

TEXT CLASS REVIEW

TEMAS A TRATAR EN EL CUE

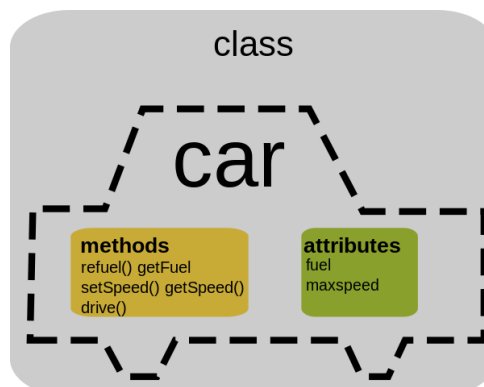
- ES6.
- Variables let y const.
- Interpolado de Strings.
- Arrow Functions.
- Node.js y npm.

PARADIGMAS DE PROGRAMACIÓN

La palabra paradigma significa “modelo” o “ejemplo”, por lo que se puede decir que un paradigma de programación es un modelo a seguir al momento de programar. Existen múltiples modelos que facilitan una programación más ordenada, para lograrlo, el paradigma nos provee de una especie de “normas”.

ABSTRACCIÓN Y PROGRAMACIÓN ORIENTADA A OBJETOS

En la programación orientada a objetos o POO, se trata de modelar la realidad humana, ya que es más fácil representar problemas tal como se presentan en la vida. Es decir, en la POO, se trata de modelar objetos (clases) que se asemejen a la realidad, los cuales pueden realizar acciones (métodos), y poseen características (propiedades).



¿QUÉ ES ES6?

Para contestar esta pregunta, primero debemos analizar la historia del lenguaje que estamos usando. JavaScript fue inventado el año 1995 por Brendan Eich. Éste fue desarrollado para un navegador por nombre Netscape 2, y se convirtió en el estándar ECMA-262 en 1997. Proviene de la especificación ECMAScript, que es una estandarizada de un lenguaje de scripting también desarrollado por Brendan Eich.

Este lenguaje siguió siendo desarrollado a través de los años, y con el tiempo se publicó **ECMAScript** 2009, también conocido como **ES5**, que fue la primera revisión importante de JavaScript. Ésta incluyó características como nuevos métodos del objeto Array, métodos JSON, y un nuevo método Date.

Seis años más tarde se revisó de nuevo esta versión de JavaScript, y se publicó ECMAScript 6, también conocido como ES6 o ECMAScript 2015. Fue la segunda revisión importante que se le hizo al lenguaje.

ES6 es compatible por la mayoría de los navegadores actuales. A continuación, una tabla extraída desde la [W3S](#).

Browser Support for ES6 (2015)

Safari 10 and Edge 14 were the first browsers to fully support ES6:

				
Chrome 58	Edge 14	Firefox 54	Safari 10	Opera 55
Jan 2017	Aug 2016	Mar 2017	Jul 2016	Aug 2018

ES6 tiene una gran variedad de actualizaciones y mejoras para JavaScript. Durante este curso, se cubrirán variados aspectos de esta revisión. En la siguiente imagen podrá apreciar cuanto ha cambiado, simplificando nuestro desarrollo:



Dentro de las todas nuevas características de ES6, empezaremos analizando una nueva manera de declarar variables con las palabras **let** y **const**. Además, veremos nuevas maneras de interpolar Strings y declarar funciones.

NODE.JS Y NPM

En este CUE también estudiaremos una pincelada de Node.js, para comprender e implementar un administrador de paquetes como es npm.

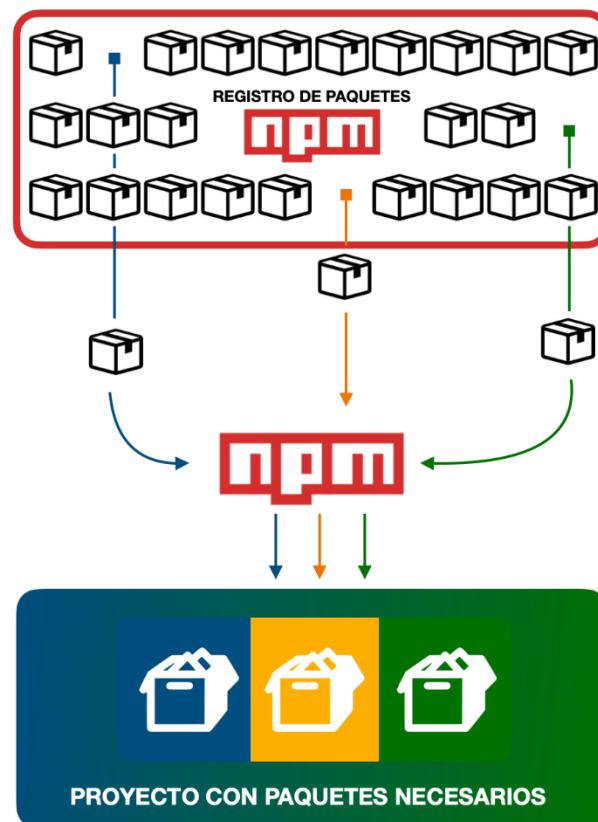


Primero, ¿Qué es node.js?: no es un lenguaje de programación ni un framework, sino un entorno de ejecución para código JavaScript, hecho con la finalidad de fabricar aplicaciones de alto rendimiento. Esto quiere decir que las aplicaciones Node pueden procesar grandes cantidades de datos optimizando recursos.

Al ser un entorno de ejecución, Node.js permite ejecutar código JavaScript en nuestro computador, sin la necesidad de utilizar un navegador. Esto se debe a que utiliza el mismo “motor V8” de JavaScript que emplea Google Chrome. Cada navegador tiene su propio “motor” de JavaScript que, en esencia, es un compilador que convierte el código JavaScript, en código de máquina. En el caso de Chrome, su motor V8 es considerado uno de los más rápidos, y esto es una de las razones por las que aplicaciones Node tienen un muy rendimiento excelente.

En cuanto a Node, solo hemos considerado un poco de su historia. Otras características que contribuyen a su rendimiento es que realiza tareas de manera asíncrona (más de un proceso a la vez), y maneja todas las solicitudes en un hilo. Ésto hace que aplicaciones Node sean ideales para cuando se requiera un alto procesamiento de datos, o cuando se necesitan procesamientos en tiempo real.

Una herramienta muy útil que viene con cada descarga de node.js es npm. Estas letras son las siglas de **Node Package Manager** (administrador de paquetes de Node), y nos permite instalar y administrar los paquetes necesarios (o dependencias) para la ejecución de cualquier proyecto. Estos paquetes o “dependencias”, son simplemente códigos que otras personas han escrito y publicado en el registro de paquetes npm (puedes buscar paquetes publicados en este registro [aquí](#)).



Los paquetes difieren en el caso de uso y en su tamaño, pero todos pueden ayudar a hacer la vida de los desarrolladores mucho más fácil, ofreciendo los componentes básicos necesarios para facilitar todos los desarrollos.