

# José Montoya Guzmán

- Desarrollador y líder de equipos Front end (bbva).
- Mentor de fundación BBVA.
- Sensei en Dev.f
- HTML/CSS, JS, React, Lit element, Express (JS), Spring (Java) y SQL.

<https://github.com/montoyaguzman>



# José Rubén Espino

- Ingeniero en Informática
- Sensei en Dev.f
- Encardo del área.
  - Fundación Mexicana para la salud.
- <https://github.com/JoseRuben91>



# Computer Science

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev.f

# ¿Qué veremos?

## 1. Lógica de programación y Algoritmia

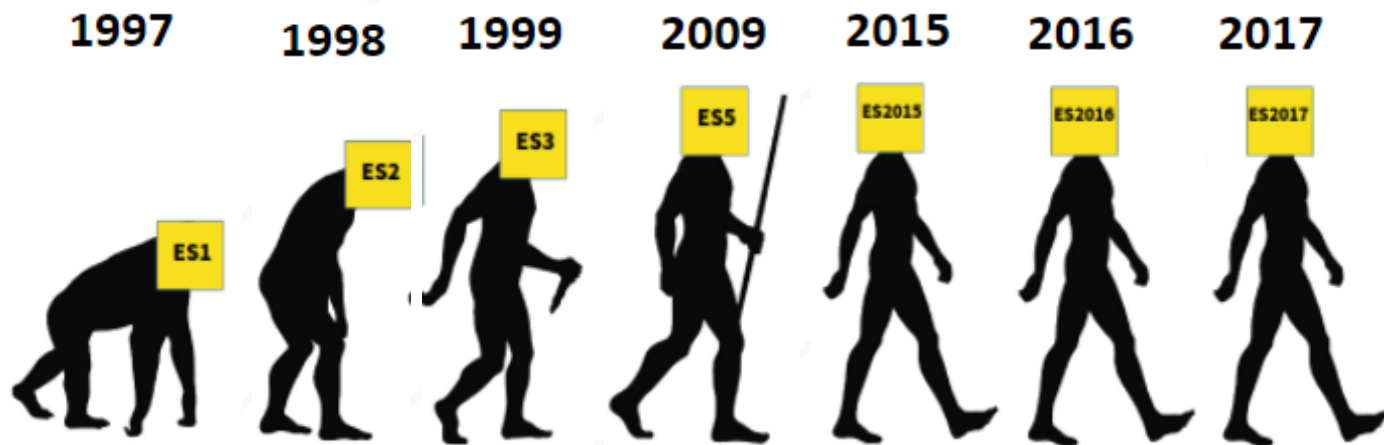
- Introducción a la lógica.
- Análisis de problemas y diseño de soluciones.
- Algoritmos.
- Lenguaje natural.
- Diagramas de flujo.
- Pseudocódigo.



# ¿Qué veremos?

## 2. ECMA

- ¿Qué es ECMA?
- Características de ECMA



# ¿Qué veremos?

## 3. Prototype y POO

- Introducción al diseño de clases.
- ¿Qué es un prototipo?
- Herencia prototipal.
- POO y sugar syntaxis.

4 || We can save memory by adding properties to the prototype that all instances can share, instead of creating new copies of that property each time.

```
function Dog(name, breed, color) {  
  this.name = name  
  this.breed = breed  
  this.color = color  
  this.bark = function() {  
    return 'Woof!'  
  }  
}
```

```
name: 'Daisy'  
breed: 'Labrador'  
color: 'black'  
bark: function(){ ... }  
__proto__
```

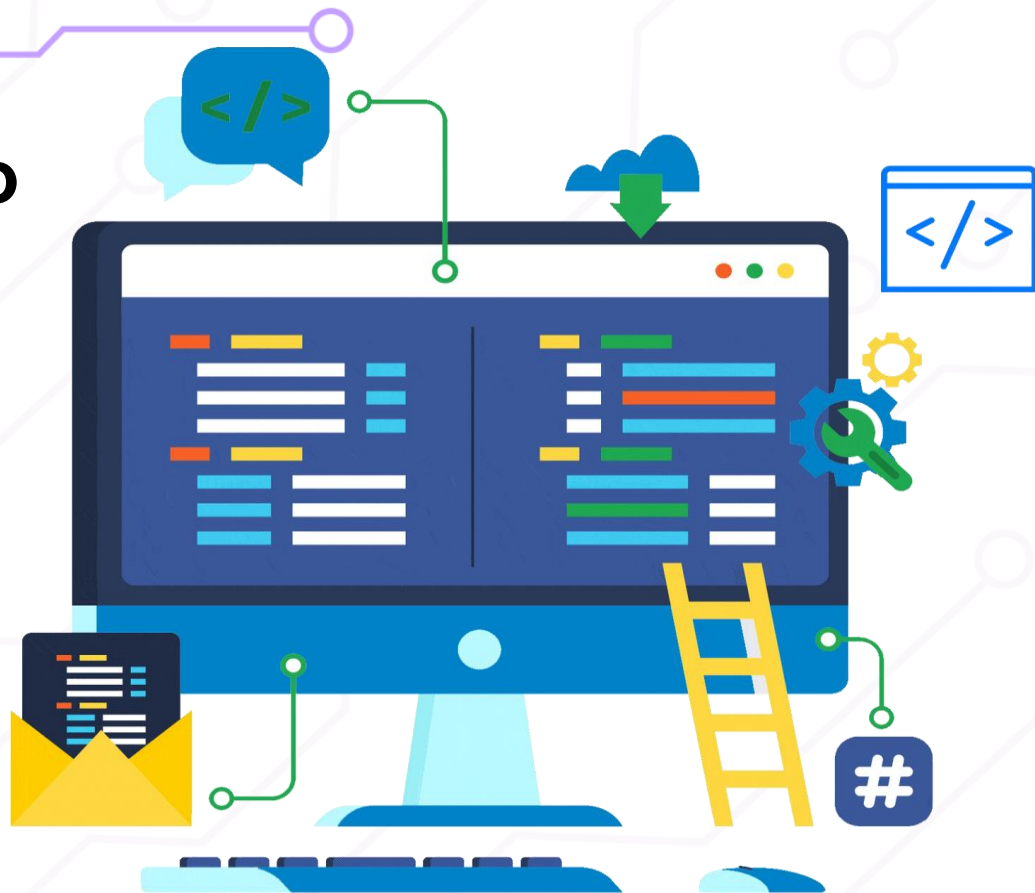
```
name: 'Jack'  
breed: 'Jack Russell'  
color: 'white'  
bark: function(){ ... }  
__proto__
```

Dog.prototype

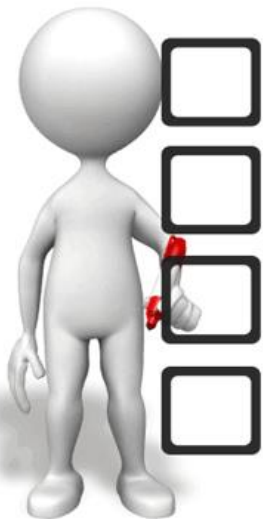
```
constructor: Dog {}
```

¿Qué veremos?

## 4. Proyecto



# Objetivos del módulo.



1. Desarrollar lógica para analizar, entender y resolver problemas del mundo real mediante programación.
2. Entender los fundamentos de ECMA.
3. Diferenciar los conceptos de herencia prototipal y POO.
4. Aplicar herencia prototipal y POO a la solución de problemas.



# Introducción a la lógica

# ¿Que es la lógica?



## Reflexiones.

...esto afirmamos: que si ha de buscarse la verdad en cada división de la filosofía, debemos, antes que nada, contar con principios confiables y métodos para el hallazgo de la verdad. La lógica es la rama que incluye la teoría de los criterios y de las demostraciones; por lo tanto, es con ella con la que debemos comenzar.

-Sexto Empirico

...tanto el mal razonamiento como el bueno son posibles; y este hecho es el fundamento del aspecto práctico de la lógica.

- Charles Sanders Peirce

# ¿Que es la lógica?

La lógica es el estudio de los métodos y principios que se usan para distinguir el razonamiento bueno ( correcto) del malo (incorrecto).



“Es la ciencia del pensamiento racional”

# ¿Que es la lógica?

**A manera de ejemplo, recordemos las siguientes:**

"La lógica es la ciencia de la demostración, pues sólo se preocupa de formular reglas para alcanzar verdades a través de la demostración"  
(Aristóteles).

"La lógica o arte de razonar es la parte de la ciencia que enseña el método para alcanzar la verdad"  
(San Agustín).

"La lógica es la ciencia de las leyes necesarias del entendimiento y de la razón"  
(Kant).

"La lógica es la ciencia de la idea pura de la idea en el elemento abstracto del pensamiento"  
(Hegel).

"La lógica es la ciencia de las aspiraciones intelectuales que sirven para estimación de la prueba"  
(J. S. Mill).

# ¿Para qué sirve la Lógica?

el programador cuenta con un conjunto de herramientas y lenguajes para construir la solución



PROGRAMADOR

HERRAMIENTAS Y LENGUAJES

análisis del  
problema

diseño de la  
solución

construcción  
de la solución

PROBLEMA

proceso

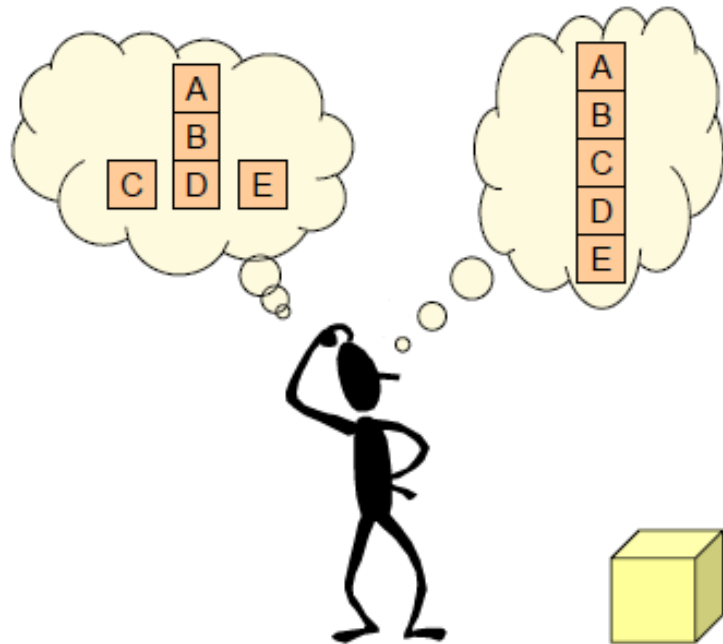
SOLUCIÓN

# Fases de análisis del problema

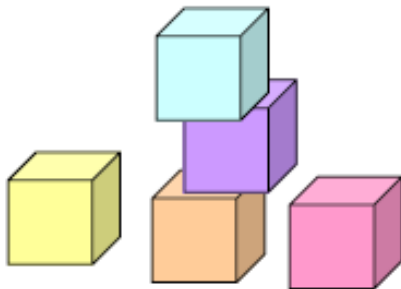
1. Definición.
2. Análisis (entradas, procesos, salidas)
3. Diseño de un algoritmo.
4. Transformación del algoritmo en código.
5. Ejecución y validación.
6. Pruebas de aceptación.



# Análisis



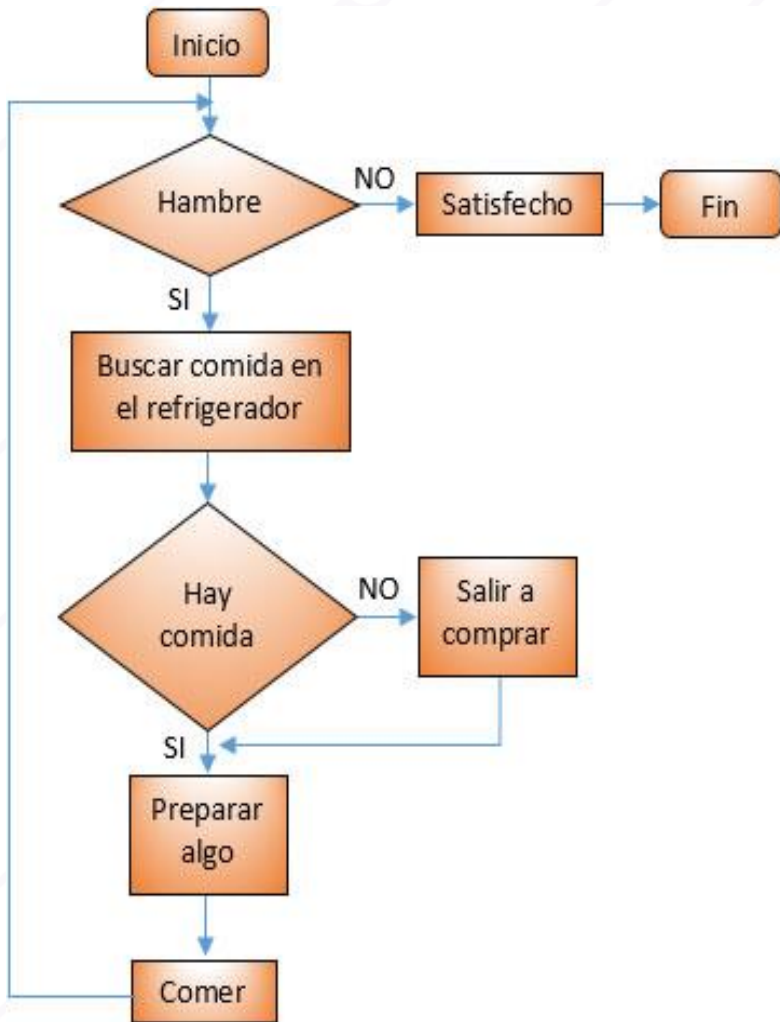
En esta etapa se debe definir y tener claro lo que el programa tiene que hacer junto con las tareas que debe realizar para llevar a cabo la solución.



# Algoritmos.

Un algoritmo es una secuencia de instrucciones secuenciales, que pueden llevarse a cabo ciertos procesos y dar respuesta a determinadas necesidades o decisiones.

Se trata de conjuntos ordenados y finitos de pasos, que nos permiten resolver un problema o tomar una decisión.





# Practica 1

- Analizar y diseñar la solución para el siguiente problema:

**DEFINICIÓN:** ¿Qué tengo que hacer para ir al cine?



## Practica 2

- Analizar y diseñar la solución para el siguiente problema:

**DEFINICIÓN:** ¿Qué tengo que hacer para lavarme las manos?

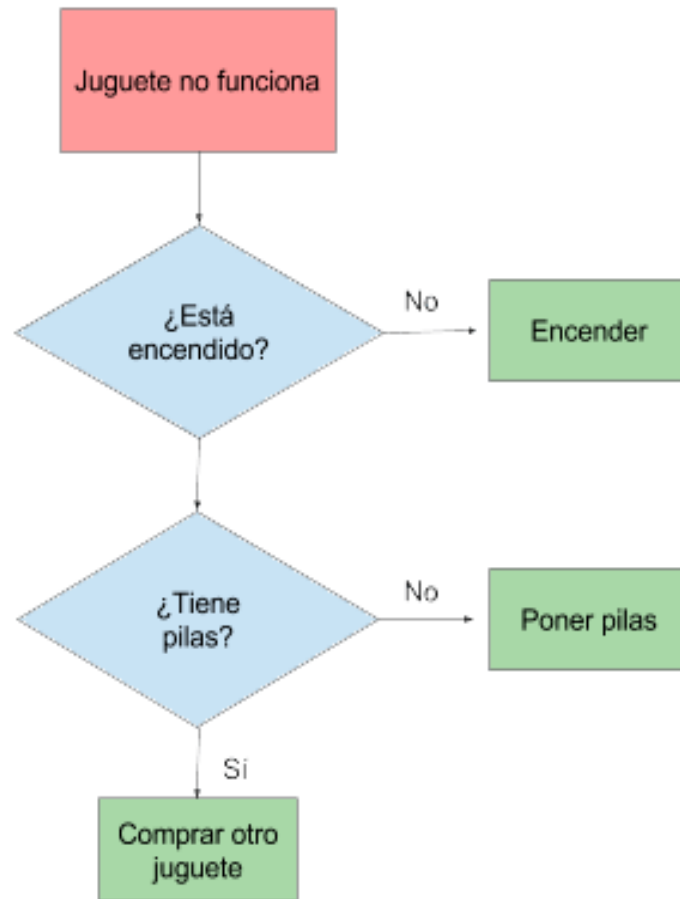


# Algoritmos

**DEV.F**  
DESARROLLAMOS(PERSONAS);

# Algoritmo

Un **algoritmo** es un conjunto de instrucciones definidas, ordenadas y acotadas para resolver un problema, realizar un cálculo o desarrollar una tarea.



# Partes de un algoritmo

Las tres partes de un algoritmo son:

- **Input (entrada).** Información que damos al algoritmo con la que va a trabajar para ofrecer la solución esperada.
- **Proceso.** Conjunto de pasos para que, a partir de los datos de entrada, llegue a la solución de la situación.
- **Output (salida).** Resultados, a partir de la transformación de los valores de entrada durante el proceso.

# Partes de un algoritmo

**Entrada:** Son los datos que se le dan al algoritmo

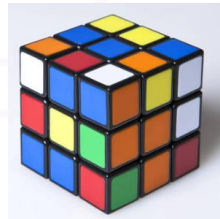
$a = 1;$

$b = 2;$



**Proceso:** Son las operaciones que se hacen con los datos

$\text{Suma} = a + b;$



**Salida:** Es el resultado final que se obtiene de las operaciones,

en este caso será 3

`document.write(Suma)`

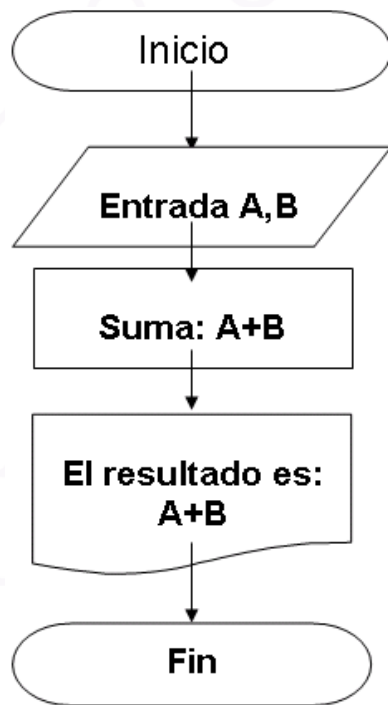
`console.log(Suma)`



# Características de un algoritmo

- **Precisos.** Objetivos, sin ambigüedad.
- **Ordenados.** Presentan una secuencia clara y precisa para poder llegar a la solución.
- **Finitos.** Contienen un número determinado de pasos.
- **Concretos.** Ofrecen una solución determinada para la situación o problema planteados.
- **Definidos.** El mismo algoritmo debe dar el mismo resultado al recibir la misma entrada.

# Como podemos representar un algoritmo.



**DDF**

**algoritmo** Sumar

**variables**

entero a, b, c

**inicio**

escribir( "Introduzca el primer número (entero): " )

leer( a )

escribir( "Introduzca el segundo número (entero): " )

leer( b )

$c \leftarrow a + b$

escribir( "La suma es: ", c )

**fin**

**Pseudocódigo**



**Lenguajes de programación**



# Diagrama de Flujo


**DEV.F**  
DESARROLLAMOS(PERSONAS);

# ¿Qué es un diagrama de flujo?

Un diagrama de flujo es una forma visual la cual nos permite representar un algoritmo o proceso. Se usan diferentes elementos y conexiones las cuales nos permiten representar el algoritmo de una manera visual, estructurada y organizada.



# Símbolos de diagramas de flujo

SÍMBOLO	NOMBRE	SÍMBOLO	NOMBRE
	Flecha de flujo.		Subproceso
	Comentario o anotación		Documento/ Impresora
	Inicio o finalización		Salida en Pantalla
	Proceso		Datos o Entrada/Salida
	Decisión		Referencia en página
			Referencia a otra página