

Computer Science

DEV.F
DESARROLLAMOS(PERSONAS);

dev

¿Qué veremos en esta kata?

- Introducción a la lógica.
 - Análisis de problemas y diseño de soluciones.
 - Algoritmos.
 - Pseudocódigo.
 - Diagramas de Flujo.
 - European Computer Manufacturers Association.
 - ECMAScript.
-
- Prototype.
 - POO.
 - Estructuras de datos.
 - Proyecto.



Introducción a lógica de programación

DEV.FX
DESARROLLAMOS(PERSONAS);

dev

¿Qué lógica?



Lógica

- Método o razonamiento en el que las ideas o la sucesión de los hechos se manifiestan o se desarrollan de forma coherente y sin que haya contradicciones entre ellas.



Fases del análisis de un problema

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



Fases del análisis de un problema

Se realizan en lenguaje natural

- Definición.
- Análisis (entradas, proceso, salidas).

Se realizan con gráficos (diagrama de flujo) y notas propias (pseudocódigo).

- Diseño de un algoritmo.

Codear en el lenguaje de programación requerido

- Transformación del algoritmo en código.
- Ejecución y validación.

Lenguaje Natural

El lenguaje natural se entiende como la lengua que usan los individuos para interactuar a través de alguna forma de comunicación sea escrita, oral o no verbal. Una base para comprender el lenguaje natural es el idioma que empleamos cuando hablamos (castellano, inglés, chino, entre otros).



Práctica 1

- Utilizar las 2 primeras fases de la resolución de un problema para solucionar el siguiente proceso:

Definición: ¿Que tengo que hacer para ir al cine?



Práctica 2

Explicar en lenguaje natural como funcionaria una calculadora que sea capaz de calcular area y perimetro de 3 figuras (cuadrado, rectángulo y triángulo equilátero). Solo debe calcular los resultados de una dimensión y de una figura a la vez.



Diagramas de Flujo

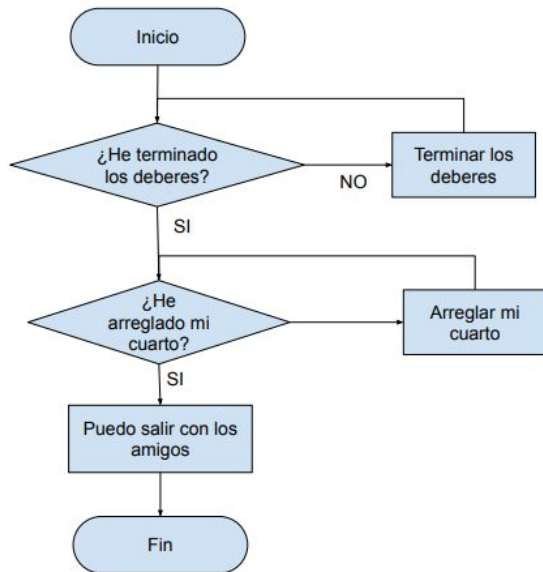
DEV.F
DESARROLLAMOS(PERSONAS);

dev

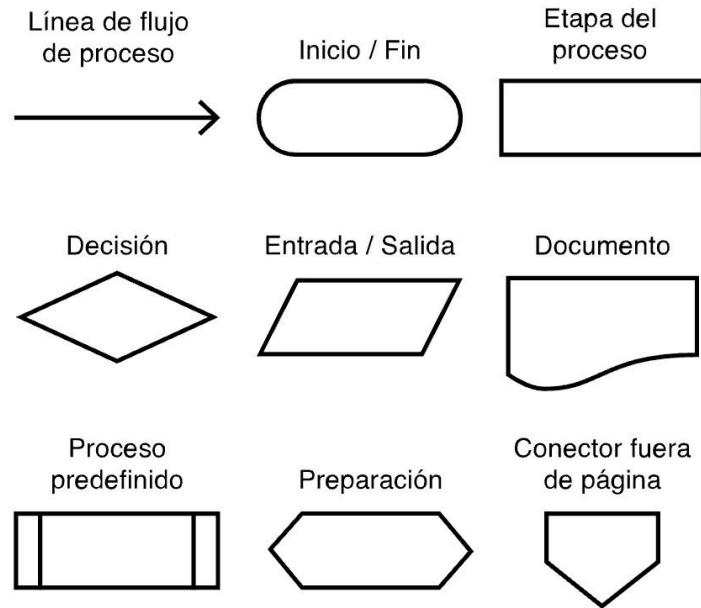
¿Qué es un diagrama de flujo?

Un diagrama de flujo es un diagrama 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



https://docs.google.com/document/d/1mdrgHLbndoxR9WHMv5dwEXOKud--_o1GJ4F4l8SkqTg/

https://drive.google.com/file/d/0B_mAPkMcdtUsMHgtUWZJMWZvYms/view?resourcekey=0-ITkYE1KVVBh_tCutVSL_ew

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.



Ejemplo de pseudocodigo

INICIO

Solicita Tipo Figura

Guarda tipo en var1

Solicita Valor a calcular

Guarda valor en var2

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

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

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

...

Guardar resultado de la formula

Imprime resultado

FIN

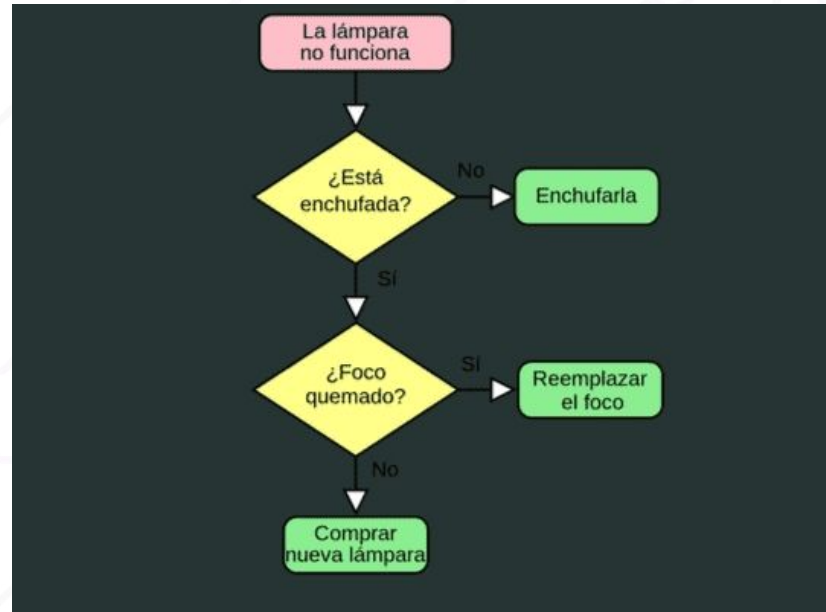
Algoritmos

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Algoritmo

- Un algoritmo es un conjunto de acciones que especifican la secuencia de operaciones realizar, 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.

Características de un algoritmo

- **Exactitud:** el algoritmo tiene que indicar un orden claro de la ejecución de cada paso, estos no pueden ser ambiguos (debe existir una confiabilidad).
- **Estar definido:** si se realiza la ejecución de un mismo algoritmo en distintas instancias utilizando la misma entrada, debe resultar en la misma salida.
- **Completo:** en la solución se deben considerar todas las posibilidades del problema.

Características de un algoritmo

- **Finito:** necesariamente un algoritmo debe tener un número finito de pasos.
- **Instrucciones entendibles:** Las instrucciones que lo describen deben ser claras y legibles.
- **General:** debe poder abarcar problemas de un mismo tema soportando las distintas variantes que se presentan en la definición del problema.

Práctica 3

Realizar la solución al problema de calcular el valor factorial de un determinado numero. Utilice:

- Lenguaje natural.
- Diagrama de flujo.
- Pseudocódigo.
- Código.
- Prueba de escritorio.



Práctica 4

Realizar la solución al problema de calcular la serie de fibonacci de un determinado número Utilice:

- Lenguaje natural.
- Diagrama de flujo.
- Pseudocódigo.
- Código.
- Prueba de escritorio.



Práctica 5

Realizar una calculadora que realice las 4 operaciones aritméticas con 2 números. Utilice:

- Lenguaje natural.
- Diagrama de flujo.
- Pseudocódigo.
- Código.
- Prueba de escritorio.



Práctica 6

Realizar una calculadora que sume, reste y multiplique dos matrices cuadradas. Utilice:

- Lenguaje natural.
- Diagrama de flujo.
- Pseudocódigo.
- Código.
- Prueba de escritorio.



ECMAScript

DEV.F
DESARROLLAMOS(PERSONAS);

dev

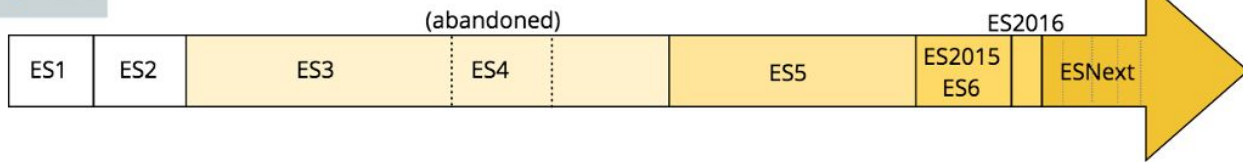
¿Qué es ECMA?

Ecma International es una organización internacional basada en membresías de estándares para la comunicación y la información. Adquirió el nombre Ecma International en 1994, cuando la European Computer Manufacturers Association (ECMA) cambió su nombre para expresar su alcance internacional.

Fue fundada en 1961 para estandarizar los sistemas informatizados en Europa.



ECMAScript



JS



ECMAScript

Es el estándar que la empresa del mismo nombre definió para JavaScript en el año 2015 (ES6) y encarga de regir como debe ser interpretado y cómo debe funcionar el lenguaje JavaScript.

Versión	Nombre	Publicación	Guía
ES11	ES2020	Junio del 2020	-
ES10	ES2019	Junio del 2019	-
ES9	ES2018	Junio del 2018	-
ES8	ES2017	Junio del 2017	-
ES7	ES2016	Junio del 2016	-
ES6	ES2015	Junio del 2015	-
ES5.1	ES5.1	Junio del 2011	-

Características más utilizadas de ECMA

- Let y const.
- Arrow functions (short and long).
- For/of.
- Map.
- Clases.
- Promises.
- Default parameters.
- Rest / Spread operator.

Características más utilizadas de ECMA

- Includes.
- Array => from, keys, find, findIndex.
- Object entries.
- JS Modules.
- Literal templates.
- Clases.
- Map y Set. => Estructuras de datos

Características más utilizadas de ECMA

- Exponentiation.
- Array => includes.

Características más utilizadas de ECMA

- Object => entries, values.
- Async.

Características más utilizadas de ECMA

- Asynchronous Iteration.
- Promise Finally.
- Object Rest Properties.
- New RegExp Features.

Arrays

```
[1, 2, 3].push(4) // [1,2,3,4]
[1, 2, 3].pop() // [1,2]
[1, 2, 3].shift() // [2,3]
[1, 2, 3].unshift(0) // [0,1,2,3]
['a', 'b'].concat('c') // ['a','b','c']
['a', 'b', 'c'].join('-') // a-b-c
['a', 'b', 'c'].slice(1) // ['a','b']
['a', 'b', 'c'].indexOf('b') // 1
['a', 'b', 'c'].includes('c') // true
[3, 5, 6, 8].find((n) => n % 2 === 0) // 6
[2, 4, 3, 5].findIndex((n) => n % 2 !== 0) // 2
[3, 4, 8, 6].map((n) => n * 2) // [6,8,16,12]
[1, 4, 7, 8].filter((n) => n % 2 === 0) // [4,8]
[2, 4, 3, 7].reduce((acc, cur) => acc + cur) // 16
[2, 3, 4, 5].every((x) => x < 6) // true
[3, 5, 6, 8].some((n) => n > 6) // true
[1, 2, 3, 4].reverse() // [4,3,2,1]
[3, 5, 7, 8].at(-2) // 7
```

Bonus

- Tablas de verdad.
- Valor por defecto de una variable.
- Cortocircuito ?.
- Paso por referencia y valor.
- Diferencia entre map y forEach.
- Tipos de funciones en JS.