	INSTITUCIÓN EDUCATIVA JOSÉ EUSEBIO CARO **Tecnología e informática**
Docente	Jesús Eduardo Madroñero Ruales
Propósito del taller	Entender la importancia de la programación y el pensamiento computacional, en situaciones cotidianas.
Competencias	Analizo el funcionamiento de prototipos conformados por artefactos y procesos como respuesta a necesidades o problemas.

Introducción a la programación

El tema a desarrollar es uno de los más importantes en la formación académica de cualquier estudiante, porque aporta **técnicas**, **habilidades** y **hábitos** que le servirán para la vida práctica, en la **solución de problemas** (no únicamente en ingeniería).

Aprender a resolver problemas que se presentan a lo largo de la existencia utilizando un orden establecido, el algoritmo adecuado y la lógica, nos lleva a un desarrollo pleno en todos los ámbitos: en el hogar, en la escuela, en la universidad y en el lugar de trabajo.

Algoritmos y programación

Muchas palabras usadas en la aritmética provienen del lenguaje árabe, debido a los grandes adelantos que tenían los árabes en esta materia. Mohamed ben Musa (780-950 d.C.) padre del álgebra, era conocido con el seudónimo de Al Jwarizmi, término que fue cambiando con el tiempo a algorismo; hasta convertirse finalmente en **algoritmo** por la influencia y similitud fonética.

Para casi todos los problemas, y hasta para las más elementales acciones que se tienen que llevar a cabo diariamente, se realizan secuencias de pasos, a veces inconscientemente.

Ejemplo 01: Suponiendo que se va a ir a una fiesta. Una de las posibles soluciones (paso a paso) del escenario anterior, se presenta en el siguiente cuadro:

Algoritmo: 1. Inicio 2. Seleccionar la ropa a ponerse. 3. Seleccionar los zapatos. 4. Bañarse. 5. Vestirse. 6. Ponerse los zapatos. 7. Peinarse. 8. Lavarse los dientes. 9. Salir de la casa. 10. Dirigirse al lugar de la fiesta. 11. Fin	Tenga en cuenta que la situación, puede ser solucionada de muchas maneras. La solución anterior es tan solo, una de muchas soluciones posibles. De esta manera se llevan a cabo todas las actividades del ser humano. Las recetas de cocina y los manuales de procedimientos son ejemplos claros de algoritmos. Las computadoras utilizan estos pasos lógicos para resolver los problemas o realizar actividades como el procesamiento de textos, los cálculos, el ordenamiento y manejo de los datos, el diseño de presentaciones, la creación de gráficas e imágenes y, en general, todo lo que se hace con los programas de cómputo y aplicaciones.

Para aplicar algoritmos a la solución de problemas, se debe entender primero lo que es un problema: La palabra problema tiene muchas acepciones, las más empleadas son:

- Situación difícil que debe resolverse.
- Cuestión que se trata de aclarar o resolver.
- Cuestión en que hay algo que averiguar o alguna dificultad.
- Cuestión en la que se conocen algunos datos mediante los cuales es posible encontrar otros valores o datos.
- Asunto del que se espera una solución.
- Situación que nos presenta la necesidad cambiar algo que tenemos por algo que deseamos; es decir, la satisfacción de las necesidades es una solución de los problemas.

Se debe tener en cuenta, que no todos los problemas representan situaciones difíciles que han de resolverse. Algunos problemas son tan triviales como asistir el sábado a una fiesta. La serie de pasos lógicos que han de llevarse a cabo para hacerlo, es lo que se conoce como algoritmo.

Los problemas se pueden resolver de diversas maneras; sin embargo, los que nos atañen, que requieren de las computadoras y un lenguaje de programación para proporcionar soluciones a los usuarios de computadoras, siempre

deben ser resueltos utilizando algoritmos y la lógica, ya que los circuitos de una computadora trabajan de esa manera; utilizando los **operadores lógicos del algebra de Boole**.

Diccionario

- **Algoritmo:** Es una serie finita de pasos o instrucciones que deben seguirse para resolver un problema.
- **Programa:** Conjunto de instrucciones, funciones y comandos, que indican a la computadora lo que debe hacer.
- **Lenguaje de programación:** Lenguaje que permite el control de las computadoras, mediante símbolos, instrucciones y enunciados, sujetos a una serie de reglas sintácticas y semánticas. Se utilizan para crear programas.
- **Lógica:** Forma de pensamiento razonado que se basa en el conocimiento científico para obtener mejores resultados.

Recursos complementarios

[1] Educar Portal (2019, 11 de julio). Microaprendizaje: ¿Qué es el pensamiento computacional? [video] https://youtu.be/ti315UlVtS4
[2] Magic Markers (2015, 21 de julio). ¿Qué es un algoritmo? [video]. https://youtu.be/U3CGMyjzlvM

Actividad

1. En sus propias palabras responder, ¿qué son los algoritmos? ¿Qué utilidades tienen en la vida cotidiana?
2. Describir dos algoritmos que utilices para resolver situaciones cotidianas.
3. ¿Qué programas (de PC) o aplicaciones (de celular) utilizas en tu vida diaria? Mencionar la utilidad y que situaciones resuelven.
4. Consultar la biografía de **George Boole** y responder: ¿Cuál fue el aporte más significativo a la programación?
5. Consultar la biográfica de **Alan Turing** y responder: ¿Cuál fue su aporte más significativo a la computación?
6. ¿La programación se utiliza únicamente en ingeniera y ciencias exactas? Argumentar la respuesta con un ejemplo.

Retos Makecode

Los siguientes retos deben ser desarrollados en grupos de máximo dos estudiantes, haciendo uso de la plataforma MakeCode (https://makecode.microbit.org/).

1. Diseñar un algoritmo para la tarjeta MicroBit, el cual debe presentar el primer nombre del estudiante al presionar el botón A. Cuando se presione el botón B, se debe presentar el primer apellido.
2. Diseñar un algoritmo para la tarjeta MicroBit, el cual realice y presente un conteo positivo (de 0 a 5) al presionar el botón A, y un conteo inverso (de 5 a 0) al presionar el botón B.

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

Características de los algoritmos

El **algoritmo** es una secuencia de pasos o instrucciones que representan la solución de un determinado tipo de problema. Cuando se quiere solucionar un problema a través de la computadora, se exige que el algoritmo muestre la secuencia de solución del mismo.

Al analizar los pasos para la solución de un problema a través de la computadora se nota que el algoritmo es bastante importante, y a él debe llegarse cuando se ha entendido el problema y se ha hecho un análisis concreto de las características de éste. Es común que las personas por ligereza omitan uno o varios pasos y luego se dan cuenta de que han invertido su tiempo en algo que está erróneo o equivocado desde el principio, debido a que no han verificado si lo que están haciendo es correcto o no. Si no se está seguro de la implementación de uno o algunos de los pasos, es preferible pedir ayuda especializada para aclarar las dudas que surjan.

Características fundamentales de los algoritmos

Las **características fundamentales** que debe cumplir todo algoritmo son:

- **Entradas:** Las entradas hacen referencia a la información proporcionada al algoritmo, la cual debe sufrir un proceso para obtener los resultados. Un algoritmo tiene cero o más datos de entrada, estos valores le son dados por medio de una instrucción o mandato que debe cumplir al ejecutarse el algoritmo.
- **Salidas:** Todo algoritmo debe proporcionar uno o más valores como resultado, una vez se ha ejecutado la secuencia de pasos que lo conforman. La salida es la respuesta dada por el algoritmo o el conjunto de valores que el programador espera que le proporcionen. Estos resultados pueden ser de cualquier tipo: uno o más valores **numéricos, valores lógicos** o **caracteres**.
- **Limitado o finito:** Todo algoritmo debe tener un número finito de instrucciones que limitan el proceso en algún momento, es decir, la ejecución debe detenerse. No puede existir un algoritmo, por muy grande que sea o por muchos resultados que produzca, que se quede en forma indefinida ejecutando sus instrucciones o repitiendo la ejecución de un subconjunto de ellas.
- **Finalización:** Un algoritmo debe indicar el orden de realización de cada uno de sus pasos. Debe mostrar la primera, la intermedia y la última instrucción que debe realizarse. Esto permite mostrar que en algún momento debe culminar la acción o tarea que hace el algoritmo.
- **Claridad:** Todo el conjunto de pasos debe ser entendible y factible de realizar, de tal manera, que, al hacer un seguimiento del algoritmo, éste produzca siempre los resultados requeridos. No puede entonces existir incertidumbre en las acciones a tomar cuando se sigue la lógica (flujo del programa) del algoritmo.
- **Estructura básica:** Generalmente, todo algoritmo se compone de tres partes:
    - **Entradas:** Información dada al algoritmo, o conjunto de instrucciones que generen los valores con que ha de trabajar, en caso de que no tenga datos de entrada.
    - **Procesos:** Cálculos necesarios para que a partir de un dato de entrada se llegue a los resultados.
    - **Salidas:** Resultados finales o transformaciones que ha sufrido la información de entrada a través del proceso.

Ejemplo: Construir un algoritmo que, dados los dos lados diferentes de un rectángulo, encuentre el perímetro y el área del mismo.

<table>
<tr>
<td>

Análisis del algoritmo: Como no se dice cuáles son los valores de los dos lados del rectángulo, se debe separar espacio en memoria a través de variables para que estos valores sean dados posteriormente, de tal manera que el algoritmo debe proveer el perímetro y el área de cualquier rectángulo. Al ser un rectángulo (se menciona en el enunciado), conociendo los valores de los dos lados diferentes, es posible obtener los resultados pedidos.

Datos de entrada:
- Valor de un lado.
- Valor del otro lado.

Datos de salida:
- El valor de perímetro.
- El valor del área del rectángulo.

Definición de variables a utilizar
- **lado1**: Valor del lado que representa la base.
- **lado2**: Valor del lado que representa la altura.
- **perímetro**: Perímetro del rectángulo.
- **area**: Área del rectángulo.

</td>
<td>

Procesos: Los cálculos necesarios para obtener los resultados partiendo de los datos de entrada, son:
- **perimetro**=suma de los cuatro lados del rectángulo.
- **area**= lado que representa la base * lado que representa la altura.

Algoritmo

Inicio
Lea: lado1, lado2
perimetro=lado1+lado1+lado2+lado2
Área=lado1 * lado2
Escriba: “El Perímetro es:”, perimetro
Escriba: “El Área es:”, area
Fin

</td>
</tr>
</table>

Prueba de escritorio: En la siguiente tabla se detalla el cambio de las variables conforme avanza la secuencia del anterior algoritmo. Para la prueba de escritorio se supondrá que los valores de los lados lado1 y lado2, son 25 y 15 respectivamente.

Secuencia	lado1	lado2	perimetro	area
1	25	15	<BASURA>	<BASURA>
2	25	15	80	<BASURA>
3	25	15	80	375
4	Se escribe en pantalla: “EL PERIMETRO ES:”, 80			
5	Se escribe en pantalla: “EL AREA ES:”, 375			

La **prueba de escritorio** consiste en dar valores a las variables que se han definido y que siguen el flujo del programa elaborado, con el objetivo de comprobar si al final el resultado es el acertado.

Recursos complementarios

[1] Educar Portal (2019, 11 de julio). Microaprendizaje: ¿Qué es un algoritmo? [video] https://youtu.be/2tlOmfoVKe0
[2] Educar Portal (2019, 11 de julio). Microaprendizaje: ¿Qué es la programación y cuáles son sus usos? [video]. https://youtu.be/EHiiNhLRIGc

Actividad

1. ¿Cuáles son las características básicas de los algoritmos? Describir cada una de éstas.
2. Teniendo en cuenta el presente documento, responder: ¿Los algoritmos pueden tener múltiples entradas y múltiples salidas? Argumentar la respuesta con un ejemplo.
3. Para los siguientes ejercicios, describir **el análisis del algoritmo, los datos de entrada, los datos de salida, los procesos, las variables a utilizar, el algoritmo y una prueba de escritorio**.
   a. Construir un algoritmo que brinde como salida: **el área** de un círculo, dado **el diámetro** como entrada.
   b. Construir un algoritmo que brinde como salidas: **la suma, la resta, la multiplicación** y **la división**, de **dos números** ingresados por el usuario.
4. En programación, ¿para qué se utiliza la prueba de escritorio? Argumentar la respuesta con un ejemplo.
5. En programación, ¿qué sucede con el algoritmo cuando la prueba de escritorio es incorrecta o errónea? Argumentar la respuesta.

	INSTITUCIÓN EDUCATIVA JOSÉ EUSEBIO CARO **Tecnología e informática**
Docente	Jesús Eduardo Madroñero Ruales
Propósito del taller	Comprender los métodos de representación, instrucciones, reglas y diagramas básicos para la representación de algoritmos. Aplicar los métodos de representación de algoritmos, en la solución de problemas cotidianos.
Competencias	Analizo el funcionamiento de prototipos conformados por artefactos y procesos como respuesta a necesidades o problemas.

Representación de Algoritmos: Diagramas de Flujo

Los algoritmos deben ser representados usando algún método que les permita ser independizados del lenguaje de programación que se requiera utilizar. Los métodos más usuales son: **diagramas de flujo, diagramas rectangulares y pseudocódigos**.

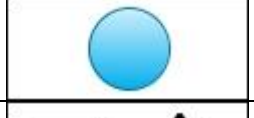
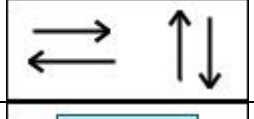
Diagrama: Un diagrama es la representación, mediante gráficos, de cada uno de los pasos que dan solución a un problema determinado. Cada gráfico utilizado representa la acción o mandato que se debe ejecutar dentro del algoritmo.

Diagramas de flujo: Se conoce como diagramas de flujo a aquellos gráficos representativos que se utilizan para esquematizar conceptos vinculados a **la programación**, la economía, los procesos técnicos y/o tecnológicos, la psicología, la educación y casi cualquier temática de análisis. Los símbolos más utilizados en los diagramas de flujo son las flechas (que indica sentido y trayectoria), el rectángulo (representa un evento o proceso), el rombo (una condición), el círculo (un punto de conexión) y otros.

Simbología de los diagramas de flujo

Los Diagramas de flujo se dibujan generalmente usando algunos símbolos estándares; sin embargo, algunos símbolos especiales pueden también ser desarrollados cuando sean requeridos. Algunos símbolos estándares, que se requieren con frecuencia para diagramar programas de computadora se muestran a continuación:

Tabla 1 - Simbología básica de un diagrama de flujo

	**TERMINAL:** Indica comienzo o final de un programa, subprograma o módulo.
	**PROCESO:** Cualquier proceso interno realizado por el ordenador como asignación de valor a variables, operaciones matemáticas, entre otros.
	**CAPTURA Y EMISIÓN DE DATOS:** Entrada de información desde un periférico y hacia el ordenador.
	**DECISIÓN MÚLTIPLE:** El dato o condición planteada que presenta distintas alternativas (o casos), a seguir.
	**CONECTOR**: Indica a través de una referencia (número, letra o texto) en que parte debe continuar un diagrama de flujo que se interrumpe.
	**LÍNEAS DE FLUJO**: Sentido del flujo de procesos. Indican que proceso viene a continuación del otro.
	**DISPLAY:** Para presentar datos o enviarlos a impresora.

Operaciones matemáticas

Los símbolos gráficos son utilizados específicamente para llevar a cabo operaciones aritméticas y relaciones condicionales. La siguiente es una lista de los símbolos más comúnmente utilizados:

Reglas básicas para dibujar diagramas de flujo

1. Los Diagramas de flujo deben escribirse de arriba hacia abajo, y/o de izquierda a derecha.
2. Los símbolos se unen con líneas, las cuales tienen en la punta una flecha que indica la dirección en que fluye la información o los procesos.
3. Se deben utilizar solamente líneas de flujo horizontales o verticales (nunca diagonales).
4. Se debe evitar el cruce de líneas, para lo cual se quisiera separar el flujo del diagrama a un sitio distinto, se pudiera realizar utilizando los conectores. Se debe tener en cuenta que solo se van a utilizar conectores cuando sea estrictamente necesario.
5. No deben quedar líneas de flujo sin conectar.
6. Todo texto escrito dentro de un símbolo debe ser legible, preciso, evitando el uso de muchas palabras.
7. Todos los símbolos pueden tener más de una línea de entrada, a excepción del símbolo final.
8. Solo los símbolos de decisión pueden y deben tener más de una línea de flujo de salida

Operaciones matemáticas y relaciones condicionales básicas

Símbolo	Función
+	Sumar
-	Menos
*	Multiplicación
/	División
=	Equivalente a
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
<>	Diferente de

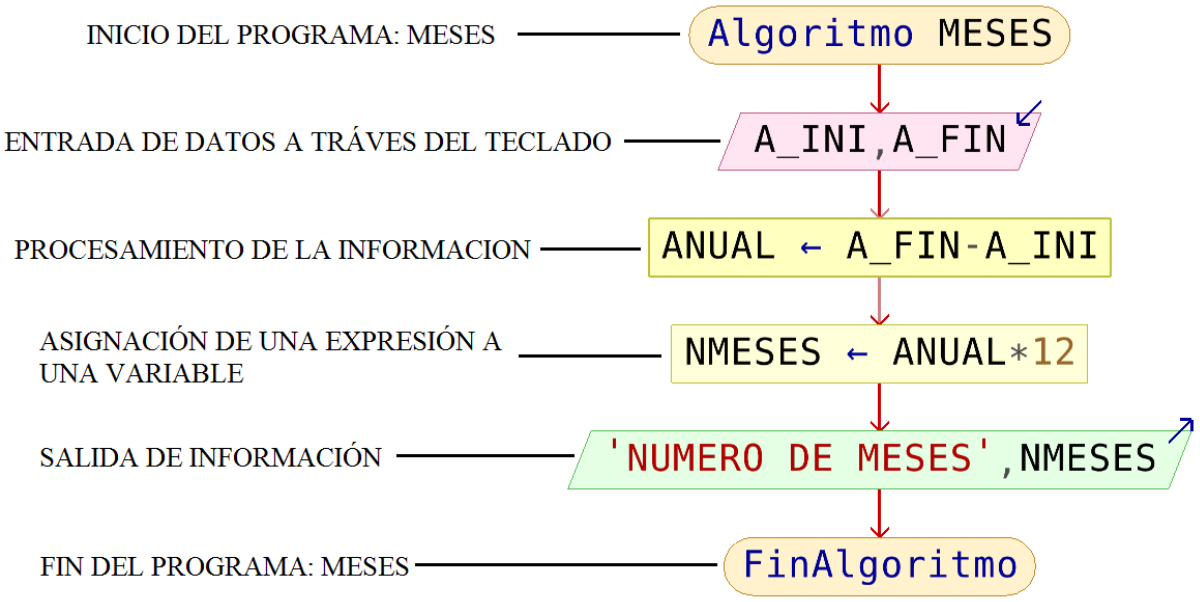
Ejemplos de diagrama de flujo:

1. Construir un algoritmo en el cual se presente cuántos meses han transcurrido entre dos años cualesquiera dados como entradas.

   Solución: Para la solución de este problema, se plantea un algoritmo que lleva por nombre **MESES**, y en el cual se hace uso de las siguientes variables:

   - **ENTRADAS**: Fecha inicial (**A_INI**), fecha final (**A_FIN**)
   - **VARIABLES PARA PROCEDIMIENTOS**: ANUAL, NMESES.
   - **SALIDAS**: Número de meses entre dos años (**NMESES**).

   El algoritmo se presenta y explica en el siguiente diagrama de flujo (**MESES**):

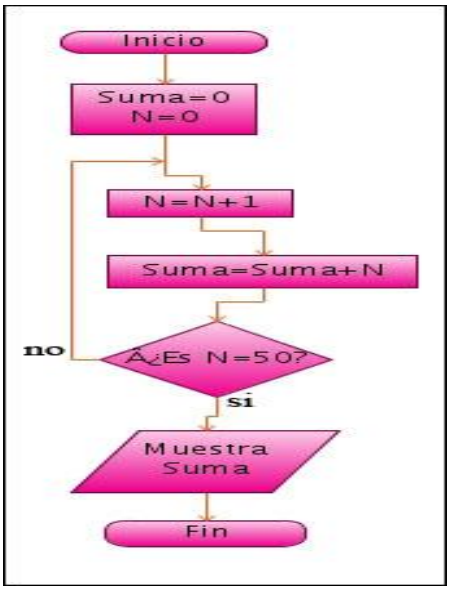


2. Construir un diagrama de flujo que encuentre la suma de los primeros 50 números naturales:

   Solución: Para la solución de este problema, se plantea un algoritmo que lleva por nombre **SUMANATURAL**, y en el cual se hace uso de las siguientes variables:

   - **ENTRADAS**: Debido a la naturaleza del problema, este algoritmo no presenta entradas ingresadas por periféricos.
   - **VARIABLES PARA PROCEDIMIENTOS**: Suma, N.
   - **SALIDAS**: Sumatoria de números naturales (**Suma**).

   El algoritmo se presenta y explica en el diagrama de flujo (**SUMANATURAL**) a la derecha:

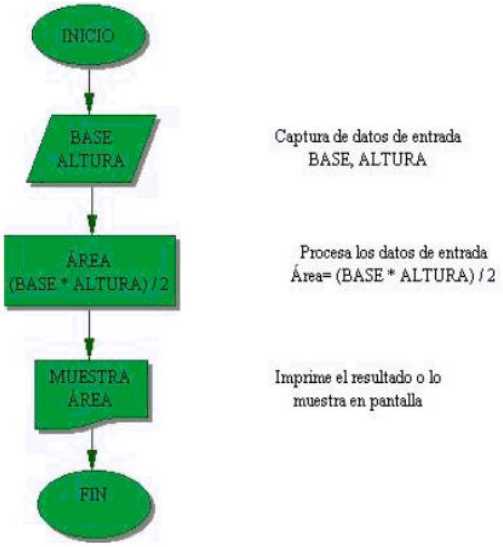


3. Diagrama de flujo que encuentra el área de un triángulo, dadas como entradas: la base y la altura del triángulo:

   Solución: Para la solución de este problema, se plantea un algoritmo que lleva por nombre **TRIÁNGULO**, y en el cual se hace uso de las siguientes variables:

   - **ENTRADAS**: Base del triángulo (**BASE**), Altura del triángulo (**ALTURA**)
   - **VARIABLES PARA PROCEDIMIENTOS**: Área del triángulo (**ÁREA**).
   - **SALIDAS**: Área del triángulo (**ÁREA**).

   El algoritmo se presenta y explica en el siguiente diagrama de flujo (**TRIÁNGULO**):



Recursos complementarios

[1] Absolute (2020, 22 de marzo). Diagramas de flujo en 2 minutos [video] https://youtu.be/u6fusP6JLgg
[2] Pasos por ingeniería (2016, 20 de julio). Diagramas de Flujo Explicación (simbología y construcción) [video]. https://youtu.be/qDttSc3RQBc
[3] Escuela Ingeniería Robótica (2016, 07 de agosto). Programación básica #4 – Diagramas de Flujo [video]. https://youtu.be/2ZjWQdSX5Go

Actividad conceptual

1. ¿Qué son los Diagramas de Flujo? ¿Para qué se utilizan?
2. Describir la simbología y reglas básicas para los diagramas de flujo.
3. Diseñar un algoritmo en diagrama de flujo que calcule la suma, la resta, la multiplicación y la división, de dos números ingresados por el usuario. Tenga en cuenta las siguientes variables para la solución:
   Datos de entrada: numero1, numero2.
   Datos de salida: suma, reste, producto, division.
4. Diseñar un algoritmo en diagrama de flujo que realice el cálculo del área de un círculo, a partir de su diámetro, que debe ser ingresado por el usuario. Tenga en cuenta las siguientes variables para la solución:
   Dato de entrada: diametro (d).
   Dato de salida: area (a).
5. Elaborar y resolver tres ejemplos de situaciones problema que requieran algoritmos para su solución. Especificar las entradas, las salidas y los respectivos procesos.

	INSTITUCIÓN EDUCATIVA JOSÉ EUSEBIO CARO **Tecnología e informática**
Docente	Jesús Eduardo Madroñero Ruales
Propósito del taller	Comprender los métodos de representación, instrucciones, reglas y diagramas básicos para la representación de algoritmos. Aplicar los métodos de representación de algoritmos, en la solución de problemas cotidianos.
Competencias	Analizo el funcionamiento de prototipos conformados por artefactos y procesos como respuesta a necesidades o problemas.

Representación de algoritmos: Diagrama Rectangular y Pseudocódigo

Los algoritmos deben ser representados usando algún método que les permita ser independizados del lenguaje de programación que se requiera utilizar. Los métodos más usuales son: diagramas de flujo (tema anterior), diagramas rectangulares y pseudocódigos.

El diagrama rectangular: No utiliza flechas de conexión, sino que se construye, fundamentalmente, con tres figuras que representan las estructuras de control de la programación estructurada.

Inicio
Escriba:" digite año inicial"
Lea: inicio
Escriba:" digite año final"
Lea: final
resultado = final –inicio
meses = resultado * 12
Escriba: "el número de meses que han pasado es:", **meses**
Fin

Un diagrama rectangular empieza con un rectángulo vacío que se va llenando de arriba hacia abajo, en la medida en que es necesario representar un determinado paso. No tiene figuras especiales para demarcar los medios de entrada y de salida, pero dentro de la figura escogida para representar la instrucción se escribe la acción a ejecutar. Los bloques de inicio y fin del algoritmo pueden omitirse ya que se asume que el orden de ejecución es siempre de arriba hacia abajo, sin embargo, se recomienda no hacerlo.

El pseudocódigo: El pseudocódigo es la representación de los pasos del algoritmo a través de palabras, utilizando una nomenclatura estandarizada para denotar el significado de cada paso.

Dentro de éste es permitido la sangría con el fin de que se visualice, en forma sencilla, el grupo de instrucciones pertenecientes a una determinada acción. Esta herramienta resulta muy útil para el seguimiento de la lógica de un algoritmo y, sobre todo, facilita la transcripción a un lenguaje de programación. Tiene la desventaja de que el programador trata de escribir los pasos del algoritmo utilizando palabras reservadas. Es necesario, entonces, al igual que en el diagrama de flujo, establecer unos parámetros o formas de expresar cada instrucción.

Inicio
    Escriba:" digite año inicial"
    Lea: inicio
    Escriba:" digite año final"
    Lea: final
    resultado = final – inicio
    meses = resultado *12
    Escriba: "el número de meses que han pasado es:", **meses**
Fin

Nótese cómo el grupo de instrucciones entre las palabras inicio y fin se han rodado un poco hacia la derecha (sangría), lo cual permite visualizar en una forma sencilla el grupo de instrucciones pertenecientes a ese bloque. El otro esquema ya visto anteriormente (ver guía pasada), es el diagrama de flujo. En la siguiente figura se detalla el diagrama de flujo para el ejemplo:

Figura 1 - Ejemplo de Diagrama de Flujo

Consejos a tener en cuenta:

- Trate de entender y hacer el mejor análisis posible del problema a solucionar.
- A todos los campos utilizados en un programa se les asigna una única dirección de memoria.
- Los campos variables (variables) se les asignan valores en el momento de ejecución, bien sea a través de una instrucción de entrada de datos o una instrucción de asignación.
- Cuando se le asigna una variable; lo que había antes en esa determinada dirección es destruido, no se puede recuperar.
- No todos los valores de partida al hacer el algoritmo son datos de entrada, sólo lo son aquellos valores que desconocemos en el momento, pero que serán suministrados en la fase de ejecución.
- Al extraer de memoria uno o más valores hacia un medio externo, éstos permanecen; no se borran de la memoria de la computadora.
- Usted aprende a construir algoritmos haciendo algoritmos. Trate de hacer el máximo número de algoritmos propuestos. Busque, si es necesario, enunciados en la red o en otros textos.

Recursos complementarios

1] DiscoDurodeRoer (2015, 27 de abril). Ejercicios PseInt – Básicos #1 – Empezando por lo básico [video] https://youtu.be/DHIi4dcaMEc
[2] JuanRa Garcia Montes (2020, 26 de marzo). Programación: Diagramas de flujo y pseudocódigo [video]. https://youtu.be/Lub5qOmY4JQ
[3] Eliezer De León (2016, 21 de septiembre). [PSEINT] Diagrama de flujo y Pseudocódigo que determina tu edad [video]. https://youtu.be/ocUWlylurUo
[4] GabyB (2018, 23 de mayo). Tutorial – Como hacer una Diagrama de flujo en PseInt [video]. https://youtu.be/RTWzhECFNyc

Actividad conceptual

Nota: Los siguientes algoritmos, deben ser resueltos de las tres formas de representación detalladas: Diagrama de flujo y pseudocódigo.

1. Realizar un cuadro comparativo entre las tres formas de representación de algoritmos: los diagramas de flujo, los diagramas rectangulares y los pseudocódigos.
2. ¿Cuál de las tres herramientas para elaborar algoritmos le parece más eficaz? ¿Por qué?
3. Elaborar un algoritmo que convierta un dato de entrada, de centímetros a pulgadas. Recuerde que una pulgada es igual a 2,54 centímetros. ¿Es posible resolver este problema en Excel? Argumentar su respuesta.
4. Elaborar un algoritmo que permita convertir grados centígrados en:
    a. Grados Fahrenheit.
    b. Grados Kelvin.
5. Elaborar un algoritmo que calcule el promedio de cinco notas ingresadas por un usuario. ¿Es posible resolver este problema en Excel? Argumentar su respuesta.

	INSTITUCIÓN EDUCATIVA JOSÉ EUSEBIO CARO **Tecnología e Informática**
Docentes	Jesús Eduardo Madroñero Ruales
Propósito del taller	Comprensión de los aspectos básicos de la plataforma Makecode. Desarrollo de algoritmos en la plataforma Makecode para solucionar situaciones problema.
Competencias	Utilizo adecuadamente herramientas informáticas de uso común para la búsqueda y procesamiento de la información y la comunicación de ideas.

Retos en Makecode

Los ejercicios pueden ser desarrollados en la plataforma Makecode (https://makecode.microbit.org/).

Reto 1: Sensor de temperatura

El sensor de temperatura integrado en la tarjeta micro:bit, detecta la temperatura ambiente en grados centígrados.

1. Diseñar un programa en la tarjeta micro:bit, que muestre constantemente en el panel LED la temperatura detectada por el sensor, en grados centígrados. Al presionar el botón A debe mostrar la temperatura en grados fahrenheit. Al presionar el botón B debe mostrar la temperatura en grados kelvin.

Reto 2: Indicador placas de hielo

En algunas regiones, la mayoría de coches incorporan en el salpicadero, junto a la pantalla que marca la temperatura exterior, un testigo con forma de copo de nieve, que avisa de la posibilidad de que haya placas de hielo en la carretera. Este testigo se suele iluminar cuando la temperatura disminuye por debajo de 3 grados centígrados.

2. Diseñar un programa, que muestre en el panel LED de la micro:bit un icono con forma de copo de nieve cuando la temperatura esté por debajo de 3 grados, para el caso contrario debe mostrar un ícono que indique la temperatura está normal.

Reto 3: Temperatura de una nevera

La temperatura óptima de un frigorífico es de 7ºC, mientras que la temperatura de un congelador debe estar entorno a los -18ºC. Algunas neveras incorporan un avisador acústico que se activa cuando la temperatura no es la óptima.

3. Mostrar en el panel LED de la tarjeta micro:bit, la temperatura del frigorífico. Se debe activar una alarma cuando la temperatura esté por encima de 7ºC. Mientras la temperatura este en el rango normal, se debe mostrar un ícono de detalle funcionamiento adecuado en la nevera.

Reto 4: ¿Qué botón has pulsado?

Hay dos botones en la cara frontal de la micro:bit (etiquetados como A y B). Puedes detectar cuando son pulsados de forma independiente o a la vez y ejecutar una acción en cada caso.

4. Crear un programa que muestre en pantalla la letra del pulsador que se ha accionado, además de cuando se presionan A y B de manera conjunta. Se debe agregar un sonido para cada evento.

Reto 5: Alarma de exposición al sol

En la actualidad hay estudios que demuestran la relación entre la exposición al sol y el riesgo de padecer cáncer de piel. Los dermatólogos recomiendan evitar la exposición al sol durante las horas de máxima radiación y el uso de cremas protectoras.

5. Al iniciar el programa se indica que hay que pulsar el botón A para que muestre el valor de la intensidad del sol. Al pulsar el botón se mostrará una carita feliz si el valor es inferior a 175 ua (unidades arbitrarias) y una carita triste si es superior, además de un sonido. Al pasar 3 segundos la tarjeta debe cambiar a modo de ahorro de energía, quedando a la espera de que se vuelva a pulsar el botón A.

	INSTITUCIÓN EDUCATIVA JOSÉ EUSEBIO CARO **Tecnología e Informática**
Docentes	Jesús Eduardo Madroñero Ruales
Propósito del taller	Entender el funcionamiento de las instrucciones básicas de un algoritmo: Leer, Asignar y Escribir. Desarrollar algoritmos haciendo uso de instrucciones básicas.
Competencias	Utilizo adecuadamente herramientas informáticas de uso común para la búsqueda y procesamiento de la información y la comunicación de ideas.

Instrucciones básicas en un Algoritmo

Leer: Esta instrucción de entrada o lectura de datos, viene a ser aquella mediante la cual se ingresa uno o más datos por medio del teclado, para luego almacenarse internamente en una variable. Este valor se requerirá para realizar cálculos y hallar la solución a un algoritmo. Esta instrucción permite:

- Solicitar un dato o datos iniciales.
- Requerir un dato o datos de entrada.

Su declaración dentro de pseudocódigo es: | **Leer** <dato1, dato2, dato3, …. >;

Asignar: Esta instrucción asigna un valor a una variable. A diferencia de la instrucción **leer,** esta no se ingresa por teclado sino mediante una instrucción en el mismo Pseudocódigo. Esta instrucción permite:

- Operar sobre el dato obtenido.
- Procesar los datos, obteniendo nuevos valores.

Su empleo dentro de un pseudocódigo es, por ejemplo, si vamos a asignar un valor 5 a la variable **Numero**: | **numero** ← 5;

Escribir: Esta instrucción de salida, se emplea para poder visualizar resultados o valores que contiene una variable, también para mostrar mensajes, por lo general el resultado se aprecia en la pantalla de la computadora o impreso en papel. Esta instrucción permite:

- Mostrar el resultado.
- Imprimir el valor resultante.

Su empleo dentro de un pseudocódigo es: | **Escribir** <Valor Resultante>

Ejemplo 1: Elaborar un algoritmo que calcule y presente el promedio de un estudiante, a partir de sus tres notas parciales.

Solución:

Variables de entrada: nota1, nota2, nota3.
Procesos: Calculo del promedio.
Salidas: Valor del promedio.

Pseudocódigo:

```
Proceso CalcularPromedioEstudiante
   Definir nota1, nota2, nota3, promedio Como Real;
   // Solicitar al usuario las tres notas parciales
   Escribir "Ingrese la primera nota:";
   Leer nota1;
   Escribir "Ingrese la segunda nota:";
   Leer nota2;
   Escribir "Ingrese la tercera nota:";
   Leer nota3;
   // Calcular el promedio
   promedio <- (nota1 + nota2 + nota3) / 3;
   // Mostrar el resultado
   Escribir "El promedio del estudiante es: ", promedio;
FinProceso
```

Ejemplo 2: Elaborar un algoritmo que permita ingresar el número de partidos ganados, empatados y perdidos, para un equipo de futbol. Se debe presentar el puntaje tota, teniendo en cuenta que por cada partido ganado se obtiene 3 puntos, por cada partido empatado se obtiene 1 punto, y cada partido perdido 0 puntos.

Solución:

Variables de entrada: ganados, empatados, perdidos.
Procesos: Calculo de los partidos ganados, empatados, perdidos.
Calculo del puntaje final
Salidas: Valor del puntaje.

Pseudocódigo:

```
Proceso CalcularPuntajeFutbol
   Definir ganados, empatados, perdidos, puntaje Como Entero;
   // Solicitar al usuario los datos
   Escribir "Ingrese el número de partidos ganados:";
   Leer ganados;
   Escribir "Ingrese el número de partidos empatados:";
   Leer empatados;
   Escribir "Ingrese el número de partidos perdidos:";
   Leer perdidos;
   // Calcular el puntaje total
   puntaje ← (ganados * 3) + (empatados * 1) + (perdidos * 0);
   // Mostrar el resultado
   Escribir "El puntaje total del equipo es: ", puntaje;

FinProceso
```

Actividad

1. En su cuaderno realizar un resumen de lo descrito en el presente documento.
2. Relacionar la instrucción con el tipo de característica correcta:

Instrucción	Característica
1. Escribir	a. Entrada
2. Leer	b. Salida
3. Asignar	c. Proceso

Nota: Para los siguientes algoritmos se debe describir los siguientes ítems: **Variables, procesos, salidas y prueba de escritorio.**

3. Elaborar un algoritmo que solicite el número de respuestas correctas, incorrectas y en blanco, correspondientes a un test de preguntas, y presente su puntaje final considerando lo siguiente:
   - Por cada respuesta correcta tendrá 2 puntos.
   - Por cada respuesta incorrecta tendrá -1 puntos.
   - Por cada respuesta en blanco tendrá 0 puntos.

4. Desarrollar un algoritmo que, dado como datos de entrada: la base y la altura de un rectángulo, calcule y presente el perímetro y la superficie de este.
   Formulas:
   Perímetro = 2 * (base + altura)
   Superficie = base * altura

5. Desarrollar un algoritmo que, dado como datos de entrada: el radio y la altura de un cilindro, calcule y presente el área y su volumen.
   Fórmulas:
   Área = 2 * pi * radio (radio + altura)
   Volumen = pi * $(radio)^2$ * altura

6. Desarrollar un algoritmo que resuelva un problema que tiene una gasolinera. Los dispensadores de esta, registran lo que surten en galones, pero el precio de la gasolina está fijado en litros. El algoritmo debe calcular y presentar lo que se debe cobrarle al cliente.
   Fórmulas:
   El precio promedio del galón en Colombia (en mayo de 2025) cuesta $ 15.827 pesos.
   Litros = galones x 3,78.