# JavaScript Herencia e Interfaces en Orientación a Objetos

## Conclusión 1: Conociendo el problema del cliente

* Recordamos los conceptos de la Orientación a objetos: Clases, métodos y atributos/propiedades
* Buenas prácticas a la hora de declarar atributos.
* Declarar los atributos privados en la medida de lo posible
* Problemas relacionados a la repetición de código

## Conclusión 2: Herencia

* Especialización del código en clases
* Herencia de atributos/propiedades y métodos
* Sobreescritura de métodos
* Compartiendo el código de forma segura.
* Evitar código duplicado usando el concepto de herencia.
* Palabra reservada super
* Métodos y atributos privados y protegidos.

## Conclusión 3: Clases abstractas

## Conclusión 4: Sistema interno

## Conclusión 5: Interfaces y Duck Type

## Anexos:

* Articulo: <https://ljcl79.medium.com/herencia-en-programación-orientada-a-objetos-370cf3f97402>

## Código:

* Extends: puede dar herencia a un hijo
* Super: se puede buscar la herencia de un padre

## Vocabulario:

* Herencia:

## Capturas:

|  |
| --- |
|  |