



# Uleam

UNIVERSIDAD LAICA  
ELOY ALFARO DE MANABÍ

Guía de  
estudio

Fundamentos de Tecnologías de  
la Información

Dirección de Bienestar, Admisión y Nivelación Universitaria

2024

UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ



## GUÍA DE ESTUDIO

### Fundamentos de tecnología de la información

Lic. Victor Geovanny Zambrano Cedeño

Ing. Junior José Zamora Mendoza

Ing. Óscar Iván Briones Maldonado

Ing. César Stalin Villavicencio Palacios

Ing. José Daniel Veloz Salcedo

Ing. Jharol Antonio Ormaza Sabando

Ing. Carmen Vanessa Moreira Loor

**Universidad Laica Eloy Alfaro de Manabí**  
Ciudadela universitaria vía circunvalación (Manta)  
www.ulead.edu.ec

**Dr. Marcos Zambrano Zambrano, PhD.**  
Rector

**Dr. Pedro Quijije Anchundia, PhD.**  
Vicerrector Académico

**Dra. Jackeline Terranova Ruiz, PhD.**  
Vicerrectora de Investigación, Vinculación y Postgrado

**Lic. Víctor Geovanny Zambrano Cedeño, Mg**  
Dirección de Bienestar, Admisión y Nivelación Universitaria

**Guía de estudio**  
**Fundamentos de tecnología de la información**

Lic. Víctor Geovanny Zambrano Cedeño  
Ing. Junior José Zamora Mendoza  
Ing. Óscar Iván Briones Maldonado  
Ing. César Stalin Villavicencio Palacios  
Ing. José Daniel Veloz Salcedo  
Ing. Jharol Antonio Ormazá Sabando  
Ing. Carmen Vanessa Moreira Loo

ISBN: 978-9942-681-17-1

Edición: Primera. Diciembre de 2024. Publicación digital

Prohibida su venta

Trabajo de edición y revisión de texto: Mg. Alexis Cuzme Espinales  
Diseño de portada: Mg. José Márquez Rodríguez

Una producción de la Universidad Laica Eloy Alfaro de Manabí, registrada en la Cámara Ecuatoriana del Libro.

Sitio Web: [ulead.edu.ec](http://ulead.edu.ec)  
Teléfonos: 2 623 026 Ext. 255

## ÍNDICE

INTRODUCCIÓN .....	6
RESULTADOS DE APRENDIZAJE.....	7
UNIDAD 1 .....	8
<b>1 INTRODUCCIÓN A LAS TECNOLOGÍAS DE LA INFORMACIÓN .....</b>	<b>8</b>
1.1 SISTEMAS DE INFORMACIÓN .....	8
1.1.1 Teoría General de los Sistemas .....	8
1.1.2 Sistema de Información .....	9
1.1.3 Tipos de sistemas de información en las organizaciones.....	11
1.1.4 Datos e Información como elementos del sistema.....	12
1.2 RECURSOS TECNOLÓGICOS Y TIC .....	13
1.2.1 Definiciones.....	13
1.2.2 Hardware y Software.....	13
1.2.3 Funciones de los dispositivos de hardware .....	14
1.2.4 Software de Sistema Operativo.....	15
1.2.5 Interfaces.....	16
1.2.6 Redes Informáticas.....	16
1.2.7 Topología de redes.....	18
<b>UNIDAD 2 .....</b>	<b>20</b>
<b>2 ALGORITMOS Y DIAGRAMAS DE FLUJO .....</b>	<b>20</b>
2.1 ALGORITMOS .....	20
2.1.1 Definición y características de un algoritmo .....	20
2.1.2 Clasificación y ejemplos .....	20
2.1.3 Estructura de un algoritmo .....	21
2.2 ALGORITMOS COMPUTABLES.....	22
2.2.1 Datos y tipos de datos.....	22
2.2.2 Operadores aritméticos y expresiones.....	23
2.2.3 Modelado computable para algoritmos cuantitativos .....	25
2.3 FORMAS DE REPRESENTAR UN ALGORITMO.....	26
2.3.1 Símbolos y reglas para elaborar un diagrama de flujo .....	27
2.3.2 Detalle de un diagrama de flujo básico con su prueba de escritorio.....	29
<b>UNIDAD 3 .....</b>	<b>32</b>
<b>3 DECISIONES EN ALGORITMOS COMPUTABLES .....</b>	<b>32</b>
3.1 DECISIONES.....	32
3.1.1 Definición .....	32
3.1.2 Operadores relacionales o de comparación.....	32
3.1.3 Estructura selectiva SI ... ENTONCES ... SINO (If ... Then ... Else) .....	33
3.2 BLOQUES DE DECISIÓN .....	36
3.2.1 Decisiones secuenciales.....	36

3.2.2 Decisiones anidadas .....	38
3.2.3 Operadores lógicos .....	40
<b>UNIDAD 4 .....</b>	<b>43</b>
<b>4 CONTADORES, ACUMULADORES Y ESTRUCTURAS REPETITIVAS .....</b>	<b>43</b>
4.1 CONTADORES Y ACUMULADORES.....	43
4.1.1 Definición .....	43
4.2 ESTRUCTURAS REPETITIVAS .....	44
4.2.1 Definición .....	44
4.2.2 Tipos de estructuras repetitivas .....	45
4.2.3 Usando la estructura repetitiva FOR .....	45
<b>UNIDAD 5 .....</b>	<b>52</b>
<b>5 COMPOSICIÓN Y COMPROBACIÓN DE ALGORITMOS .....</b>	<b>52</b>
5.1 USO DEL ENTORNO INTEGRADO DE DESARROLLO .....	52
5.1.1 Generalidades .....	52
5.1.2 Variables y tipos de datos.....	52
5.1.3 Opciones especiales para la comprobación de un algoritmo .....	54
5.2 ANÁLISIS ORIENTADO A MODELAR UN PROBLEMA COMPUTABLE.....	55
5.2.1 Recomendaciones generales para analizar y resolver un problema .....	55
5.2.2 Ejemplo de análisis y propuesta de solución a problemas que involucran decisiones .....	56
<b>6 Bibliografía.....</b>	<b>59</b>

## INTRODUCCIÓN

Los contenidos de la presente guía de estudios han sido preparados en base al requerimiento realizado por la Facultad de Ciencias de la Vida y Tecnologías, por lo que el objetivo más importante de esta asignatura es prepararlo(a) para superar los desafíos que encontrará en el primer semestre de esta apasionante carrera que usted ha escogido.

Esperamos que este instrumento de apoyo constituya un valioso aporte en el inicio de su formación profesional y por este motivo se lo ha “condimentado” con mucha dedicación y cariño para usted.

¡Éxitos!

*«Dejadme practicar las buenas costumbres y les devolveré libertad y gloria».*

***Eloy Alfaro Delgado***



## RESULTADOS DE APRENDIZAJE

Unidad 1 Comprende los principios básicos que permiten identificar y clasificar los diferentes elementos que proponen las Tecnologías de la Información.



Unidad 2 Desarrolla el pensamiento lógico computacional aplicando los principios de la algoritmia y su representación mediante diagramas de flujo para resolver problemas computables.



Unidad 3 Desarrolla el pensamiento lógico computacional aplicando los principios de la algoritmia y su representación mediante diagramas de flujo para resolver problemas que requieren emplear decisiones y bloques de decisiones para proponer una solución computable.



Unidad 4 Desarrolla el pensamiento lógico computacional aplicando los principios de la algoritmia y su representación mediante diagramas de flujo a problemas que requieren emplear contadores, acumuladores y estructuras repetitivas para proponer una solución computable.



Unidad 5 Fortalece el pensamiento lógico computacional aplicando los principios de la algoritmia mediante el uso de la herramienta digital DFD para representar y comprobar cualquier propuesta de solución a problemas computables que involucren secuencias, decisiones y ciclos.



Resultados  
de las  
Unidades

## UNIDAD 1

### 1 INTRODUCCIÓN A LAS TECNOLOGÍAS DE LA INFORMACIÓN

#### 1.1 SISTEMAS DE INFORMACIÓN

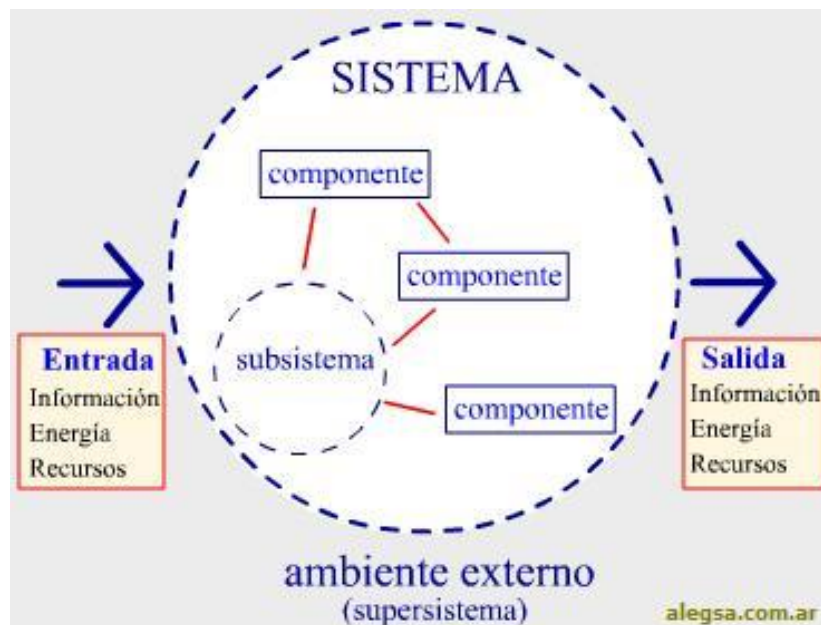
##### 1.1.1 Teoría General de los Sistemas

Según Beynon-Davies (2018) un sistema es un conjunto de componentes organizados que interactúan entre sí para lograr un objetivo común.

La interrelación entre los componentes de un sistema es muy variada y no siempre es sencillo definirla, sin embargo, cada componente debe cumplir una función básica de **Entrada, Proceso y Salida** para que el sistema logre sus objetivos.

Para comprenderlo mejor, veamos a continuación el siguiente vídeo de corta duración:

Fuente: Psicología Animada



*Ilustración 1. Funciones genéricas de los componentes de un sistema*



(<https://sites.google.com/site/teoriageneraldesistemasuco/2-sistemas-y-subsistemas/subsistemas/>)

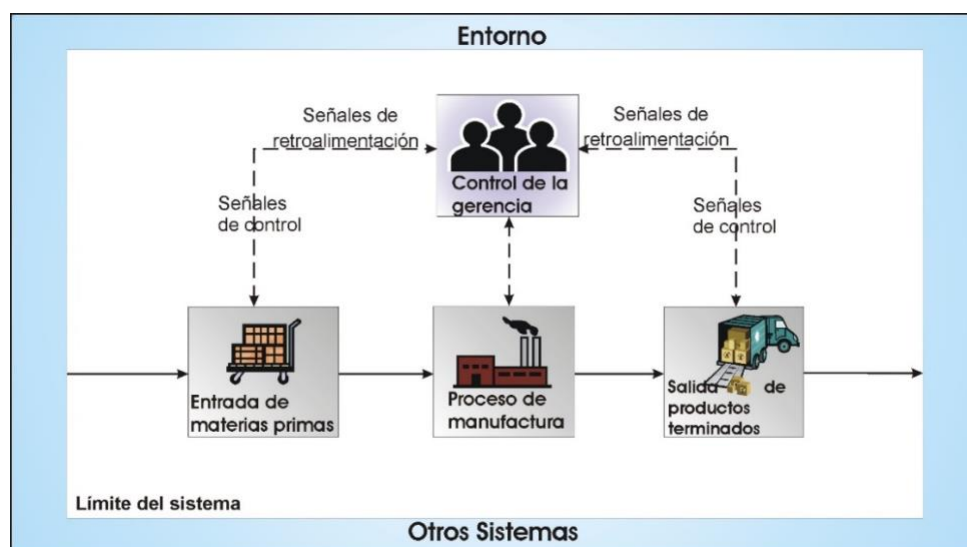


Ilustración 2. Ejemplo de las funciones realizadas por un sistema de fabricación (<https://walfaera.wordpress.com/2008/03/10/fundamentos-de-los-sistemas-de-informacion/>)

Para una explicación más detallada del enfoque sistémico, observemos el siguiente vídeo de corta duración:

Fuente: Bruno Jerardino Wiesenborn



### Actividad 1

Seleccione un sistema o subsistema que usted identifique y complete la siguiente tabla:

Sistema seleccionado:	(Nombre descriptivo de referencia común)		
Funciones:	Entrada	Proceso	Salida
Nombre del Componente:			
Detalle de la forma en que cumple su función:			

## 1.1.2 Sistema de Información

Es un conjunto organizado de componentes que se interrelacionan para manipular de manera estructurada datos e información relevante permitiendo establecer un flujo automatizado de procedimientos y controles que agregan valor para una organización (Publicaciones Empresariales UNAM, 2016).

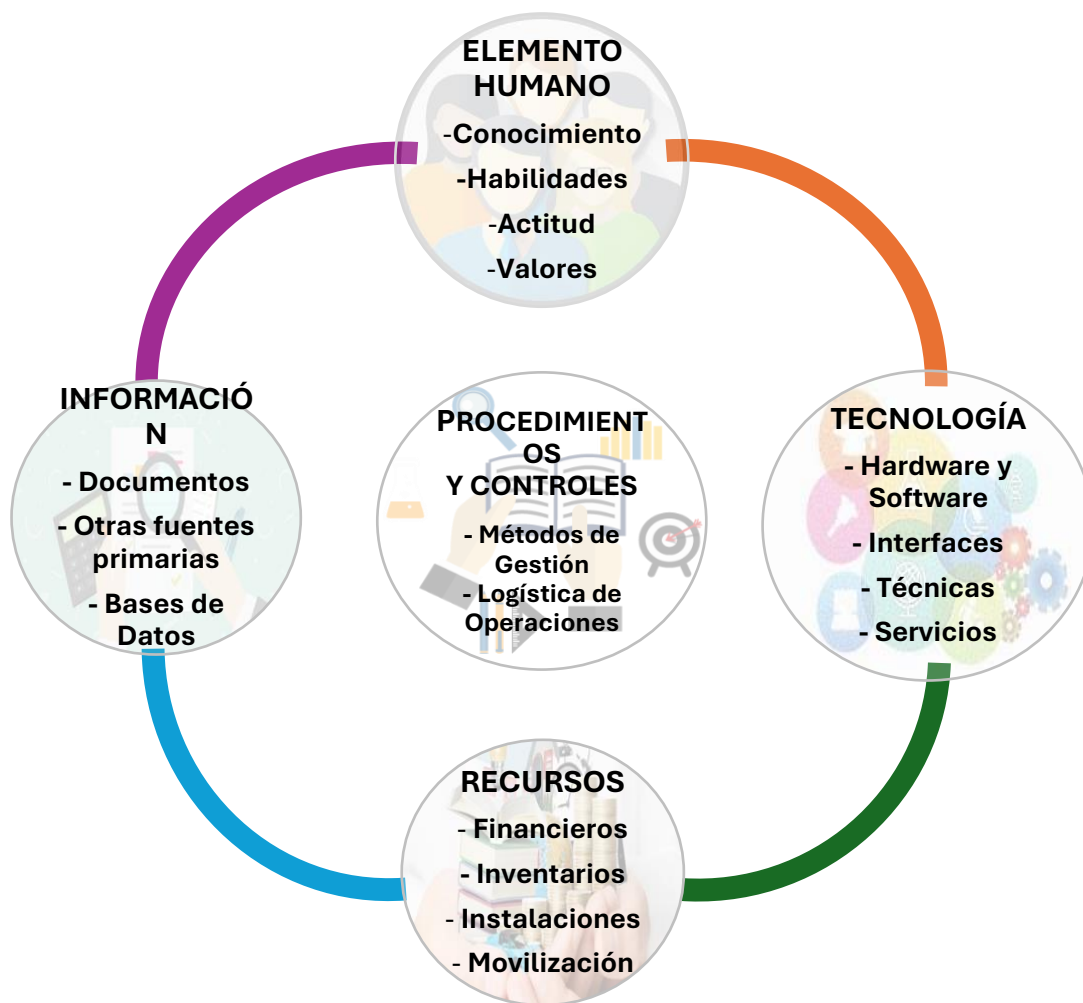




Ilustración 3. Componentes de un sistema de Información.

Es importante destacar que todo sistema de información surge de la necesidad de obtener información precisa y generar conocimiento para cualquier organización. De esta manera, se elimina la incertidumbre permitiendo que la toma de decisiones se apoye en indicadores confiables.

	Temática del vídeo	Fuentes
	“Sistema de información”	UCAM Universidad Católica de Murcia
	“Ejemplos de sistemas de información”	Intor Huaripata

### 1.1.3 Tipos de sistemas de información en las organizaciones

Tabla 1. Tipos de Sistemas de Información


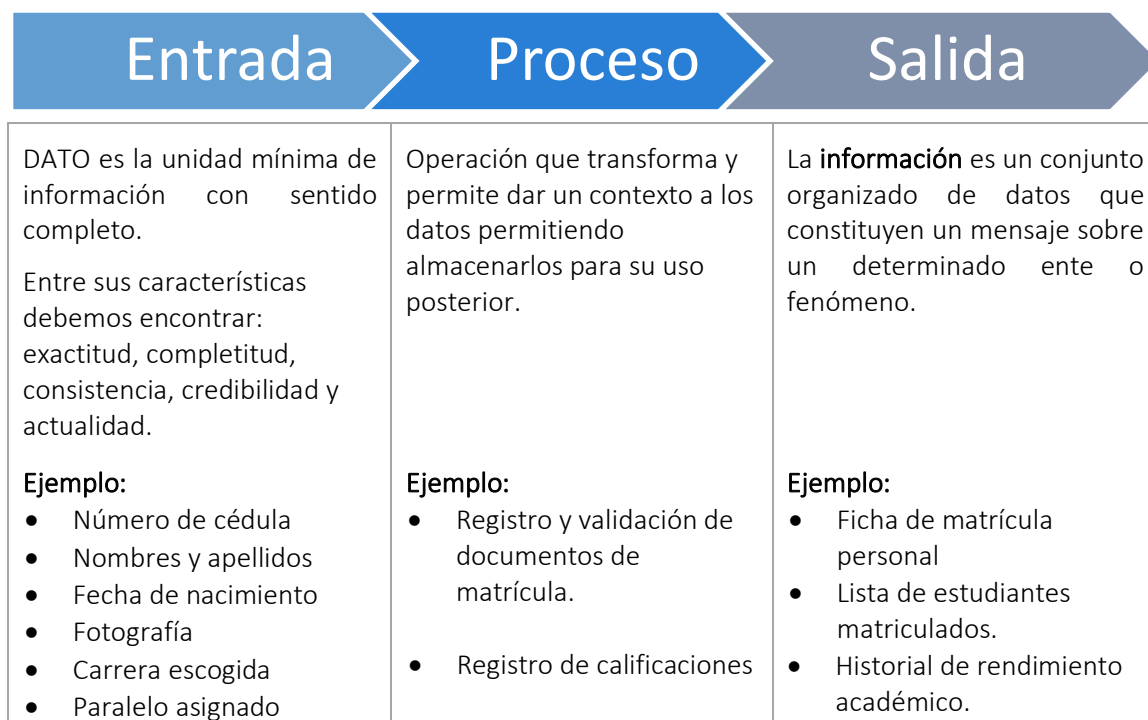
 <p>Sistemas transaccionales</p>	 <p>Sistemas de soporte para la toma de decisiones</p>	 <p>Sistemas estratégicos</p>
<ul style="list-style-type: none"> <li>✓ Son recolectores de grandes volúmenes de datos.</li> <li>✓ Permiten un ahorro significativo en mano de obra, porque permiten automatizar tareas operativas de la organización.</li> <li>✓ Su función principal es procesar transacciones como pagos, cobros, pólizas, etc.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Producen información para apoyar la toma de decisiones.</li> <li>✓ Están basados en recursos de computación, donde combinan modelos y datos para resolver problemas no estructurados.</li> <li>✓ Utilizan interfaces amigables para el usuario, por lo que suelen tener altos costos.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Ayudan a la organización a obtener ventajas competitivas.</li> <li>✓ Apoyan la innovación de productos y procesos dentro de la organización.</li> <li>✓ Monitorean de manera permanente el mercado advirtiendo cambios que pudiesen afectar a la organización.</li> </ul>



Ilustración 4. Relación entre los tipos de sistemas de información y los niveles de una organización (Mazón, Pan y Tinoco, 2018, p.248)

### 1.1.4 Datos e Información como elementos del sistema

El sistema de información se utiliza de forma habitual en lo académico y en las organizaciones. El dato es su materia prima que, luego de ser obtenido, será procesado y almacenado para ser transformado en información.



“El conocimiento es información combinada con experiencia, contexto, interpretación y reflexión” (Davenport, 1998, como se citó en Publicaciones Empresariales UNAM, 2016).

El siguiente vídeo de corta duración nos expone otros ejemplos:



**Fuente:** Claudio J. Varela A.



#### Actividad 2

Seleccione un sistema o subsistema y complete la siguiente tabla desde la perspectiva de “sistema de información”:

Sistema seleccionado:	(Descripción de referencia común)	
Datos de Entrada	Proceso	Salidas

## 1.2 RECURSOS TECNOLÓGICOS Y TIC

### 1.2.1 Definiciones

Podemos definir a las Tecnologías de la Información y Comunicación (TIC) como un conjunto de elementos compuesto por herramientas, métodos y técnicas que son utilizados para el tratamiento, procesamiento, almacenamiento y transmisión de datos con la finalidad de estructurarlos en información útil para generar conocimiento y solucionar problemas (Publicaciones Empresariales UNAM, 2016).

Por otra parte, los recursos tecnológicos son un conjunto de dispositivos, interfaces o aplicaciones de software que pueden integrarse de manera tangible o intangible para permitir, por ejemplo:

- Obtener una funcionalidad determinada para un dispositivo físico como un PC o un TV inteligente.
- Desplegar un servicio automático como las transacciones en línea, la biblioteca virtual, etc.
- Interactuar de manera automática o semiautomática con otros elementos de un sistema (maquinaria, procesos organizacionales, gestión de recursos, etc.).

Podemos resumir que **los recursos tecnológicos son elementos de hardware y de software** que permiten la gestión de la información por parte del ser humano.

### 1.2.2 Hardware y Software

Cualquier dispositivo electrónico o electromecánico, interno o externo, que sea tangible, será reconocido como hardware.

Al conjunto de instrucciones reunidas en lo que se conoce como “programa” o “aplicación”, se reconocerá como software. Es intangible.

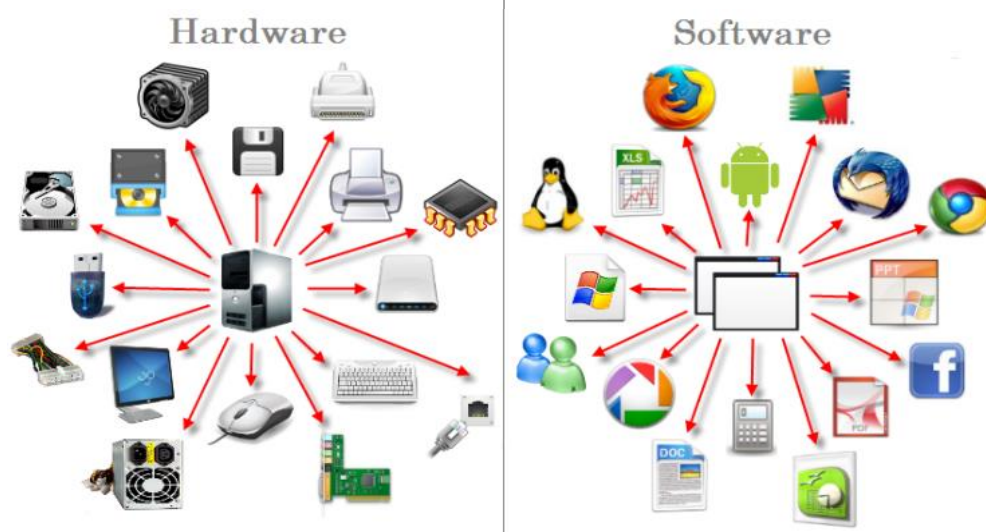





Ilustración 5. Hardware y software (<https://juanhernandez6.wordpress.com/2015/11/17/hardware-y-software/>)

El software provee las instrucciones que el hardware necesita para realizar las funciones para las que ha sido diseñado y fabricado. En este tipo especial de software encontraremos al firmware, controladores de dispositivo (también conocidos como “drivers”) y al sistema operativo.

Otro tipo de software cumple actividades más específicas y cercanas a las necesidades del usuario final; por lo que se denomina software de aplicaciones. En este grupo encontramos a las aplicaciones ofimáticas como el procesador de textos, la hoja electrónica, el presentador de diapositivas, entre otros.



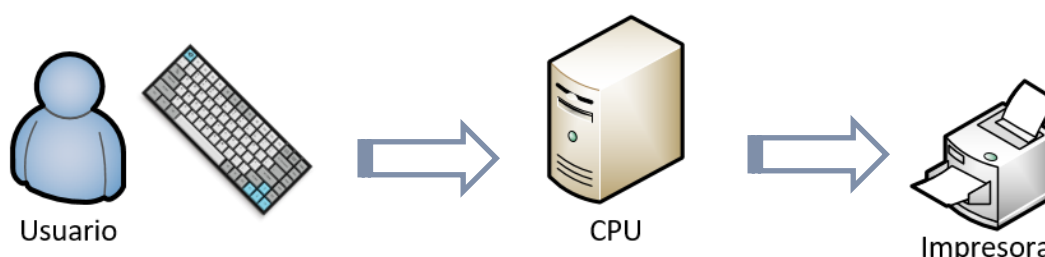
Para crear nuevas aplicaciones se utilizan lenguajes de programación, gestores de bases de datos, entre otras herramientas de software de base. La adecuada combinación de éstas permitirá, por ejemplo, automatizar procesos que forman parte de un sistema de información.

	Temática del vídeo	Fuentes
	“Qué es software y qué es hardware”	GCFAprendeLibre
	“Tipos de Software”	Pase de lista

### 1.2.3 Funciones de los dispositivos de hardware

Tomaremos como ejemplo al computador personal, que está compuesto por una CPU o Unidad Central de Procesos y los dispositivos periféricos.

- **Periféricos de entrada:** Son los que permiten al usuario proporcionar información al dispositivo: teclado, ratón, micrófono, entre otros.
- **CPU o microprocesador:** Realiza su trabajo gracias a dos componentes fundamentales: la unidad de control y la unidad aritmético/lógica.
- **Periféricos de salida:** Nos permiten recibir la información que procesa el dispositivo: monitor, impresora, parlantes, etc.




Con ayuda de esta ilustración, describiremos una “tarea” añadiéndole una perspectiva sistémica:

El usuario envía una **petición a través del teclado (ENTRADA)**. La **CPU** recibe la orden y, con ayuda del sistema operativo, la **traduce y desglosa en varias operaciones (PROCESO)** que el dispositivo interpreta y ejecuta como una **lectura de datos (ENTRADA)** de un pen drive para **enviarlos hacia la impresora (SALIDA)**. Finalmente, el usuario decide **guardar (SALIDA) los datos actualizados** en el pendrive.

**Periféricos de entrada y salida:** Son los que integran ambas funciones en un mismo dispositivo: el auricular con micrófono, un equipo multifunción (impresora y escáner), la pantalla táctil, entre otros.

También entran en esta categoría todos los dispositivos de almacenamiento: tarjetas de memorias, discos, pen drive, grabador de DVD, etc., así como los dispositivos que permiten el intercambio de datos: módem, tarjeta de red, adaptador de conexión (WiFi, Bluetooth, infrarrojo, etc.).

	Temática del vídeo	Fuentes
	“Entrada, procesamiento y salida de la información en un dispositivo de cómputo”	Nahuel Esquivel

### **Actividad 3**

Partiendo de los ejemplos mostrados en el vídeo anterior, identifique los componentes de hardware integrados en un teléfono móvil y desglose de manera breve una tarea sencilla que Ud. realiza con dicho dispositivo completando la tabla siguiente:


Tarea que realiza:	(Descripción breve de referencia común)		
Funciones:	Entrada	Proceso	Salida
Componente de hardware:			
Forma en la que el hardware interviene en la “tarea que realiza”:			

#### 1.2.4 Software de Sistema Operativo

“Es el software que controla el hardware y sirve de base para el funcionamiento de otros programas y aplicaciones” (Publicaciones Empresariales UNAM, 2016, p. 15). Es imprescindible para el correcto funcionamiento de cualquier dispositivo con capacidades de computación. La ilustración 6 muestra varios sistemas operativos de uso común.



Ilustración 6. Sistemas Operativos (<https://sites.google.com/site/faseno3postareagrupo17/sistemas-operativos>)

	<p>El sistema operativo tiene varios subcomponentes con funciones muy específicas. A continuación se resume las funciones más comunes que realizará cualquier sistema operativo:</p> <ul style="list-style-type: none"> <li>✓ Administra los recursos ofrecidos por el hardware.</li> <li>✓ Actúa como intermediario entre el usuario y el dispositivo de cómputo.</li> <li>✓ Coordina y controla la ejecución del software de aplicaciones.</li> <li>✓ Organiza la información en la estructura básica de archivos y carpetas.</li> </ul>
---	--

Para comprender mejor la forma en que trabaja un sistema operativo, compartimos el siguiente recurso audiovisual:

Fuente: El Salón de Informática



### 1.2.5 Interfaces

Las interfaces son elementos tecnológicos que permiten la interconexión entre dos dispositivos o sistemas para que trabajen de manera integrada. La necesidad primaria de intercambio de datos puede ser a nivel de hardware, de software, o en ambos entornos, tal como sucede en las telecomunicaciones.

A nivel de hardware existen diversas especificaciones técnicas que posibilitan la interconexión entre un origen (emisor) y un destino (receptor). Por este motivo, toda interfaz debe disponer de un conector físico; por ejemplo, HDMI, USB y SATA, son conectores de interfaces de hardware. La “forma lógica” de establecer de qué manera se realizará el intercambio de información recibe el nombre de **protocolo**. Algunos protocolos forman parte del **firmware**, que es software integrado dentro de los circuitos del dispositivo y en muchos casos, podrá ser actualizado si el fabricante así lo determina.



Fuente: Fredy García

### 1.2.6 Redes Informáticas

Una red es un conjunto de dispositivos (denominados “nodos”) y software conectados entre sí mediante una NIC (**N**etwork **I**nterface **C**ard) para enviar y recibir datos en forma de impulsos eléctricos, ondas electromagnéticas o impulsos luminosos con la finalidad de compartir información, recursos y servicios que en esencia se reducen a hardware y software.



Fuente: Manage Engine LATAM

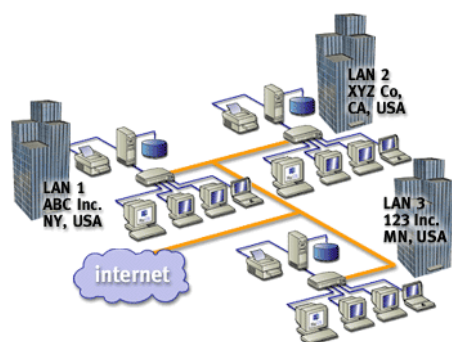
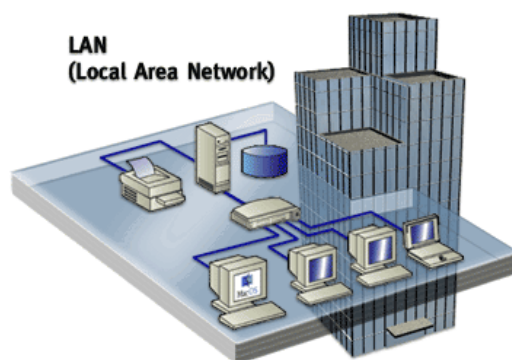


Tabla 2. Tipos de redes según su amplitud o alcance geográfico.

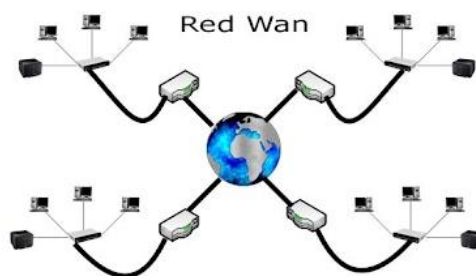
**PAN:** Red de área personal. Es una red pequeña de pocos metros de alcance, pueden ser construidas con cables o de manera inalámbrica, por ejemplo: en una habitación se interconectan un ordenador, una impresora y un smartphone.



**LAN:** Red de área local. Es un conjunto de equipos que pertenecen a la misma organización y están dentro de un área geográfica pequeña, por ejemplo: un colegio con un laboratorio de computación y sus oficinas (rectorado, secretaría, etc.).



**MAN:** Red de área metropolitana. Está formada por equipos que se encuentran en distintos edificios de una misma ciudad, ejemplo: el municipio y el registro de la propiedad.



**WAN:** Red de área extensa. Conecta varios equipos, a grandes distancias geográficas, como distintas ciudades e incluso distintos continentes. Generalmente utiliza internet. Por ejemplo: Uleam y sus extensiones.



La accesibilidad a modernos dispositivos de red y servicios de conectividad más rápidos gracias a la fibra óptica y la tecnología 5G, ha permitido reemplazar interfaces de red a nivel de MAN empleando INTERNET como medio de conexión principal.

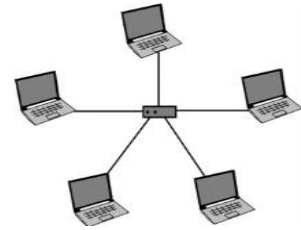


Fuente: YACKLYON

## 1.2.7 Topología de redes

La topología es el mapa o la forma en que está diseñada la red a nivel físico o lógico. Existen varias topologías y a continuación se exponen dos de uso común:

- ✓ **Estrella:** Cada dispositivo está conectado directamente a un punto central y todas las comunicaciones se hacen necesariamente a través de ese punto (conmutador, repetidor o concentrador). Entre sus características destacan las siguientes:



- Posee un sistema que permite agregar nuevos dispositivos fácilmente.
- Reconfiguración rápida.
- Fácil de prevenir daños y/o conflictos, ya que no afecta a los demás dispositivos si falla alguno de ellos.
- Centralización de la interconexión y administración de la red.
- Fácil de encontrar fallos.

- ✓ **Estrella extendida:** Esta topología se presenta cuando un concentrador se conecta a uno o varios concentradores, por lo que también se conoce como “conexión en cascada” ya que, al conectar un puerto de red de un concentrador “principal” con un concentrador “secundario”, este último permitirá a otros dispositivos enlazarse a la misma red como si estuvieran conectados al concentrador principal. Entre sus características destacan las siguientes:

- Mejorar el rendimiento de la red sin reemplazar el concentrador principal.
- Extender el alcance de la red alámbrica o inalámbrica.
- Conectar más dispositivos.
- Aislar el tráfico de la red.





#### Actividad 4

Encuentre información sobre uno de los componentes de un sistema operativo (se mencionan a partir del minuto 2:12 del vídeo “[¿Cómo funciona un sistema operativo?](#)”) para completar la siguiente tabla:

Componente de Sistema Operativo:	(Nombre común del componente de sistema operativo)
¿De qué se encarga? :	(Breve descripción de la tarea/operación general que realiza el componente)
Ejemplo de tarea que realiza:	(Ejemplo de una tarea/operación específica en la que interviene este componente)

# UNIDAD 2

## 2 ALGORITMOS Y DIAGRAMAS DE FLUJO

### 2.1 ALGORITMOS

#### 2.1.1 Definición y características de un algoritmo

“Un algoritmo es un conjunto de acciones o pasos finitos, ordenados de forma lógica y que se utilizan para resolver un problema o para obtener un resultado” (Herrera Morales, J., Gutiérrez Posada, J. y Pulgarín Giraldo, R., 2017, p. 45).

Tabla 3. Características de un algoritmo

Preciso	Definido	Finito
Instrucciones claras en cada uno de sus pasos, sin ambigüedades. No realiza pasos o cálculos que no consten en el algoritmo.	Si se siguen los pasos dos o más veces, siempre nos llevará al mismo resultado partiendo de una misma situación inicial. Se debe especificar fórmulas en caso de requerirlas.	Tiene un determinado número de pasos, es decir, tiene principio y fin.

#### 2.1.2 Clasificación y ejemplos

Tabla 4. Clasificación de los algoritmos según sus signos (Conde, 2009)

Cualitativos o No numéricos	Cuantitativos o Numéricos
Son instrucciones paso a paso que se dan de forma oral o escrita, por ejemplo, una <b>receta de cocina para preparar un té aromático</b> : <ol style="list-style-type: none"><li>1. INICIO</li><li>2. Tomar 5 clavos de olor, 3 ramas de canela y dos tazas de agua.</li><li>3. Verter los ingredientes en una olla y poner al fuego hasta que el agua alcance el punto de ebullición.</li><li>4. Mantener la cocción por un minuto adicional y luego apagar.</li><li>5. Dejar reposar hasta temperatura apta para beber.</li><li>6. Servir endulzado con miel o panela de acuerdo con el gusto.</li><li>7. FIN</li></ol>	Se basa en cantidades que servirán para encontrar el resultado de algún cálculo o ecuación, por ejemplo, <b>encontrar el área de un cuadrado</b> : <ol style="list-style-type: none"><li>1. INICIO</li><li>2. Obtener el valor que represente la medida del lado del cuadrado.</li><li>3. Multiplicar consigo mismo el valor obtenido en el paso anterior.</li><li>4. Mostrar el resultado del cálculo realizado.</li><li>5. FIN</li></ol>



Las cantidades que expresan unidades de medida pueden requerir conversión a otras unidades de medida al momento de interpretar una situación o antes de aplicar una fórmula con dichas cantidades, tal cual sucede en física y en química.



### Actividad 5

*Redacte el algoritmo cualitativo para resolver un cuestionario en el aula virtual Moodle.*

### 2.1.3 Estructura de un algoritmo

Los algoritmos son esenciales en muchos aspectos de nuestra vida moderna, especialmente en la automatización de procesos que requieren un tratamiento automático de los datos que utilizan. De allí que un enfoque sistémico proporcione las bases para su diseño:

Entrada	Proceso	Salida
Son aquellos elementos requeridos para realizar el proceso y que ayudarán a obtener el resultado.	Se trata de las operaciones a realizar con los elementos de entrada y que permiten obtener el resultado deseado.	Son los resultados de las operaciones realizadas durante el proceso.
<ul style="list-style-type: none"> <li>• Ingredientes: 5 clavos de olor, 3 ramas de canela y dos tazas de agua.</li> </ul>	<ul style="list-style-type: none"> <li>• Verter los ingredientes en una olla y poner al fuego hasta que el agua alcance el punto de ebullición.</li> <li>• Mantener la cocción por un minuto adicional y luego apagar.</li> <li>• Dejar reposar hasta temperatura apta para beber.</li> </ul>	<ul style="list-style-type: none"> <li>• Servir endulzado con miel o panela de acuerdo con el gusto.</li> </ul>
<ul style="list-style-type: none"> <li>• El valor que representa la medida del lado del cuadrado.</li> </ul>	<ul style="list-style-type: none"> <li>• Multiplicar consigo mismo el valor indicado como entrada.</li> </ul>	<ul style="list-style-type: none"> <li>• Mostrar el resultado del cálculo realizado.</li> </ul>

Para comprender mejor cómo funciona un algoritmo, le invitamos a revisar el siguiente recurso audiovisual compartido por Magic Markers:





### Actividad 6

Identifique los elementos necesarios para obtener la superficie de un triángulo (considerando las medidas de la base y la altura) y clasifíquelos en la siguiente tabla:

Entrada	Proceso	Salida

## 2.2 ALGORITMOS COMPUTABLES

### 2.2.1 Datos y tipos de datos

Un dato es una representación simbólica en forma de números, letras, imágenes, entre otras, de una característica de un elemento (Herrera y otros, 2017, p. 22), misma que determinará una forma conveniente para su almacenamiento y procesamiento dentro de un algoritmo computable.

Tabla 5. Tipos de datos elementales

Numérico	Alfanumérico o Alfabético	Lógico
Contiene números reales (enteros o racionales) que permiten realizar cálculos. Por ejemplo: 1593 -10,1 0,0095	Pueden contener letras, números y símbolos en forma combinada. Por ejemplo: Ecuador MBA-123 admision@uleam.edu.ec 1425-A\$#%	Solo puede contener un valor VERDADERO o FALSO. También se puede caracterizar como: 0 => Falso 1 => Verdadero

Compartimos un vídeo que nos demuestra cómo definir convenientemente a un dato:

Fuente: Canal de Charly Cimino



Además de un tipo, cada dato a ser utilizado requiere ser identificado con un nombre único conformado por letras o letras combinadas con números (en este caso considerar que siempre deberá empezar por una letra). Ambos requisitos le permitirán reservar un espacio de memoria para su tratamiento en cualquier lenguaje de programación de alto nivel. Se recomienda

emplear un nombre descriptivo al momento de asignar un identificador al campo de memoria. A continuación, se listan cuatro ejemplos de identificadores válidos: **A**, **Edad**, **N1**, **N2**


	<p>Todo algoritmo numérico podrá convertirse en un programa de computación mediante su transcripción a un lenguaje de programación. Los algoritmos computables se caracterizan por procesar grandes volúmenes de datos con mucha rapidez.</p>
---	---

Tabla 6. Datos constantes y datos variables

Constante	Variable
<p>Una constante almacena un dato cuyo valor no puede cambiar durante la ejecución de un programa. Aquí dos ejemplos:</p> <p>Si el valor de <math>\pi</math> es 3.14159265359, podremos identificarlo y guardar su valor de la siguiente manera:</p> <p style="text-align: center;"><b>PI = 3.14159265359</b></p> <p>Si el porcentaje establecido por el IESS para calcular el monto que debe cancelar un empleado del sector privado es de 9,45%, podremos identificarlo y guardar su valor de la siguiente manera:</p> <p style="text-align: center;"><b>PIESS = 0,0945</b></p>	<p>Es un campo de memoria que puede tomar cualquier valor como dato y cambiar su contenido posteriormente mediante asignación o ingreso directo del usuario durante la ejecución de un programa. Aquí un ejemplo: Fórmula para calcular la longitud de una circunferencia a partir de un diámetro cualquiera:</p> <p style="text-align: center;"><b>LC = PI × D</b></p> <p>Donde: PI =&gt; constante <math>\pi</math>  D =&gt; variable que guarda el valor de la medida del diámetro.  LC =&gt; variable que guardará el resultado de la fórmula.</p>

Compartimos el siguiente recurso respecto del uso de variables e identificadores:

Fuente: Canal de Charly Cimino



## 2.2.2 Operadores aritméticos y expresiones

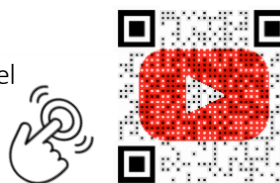
Para realizar cálculos con datos numéricos, los algoritmos computacionales simbolizan cada operador aritmético y también aplican la jerarquía matemática en cada operación. La combinación de operadores, constantes, variables y “funciones especiales” permiten formular una expresión cuyo resultado se puede almacenar en la memoria empleando una variable y el operador de asignación (=, ←) como nos muestra la Tabla 7.

Tabla 7. Operadores aritméticos y expresiones

Símbolo	Nombre	Operación	Ejemplos de expresiones		Interpretación
=	Igual	Asignación	A = 10	C = 5+A	
+	Signo más	Suma	3 + 16	P = A+B	
-	Signo menos	Resta	8 - 5	A = C-D	
*	Asterisco	Multiplicación	4 * 3	S = B*C	
/	Barra oblicua (slash)	División	7 / 3	C = L/A	
^	Acento circunflejo	Potenciación	2 ^ 8	L = A^B	
()	Paréntesis	Agrupación	(A+3) * B	R = (A+3)*B	

Aplicar la jerarquía de los operadores aritméticos es fundamental para escribir fórmulas matemáticas que calculen un valor de manera correcta. A continuación, compartimos un corto vídeo que lo explica con ejemplos:

Fuente: Asesor Juan Manuel



### Actividad 7

- Complete la última columna de la Tabla 7 con lo que usted interpreta de la expresión de la penúltima columna.
- Lea los siguientes planteamientos e Identifique constantes y variables con su correspondiente tipo de dato:
  - La empresa ABC Seguridad define turnos rotativos: de 00h00 a 08h00, de 08h00 a 16h00 y de 16h00 a 00h00.
  - Jorge recorre varios kilómetros diariamente para vender varias fundas de dulces artesanales a un dólar cada funda.
- Transcriba a expresión computacional cada una de las siguientes expresiones aritméticas asignando su resultado a una variable cualquiera:

$$\frac{3 + b^2}{a - b}$$

$$\frac{\frac{x}{y} - 2}{x^4 + 3}$$

$$\frac{2a + b}{\frac{3k^2}{b + a}}$$



### 2.2.3 Modelado computable para algoritmos cuantitativos

Para proponer la solución a un problema específico, se debe considerar el dominio del área a la que pertenece el problema, de esta manera se podrá analizar y modelar una solución computable de manera apropiada.



Una herramienta básica para el análisis y diseño de algoritmos es el reconocimiento de los elementos que conforman estructura: se recomienda iniciar determinando las **salidas**, luego establecer el **proceso** que permitirá obtener cada una de las salidas y, de acuerdo a lo que necesite ser procesado, identificar las **entradas** que serán requeridas.

Revisemos el siguiente ejemplo: **Elaborar un algoritmo que me permita obtener el total de una suma entre dos números enteros cualquiera.**

Un caso de ejemplo para este ejercicio es el siguiente: **8+11=19**, sin embargo, nuestro algoritmo deberá resolver otros casos, es decir, debemos considerar que los números a sumar podrían ser muy diferentes al ejemplo dado, por lo que utilizaremos variables:

Entrada	Proceso	Salida
Dos números enteros cualquiera => <b>dos variables</b> para almacenar los números.	<b>Calcular la suma</b> de los dos números previamente almacenados y guardar este resultado en la variable <b>R</b> .	<b>Mostrar el resultado</b> guardado en la variable <b>R</b> del paso anterior.

Con los elementos identificados, podemos describir el algoritmo en el orden necesario para cumplir con el objetivo (salida) requerido:

Pasos	Detalle del algoritmo
1	INICIO
2	Ingresar dos números enteros y guardarlos en las variables correspondientes: <b>N1</b> y <b>N2</b>
3	Sumar los dos números: <b>N1 + N2</b> y guardar el resultado en la variable <b>R</b> .
4	Mostrar el resultado guardado en la variable <b>R</b>
5	FIN

Ahora realicemos otro ejercicio con un enfoque “menos explícito” pero más centrado en aquellos elementos que no pueden faltar: **Redactar un algoritmo que permita obtener el precio final de un artículo conociendo su valor base o precio original antes de agregarle el 15% de impuesto IVA. Al final se mostrarán todos los valores involucrados.**

Entrada	Proceso	Salida
VALOR: una variable para guardar el precio original del artículo.	$IVA = VALOR * 0,15$ $TOTAL = VALOR + IVA$	VALOR IVA TOTAL

Pasos	Detalle del algoritmo
1	INICIO
2	Obtener una cantidad para VALOR
3	$IVA = VALOR * 0,15$
4	$TOTAL = VALOR + IVA$
5	Mostrar el contenido de VALOR, IVA y TOTAL
6	FIN

Para mayor comprensión respecto del análisis estructural que involucra el diseño de un algoritmo computacional, compartimos el siguiente recurso audiovisual:



Fuente: Eddy Ordoñez



### Actividad 8

Clasifique en la tabla siguiente los elementos que permitirán obtener la superficie de un círculo a partir de la medida de su radio y luego describa con sus propias palabras el algoritmo completo:

Entrada	Proceso	Salida

## 2.3 FORMAS DE REPRESENTAR UN ALGORITMO

Un algoritmo se puede representar de tres maneras diferentes: pseudocódigo, diagrama de cajas rectangulares o diagrama de flujo.

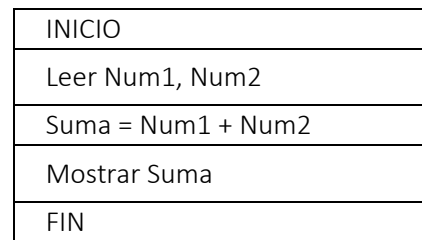
Tabla 8. Ejemplos de representación de un algoritmo

**Pseudocódigo:** Expresa las instrucciones que ejecuta un algoritmo de forma similar a un lenguaje de programación de alto nivel. Sin embargo, este “falso código” está más orientado a la interpretación humana antes que a la ejecución en un dispositivo de computación (Ordoñez, Luis, 2019).

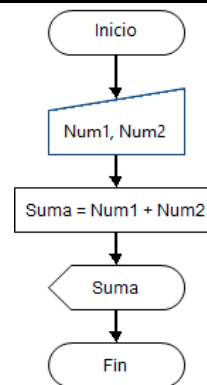
**Algoritmo que suma dos números**

**Leer** Num1, Num2  
Suma = Num1 + Num2  
**Escribir** Suma  
**FinAlgoritmo**

**Diagrama de Cajas Rectangulares o diagrama Nassi-Schneiderman (N/S):** Es similar al pseudocódigo pero cada paso o instrucción se ubica dentro de un rectángulo.










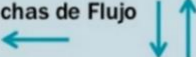
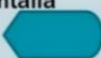
**Flujograma o Diagrama de flujo:** Es la representación gráfica de un algoritmo que utiliza símbolos para expresar las instrucciones correspondientes a entradas, procesos o salidas. Cada símbolo se une a otro mediante una flecha que indicará el camino a seguir desde el inicio hasta el final del algoritmo.



### 2.3.1 Símbolos y reglas para elaborar un diagrama de flujo

De acuerdo con el sitio web de Asana (2023), los diagramas de flujo son la representación gráfica y de manera secuencial de un flujo de procesos o actividades necesarias para cumplir una meta.

Tabla 9. Simbología estándar utilizada para representar diagramas de flujo (Smartdraw, s.f.)

símbolo	Función	Símbolo	Función
Terminal 	Indicar el inicio y fin del diagrama	Teclado 	Introducir datos manualmente por el teclado
Entrada/salida 	Entrada o salida simple de información	Decisión 	Indica operaciones lógicas o de comparación y tienen dos salidas dependiendo del resultado.
Proceso 	Realizar cualquier operación o calculo con la información	Conectores 	Une dos partes del diagrama a la misma o diferente página
Salida a Impresora 	Salida de información a la impresora	Flechas de Flujo 	Indica la dirección del flujo de la información
Salida a Pantalla 	Mostrar información de salida a la pantalla		

### Reglas para elaborar diagramas de flujo

1. Los diagramas de flujo deben elaborarse de arriba hacia abajo y/o de izquierda a derecha.
2. Los símbolos se unen con líneas que culminan con una flecha para indicar la dirección del flujo. Se deben de utilizar líneas horizontales o verticales, no diagonales.
3. Toda línea debe salir de un símbolo y llegar a otro. No debe haber "líneas en el aire".
4. Todo símbolo tendrá una línea de entrada y al menos una línea de salida. Se exceptúan de esta regla los símbolos de Inicio y Fin.
5. Los símbolos de decisión son los únicos que podrán tener más de una línea de flujo de salida.
6. Todo texto escrito dentro de un símbolo debe ser legible y preciso, evitando el uso de palabras o expresiones innecesarias.
7. Se debe evitar el cruce de líneas. Se recomienda el uso de conectores si una línea es muy extensa o en caso de que el diagrama requiera continuar en otra página, sin embargo, el abusar de este recurso restará legibilidad al diagrama.



Hay varias herramientas digitales para elaborar diagramas de flujo, sin embargo, algunas como DFD, Flowgorithm, PSeInt (pseudocódigo), entre otras, además de cumplir las reglas señaladas de forma automática, también nos ayudarán a comprobar los resultados de cualquier algoritmo computable.

Para aprender rápidamente cómo utilizar una sencilla herramienta como DFD, observemos el siguiente vídeo de corta duración:

**Fuente:** Fernando Naranjo



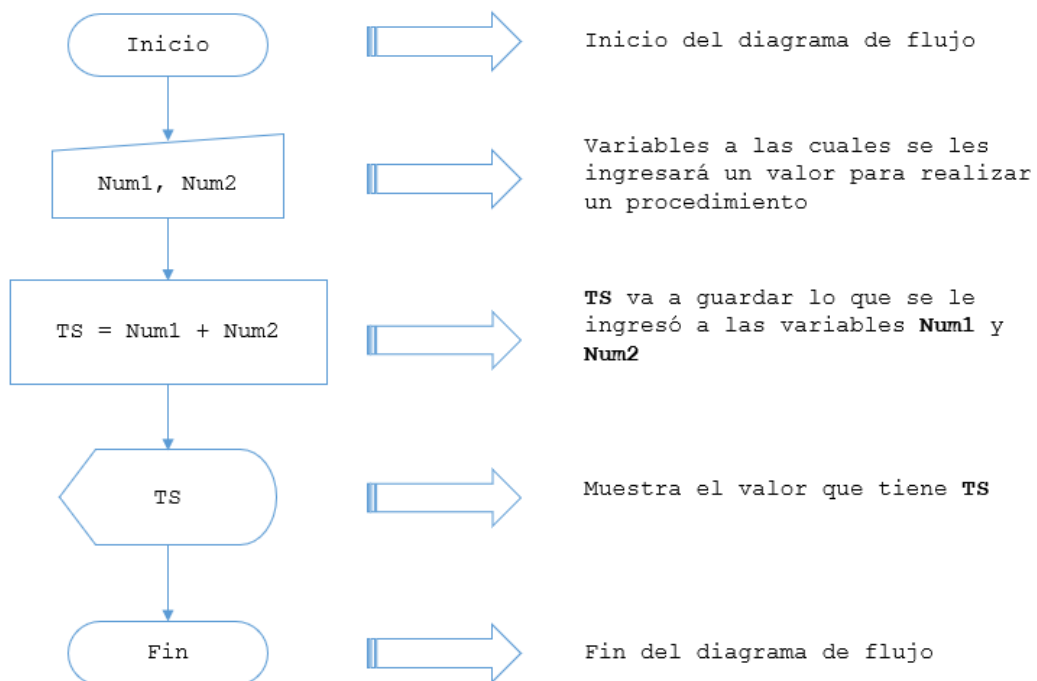
Enlace para descargar desde el SharePoint de la herramienta DFD.



### 2.3.2 Detalle de un diagrama de flujo básico con su prueba de escritorio

En el tema de algoritmos computables, ya tenemos realizados algunos ejercicios. Ahora, vamos a utilizar estos mismos ejemplos para representarlos en un diagrama de flujo.

Realizar un diagrama de flujo que me permita obtener el total de una suma de dos números y mostrar el resultado.



Las pruebas de escritorio servirán para comprobar que el algoritmo entrega las salidas esperadas. A mayor complejidad del problema, habrá que realizar más pruebas para asegurar que nuestra "propuesta algorítmica" resuelve el problema planteado.

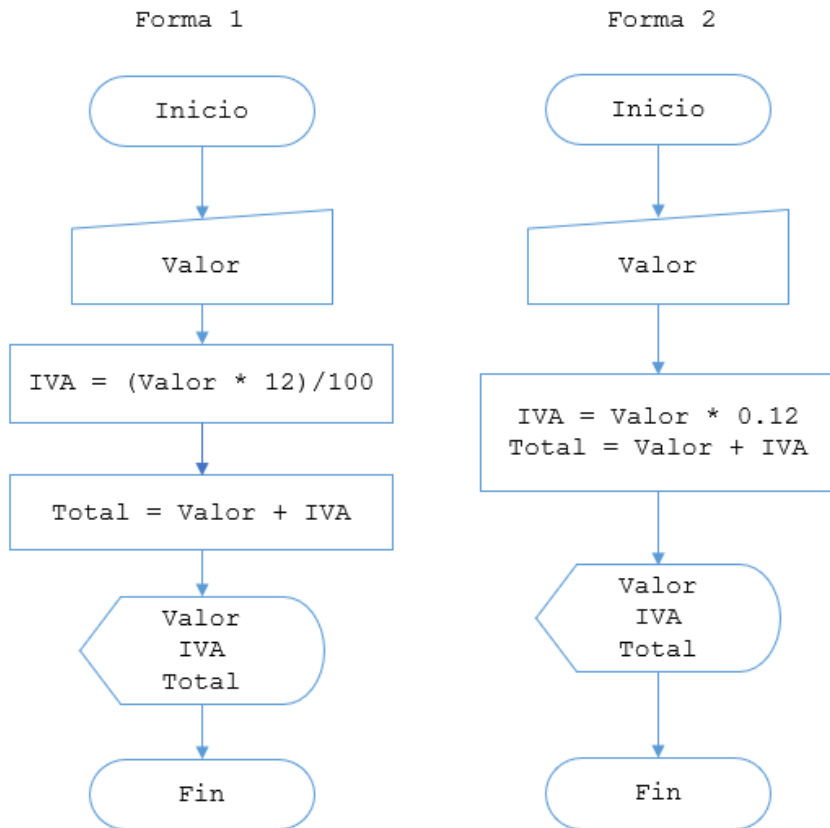
#### Prueba de escritorio

A continuación, aplicaremos la prueba de escritorio al ejercicio anterior:

Num1	Num2	TS	Salidas
15	20	35	35

**Procedimiento**  
 $TS = Num1 + Num2$   
 $15 + 20$

Realizar un diagrama de flujo que me permita ingresar un valor al que se le añadirá un IVA del 12%. Al final se deberá mostrar el valor ingresado, el IVA y el valor total con el IVA agregado.



**Prueba de escritorio**

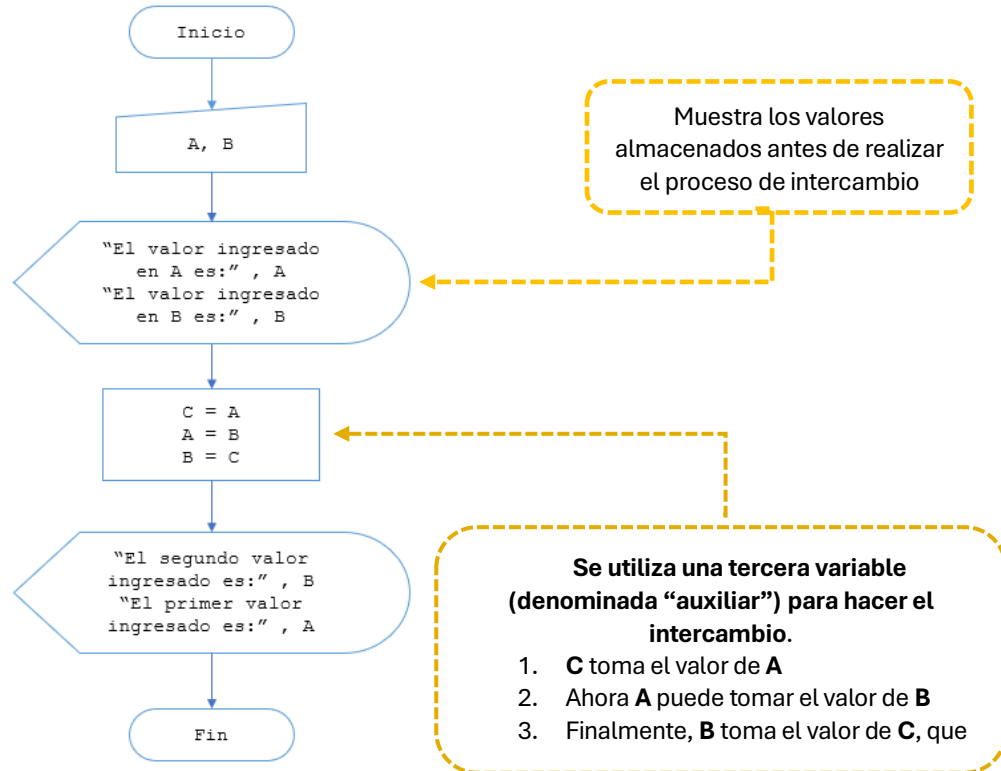
<b>Valor</b>	<b>IVA</b>	<b>Total</b>	<b>Salidas</b>
157	18,84	175,84	157 18,84 175,84

Para una explicación detallada respecto de cómo realizar adecuadamente una prueba de escritorio sencilla, se invita a observar el siguiente vídeo de corta duración:

**Fuente:** Saúl Quispe Valdez



Elaborar un diagrama de flujo que permita ingresar dos números distintos, uno en A y otro en B, se solicita que en el proceso se realice un intercambio de valores en las variables: el número de A se guarde en B y el número de B se guarde en A. Se deberá mostrar el valor de las variables antes y después del intercambio.



### **Actividad 9**

- Empleando una herramienta digital para flujogramas, replique el diagrama anterior (intercambio de valores entre dos variables), verifique su funcionamiento y realice una prueba de escritorio.
- Empleando la herramienta digital de su preferencia, elabore un diagrama de flujo a partir del algoritmo diseñado para calcular la superficie de un círculo a partir de la medida de su radio. Realice dos pruebas de escritorio a este diagrama de flujo.

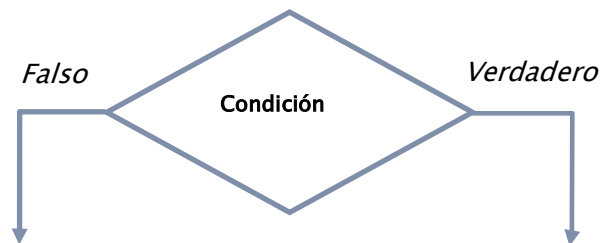
## UNIDAD 3

### 3 DECISIONES EN ALGORITMOS COMPUTABLES

#### 3.1 DECISIONES

##### 3.1.1 Definición

Una **decisión o condición** determinará un “camino alternativo” que bifurcará el flujo de un algoritmo a partir de una evaluación **Verdadera** o **Falsa** realizada mediante una expresión condicional. Su representación en un diagrama de flujo se realiza de la siguiente manera:



Las decisiones son importantes en las ENTRADAS porque nos permiten garantizar que un dato cumple los requisitos para continuar con su procesamiento. Esto se denomina comúnmente “validación”.

En el PROCESO, condicionar la ejecución u omisión de pasos importantes nos permitirá obtener las SALIDAS adecuadas.

##### 3.1.2 Operadores relacionales o de comparación

Los operadores relacionales son símbolos que se usan para comparar dos valores o expresiones de un mismo tipo de dato. El resultado de la comparación puede ser **Verdadero** o **Falso** y constituye la forma más básica para establecer una condición.



Tabla 10. Operadores de relación y ejemplos de uso

Operador	Denominación	Ejemplo	Resultado
>	Mayor que	6 > 6 5,1 > 5	FALSO VERDADERO
<	Menor que	A < A B < 0,01	FALSO VERDADERO
>=	Mayor o igual	-5 >= A B >= 1	VERDADERO FALSO
<=	Menor o igual	A <= B B <= A	VERDADERO FALSO
=	Igual	A = B	FALSO
==		X = Y	FALSO
<>	Diferente	A <> B	VERDADERO
!=		X != "María"	VERDADERO

A	B
-5	0

X	Y
Miguel	Ariel

### **Actividad 10**

Con los mismos datos de la tabla anterior, evalúe las siguientes expresiones:

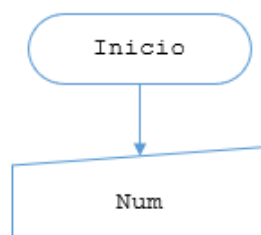
Condición	Resultado evaluación	Condición	Resultado evaluación
B > A		Y > B	
B >= B		Y <> "Ariel"	

### 3.1.3 Estructura selectiva SI ... ENTONCES ... SINO (If ... Then ... Else)

Realizar un diagrama de flujo que me permita ingresar un número. Si el número es mayor a 10, sumarle 5 y si el número es menor o igual a 10, restarle 1.

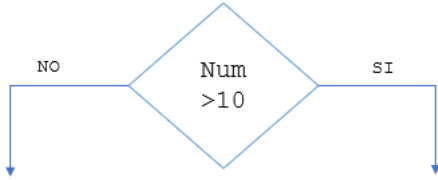
Desarrollaremos este ejercicio explicando cada paso:

1. Se ingresará el número en una variable, tal como hemos realizado en ejercicios anteriores:

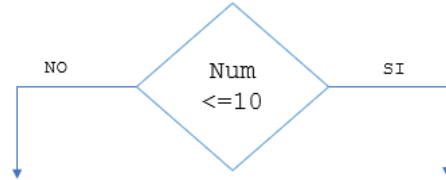


2. Basado a lo indicado en el planteamiento, definimos una condición utilizando el/los operadores relacionales. En la imagen siguiente se puede observar que podemos emplear dos formas de comparar o preguntar:

a. Podemos preguntar si el número ingresado en **Num** es mayor a 10.

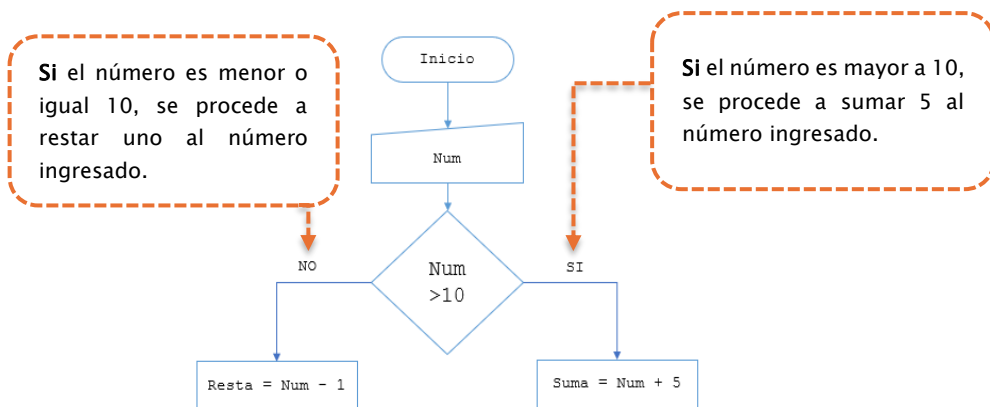


b. Podemos preguntar si el número ingresado en **Num** es menor o igual a 10.

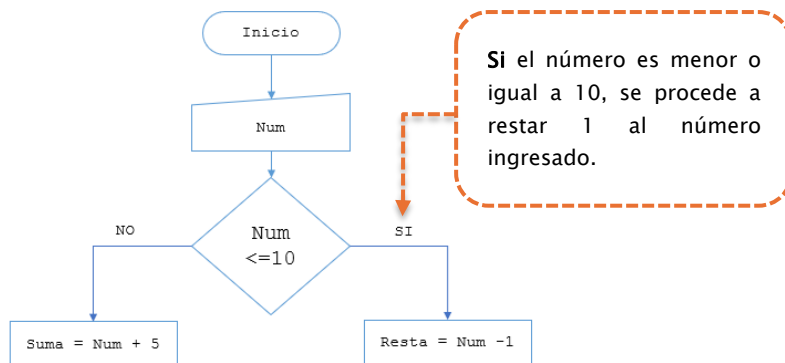


3. Realizamos el proceso en base a la condición elegida, puesto que la evaluación se puede establecer correctamente de dos maneras diferentes:

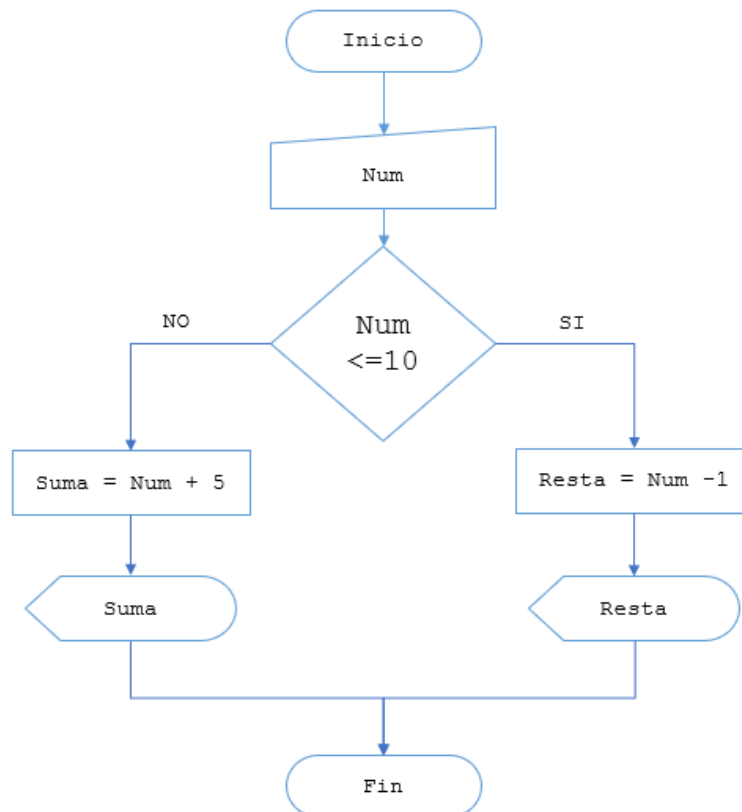
Alternativa a



Alternativa b



4. Finalmente, mostramos el diagrama final y realizamos algunas pruebas de escritorio:



**Prueba de escritorio**





Num	Suma	Resta	Salidas
5	-	4	4
100	105	-	105
10	-	9	9
10,1	15,1	-	15,1


**(Explicación)**

En este caso el 5 **SI** es menor o igual a 10  
 En este caso 100 **NO** es menor o igual a 10  
 En este caso el 10 **SI** es menor o igual a 10  
 En este caso 10,1 **NO** es menor o igual a 10



Cuando utilizamos decisiones, es conveniente **realizar al menos una prueba por cada alternativa de flujo o caso de uso**, especialmente con aquellos datos que podrían tener “conflictos de evaluación” por aproximarse al cumplimiento o incumplimiento de la condición establecida durante el análisis del problema.

	Temática del vídeo	Fuente
	"Estructura selectiva"	FACOMSYS
	"Operadores aritméticos especiales para calcular división entera y residuo de división entera"	Pablo Dechia
	"Uso de condiciones en diagramas DFD"	Programación ATS

 **Actividad 11**

Elabore un diagrama de flujo que permita verificar si un número ingresado es divisible exactamente por 7 (es decir, que al dividir por 7 no queda residuo alguno). Realice el análisis correspondiente y la comprobación de escritorio.

**NOTA:** En DFD no existe el operador DIV, pero se puede emplear la función TRUNC que suprime la parte decimal del resultado de cualquier operación aritmética. Por ejemplo:

TRUNC(4.7) devolverá 4  
TRUNC(75/8) devolverá 9

**EJEMPLOS CON LOS OPERADORES DIV Y MOD**

$$\begin{array}{r} 75 \\ 72 \overline{) 8} \\ \underline{3} \end{array}$$

$$\begin{array}{r} 75 \text{ DIV } 8 \Rightarrow 9 \\ 75 \text{ MOD } 8 \Rightarrow 3 \end{array}$$

$$\begin{array}{r} 99 \\ 90 \overline{) 15} \\ \underline{9} \end{array}$$

$$\begin{array}{r} 99 \text{ DIV } 15 \Rightarrow 6 \\ 99 \text{ MOD } 15 \Rightarrow 9 \end{array}$$

$$\begin{array}{r} 154 \\ 144 \overline{) 12} \\ \underline{10} \end{array}$$

$$\begin{array}{r} 154 \text{ DIV } 12 \Rightarrow 12 \\ 154 \text{ MOD } 12 \Rightarrow 10 \end{array}$$

## 3.2 BLOQUES DE DECISIÓN

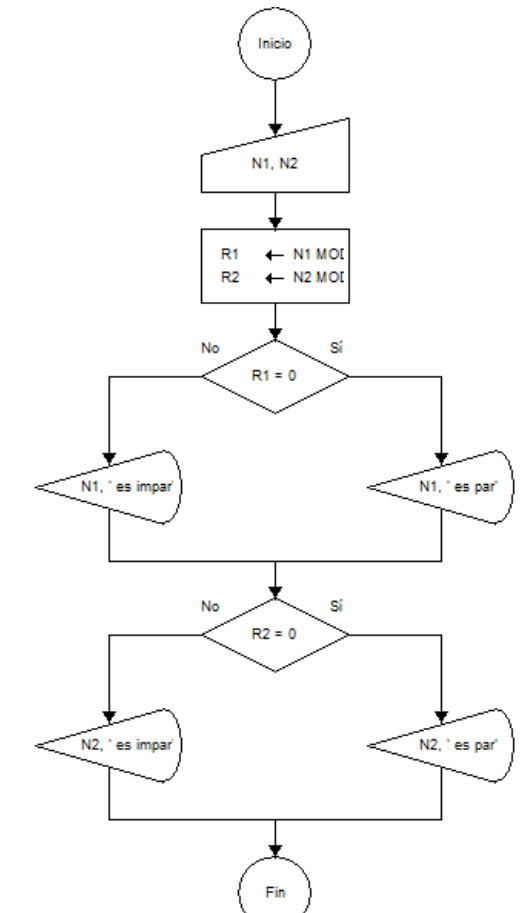
### 3.2.1 Decisiones secuenciales

Este tipo de estructura muestra una decisión a continuación de otra debido a que ambas deben evaluarse en el mismo flujo independientemente del resultado de cada una. Ejemplo:

**Ingresar dos números enteros y comprobar si ambos son pares o impares**

Casos tomados como ejemplos: 9 => "Número impar"  
8 => "Número par"

ENTRADAS	PROCESOS	SALIDAS
(Primer número a ser evaluado) <b>N1</b>  <b>Casos posibles</b> <b>N1</b> <b>(Número impar)</b> <input type="text" value="9"/>  (Número par)  (Segundo número a ser evaluado) <b>N2</b>  <b>Casos posibles</b> <b>N2</b> <b>(Número impar)</b> <input type="text" value="8"/>  <b>(Número par)</b>	$R1 = N1 \text{ MOD } 2$ <input type="text" value="1"/> <- $9 \text{ MOD } 2$  $\text{¿ } R1 = 0 \text{ ? (SI)}$ $\text{(NO)}$  $R2 = N2 \text{ MOD } 2$ <input type="text" value="0"/> <- $8 \text{ MOD } 2$  $\text{¿ } R2 = 0 \text{ ? (SI)}$ $\text{(NO)}$	   9 es IMPAR         8 es PAR



Salida por pantalla



Salida:

11 es impar

Salida por pantalla



Salida:

12 es par

Salida por pantalla



Salida:

6 es par

Salida por pantalla



Salida:

7 es impar

Prueba de escritorio

N1	N2	R1	R2	(Salidas)
9	8	1	0	9 es impar 8 es par
12	11	0	1	12 es par 11 es impar



**Actividad 12**

Elabore un diagrama de flujo que determine si un número distinto de cero (considerar esta excepción) es positivo o negativo. Determine además si es entero o si tiene parte decimal e imprima la parte entera del número.

**3.2.2 Decisiones anidadas**

Este tipo de bloque utiliza una decisión dentro de otra debido a que necesitará avanzar hacia la condición interna habiendo evaluado la condición externa que la contiene. Generalmente, esta estructura resulta muy conveniente para ir encontrando salidas específicas.

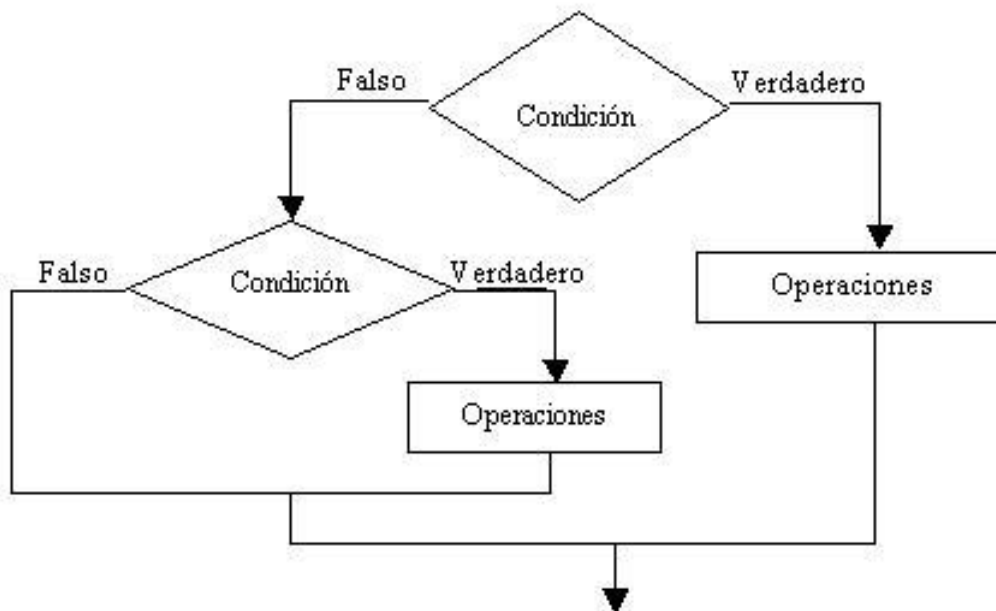
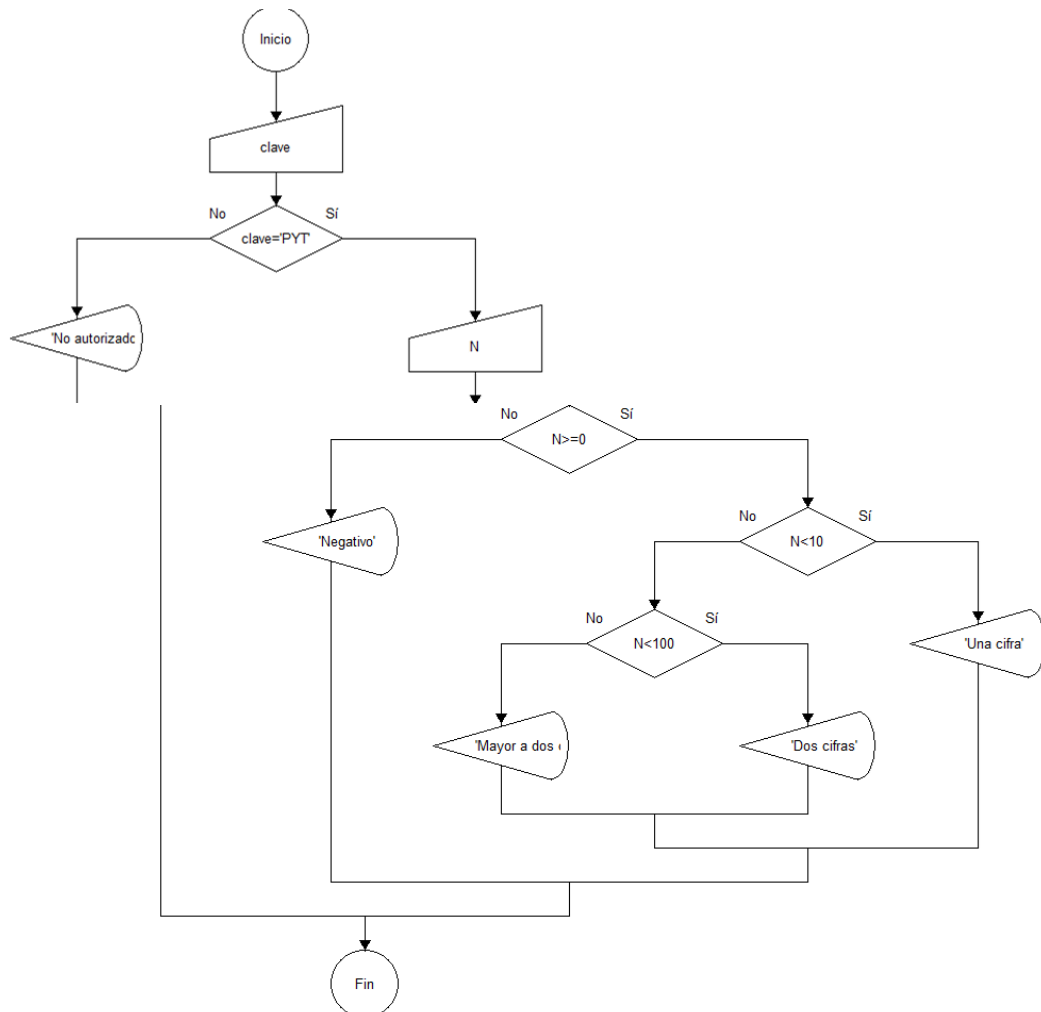






Ilustración 7. Estructuras condicionales anidadas  
(<https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=81&>)

Este tipo de bloque también se conoce como “decisiones en cascada”. A continuación, mostraremos un ejemplo:



A continuación, revisemos tres vídeos de corta duración que explican el uso de este tipo de bloque:

	Temática del vídeo	Enlace para visualización
	“Estructura selectiva con anidamiento”	Programación ATS
	“Condicionales: Número par, impar o cero realizado en DFD”	Programación ATS
	“Decisiones anidadas para obtener el mayor de tres números en DFD”	Asesor Juan Manuel

### **Actividad 13**

Compruebe el flujograma de la página anterior con la herramienta digital de su preferencia. Luego, redacte lo que usted interpreta que realiza dicho algoritmo.

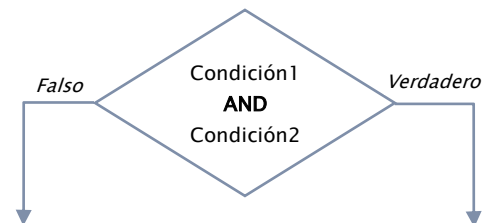
### 3.2.3 Operadores lógicos

Los operadores lógicos permiten evaluar dos o más expresiones en una misma condición evitando emplear decisiones separadas, lo que puede reducir, en algunos casos, el tamaño y complejidad de un algoritmo. Su aplicación en forma secuencial o anidada dependerá del problema a resolver. Los operadores utilizados con mayor frecuencia son los que se describen a continuación:

#### Operador AND - &&

Devuelve VERDADERO si todas las condiciones son Verdaderas, pero devolverá FALSO si encuentra una condición Falsa.

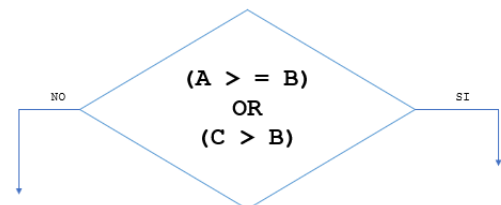
Condiciones	Evaluación	Evaluación AND
5 > 4	VERDADERO	VERDADERO
8 < 10	VERDADERO	
"A" = "A"	VERDADERO	



#### Operador OR - ||

Devuelve VERDADERO si al menos una de las condiciones es Verdadera, pero devolverá FALSO si todas las condiciones son Falsas.

Condiciones	Evaluación	Evaluación OR
5 < 4	FALSO	VERDADERO
5 > 5	FALSO	
"A" <> "B"	VERDADERO	



Estos operadores también tienen una jerarquía: primero se evalúa "AND" y luego "OR", sin embargo, podemos usar paréntesis para cambiar esto tal como hacemos con las expresiones aritméticas:

$$(X > 5) \text{ OR } (X > B) \text{ AND } (X > 4) \implies (X > 5) \text{ OR } \text{RESULTADO\_LOGICO\_AND}$$

$$\left( (X > 5) \text{ OR } (X > B) \right) \text{ AND } (X > 4) \implies \text{RESULTADO\_LOGICO\_OR} \text{ AND } (X > 4)$$



Fuente: Programación ATS

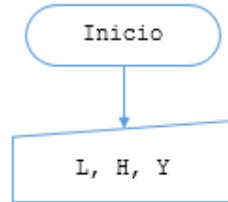


Fuente: Operadores de FreeDFD

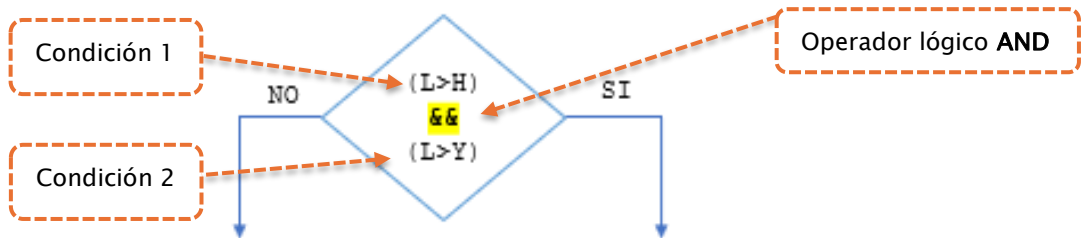


Realizar un flujograma que permita ingresar el valor de tres productos: Arroz, Harina y Yogurt. Muestre el valor del producto más costoso.

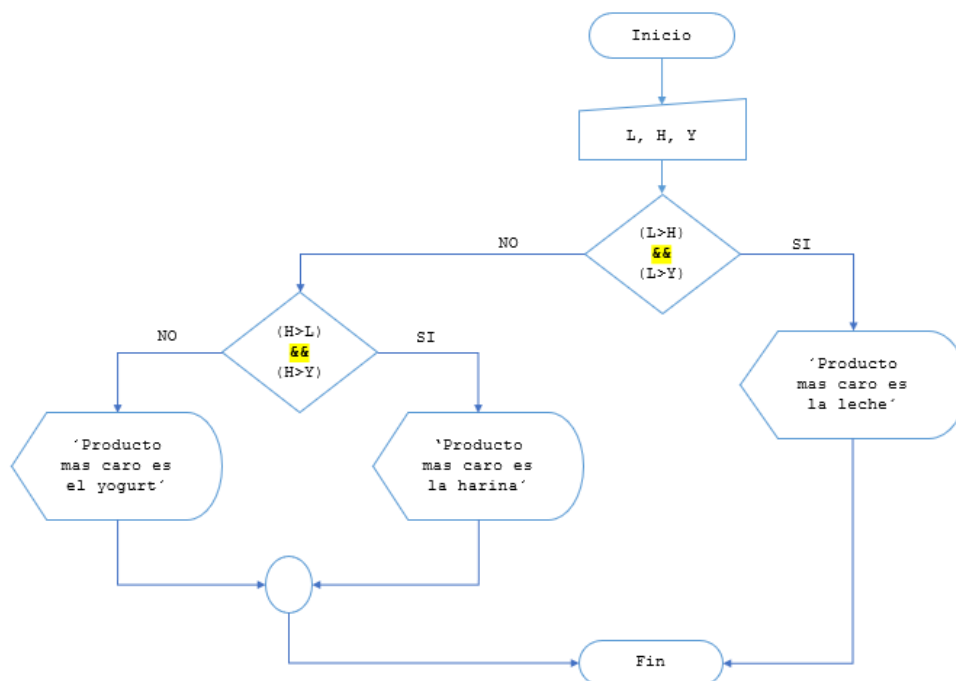
1. Ingresar el costo del arroz, harina y yogurt, en las variables L, H, Y, respectivamente.



2. La primera condición pregunta si **L es mayor que H** **Y** si **L es mayor que Y**:



3. La condición aplicada en el punto anterior permite determinar que el producto más costoso es la leche ya que es el valor que cumple las dos condiciones. Por lo tanto, se debe mostrar en pantalla el nombre del producto más costoso: "Leche".
4. En caso de **no** cumplir **una** de las dos comparaciones de esta primera condición (lo que se evaluará como Falso), se avanzará a la siguiente condición como demuestra el flujograma completo:



**Prueba de escritorio**

<i>L (Leche)</i>	<i>H (Harina)</i>	<i>Y (Yogurt)</i>	<i>(Salidas)</i>
10	5	16	Producto más caro es el yogurt
53	5	16	Producto más caro es la leche
5	10	4	Producto más caro es la harina

En el siguiente recurso audiovisual encontraremos dos ejemplos más de la aplicación de estos operadores para resolver un problema utilizando DFD:



Fuente: Sistematts

 **Actividad 14**

*Elabore un diagrama de flujo que permita ingresar un número entre 1 y 12 para identificar un mes del año. Determine el máximo de días que trae dicho mes (para febrero se considerará sólo 28 días).*

## UNIDAD 4

### 4 CONTADORES, ACUMULADORES Y ESTRUCTURAS REPETITIVAS

#### 4.1 CONTADORES Y ACUMULADORES

##### 4.1.1 Definición

Herrera et al. (2017) definen contador y acumulador de la siguiente manera:

Contador	Acumulador
Es una variable cuyo valor se incrementa o decrementa en una cantidad fija cada vez que se ejecuta un proceso (generalmente dentro de una estructura repetitiva). Por lo tanto, un contador puede definirse de la siguiente manera:	Es una variable que almacena un valor como resultado de operaciones aritméticas sucesivas (sumas, restas, multiplicaciones, divisiones). Es similar a un contador, pero el incremento o decremento no es una cantidad fija sino una cantidad variable:
Creciente: $C = C + 1$	$AC = AC + X$
Decreciente: $D = D - 5$	$ACUM = ACUM - Y$



Una variable contadora o acumuladora debe tener un valor de inicio (generalmente cero) antes de empezar a realizar su trabajo, sin embargo, este valor inicial dependerá de la operación aritmética involucrada. Por ejemplo, un acumulador para multiplicaciones sucesivas deberá iniciar en 1:

$$AC = 1$$

⋮

$$AC = AC * X$$



Fuente: EducacionBA.



### **Actividad 15**

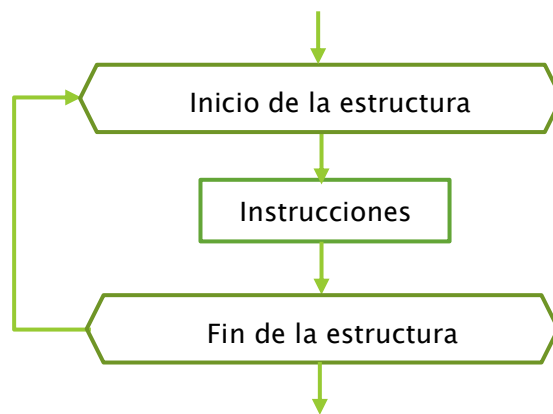
*Determine en qué caso debería utilizar un contador o un acumulador para la siguiente situación:*

Se atiende a varios clientes en un restaurante y al finalizar la jornada se debe informar cuántos clientes cancelaron mediante efectivo y cuántos mediante transferencia electrónica, así como el monto recaudado por cada forma de pago.

## **4.2 ESTRUCTURAS REPETITIVAS**

### **4.2.1 Definición**

Las estructuras repetitivas permiten definir lo que se reconoce también como bucles, ciclos, lazos o iteraciones dentro de un algoritmo. Estas estructuras se emplean para ejecutar un conjunto de instrucciones varias veces de acuerdo con una condición que permitirá iniciar, iterar o salir del bucle según la estructura repetitiva que se utilice.



*Ilustración 8. Esquema general de una estructura repetitiva*



Por **instrucciones** nos referimos a lecturas, procesos, salidas, condiciones e incluso otras estructuras repetitivas. Esto último daría lugar a estructuras repetitivas anidadas.

## 4.2.2 Tipos de estructuras repetitivas

Tabla 11. Estructuras repetitivas

FOR – Para	WHILE – Mientras
Se utiliza cuando podemos determinar la cantidad exacta de iteraciones que se van a realizar. Por esto también se lo denomina “contador automático”.	Evalúa una condición antes de entrar al ciclo. Se utiliza cuando no se puede determinar la cantidad exacta de iteraciones que se van a realizar y por tanto será requerida una condición que determine la continuidad o finalización del ciclo.
<b>FOR</b> C = Inicio, Fin, Incremento	<b>WHILE</b> <condición>
<p>Donde:</p> <p><b>C</b> =&gt; variable contadora o de recorrido.</p> <p><b>Inicio</b> =&gt; valor inicial asignado a la variable contadora</p> <p><b>Fin</b> =&gt; valor que debe alcanzar la variable contadora y que determinará el final del ciclo.</p> <p><b>Incremento</b> =&gt; valor que se aumentará o disminuirá en la variable contadora con cada iteración.</p>	<p>Donde <b>condición</b> puede ser cualquier comparación o grupo de comparaciones unidas por operadores lógicos. Mientras condición sea <i>Verdadera</i>, entrará a ejecutar todas las instrucciones hasta llegar al final del bucle y regresará a evaluar la condición. Cuando la condición sea <i>Falsa</i>, entonces saldrá del bucle.</p>
<p>Por ejemplo, para el procesamiento de 5 elementos indicamos:</p> <p style="text-align: center;"><b>FOR C = 1, 5, +1</b></p> <p>Que se interpreta como: para la variable <b>C</b> contar desde 1 hasta 5 aumentando 1 en cada iteración.</p>	<p>Por ejemplo, para el procesamiento de cinco elementos indicamos:</p> <p style="text-align: center;"><b>C = 1</b></p> <p style="text-align: center;"><b>WHILE C &lt;= 5</b></p> <p style="text-align: center;">⋮</p> <p style="text-align: center;"><b>C = C + 1</b></p>



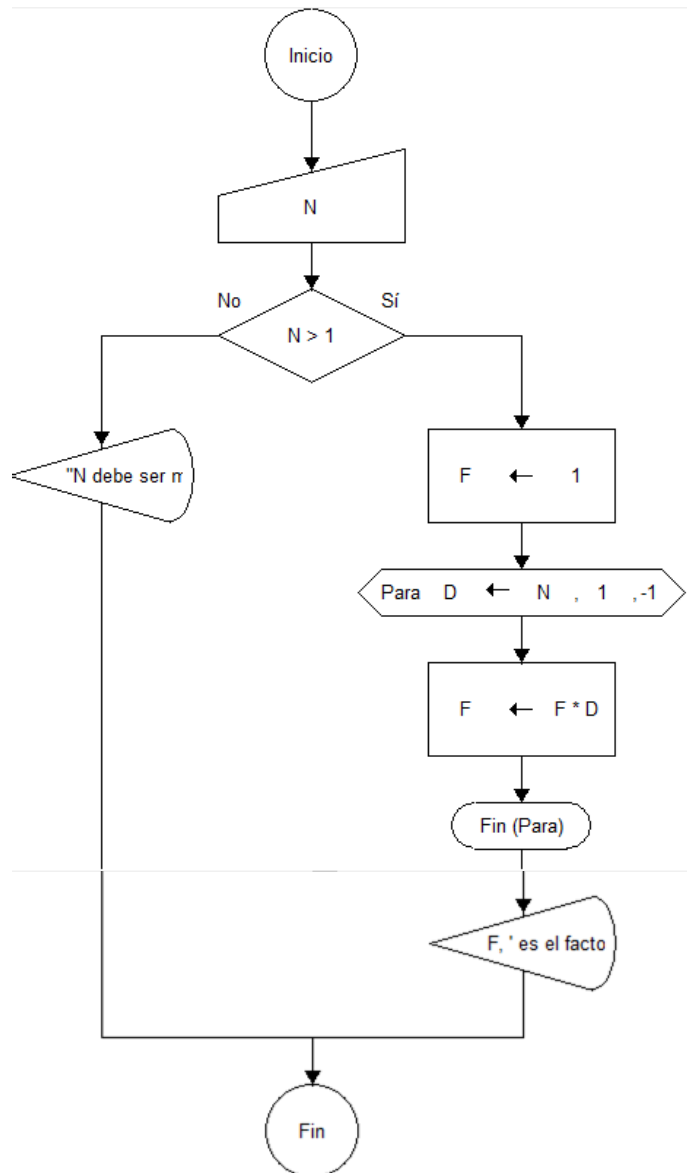
Fuente: Programación ATS

## 4.2.3 Usando la estructura repetitiva FOR

Ejercicio 1. Calcular el factorial de un número entero mayor a 1.


Ejemplo:  $5! = 5 * 4 * 3 * 2 * 1 = 120$

ENTRADAS	PROCESOS	SALIDAS
N => número entero mayor a 1	<p>Iniciar acumulador <b>F = 1</b></p> <p><i>Contar</i> para <b>D</b> desde <b>N</b> hasta 1 disminuyendo en 1</p> <p style="padding-left: 40px;">Calcular <b>F = F * D</b></p> <p><i>Fin-Contar D</i></p>	<p>N debe ser mayor a 1</p> <p>F es el factorial de N</p>



**Primera prueba de ejecución:**

Entrada de valores por teclado




Valor a ingresar:

1

---

Salida por pantalla




Salida:

N debe ser mayor a 1

**Segunda prueba de ejecución:**

Entrada de valores por teclado




Valor a ingresar:

5

---

Salida por pantalla



Salida:

120 es el factorial de 5

N	F	D	(Salidas)
1			N debe ser mayor a 1
5	<del>1</del> <del>5</del> <del>20</del> <del>60</del> 120	<del>5</del> <del>4</del> <del>3</del> <del>2</del> 1	120 es el factorial de 5



### **Actividad 16**

*Realice tres pruebas de escritorio para el flujograma anterior con otros números diferentes de los utilizados en la prueba de escritorio previa.*



**Fuente:** Programación ATS

#### **Ejercicio 2:**

Se solicita procesar N estudiantes (este valor no puede ser menor a 1). Por cada estudiante se debe ingresar los siguientes datos: Nombre, Asignatura, y 3 calificaciones trimestrales entre 0 a 20 puntos (se asumirá que cada calificación se ingresa en el intervalo señalado por lo que no será necesario validarlas).

El ingreso a supletorio dependerá de la suma de las tres calificaciones ingresadas:

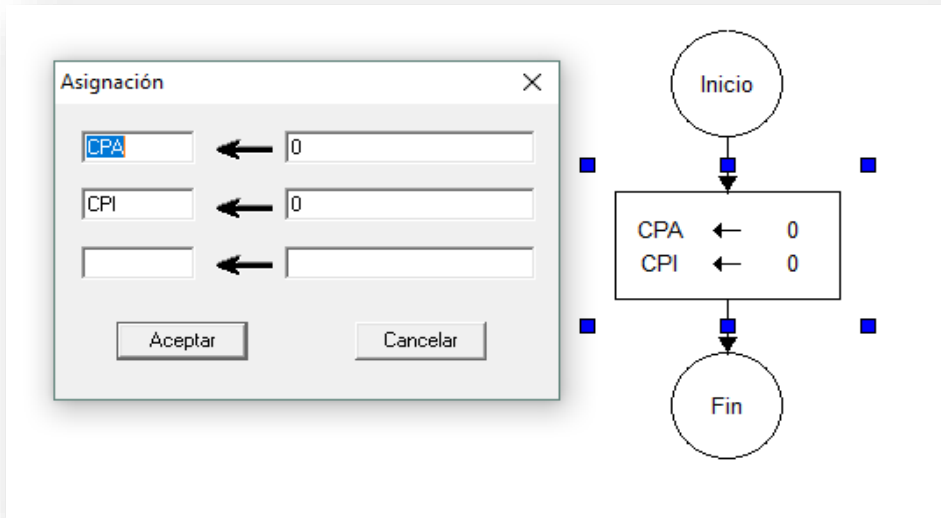
- Si la suma es mayor o igual a 40, el estudiante aprobará la asignatura de manera directa.
- Si la suma es menor a 10, reprobará inmediatamente.
- Si no se cumplen las dos condiciones anteriores, se ingresará una nueva calificación y si dicha calificación es menor a 18, el estudiante reprobará la asignatura, pero si es mayor o igual a 18, el estudiante habrá aprobado finalmente.

Como resultado final se nos solicita informar cuántos estudiantes aprobaron y cuántos reprobaron.

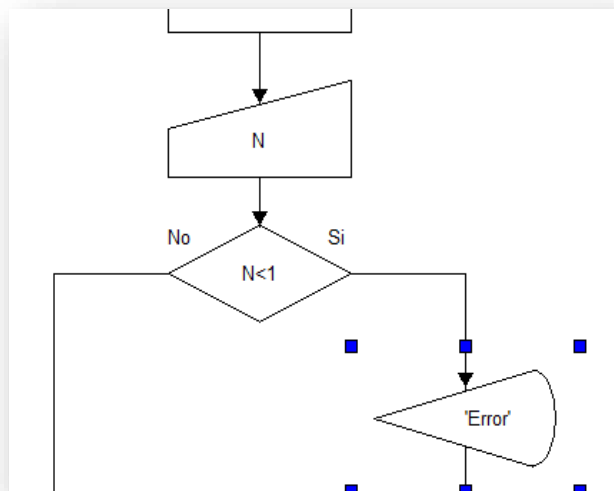
#### **Resolución:**

En este punto debemos recordar que siempre será conveniente tomarnos un tiempo para determinar las entradas, procesos y salidas antes de comenzar la construcción del flujograma. Con la finalidad de resolverlo de manera explicativa y directamente en DFD, iremos detallando los elementos relevantes que conforman la propuesta de solución. Mucha atención porque realizaremos una práctica con este ejercicio:

1. SALIDAS: El ejercicio nos solicita informar cuántos estudiantes aprueban y cuántos reprobaban, por lo que utilizaremos dos contadores: CPA (contador para aprobados) y CPI (contador para reprobados). Ambos deberán iniciar con cero.

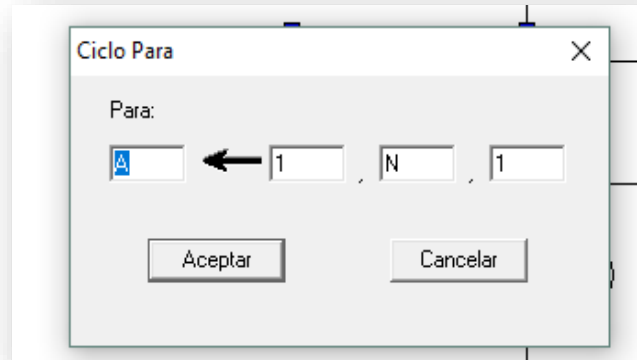


- Al revisar el planteamiento, nos percatamos que se van a procesar N estudiantes, por lo que debemos de verificar que este valor no sea menor a uno:

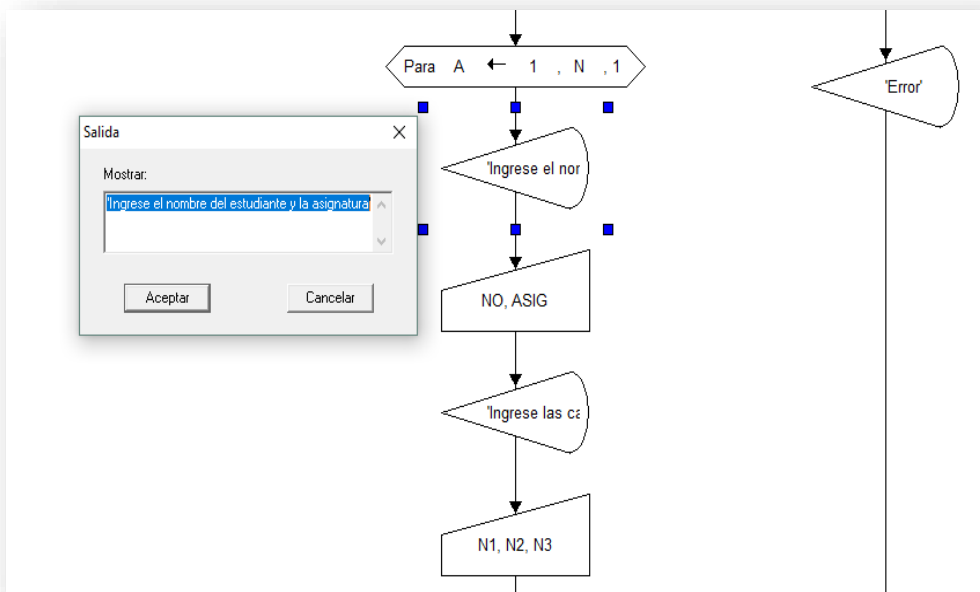


- A continuación, debemos considerar que una estructura repetitiva nos ayudará a reproducir el proceso para cada estudiante, por lo que definiremos una estructura repetitiva FOR indicando que recorrerá desde 1 hasta N con incremento de uno en cada iteración:

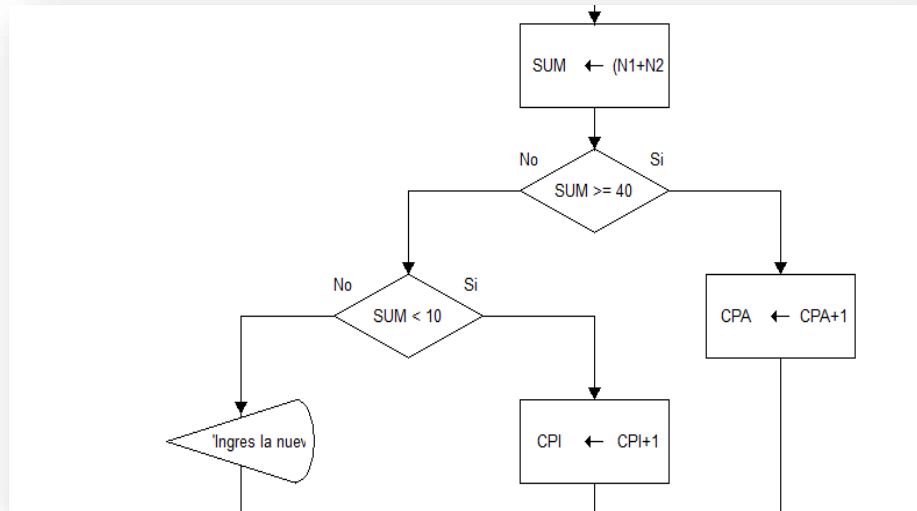




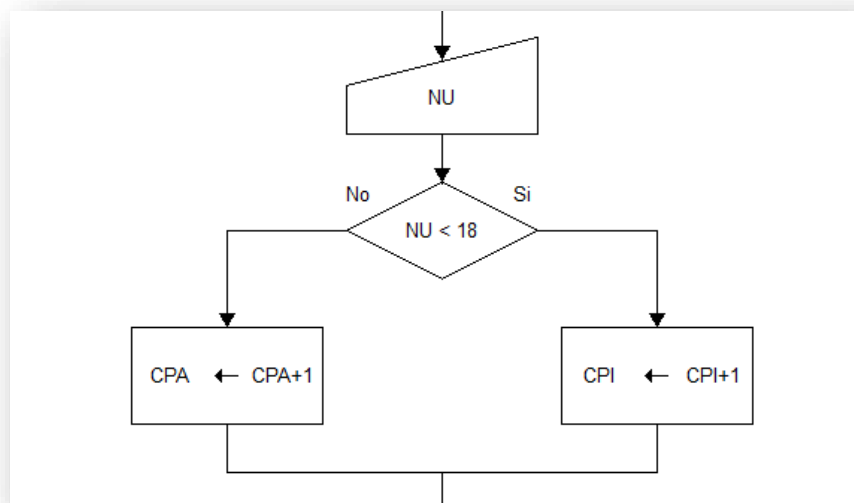
4. Recordemos que el planteamiento nos indica que es necesario ingresar Nombre, Asignatura y las calificaciones considerando que éstas últimas se digitarán con el intervalo apropiado (de 0 a 20). Podemos mostrar mensajes indicando lo que se debe ingresar tal como se define a continuación:



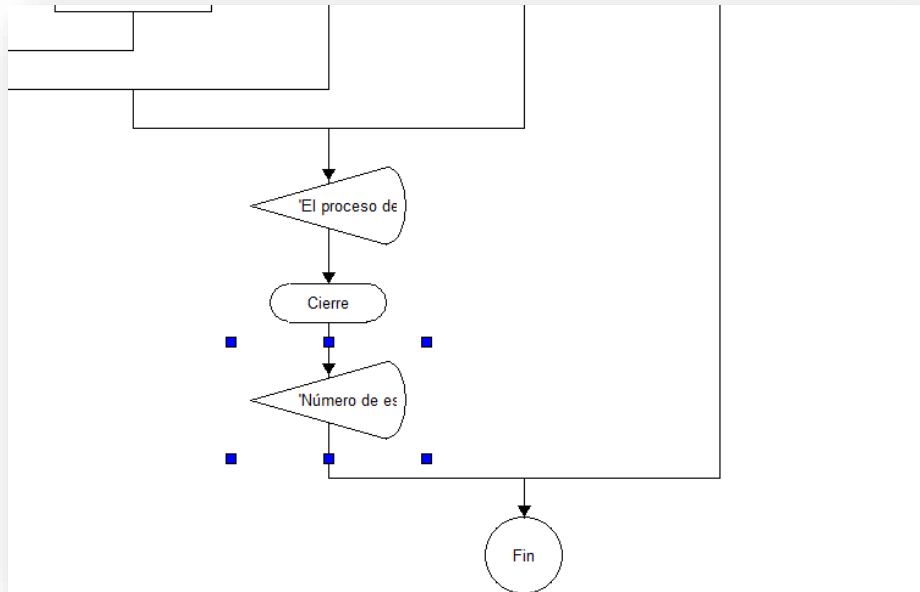
5. Ahora calcularemos la suma de las calificaciones ingresadas y continuaremos el procesamiento acorde a las condiciones establecidas: si la suma es igual o mayor a 40, el estudiante aprobará directamente, por lo que activamos el contador CPA para que cuente a cualquier estudiante que cumpla con esta condición, caso contrario, se preguntará si la suma (guardada en la variable SUM) es menor a 10 y de ser así, se activará el contador CPI tal como nos demuestra la siguiente imagen:



6. Tal como nos indica el ejercicio, de no cumplirse las condiciones anteriores, procederemos a ingresar una nueva calificación que determinará, en última instancia, si el estudiante aprueba o reprueba la asignatura:



7. Y por último, justo antes de terminar el ciclo FOR, podremos colocar un mensaje en el que se indique que el registro del estudiante se ha completado. Con esto facilitamos la operatividad y lograremos que nuestro “proceso computable” sea mucho más amigable para el usuario. Posterior al cierre del ciclo FOR, y antes de finalizar el flujograma, debemos mostrar el resultado de los contadores para cumplir con los informes requeridos al final del planteamiento del ejercicio:



### Actividad 17

- En la herramienta digital de su preferencia, reproduzca el flujograma completo de este último ejercicio y realice una prueba de escritorio para procesar cinco estudiantes.
- Elabore un flujograma que permita ir mostrando los  $N$  primeros múltiplos de cualquier número ingresado en el intervalo de 2 a 10 (validar esta entrada).

## UNIDAD 5

### 5 COMPOSICIÓN Y COMPROBACIÓN DE ALGORITMOS

#### 5.1 USO DEL ENTORNO INTEGRADO DE DESARROLLO

##### 5.1.1 Generalidades

Cualquier problema que utilice la algoritmia computacional como método de solución, podrá traducir su algoritmo a un lenguaje de programación para ser comprobado y utilizado en un dispositivo de cómputo. Esta traducción se conoce como “codificación” y consiste en escribir las instrucciones del algoritmo respetando las “reglas gramaticales” propias del lenguaje de programación a utilizar.



A pesar de ser muy sencillas y utilizar gráficos, nuestras herramientas de modelado computacional se aproximan mucho a los lenguajes de programación porque, además de aplicar los fundamentos de la algoritmia, también nos permiten comprobar nuestros modelos aplicando símbolos y reglas básicas como elementos gramaticales.

##### 5.1.2 Variables y tipos de datos

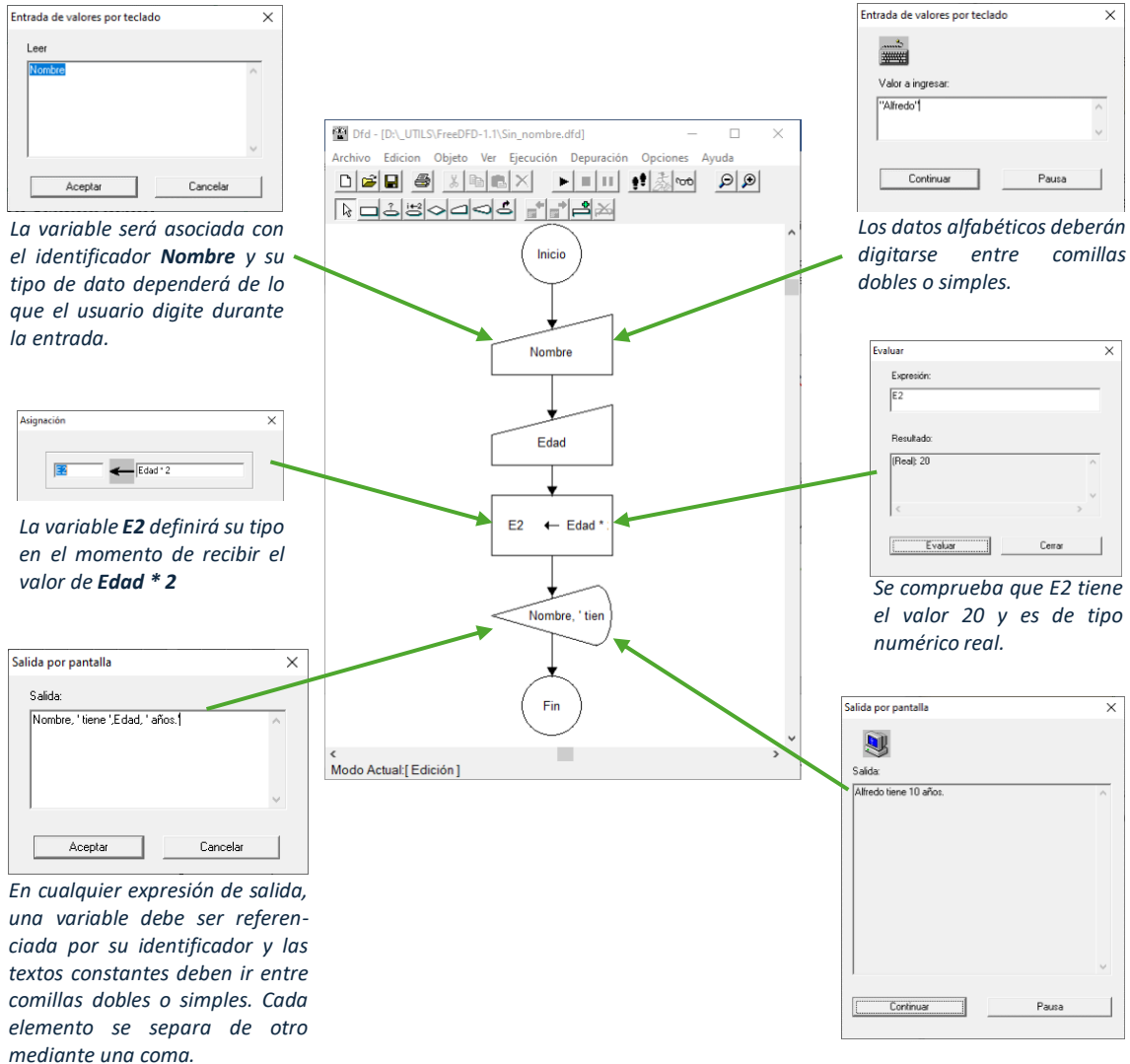
En cualquier lenguaje de programación de alto nivel, una variable es un espacio de memoria que se reserva para almacenar un dato de un tipo específico (numérico, alfabético, lógico, ...) que se asocia de manera exclusiva a un identificador cuyo “nombre descriptivo” generalmente deberá empezar con una letra.



En DFD y varios lenguajes de programación, el tipo de dato de una variable se define en el momento de invocar a dicha variable, ya sea por asignación directa o durante una operación de entrada. Esta característica se conoce como **tipado dinámico**.

**Consideraciones durante  
durante  
el modo Edición**

**Consideraciones  
el modo Ejecución**



*Ilustración 9. Variables y tipos de datos en DFD según el modo de trabajo*

### 5.1.3 Opciones especiales para la comprobación de un algoritmo

Comprobar una secuencia de instrucciones con un camino simple no representa mayor dificultad. Sin embargo, cuando el flujo se bifurca o se repite, las variables pueden cambiar su contenido, influyendo en los resultados de nuestra solución computacional. Esta situación nos exige comprobar que nuestro algoritmo resuelve el problema o depurar los errores.



Un buen Entorno Integrado de Desarrollo (IDE en inglés), para cualquier lenguaje de programación, deberá incluir alguna opción que permita verificar el contenido de las variables conforme se va ejecutando de manera paulatina cada instrucción del “guion de código fuente” (también conocido como *script*).

En el caso de nuestras herramientas, y particularmente en DFD, disponemos de las opciones *Paso simple* y *Depurador* que se pueden combinar para comprobar nuestro flujograma paso a paso, tal como demuestra el siguiente vídeo:



Fuente: HUGOVICTORTC



#### **Actividad 18**

*Comprobar cualquier diagrama de flujo que utilice decisiones con ayuda de las opciones Paso simple (Paso a Paso) y Depurador (Vista de variables). ¿Qué ventajas encuentra en la utilización de estas opciones?*

## 5.2 ANÁLISIS ORIENTADO A MODELAR UN PROBLEMA COMPUTABLE

### 5.2.1 Recomendaciones generales para analizar y resolver un problema

El desarrollo del pensamiento lógico computacional empleado en el modelado y resolución de problemas de este tipo requiere persistencia y práctica. Ante todo, la fase de análisis determinará el éxito o fracaso de cualquier solución a proponer.

De manera general, se recomiendan las siguientes pautas:

1. Leer el problema tantas veces como sea necesario procurando identificar las salidas, ya que estas determinan el objetivo.
2. Determinar los procesos o condiciones necesarias para obtener cada una de las salidas que plantea el problema.
3. De ser posible, trace un gráfico con los elementos que propone el ejercicio o cualquier otro elemento no mencionado que sirva para obtener alguna de las salidas, como una fórmula o algún dato especial que deba ser considerado.
4. Mientras menos variables se utilicen, más sencilla será la solución y su comprobación. Se recomienda reutilizar una misma variable tantas veces como sea posible.
5. Establecer las entradas que son requeridas para que el proceso pueda ejecutarse. Generalmente, a cada entrada deberá asignar una variable diferente.
6. Asigne “datos reales” a las variables de entrada para ejemplificar los casos a ser evaluados. Es muy útil llenar una tabla con estos datos y su correspondiente salida para reutilizarlos en la fase de comprobación. Considerar aquellos datos o casos que generan error para validarlos convenientemente.
7. Al diseñar decisiones que validan intervalos numéricos (desde – hasta) considerar que los operadores seleccionados y el orden en que se comprueben dichos intervalos incidirá en la complejidad de la solución propuesta, así como en la cantidad de condiciones a ser evaluadas. No siempre un bloque de decisiones extenso garantiza un resultado correcto.
8. Finalmente, comprobar que nuestra propuesta de solución responde a todas las salidas requeridas por el problema o casos a ser verificados. Esto implica realizar las pruebas de escritorio que sean necesarias para garantizar el recorrido por cada alternativa de flujo establecida en el diagrama.



Fuente: Programación ATS



#### **Actividad 19**

- a. *En la herramienta de su preferencia, reproducir el ejercicio del vídeo anterior y comprobar su funcionamiento utilizando las opciones de Paso Simple y Depurador.*
- b. *¿Qué condiciones alternativas propondría Ud. para obtener decisiones más simples sin afectar el resultado obtenido en dicho ejercicio?*

### 5.2.2 Ejemplo de análisis y propuesta de solución a problemas que involucran decisiones

Determinar el valor de comisión que se cancelará a un vendedor a partir de las ventas de un producto A y un producto B bajo las siguientes consideraciones:

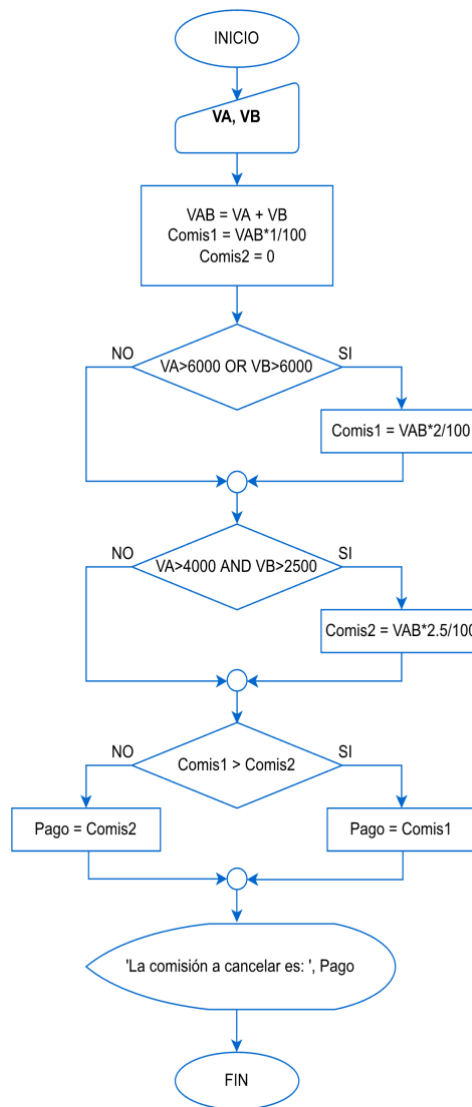
- Si las ventas de cualquiera de los productos supera los \$6000, comisionará el 2% de la venta total de ambos productos.
- Si las ventas de A superan los \$4000 y las ventas de B superan los \$2500, comisionará el 2,5% del monto total vendido.
- De cumplirse ambas condiciones previamente indicadas, se deberá cancelar el valor más favorable al vendedor.
- De no cumplirse alguna de las condiciones indicadas, el vendedor comisionará el 1% del total vendido.

Análisis preliminar:

Entradas	Procesos	Salidas
<p><b>VA =&gt;</b> Ventas del producto A</p> <p><b>VB =&gt;</b> Ventas del producto B</p>	<p>Todas las comisiones se calculan a partir de las ventas totales de ambos productos, por tanto, conviene mantener la suma de ambos valores en un campo de memoria:</p> $\mathbf{VAB = VA + VB}$ <p>La comisión más baja (1%) siempre será una alternativa para el pago, lo que permite establecer el caso predeterminado:</p> $\mathbf{Comisión1 = VAB * 1/100}$ <p>Los otros dos casos dependen de las condiciones establecidas en el planteamiento y conviene que sean evaluadas de manera independiente. Por este motivo las estructuramos en secuencia:</p> $\mathbf{¿VA > 6000 \text{ OR } VB > 6000? \Rightarrow VAB * 2/100}$ $\mathbf{¿VA > 4000 \text{ AND } VB > 2500? \Rightarrow VAB * 2,5/100}$ <p>Cualquiera de estas alternativas calculará un valor más alto que el primer caso, sin embargo, se deberá evaluar cuál de estas dos fórmulas nos devuelve un valor mayor para cumplir con la salida requerida en el planteamiento. Esto nos obliga a guardar en memoria ambos valores para poder compararlos antes de entregar el resultado final.</p>	<p>Valor de comisión más favorable para el vendedor.</p>

A continuación, el flujograma que implementa la propuesta de solución con su correspondiente prueba de escritorio:





**Prueba de escritorio**

VA	VB	VAB	Comis1	Comis2	Pago	(Salida)
4000	2500	6500	65	0	65	La comisión a cancelar es: 65
4001	2501	6502	65.02	162.55	162.55	La comisión a cancelar es: 162.55
6001	2500	8501	170.02	0	170.02	La comisión a cancelar es: 170.02
6001	2501	8502	170.04	212.55	212.55	La comisión a cancelar es: 212.55
7000	2500	9500	190	0	190	La comisión a cancelar es: 190
6999	2501	9500	190	237.5	237.5	La comisión a cancelar es: 237.5



### **Actividad 20**

- a. *Utilizando la herramienta de su preferencia, reproduzca el flujograma anterior intercambiando la ubicación de las dos primeras estructuras condicionales. Replique la prueba de escritorio propuesta utilizando Depurador y Paso simple.*
- b. *¿Hubo algún cambio en los resultados obtenidos? ¿Por qué motivo considera que se produjo o no el cambio en los resultados?*
- c. *¿Qué criterio considera Ud. que se ha tomado en cuenta para seleccionar los datos utilizados en la prueba de escritorio?*

*“Some people think that if they can write the best code, they will change the world. But you have to be able to listen. You have to be able to communicate with your future users and learn from them. **And you have to be able to communicate your ideas to them**”. Bjarne Stroustrup, creator of the C++ programming language.*

## 6 Bibliografía

- Beynon–Davies, P. (2018). *Sistemas de Información : Introducción a la Informática en Las Organizaciones*. Editorial Reverté ProQuest Ebook Central.  
<https://www.proquest.com/legacydocview/EBC/6795627?accountid=151317>
- Herrera Morales, J., Gutiérrez Posada, J. y Pulgarín Giraldo, R. (2017). *Introducción a la Lógica de Programación*. EIZCOM S.A.S.  
<https://uleam.sharepoint.com/:b:/s/79a2a5ba4/EYJfAlik3ZxCj7iqINAsIK4B9ID55FWntBYcjWD9oAoHlw?e=uKOnVU>
- Ordoñez, Luis. (18 de Junio de 2019). *OpenWebinars*.  
<https://es.calameo.com/read/0025477318a412d351788>
- Pinales Delgado, F.J. y Velázquez Amador C.E. (2014). *Algoritmos resueltos con diagramas de flujo y pseudocódigo*. Universidad Autónoma de Aguas Calientes.  
<https://issuu.com/editorialuaa/docs/algoritmos>
- Publicaciones Empresariales UNAM. (2016). *Tecnologías de Información y Comunicación en las Organizaciones*. Publicaciones Empresariales UNAM FCA Publishing.

ISBN: 978-9942-681-17-1



9789942681171



**Uleam**  
UNIVERSIDAD LAICA  
ELOY ALFARO DE MANABÍ