

	GIMNASIO SABIO CALDAS (IED) Nuestra escuela: una opción para la vida GUÍAS DE APRENDIZAJE – PLAN ESCOLAR	Código	PENP - 01
		Versión	001
		Fecha	18/03/2020
		Proceso	Gestión Académica

DOCENTE	Yudi Adriana Soler Franco		Grado	NOVENO
ASIGNATURA	PROGRAMACIÓN			
Correo electrónico de contacto	yudi.soler@sabiocaldas.edu.co			
Periodo académico	Segundo Periodo			
Tiempo de ejecución de la actividad	9 de agosto al 20 de agosto			
¿Qué competencia(s) debo alcanzar?	Solución de problemas con tecnología Identifico y comparo ventajas y desventajas en la utilización de artefactos y procesos tecnológicos en la solución de problemas de la vida cotidiana.			
Temáticas mediadoras	APLIQUEMOS LO APRENDIDO - Solución de ejercicios prácticos - algoritmos: estructuras simples, condicionales y bucles.			
Metas	Socio-afectiva: Reconoce las dificultades que presentó durante el periodo, y asume actitudes de compromiso y responsabilidad.			
	Metas de aprendizaje: Aplica los conceptos vistos en clase, para la solución de situaciones planteadas a través de los algoritmos.			

CRITERIOS DE EVALUACIÓN:

¿QUÉ SE VA A EVALUAR?	¿CÓMO SE VA A EVALUAR?	¿CUÁNDO SE VA A EVALUAR? Fechas
Comprender y aplicar los conceptos vistos en clase, para la solución de algoritmos a través del programa Pseint.	Resolución de problemas dados, a través de los procesos vistos en clase.	Primera semana: 10 de agosto
		Segunda semana 17 de agosto

SEMANA 1 (9 al 13 de agosto)

ACTIVIDAD INICIAL:

Responde

- Menciona 5 características aprendidas sobre la solución de algoritmos
- Menciona qué diferencias identifican entre algoritmos son estructuras sencillas, condicionales y ciclos.

CONTEXTUALIZACIÓN:

Importancia de los Algoritmos



Cuando realizamos un programa de computadora, no podemos empezar a escribir sentencias (órdenes, instrucciones) sin más. Antes, tenemos que planificar (y más concienzudamente cuanto mayor y más difícil sea el programa que queremos construir) qué es lo que va a hacer exactamente el programa y cómo va a hacerlo, y luego traducir este algoritmo a instrucciones.

Definimos un algoritmo como una secuencia finita y estructurada de órdenes que constituyen las instrucciones para una

computadora u otra máquina análoga con capacidad de procesamiento (como un smartphone).

El algoritmo no es el programa en sí, sino más bien este último es la concreción del algoritmo en un lenguaje que la máquina que debe ejecutarlo pueda entenderlo.

Y aquí, una pequeña aclaración: los algoritmos son generalmente implementados en las computadoras mediante lenguajes de programación de alto nivel, que no son directamente ejecutables, pero que sí son "compilables" (un proceso que genera un *software* que la computadora puede entender y ejecutar).

Este lenguaje de programación de alto nivel es un punto intermedio entre algo comprensible para un humano de a pie (aunque requiere ciertos conocimientos técnicos) y lo que puede entender y ejecutar directamente una computadora.

Si buscamos una definición enciclopédica, los algoritmos son tan viejos como las primeras instrucciones plasmadas para llevar a cabo una tarea.

Pongamos un ejemplo: ordeñar una vaca. primero tranquilice al animal. Luego ponga un recipiente bajo las ubres. Seguidamente, siéntese mirando a las ubres de la vaca. Ponga sus manos alrededor de dos de las cuatro ubres del animal y oprima ligeramente la base de cada una de las ubres. Repita el movimiento hasta que de las ubres no quede leche. Ahora haga lo mismo con las dos ubres restantes hasta que también estén vacías. Retire el recipiente lleno de leche. Ha acabado.

Repasemos: tenemos un conjunto de instrucciones concretas y que acaban en un punto determinado. Tenemos un algoritmo para ordeñar vacas.

En todos los algoritmos encontramos elementos que se repiten, como bucles (repetir el movimiento), toma de decisiones (¿está la ubre vacía? No -> continuar ordeñando; si -> pasar a las otras dos o finalizar), y acciones.

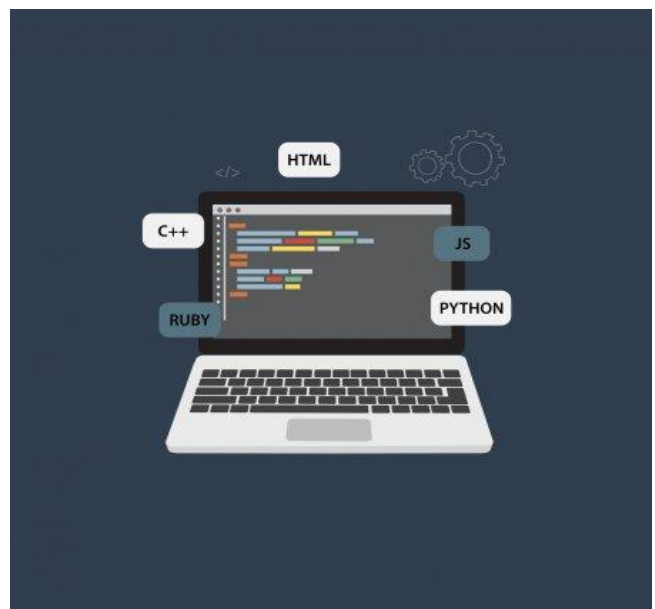
Estos elementos, bucles, condiciones, saltos y acciones, tienen sus correspondencias en las estructuras de los lenguajes de programación, aunque estos disponen de mayor número de estructuras para poder hacer frente a la resolución de los problemas concretos que se presentan.

El algoritmo es una abstracción de la resolución de un problema, que al implementarse en un lenguaje de programación necesita de unas estructuras más complicadas.

Este es el caso de las funciones, que realizan una tarea determinada y que se pueden llamar desde diferentes partes del programa.

Existen diferentes formas de expresar algoritmos.

La más intuitiva es la del lenguaje natural, como en el caso del ordeñamiento de una vaca que he explicado antes. Simplemente nos basta describir de una forma concisa las instrucciones para llevar a cabo una tarea. Sin embargo no es una forma estandarizada, solamente funciona para uno mismo o entre un grupo muy reducido de personas, y para tareas



realmente simples.

Si buscamos algo reglado y estandarizado tenemos los diagramas de flujo, una forma gráfica muy visual que representa las principales acciones y decisiones que se toman en la ejecución del algoritmo.

El pseudocódigo es una representación formal en texto mucho más parecida a un lenguaje de programación, que dispone de los elementos de este y que es fácilmente trasladable a un código compilable.

ACTIVIDAD DE AFIANZAMIENTO:

Resuelve

- A. Se ingresa por teclado un valor entero, mostrar un mensaje que indique si el número es positivo, nulo o negativo.
- B. Elaborar un algoritmo que permita ingresar el número de partidos ganados, perdidos y empatados, por algún equipo en el torneo de apertura, se debe de mostrar su puntaje total. Teniendo en cuenta que por cada partido ganado obtendrá 3 puntos, empatados 1 punto y perdidos 0 puntos.
- C. Diseñe un algoritmo para calcular el promedio de tres números. N_1, N_2, N_3

VERIFICACIÓN DE APRENDIZAJES:

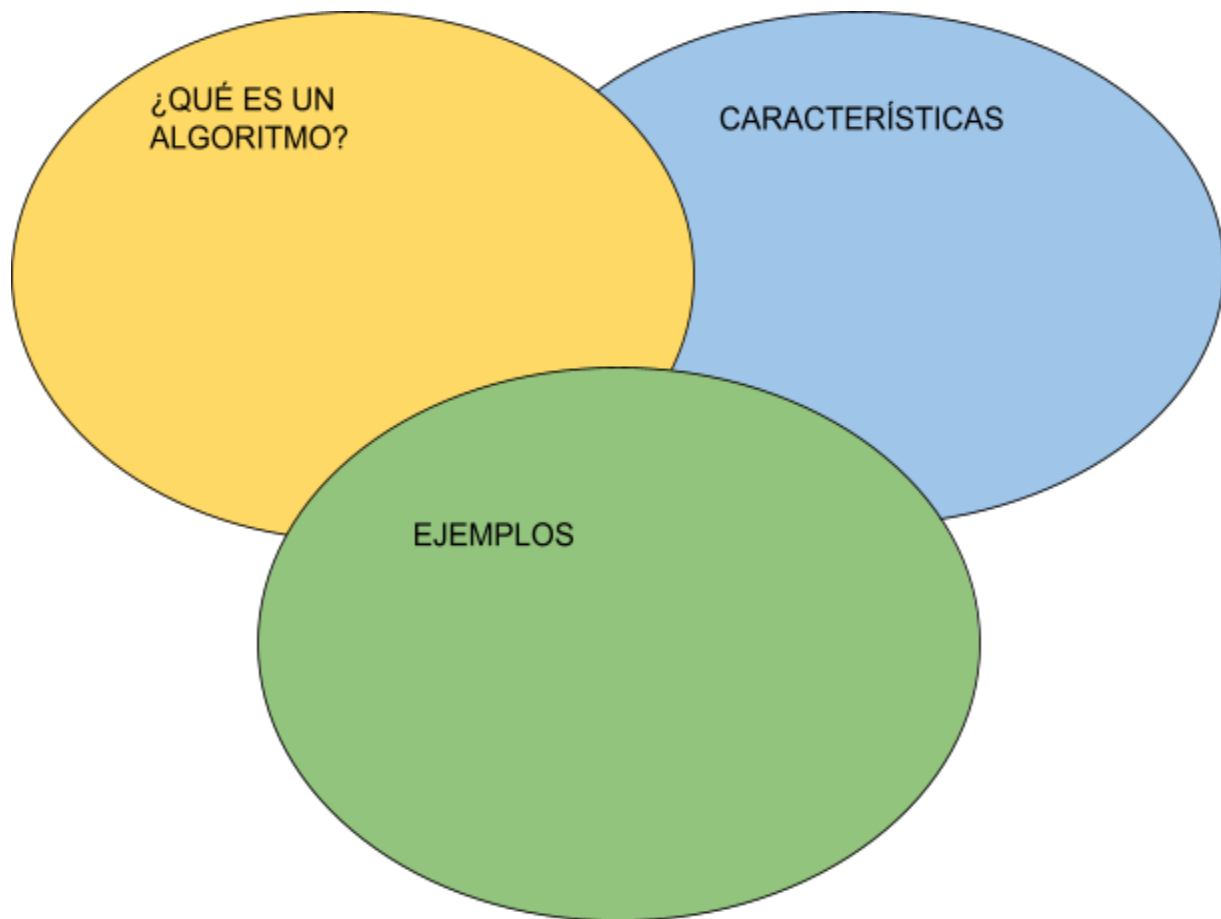
¿Qué comprendiste del tema visto?

¿Qué dudas tengo acerca del tema visto?

SEMANA 2 (16 de agosto al 20 de agosto)

ACTIVIDAD INICIAL:

Realiza en tu cuaderno el siguiente diagrama.



CONTEXTUALIZACIÓN:

Hacer un algoritmo en Pseint para calcular la suma de los primeros cien números con un ciclo mientras.

```
Algoritmo Suma_Numeros_Mientras
Definir suma, conta Como Entero;
conta <- 0; suma <- 0;
Mientras (conta<100) Hacer
    conta <- conta+1;
    suma <- suma+conta;
FinMientras
Escribir 'La suma es: ', suma;
FinAlgoritmo
```

Algoritmo

Algoritmo Suma_Numeros_Mientras
Definir suma, conta Com...
conta ← 0; suma ← 0
Mientras (conta<100)
conta ← conta+1
suma ← suma+conta
Escribir 'La suma es: ', suma
FinAlgoritmo

```
Suma_Numeros_Mientras
Definir suma, conta Como Entero;
conta <- 0; suma <- 0;
Mientras (conta<100) Hacer
    conta <
```

ACTIVIDAD DE AFIANZAMIENTO:

Soluciona los siguientes ejercicios, según lo explicado en clase.

1. Escribe los números del 1 al 200, con un ciclo mientras
2. Hacer un algoritmo en Pseint para calcular la suma de los primeros cien números con un ciclo para.
3. Con un ciclo for, representa la tabla de multiplicar del número 7.

VERIFICACIÓN DE APRENDIZAJES:

¿Qué tanto aprendí del tema?

¿Qué dudas me surgen del tema?

REFERENCIAS: WEBGRAFÍA.

<https://www.importancia.org/algoritmos.php>

<https://www.aprenderaprogramar.pro/2017/07/ejercicios-algoritmos.html>