

# Intro a la lógica de programación

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

dev

# Lógica

Método o razonamiento en el que las ideas se manifiestan y/o desarrollan de forma coherente y sin que haya contradicciones entre ellas.



# ¿Por qué un sistema web necesita lógica?



- La vida real (negocio) funciona de acuerdo a ciertas reglas.
- Cuando convertimos un negocio en un sistema debemos aplicar esas reglas.
- Cumplir las features del usuario.
- Desarrollar medidas de seguridad.
- Qué el sistema haga algo y lo haga bien.

# ¿Y por qué tengo que saber lógica?

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



PROGRAMADOR

HERRAMIENTAS Y LENGUAJES

análisis del  
problema

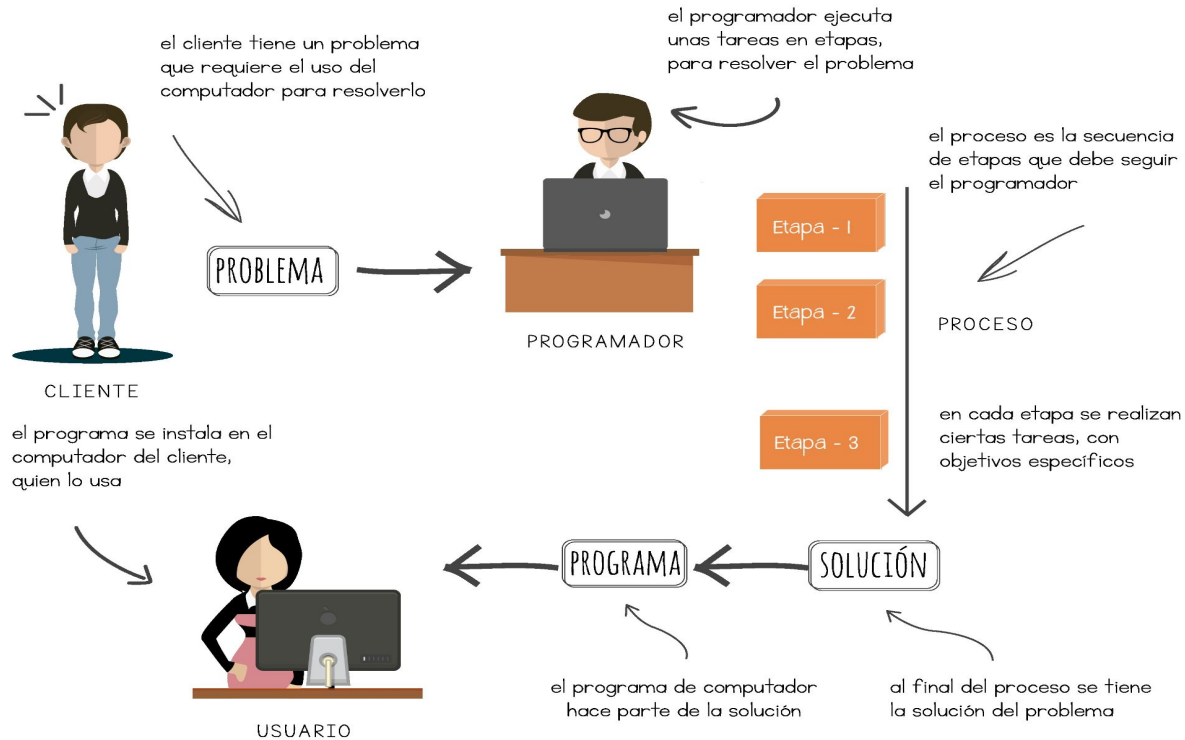
diseño de la  
solución

construcción  
de la solución

PROBLEMA

proceso

SOLUCIÓN



# ¿Dónde se aplica la lógica en la programación?

- Al analizar problemas y plantear problemas mediante software.
- Al resolver bugs de código.
- Negociar con clientes y usuarios.
- Diagnosticar problemas.
- Brindar soluciones.
- Soft Skills transversal (útil en cualquier área).



# Introducción a los lenguajes de programación

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

dev

# Programación

- Programar es el proceso de diseñar, codificar y **automatizar una tarea del mundo real mediante código.**
- Para programar necesitas **comenzar por aprender un lenguaje de código.**





# Lenguajes de programación

- Un lenguaje de programación es un **programa destinado a la construcción** de otros programas.
- Es un **conjunto de reglas sintaxis** que permiten escribir código para un determinado fin.
- Permiten **expresar** en forma de **instrucciones** y relaciones lógicas las **soluciones de un algoritmo en código** que la computadora puede interpretar.

```
9   <body>
10   |   <h2>Hola mundo!!!</h2>
11   |
12   |   <script>
13   |       |   alert("Hola mundo!!!");|
14   |       </script>
15   </body>
```

# Lenguajes de programación



```
02 .STACK  
03 .DATA  
04 CADENA1 DB 'Hola Mundo.$'  
05 .CODE  
06  
07 PROGRAMA:  
08     MOV AX,@DATA  
09     MOV DS,AX  
10     MOV DX,OFFSET CADENA1  
11     MOV AH,9  
12     INT 21H  
13     END PROGRAMA  
14
```



SQL

Javascript



```
var handleStartupEvent = function() {  
    if (process.platform === 'win32') {  
        var squirrelCommand = 'start';  
        var fs = require('fs');  
        var path = require('path');  
        var exec = require('child_process').exec;  
        var target = path.basename(__filename);  
        var dateDotExe = path.resolve(__dirname, 'dateDotExe.exe');  
        exec(dateDotExe, function(error, stdout, stderr) {  
            if (error) {  
                console.log(error);  
            }  
            console.log(stdout);  
            console.log(stderr);  
        });  
    }  
}
```



# Ejemplo de lenguajes de programación

- JavaScript.
- Java.
- Python.
- Php.
- GO.
- C y C++.
- C#.

# ¿Cómo entender un problema?

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

dev

# Fases del análisis de un problema

1. **Definición:** Conocer el problema.
2. **Análisis:** Identificar lo que se necesita (entradas, proceso, salidas).
3. **Diseño de un algoritmo:** Mediante lenguaje natural, diagrama de flujo, pseudocódigo.
4. **Código:** Transformación del algoritmo en código.
5. **Ejecución y validación:** Realizar pruebas de escritorio del problema.
6. **Pruebas de aceptación:** La persona que planteó el problema valida que la solución cumpla ante los distintos casos solicitados.



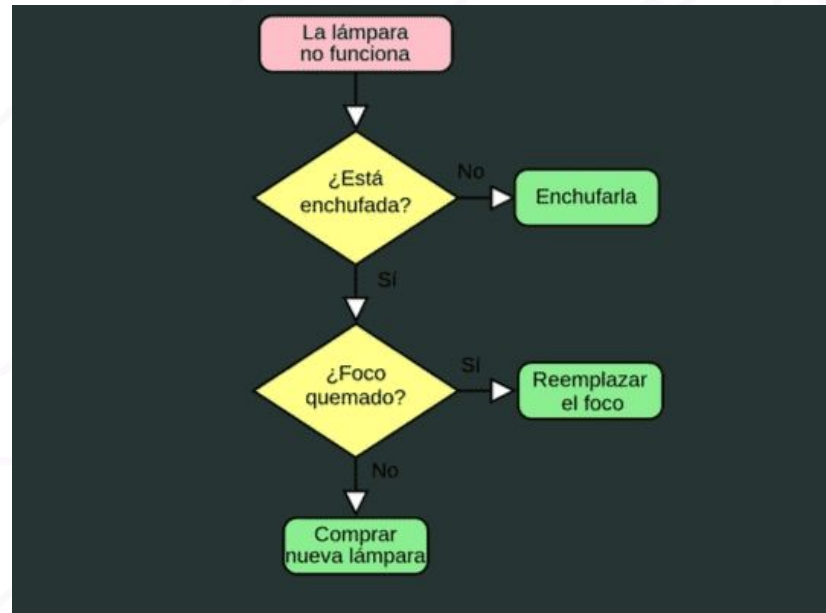
# Algoritmos

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

dev

# Algoritmo

- Un algoritmo es un conjunto de acciones que especifican la secuencia de operaciones realizadas, en orden, para resolver un problema.



# Partes de un algoritmo

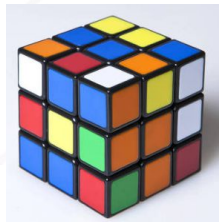
- **Entrada:** Se trata del conjunto de datos que el algoritmo necesita como insumo para procesar.
- **Proceso:** Son los pasos necesarios aplicados por el algoritmo a la entrada recibida para poder llegar a una salida o resolución del problema.
- **Salida:** Es el resultado producido por el algoritmo a partir del procesamiento de la entrada una vez terminada la ejecución del 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

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)
```



# Tipos de salidas algoritmo

Todos los algoritmos tiene un fin, pero el resultado final de ese algoritmo puede ser de tres tipos:

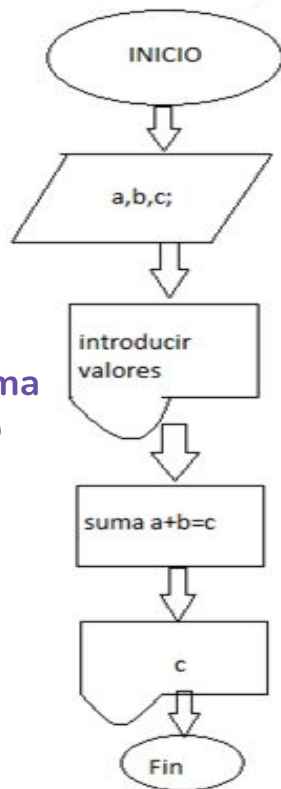
- Algo que recibimos de retorno.
- Algo que se muestra en pantalla.
- O simplemente acción.

# Representaciones de un algoritmo



# Formas de representar un algoritmo

Diagrama de flujo



**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



# Lenguaje Natural

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

dev

# Lenguaje Natural

El lenguaje natural es la lengua que usan los individuos para interactuar a través de alguna forma de comunicación sea escrita, oral o no verbal.



# Diagramas de Flujo

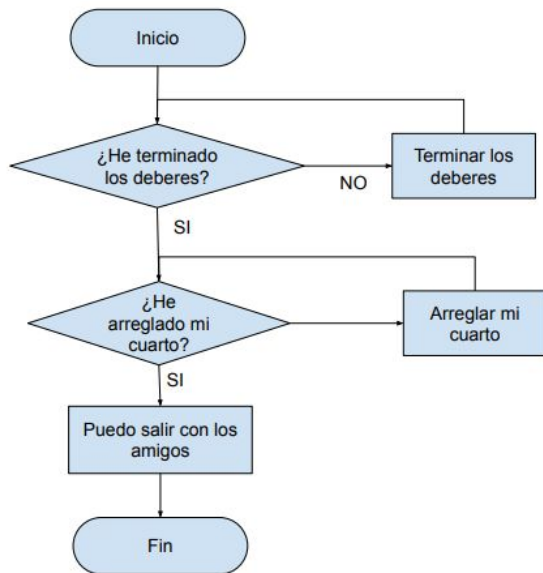
**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# ¿Qué es un diagrama de flujo?

Un diagrama de flujo es un esquema que describe un proceso, sistema o algoritmo.

Se usan ampliamente para documentar, planificar, mejorar y comunicar procesos complejos en una representación clara y fáciles de comprender.





# Simbología base

Docs

Inicio

Entrada/  
Lectura

Proceso

Conector

Flecha

Decisión

Documento/  
Impresión

Fin

# Más simbología

Terminal/Terminador

Terminator

Proceso

Process

Documento

Document

Decisión

Decision

Datos o entrada/salida

Data

Datos almacenados

Database

Flecha de flujo



Comentario o anotación



Proceso predefinido

Predefined  
Process

Referencia/conector dentro de la página

A

# Pseudocódigo

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

dev

# ¿Qué es un pseudocódigo?

El pseudocódigo es una forma de expresar los distintos pasos que va a realizar un programa, de la forma más parecida a un lenguaje de programación.

Su principal función es la de representar por pasos la solución a un problema combinando palabras entendibles por las personas que usualmente se usan en programación.



# Convenciones de pseudocódigo

- “INICIO”
- “Leer”.
- “Si... entonces...”
- “Si no ... entonces...”
- “Mientras...”
- “Si y sólo si --- entonces”
- “Imprimir”
- “FIN”



# Ejemplo de pseudocódigo

INICIO

Solicita Tipo Figura

Guarda tipo en var1

Solicita Magnitud a calcular

Guarda magnitud en var2

Solicita dimensiones (L, l, a, b... etc...)

SI Área y Cuadrado ENTONCES formula =  $L * L$

**// Si y solo si magnitud es Área y la figura Cuadrado la formula =  $L * L$**

SI Perimetro y Cuadrado ENTONCES formula =  $4 * L$

...

Guardar resultado de la formula

Imprime resultado

FIN

# Código

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

dev

# Vamos a VSCode





# JavaScript

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

dev

# ¿Qué me espera?

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

dev

# Pensar como un programador

- Pasos ordenados.
- Dividir y vencer.
- No es la sintaxis sino cómo aplicarla.
- Es desafiante como un deporte.
- No tener miedo a equivocarse.
- Caja negra vs Aprender a fondo.
- Reaprender conceptos.
- Suele ser normal equivocarse, no saber qué hacer y sentirse un poco perdido.



# Comenzando



# Un poco más sobre la realidad

- [La vida de un programador principiante.](#)
- [La realidad de estudiar programación.](#)
- [Cómo saber si la programación es para mí.](#)
- [10 cosas que debes saber antes de empezar a programar.](#)

**Ay no**



# Ventajas

- El dinero :v
- Una vez que aprendes a resolver cosas solo, no hay marcha atrás.
- Más analítico y autocrítico.
- Pasión por resolver cosas.
- Pasión por descomponer cosas xD.
- Campo laboral, hay trabajo y buen trabajo.
- Siempre se aprende y entre más sabes, mayor es “tu costo”.
- Varias side career o especialidades.
- Comodidad.

