir evolucionando y siendo el futuro para el desarrollo de aplicaciones y páginas web. Si bien es posible que otros lenguajes o tecnologías aparezcan en el futuro sustituyendo a esta, conocer a fondo este lenguaje de programación, permitirá a los alumnos tener los medios para ofrecer soluciones profesionales y crecer como informáticos a la vez que el panorama cambia.

Como se ha mencionado en el apartado 5 (véase apartado 5. Conclusiones), el perfil tecnológico con conocimientos de desarrollo web es de los más solicitados, por lo que es casi mandatorio impartir la enseñanza de este lenguaje de programación en un módulo orientado a cubrir dichos perfiles con el alumnado que lo curse.

Remarcando aún más estos argumentos, las empresas que acogen a los/as alumnos/as en prácticas suelen requerirles conocimientos básicos de HTML, CSS (especialmente Bootstrap, aunque no se profundizará en este tema) y JavaScript. De modo que lo aprendido en la asignatura puede ser puesto en práctica inmediatamente durante el período de prácticas, pudiendo ampliar su conocimiento y dominio del lenguaje.

Enseñar JavaScript en Formación Profesional es esencial para poder garantizar una educación y formación de calidad para los alumnos que cursen FP, además de una buena oportunidad para conseguir ofertas laborales en las empresas que realizan las prácticas al demostrar su dominio sobre las tecnologías web.

11. Bibliografía

Christian de Looper, Andrew Martinok (20 mayo 2021). *What is 5G? Everything you need to know.* Digital Trends. Recuperado el día 29 de mayo de 2021 de https://www.digitaltrends.com/mobile/what-is-5g/.

Mozilla Developer (24 mayo 2021). *World Wide Web*. MDN Web Docs Glossary: Definitions of web related terms. Recuperado el día 24 de mayo de 2021 de https://developer.mozilla.org/en-US/docs/Glossary/World_Wide_Web.

Berners-Lee, T. J., & Cailliau, R. (1990). WorldWideWeb: Proposal for a HyperText project.

CERN (2 noviembre 2020). *Software release of WWW to public domain.* CERN document server. Recuperado el día 1 de junio de 2021 de http://cds.cern.ch/record/1164399.

Chandra, K., Chandra, S. S., & Chandra, S. S. (2003). A comparison of VBscript, Javascript, and Jscript. *Journal of Computing Sciences in Colleges*, *19*(1), 323-335.

Mozilla Developer (8 junio 2021). *Recursos de lenguaje JavaScript*. Tecnologías para desarrolladores web. Recuperado el día 8 de junio de 2021 de https://developer.mozilla.org/es/docs/Web/JavaScript/Language_Resources.

ECMA 262 - 1st edition (1997). ECMAScript: A general purpose cross-platform programming language*. Standard ECMA-262.*

ECMA 262 - 9th edition (4 junio 2021). ECMASCript 2022 Language Specification. *Draft ECMA-262.* Recuperado el 4 de junio de 2021 de https://tc39.es/ecma262/.

IONOS (10 mayo 2019). *Las librerías y los frameworks JavaScript más populares.* Digital guide IONOS. Recuperado el 4 de junio de 2021 de https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/frameworks-javascript-y-librerias-populares/.

Chaffer, J., & Swedberg, K. (2011). *Learning jQuery*. Packt Publishing Ltd.

Wellman, D. (2009). *jQuery UI 1.6: the user interface library for jQuery*. Packt Publishing Ltd.

Holzner, S. (2008). *The Dojo Toolkit: Visual Quickstart Guide*. Peachpit Press.

Fedosejev, A. (2015). *React. js essentials*. Packt Publishing Ltd.

Pointer, I. (2013). *Instant Zepto. js*. Packt Publishing Ltd.

Mehrabani, A. (2014). *Getting Started with CreateJS*. Packt Publishing Ltd.

Darwin, P. B., & Kozlowski, P. (2013). *AngularJS web application development*. Packt Publ..

Kelonye, M. (2014). *Mastering Ember. js*. Packt Publishing Ltd.

Strack, I. (2015). *Getting Started with Meteor. js JavaScript Framework*. Packt Publishing Ltd.

Oliveira, F. L., & Mattos, J. (2019, November). State-of-the-art javascript language for internet of things. In *Anais Estendidos do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais* (pp. 149-154). SBC.

Sacha Greiff & Raphaël Benitte (2020). *State of JavaScript 2020.* State of JS 2020. Recuperado el 6 de junio de 2021 de https://2020.stateofjs.com/es-ES/.

Bierman, G., Abadi, M., & Torgersen, M. (2014, July). Understanding typescript. In *European Conference on Object-Oriented Programming* (pp. 257-281). Springer, Berlin, Heidelberg.

Delcev, S., & Draskovic, D. (2018, May). Modern JavaScript frameworks: A survey study. In *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)* (pp. 106-109). IEEE.

Juan Cabrera (8 enero 2021). *Los perfiles tecnológicos más buscados en 2021.* Channel Partner. Recuperado el 13 de junio de 2021 de https://www.channelpartner.es/negocios/noticias/1122942002202/perfiles-tecnologicos-mas-buscados-2021.1.html

Pavel Ramírez (3 marzo 2020). *El future del empleo: los 5 lenguajes más demandados en el mercado tecnológico.* Recuperado el 13 de junio de 2021 de https://www.lainformacion.com/management/empleo-lenguajes-programacion-demandados/6548089/

Flanagan, D. (2011). *Java-Script: The Definitive Guide (6th Edition)*.

Korpela, J. K. (2006). *Unicode explained.* " O'Reilly Media, Inc.".

Mozilla Developers (15 junio 2021). *Let*. MDN Web Docs Glossary: Definitions of web related terms. Recuperado el día 15 de junio de 2021 de https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/let

Mozilla Developers (15 junio 2021). *Funciones flecha*. MDN Web Docs Glossary: Definitions of web related terms. Recuperado el día 15 de junio de 2021 de https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Functions/Arrow_functions

Matthieu Faou (14 septiembre 2017). *Minería de criptomonedas en navegadores: la unión hace la fuerza (y la ganancia)*. We Live Security. Recuperado el 15 de junio de 2021 de https://www.welivesecurity.com/la-es/2017/09/14/mineria-de-criptomonedas-navegadores/

Google Code (14 junio 2021). *How does draft SES (Secure EcmaScript) differ from ES5?* Google Code Archive. Recuperado el 15 de junio de 2021 de https://code.google.com/archive/p/es-lab/wikis/SecureEcmaScript.wiki

Bisht, P., & Venkatakrishnan, V. N. (2008, July). XSS-GUARD: precise dynamic prevention of cross-site scripting attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 23-43). Springer, Berlin, Heidelberg.

Dupal, J. (Otoño 2016) *Concurrency support for PureScript*. Masaryk University, Faculty of Informatics.

ANEXOS

Anexo I

En el presente anexo se muestran las relaciones de ejercicios a desarrollar en cada sesión, así como la descripción del proyecto individual a realizar durante toda la Unidad Didáctica.

Proyecto

Realiza una página web básica de una única página html que contenga al menos un ejemplo de cada elemento visto en clase. A modo orientativo, realizando el ejercicio propuesto para el proyecto en la relación de cada sesión, se cumplen los requisitos para completar el proyecto.

La página web tendrá la temática de una página web de telefonía móvil. Deberá incluir a su vez estilos con una gama de colores aptos para personas daltónicas, contemplando todos los tipos de daltonismo.

Sesión 1

1. Ejecuta el siguiente código

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
	<head>
	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
	<title>Hola Mundo con Javascript</title>
	<script type="text/javascript">
	alert("Hola Mundo!");
	</script>
	</head>
	<body>
	<noscript>
<p>Esta página necesita JavaScript para funcionar correctamente. Comprueba si JavaScript está habilitado en el navegador.</p>
	</noscript>
	<p>Un script con la frase más famosa de la programación.</p>
	</body>
</html>
```

    a. ¿Qué ocurre si desactivas Javascript en el navegador? ¿Y si lo habilitas?
    b. **(Ampliación)** Prueba a incluir el script en un fichero con extensión .js e importarlo a la página
2. Escribe texto en un lugar de la página utilizando la función document.write()
3. Modifica texto de la página a partir del id del componente.

Ejercicios propuestos para el proyecto:

1. Incluye un script que muestre un mensaje de bienvenida al entrar en la página.
2. Incluye en tu página web una tabla con nombres de tarifas telefónicas. Después escribe un código en Javascript que modifique el nombre de las tarifas.

Sesión 2

1. Escribe un programa de una sola línea que escriba en la pantalla el resultado de sumar 3 + 5.
2. Escribe un programa de dos líneas que pida el nombre del usuario con un *prompt* y escriba un texto que diga "Hola <nombreUsuario>"
3. Escribe un programa de tres líneas que pida un número, pida otro número y escriba el resultado de sumar estos dos números.
4. Declara dos variables de tipo numérico (con valores asignados) y muestra el valor de dividir ambas con y sin decimales en el resultado.
5. **(Ampliación)** Muestra el resultado de una división con 4 decimales.

Ejercicios propuestos para el proyecto

1. Crea tantas variables como ofertas telefónicas haya y escribe, mediante Javascript, el precio correspondiente a cada oferta.
2. Crea un pequeño cuadro de texto para que el cliente pueda escribir su nombre y aparezca el texto en pantalla.
3. Crea un apartado en la página en el que el usuario pueda realizar sumas de los precios mostrados.

Sesión 3

1. Escribe un programa que pida dos números y escriba en la pantalla cual es el mayor.
2. Escribe un programa que pida 3 números y escriba en la pantalla el mayor de los tres.
3. Escribe un programa que pida un número y diga si es divisible por 2.
4. Escribe un programa que pida una frase y escriba cuantas veces aparece la letra a.
5. Escribe un programa que pida una frase y escriba las vocales que aparecen.
6. Escribe un programa que pida una frase y escriba cuántas de las letras que tiene son vocales.
7. Escribe un programa que pida una frase y escriba cuántas veces aparecen cada una de las vocales.
8. Escribe un programa que pida un número y nos diga si es divisible por 2, 3, 5 o 7 (sólo hay que comprobar si lo es por uno de los cuatro).

9. Añadir al ejercicio anterior que nos diga por cual de los cuatro es divisible (hay que decir todos por los que es divisible).
10. Escribir un programa que escriba en pantalla los divisores de un número establecido previamente.
11. Escribir un programa que escriba en pantalla los divisores comunes de dos números establecidos previamente.
12. Escribir un programa que nos diga si un número dado es primo.
13. **(Ampliación)** ¿Qué ocurre si comparamos utilizando '==' las variables A y B, siendo A="7" y B=7?¿Y utilizando '==='?¿Por qué ocurre esto?

Ejercicios propuestos para el proyecto

1. Reescribe todo el código anterior agrupando el código en funciones.
2. Añade al campo de entrada del nombre de usuario las siguientes restricciones:
    a. No puede tener espacios.
    b. No puede tener menos de 4 caracteres.
    c. No puede contener números, ni comas, ni puntos, ni puntos y comas, ni espacios.
    d. No puede tener más de 20 caracteres.
3. **(Opcional)** Añade la restricción al apartado anterior de eliminar las tildes y acentos de las vocales en el nombre del usuario, sustituyéndolas por las vocales normales, y manteniendo las vocales sin acentos intactas. (Por ejemplo, à -> a; é -> e; o -> o).

Sesión 4

1. Crea un objeto genérico y un objeto del tipo "Alumno", en el cual definas unos atributos. Añade valores a los atributos (del mismo nombre) en cada objeto. ¿Qué diferencia hay entre uno y otro?
2. Crea un objeto mediante el uso de *créate()*.
3. Crea una clase con un constructor genérico y dos constructores con distintos parámetros.
4. Añade métodos a la clase que muestren la longitud de cada uno de los campos.
5. Añade un método que, al finalizar el constructor, muestre una alerta si alguno de los campos no está relleno.
6. **(Ampliación)** ¿Qué ocurre si creamos dos constructores con distintos parámetros en una clase, pero con el mismo número? Por ejemplo, para la clase "Alumno" tenemos los constructores "Alumno(nombre, apellidos)" y "Alumno(curso, clase)". ¿Cómo podemos diferenciarlos a la hora de llamarlos en código?

Ejercicios propuestos para el proyecto

- Crea la clase "Tarifa" y carga los datos en la página a partir de los objetos definidos en código Javascript, manteniendo las funcionalidades implementadas hasta ahora.

Sesión 5

1. Implementa tantos botones como eventos haya del tipo MouseEvent y describe los métodos más similares en cuanto a funcionamiento.
2. Implementa varios *radiobutton* con distintos colores que, al ser pulsados, coloreen el fondo de las tarifas del color seleccionado. Recuerda utilizar colores aptos para daltónicos.
3. **(Ampliación)** Implementa un botón que cambie de posición cada vez que se vaya a pulsar (cada vez que pasa el ratón por encima). Implementa también una función que muestre una alerta si consigue hacer clic en el ratón.

Ejercicios propuestos para el proyecto

1. Añade un *checkbox* a cada tarifa de la página. Posteriormente añade los siguientes botones:
    a. Un botón que muestre la tarifa más barata de las seleccionadas.
    b. Un botón que muestre la tarifa más cara de las seleccionadas.
    c. Si no está seleccionada ninguna, al pulsar el botón debe mostrarse una alerta que avise al usuario de que no tiene ninguna tarifa seleccionada.
2. Añade un evento al nombre de cada tarifa de modo que, al pasar el ratón por encima del nombre, muestre el precio de cada una.
3. Añade otro *checkbox* al lado de cada tarifa (con texto descriptivo) de forma que, si está seleccionado, muestre el precio anual (multiplicado por 12) de la tarifa al pasar el ratón por encima del nombre.
4. **(Opcional)** Busca la manera de implementar la funcionalidad del apartado a) utilizando únicamente un botón. Puedes añadir los elementos que sean necesarios, pero solo puede haber un botón de “Calcular”.
5. **(Opcional)** Implementa una paleta de colores que al seleccionar con el ratón un color cambie el color de fondo de la página. Recuerda utilizar colores aptos para daltónicos.