# JavaScript: Introducción a la Orientación a Objetos JavaScript: Introducción a la Orientación a Objetos

## Conclusión 1: Repetición de código

* Creación de clases
* Qué son los atributos/propiedades de una clase
* Instancia/Objeto de una clase
* Palabras reservadas class y new

## Conclusión 2: Agregando métodos

* Creación de métodos (comportamientos de las clases)
* Palabra clave this
* Constructores de las clases
* Encapsulamiento: El patrón módulo trata de encapsular las variables de tal manera que las diferenciamos entre privadas y públicas. Con esto conseguimos el efecto de ocultar lo privado al exterior, no tanto por motivos de seguridad sino para aclarar la vista o perspectiva que desde el exterior del módulo se tiene sobre los datos y métodos del mismo. <https://www.wextensible.com/temas/javascript-closure/encapsulamiento.html>
* Propuesta de atributos privados en clases
* Retorno de resultados con return

## Conclusión 3: Modularizando el código

* Conocer y entender el archivo package.json
* Crear módulos en Javascript
* importación y exportación de módulos
* Ventajas de organizar el código por clases y composición de clases
* Parámetros por valor y por referencia
* Cambiar dinámicamente un objeto
* Extra:
  + Es bueno separar cada clase en módulos, es decir que cada clase vaya en su propio archivo de js (buena práctica)
  + Se puede crear json desde la terminal

## Conclusión 4: Accendiendo a atributos privados

* Diferencias entre null y undefined
* Gestión de la memoria con mayor detalle.
* Referencias a espacios de memoria
* Asistentes: get y set
* Protección de nuestros atributos
* Mejor encapsulamiento de las clases
  + Extras
  + Set y get protegen al Código de daños

## Conclusión 5: Constructores y atributos estáticos

* Constructores
* Encapsulamiento
* Atributos estáticos

## Anexos:

* https://ljcl79.medium.com/el-paradigma-de-la-programación-orientada-a-objetos-d8a662a79fb9

## Código:

* Class
* New
* Null
* Undefined
* Constructor
* Static//impide acceder a el como this y lo deja fijo

## Vocabulario:

* Modelado de datos: Es una estructura que se repetirá N veces, por lo tanto, ayudará a buscar una única definición para una serie de datos y así repetirlos cuanto sea necesario.
* Clase: Es un tipo de molde en js, generan instancias.
* Atributos o propiedades: variables internas del molde o clase en js
* Instancia u objeto: Una instancia u objeto de una clase es una representación concreta y específica de una clase y que reside en la memoria del ordenador. Los atributos son las caracterísiticas individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades.

|  |
| --- |
| poo - Definición y diferencia de objeto e instancia en Java - Stack  Overflow en español |

* Métodos: son los comportamientos de una clase. Ej: En un js para banco el comportamiento deposito seria una clase, o el comportamiento retiro.
* Moldes de clases: son nuestras clases, y generábamos piezas o instancias de esas clases que son nuestros objetos.
* Módulos: en los últimos años se ha comenzado a pensar en proporcionar mecanismos para dividir programas JavaScript en módulos separados que se puedan importar cuando sea necesario. Node.js ha tenido esta capacidad durante mucho tiempo, y hay una serie de bibliotecas y marcos de JavaScript que permiten el uso de módulos. Vamos a crear un archivo para cada clase, donde cada clase va a ser un módulo, cada archivo va a ser un módulo.
* Parámetros por valor: valor primitivo donde se da acceso al valor puntual y no a toda la matriz de datos (lo copia).
* Parametro por referencia o instancia objeto o pieza: js no pasa un valor, si no que pasa una referencia del mismo (le da acceso a toda la matriz del array).
* Null: ayuda a no dejar vacio el parámetro ni undefined.
* Constructor: The constructor method is a special method of a class for creating and initializing an object instance of that class. Es la función que se ejecuta cada vez que se ejecuta la instancia
* Atributos estáticos: