

	INSTITUCIÓN EDUCATIVA JOSÉ EUSEBIO CARO Tecnología e informática
Docente	Jesús Eduardo Madroñero Ruales
Propósito del taller	Comprender las generalidades de los algoritmos, en diferentes situaciones cotidianas.
Competencias	Analizo el funcionamiento de prototipos conformados por artefactos y procesos como respuesta a necesidades o problemas.

Generalidades sobre los algoritmos

La lógica: Cuando se desarrolla un algoritmo que da solución a un problema determinado, previamente se han debido cumplir los pasos anteriores a él. Como estos son previos a la construcción del programa que ejecutará la computadora, debe haber **coherencia** y **concatenación** en cada uno de los pasos seleccionados para la solución del problema.

El orden en que se disponen los pasos del algoritmo debe ser riguroso; esto implica que deben existir unos pasos antes que otros u otros antes que unos. Por ejemplo: **no se podrá multiplicar A por B si, previamente, no se conocen sus respectivos valores.**

El **algoritmo** es la antesala del programa que ejecutaría la computadora, y cuando éste se traslada al lenguaje escogido para representarlo se debe conservar el orden preestablecido en él, independientemente del lenguaje seleccionado. Un algoritmo, una vez construido, puede expresarse en lenguajes diferentes de programación y ejecutarse en computadoras distintas; sin embargo, el algoritmo será siempre el mismo. De ahí que los **errores lógicos** que se cometan en la elaboración de éste pasarán al lenguaje y, por ende, a la computadora, el cual reproducirá exactamente lo que se le ha mandado; éste no tiene poder para detectar **errores humanos**.



Los procedimientos: Un **procedimiento** es un conjunto de instrucciones o pasos descritos mediante palabras, para llegar a la solución o resultado (s) de un problema que no involucra cálculos matemáticos. Sin embargo, la descripción de cada paso rigurosamente debe conservar un orden, la **entendibilidad** o **generalidad** de éste depende en forma exclusiva de la persona que lo construye. Posiblemente, una persona distinta divida un paso en varios o condense dos o más pasos en uno solo, según la persona lo entienda.

Las expresiones: Una **expresión** es un grupo de operadores que actúan sobre operandos, para proporcionar un único resultado. La expresión puede ser:

1. **Aritmética:** En este caso los operandos son constantes o variables numéricas unidas a través de operadores aritméticos, donde el resultado obtenido de la expresión es un número. Las expresiones aritméticas que involucran más de un operador son evaluadas dependiendo de la prioridad que tenga el operador, de acuerdo con la siguiente tabla:

OPERADOR	PRIORIDAD	SIGNIFICADO
** , \circ , $^{\wedge}$	MAYOR	POTENCIACIÓN
$*$, $/$, $\%$	MENOR	MULTIPLICACIÓN, DIVISIÓN, MÓDULO
$+$, $-$	MUCHO MENOR	SUMA, RESTA

Si dos o más operadores consecutivos tienen la misma prioridad, las operaciones se ejecutarán en la instrucción de izquierda a derecha.

Ejemplo: Si se tiene la expresión:

$$A \ ** \ 2 / 5 * B - 5$$

Y los valores almacenados en A y B son 5 y 20, respectivamente, la evaluación de acuerdo al orden de prioridad será:

$$5 ** 2 = 25$$

$$25 / 5 * 20 = 100$$

$$100 - 5 = 95$$

Si se requiere que una o más operaciones se realicen primero que otras, entonces éstas se encierran entre paréntesis y dentro de éstos se conserva la jerarquía de los operadores.

Ejemplo: La operación:

$$\frac{A + B}{C - A} + 20$$

Debe representarse como: $(A + B) / (C - A) + 20$

2. **Lógica:** En este tipo de expresiones existe por lo menos un operador lógico actuando sobre variables **numéricas, lógicas o caracteres**. El resultado siempre será uno de los valores falso o verdadero. Los operadores lógicos son de dos clases: **relacionales o booleanos**.

Los **operadores relationales** permiten realizar comparaciones de tipo numérico, carácter o lógico, y tienen la misma prioridad en su evaluación. Estos se muestran en la siguiente tabla:

OPERADOR	SIGNIFICADO
=	IGUAL
< >	DIFERENTE DE
< =	MENOR IGUAL QUE
> =	MAYOR IGUAL QUE
<	MENOR QUE
>	MAYOR QUE

Ejemplos: Si las variables X y Z tienen almacenadas internamente los valores 30 y 40, respectivamente, el resultado de las siguientes expresiones será:

EXPRESIÓN	SIGNIFICADO
X = Z	F
X > Z	F
(X + 20) < > (Z + 1)	V
Z > = X	V

Los **operadores booleanos** operan sobre información lógica, uniendo condicionales simples para formar condiciones compuestas. Estos operadores son:

OPERADOR	PRIORIDAD	SIGNIFICADO
~ (NOT)	MAYOR	NEGACION
^ (AND)	MENOR	'Y', LOGICA
v (OR)	^MUCHO MENOR	'O', LÓGICA

Donde el primero es llamado **operador unario**, porque actúa sobre un solo valor de verdad; y los dos siguientes, **operadores binarios**, puesto que operan sobre dos valores de verdad.

Ejemplo: si a y b son condiciones de tipo lógico, los resultados al aplicarles los operadores booleanos se muestran en la siguiente tabla:

a	b	$\sim a$	$a \wedge b$	$a \vee b$
V	V	F	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	F

Es decir, si se unen dos condiciones a través del operador \wedge , el resultado solamente será verdadero si las condiciones son verdaderas; y si se unen a través del operador \vee , el resultado será falso si las dos condiciones son falsas.

Recursos complementarios

[1] Originpath Academy (2017, 14 de noviembre). Diccionario Coder - ¿Qué es un programa? [video]. <https://youtu.be/8CfnPapC0kc>

[2] Escuela de Videojuegos (2014, 27 de octubre). Curso Programación [#10] Expresiones [video]. <https://youtu.be/JneEZ5GbI5g>

[3] Escuela de Videojuegos (2014, 27 de octubre). Curso Programación [#11] Tipos de operadores [video]. https://youtu.be/P_cpMPfto_Q

Actividad

1. En programación, ¿para qué sirven, los procedimientos, las expresiones y los operadores?
2. Explicar en sus propias palabras la siguiente frase:

*“Como estos son previos a la construcción del programa que ejecutará la computadora, debe haber **coherencia** y **concatenación** en cada uno de los pasos seleccionados para la solución del problema.”*

3. En programación, ¿a qué se denomina **operador unario**? ¿En qué se diferencia de un **operador binario**?
4. Describir los diferentes tipos de expresiones y operadores detallados en el presente documento.
5. Elaborar un algoritmo que resuelva las siguientes situaciones:
 - Muestre los pasos necesarios para conectar un celular a un parlante Bluetooth.
 - Resolver una sopa de letras.
 - Cambiar un bombillo dañado.
 - Ingresar a su correo electrónico.
 - Preparar un sándwich.
6. Consultar información acerca de las funciones: **AND**, **OR** y **NOT**. ¿Para qué se utilizan?

Retos Makecode

Los siguientes retos deben ser desarrollados en grupos de máximo dos estudiantes, haciendo uso de la plataforma MakeCode (<https://makecode.microbit.org/>). Luego de ser desarrollados, deben ser enviados a través del formulario alojado en: <https://intec-iec.webnode.es/> o por mensaje en Edmodo.

1. Diseñar un algoritmo para la tarjeta MicroBit, el cual al presionar el botón A, debe presentar en la matriz de leds, la letra “F”. Si se presiona el botón B, debe presentar la letra “V”.
2. Diseñar un algoritmo para la tarjeta MicroBit, el cual al presionar el botón A, debe generar un número aleatorio binario. Si se presionar el botón B, el valor del numero binario generado debe cambiar al otro valor binario.