

Universidad Nacional de Santiago del Estero
Facultad de Ciencias Exactas y Tecnologías

INGRESO

LIC. EN SISTEMAS DE INFORMACIÓN (LSI)

PROFESORADO EN INFORMÁTICA (PI)

INTRODUCCIÓN A LA INFORMÁTICA

AÑO 2016

1. PROBLEMAS DE COMPUTACION

1.1 Objetivo

La resolución de problemas utilizando como herramienta una computadora requiere contar con la capacidad de expresión suficiente como para indicar a la máquina lo que debe llevar a cabo.

Se comenzara resolviendo situaciones del mundo real tratando de utilizar determinados elementos que caracterizan a una secuencia de ordenes que una computadora puede comprender.

El tema central de este curso es la definición del concepto de algoritmo y los elementos que lo componen.

1.2 Introducción

La **Informática** es la ciencia que estudia el análisis y resolución de problemas utilizando computadoras.

La palabra ciencia se relaciona con una metodología fundamentada y racional para el estudio y resolución de los problemas.

Si se busca en el diccionario una definición de la palabra problema podrá hallarse alguna de las siguientes:

- Cuestión o proposición dudosa, que se trata de aclarar o resolver.
- Enunciado encaminado a averiguar el modo de obtener un resultado cuando se conocen ciertos datos.

La resolución de problemas mediante una computadora consiste en dar una adecuada formulación de pasos precisos a seguir.

1.3. Resolución de Problemas

Si se piensa en la forma en que una persona indica a otra como resolver un problema, se vería que habitualmente se utiliza un lenguaje común y corriente para realizar la explicación, quizá entremezclado con algunas palabras técnicas. Esto es un riesgo muy grande. Los que tienen cierta experiencia al respecto saben que es difícil transmitir el mensaje y por desgracia, con mucha frecuencia se malinterpretan las instrucciones y por lo tanto se "ejecuta incorrectamente" la solución, obteniéndose errores.

Cuando de una computadora se trata, no pueden utilizarse indicaciones ambiguas. Ante cada orden resulta fundamental tener una única interpretación de lo que hay que realizar. Una máquina no posee la capacidad de decisión del ser humano para resolver situaciones no previstas.

Si al dar una orden a la computadora se produce una situación no contemplada, será necesario abortar esa tarea y recomenzar todo el procedimiento nuevamente.

Además, para poder indicar a la computadora las ordenes que debe realizar es necesario previamente entender exactamente lo que se quiere hacer. Es fundamental conocer con qué información se cuenta y qué tipo de transformación se quiere hacer sobre ella.

Una vez que se comprende un problema, se debe decidir que tipo de problema es. Dos tipos de problemas comunes son:

- Los **problemas que buscan respuestas**: si un ejercicio implica el cálculo del número de palabras que hay en un libro o la bisección de una línea utilizando regla y compás, se trata de un trabajo en que se debe encontrar algo que se desconoce. La forma en que esto se haga no es de particular importancia siempre y cuando se obtenga la respuesta correcta (aunque es deseable que sea por un medio fácil).
- Los **problemas que buscan pruebas**: cuando se pide que se pruebe que hay 720 formas de colocar seis libros en una fila, se está dando la respuesta. La tarea es distinta a la de encontrar una respuesta porque ya se sabe lo que se desea. Todo lo que se tiene que hacer es determinar la relación entre los datos y la respuesta.

Es importante observar que en un problema en que se buscan respuestas se tiene que elaborar una solución, mientras que cuando se pide que se pruebe sólo es necesario demostrar que existe (o no existe) una solución, sin crearla. Los problemas de computación no pueden ser problemas en que se busquen pruebas porque el propósito del trabajo de las computadoras es encontrar respuestas que no se conocen de antemano. Pero tampoco se les puede considerar como problemas en que se busquen respuestas porque es la computadora misma la que determina la respuesta, no la persona que trabaja con ella.

Los problemas de computación pertenecen a una tercera clase: los **problemas que buscan métodos**, aquí se busca un método mediante el cual se pueda derivar una respuesta.

El proceso de resolver problemas mediante computadoras se describe en la figura 1. Se trata de encontrar un método por medio del cual se pueda resolver un problema. Una vez que se haga esto, la computadora se hace cargo del mismo y suministra las respuestas a la pregunta.



Figura 1.

Esta es una simplificación porque una vez que se tiene un método es necesario expresar este método en una forma en que la computadora pueda operarlo (un programa).

Una forma de expresar un método de resolución de problemas es el siguiente:

Etapa 1: **Identificar** el problema, dando respuesta a **Que resultado** se debe obtener, **Que datos** están disponibles, y **Que condiciones o restricciones** se deben tener en cuenta para la resolución.

Etapa 2: **Definir y representar** el modo de resolver el problema.

Etapa 3: **Explorar** las posibles estrategias para solucionar el problema y seleccionar la más conveniente.

Etapa 4: **Probar** la estrategia de solución seleccionada, evaluando los efectos y comprobar si resuelve el problema.

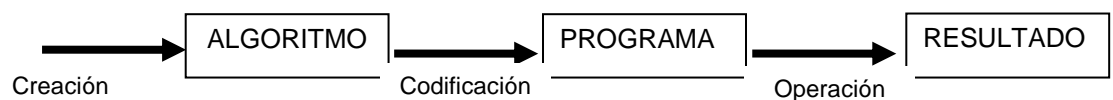
Etapa 5: **Implementar** la solución.

Como se dijo anteriormente, el proceso de resolver problemas mediante computadores, consiste en encontrar un método por medio del cual se pueda resolver un problema. Una vez encontrado (diseñado) el método, el computador se encarga de ejecutarlo y suministrar la respuesta/ solución al problema planteado.

El proceso así presentado, está muy simplificado, ya que una vez obtenido el método (o resuelto como hacerlo) es necesario expresarlo en una forma que pueda ser operado/ejecutado por el computador. Por tanto el profesional informático deberá realizar tres trabajos: **Creación** del algoritmo; **Codificación** del algoritmo creado y **Operación** del algoritmo en el computador.

En este momento surgen nuevos conceptos: ALGORITMOS, CODIFICACIÓN y OPERACIÓN. Estos tres conceptos ordenados en secuencia constituyen precisamente el Método para resolver problemas mediante computadoras. A continuación se explican cada unos de ellos.

1. **Creación** del algoritmo: significa crear un método de resolución de un problema determinado. Como resultado de este paso se obtiene un Algoritmo (Es el único Paso que veremos en este curso)
2. **Codificación** del algoritmo creado: una vez obtenido el algoritmo, debe ser escrito en un lenguaje de programación, de manera que pueda ser interpretado por el computador. Como resultado de este paso se obtiene un programa.
3. **Operación** del algoritmo en el computador, implica la ejecución del programa en el computador para obtener el **resultado** o **solución** del problema.



2. ALGORITMOS

2.1. Introducción

La etapa vital de la solución de un problema con una computadora es el diseño del algoritmo y de la estructura fundamental de datos. Un algoritmo es un procedimiento expresado precisamente para obtener la solución del problema, la que se presenta de manera subsecuente a una computadora en el lenguaje de programación seleccionado. Los algoritmos se presentan de una manera conveniente para un lector humano, mientras que los programas sirven a las necesidades de las computadoras.

Es importante recordar mientras diseñamos un algoritmo que una computadora sólo sigue las instrucciones y no puede actuar si no se le ha ordenado de manera explícita. Por lo tanto, el solucionador de problemas debe prever cualquier aspecto del problema en el propio algoritmo.

La palabra **algoritmo** deriva del nombre de un matemático árabe del siglo IX, llamado Alkhuwarizmi, quien estaba interesado en resolver ciertos problemas de aritmética y describió varios métodos para resolverlos. Estos métodos fueron presentados como una lista de instrucciones específicas (como una receta de cocina) y su nombre se utiliza para referirse a dichos métodos.

2.2. Definición

Un algoritmo es, en forma intuitiva, una receta, un conjunto de instrucciones o de especificaciones sobre un proceso para hacer algo. Ese “algo” generalmente es la solución de un problema de algún tipo. Formalmente un algoritmo se puede definir de la siguiente forma:

Un algoritmo puede definirse como una secuencia ordenada de pasos elementales, exenta de ambigüedades, que lleva a la solución de un problema dado en un tiempo finito.

Para comprender la definición anterior se clarifica algunos conceptos.

Ejemplo: escriba un algoritmo que permita preparar una tortilla de papas de tres huevos.

Si la persona que resuelva el problema es un cocinero lo resuelve sin mayor nivel de detalle, pero si no lo es, se deben describir los pasos necesarios para realizarlo:

Paso 1: Mezclar papas fritas, huevos y una pizca de sal en un recipiente.

Paso 2: Freir.

Esto podría resolver el problema, pero si la persona que lo ejecute no sabe cocinar, se debe detallar, cada uno de los pasos mencionados, pues estos no son lo bastante simples para un principiante. De esta manera el primer paso se puede descomponer en:

Paso 1: Pelar las papas

Paso 2: Cortar las papas en cuadraditos

Paso 3: Freír las papas

Paso 4: Batir los huevos en un recipiente

Paso 5: Verter las papas en un recipiente y echar una pizca de sal

El tercer paso puede descomponerse en:

- Calentar el aceite en la sartén
- Verter las papas en la sartén
- Sacar las papas ya doradas en un recipiente

Nótese que si la tortilla la va a ejecutar un niño, alguna tareas, por ejemplo batir huevos, pueden necesitar una mejor especificación.

Con este ejemplo se pretende mostrar que la lista de **pasos elementales** que compongan nuestro algoritmo depende de quien sea el encargado de ejecutarlo. Si en particular, el problema va a ser resuelto utilizando una computadora, el conjunto de pasos elementales conocidos es muy reducido, lo que implica un alto grado de detalle para los algoritmos.

Se considera entonces como un paso elemental aquel que no puede volver a ser dividido en otros más simples.

Otro aspecto importante de un algoritmo es su nivel de detalle, que no debe confundirse con el concepto de paso elemental. En ocasiones, no se trata de descomponer una orden en acciones más simples sino que se busca analizar cuáles son las órdenes relevantes para el problema. Para comprender lo expuesto se lo analiza con un ejemplo:

Ejemplo: escriba un algoritmo que describa la manera en que Ud. se levanta todas las mañanas para ir al trabajo.

Paso 1: Salir de la cama

Paso 2: Quitarse el pijama

Paso 3: Ducharse

Paso 4: Vestirse

Paso 5: Desayunar

Paso 6: Arrancar el auto para ir al trabajo

La solución del problema se expresó en seis pasos, descartando aspectos que se suponen o sobreentienden, como “colocarse los zapatos al vestirse” o “abrir la puerta del auto” pues a nadie se le ocurriría ir a trabajar descalzo. En cambio existen aspectos que no pueden obviarse o suponerse porque el

algoritmo perdería lógica, por ejemplo el paso “vestirse”, no puede ser omitido. Puede discutirse si se requiere un mayor nivel de detalle o no, pero no puede ser eliminado del algoritmo.

Un buen desarrollador de algoritmos deberá reconocer esos aspectos importantes y tratar de simplificar al mínimo su especificación de manera de seguir resolviendo el problema con la menor cantidad de órdenes posibles.

Además, en la definición de algoritmo se hace referencia a la ambigüedad y tiempo de respuesta, debido a que todo algoritmo debe cumplir con ciertas propiedades para que se lo considere como tal y proporcione el resultado deseado cuando un programa basado en él se presenta a una computadora.

Un algoritmo debe cumplir las siguientes propiedades:

Ausencia de Ambigüedad: si se trabaja dentro de cierto marco o contexto, la representación de cada paso de un algoritmo debe tener una única interpretación.

Ejemplo: indique la forma de condimentar una salsa.

Incorrecto: ponerle algunas especies a la salsa.

Correcto: ponerle sal, pimienta y orégano a la salsa.

Generalidad: un algoritmo se puede realizar para varios problemas que se relacionan entre si, es decir, se debe aplicar a un problema o clase de problemas específicos.

Ejemplo: indique la forma de marcar un número de teléfono.

Incorrecto: si la solución del algoritmo sirve para marcar solamente el número 4220234, solo tendrá valor para ese número.

Correcto: si la solución es un método para marcar cualquier número, entonces es útil. Por supuesto, debe haber alguna restricción a la generalidad de un algoritmo.

Tiempo de respuesta: la ejecución de un algoritmo debe finalizar después de que se haya llevado a cabo una cantidad finita de pasos. De otra manera, no se puede exigir que la ejecución produzca una solución.

Ejemplo: Llene la zanja con ese montón de arena

Algoritmo: tome una pala y empiece a echar arena en la zanja. Cuando se llene la zanja deténgase.

(se está seguro que en algún momento parará, aunque no se sabe cuanto tardará).

2.3. Dominio de un Algoritmo

La clase o el conjunto de datos y las condiciones para las cuales un algoritmo trabaja correctamente se llama dominio. Cuando se trata de resolver cualquier problema es necesario definir el dominio del algoritmo y después verificar que trabaja para todos los casos que se encuentran dentro de ese dominio.

Al decidir el dominio de un algoritmo es necesario incluir todas las situaciones similares, pero los casos remotos o poco probables se pueden omitir.

2.4. Errores en la Construcción de un Algoritmo

En la construcción de algoritmos se consideran dos tipos de errores:

- **Errores de Dominio:** se presentan cuando no se han especificado todas las situaciones que se pueden presentar en la práctica o se ha descuidado la apreciación de su importancia. Las pruebas más difíciles son aquellas que verifican que se ha seleccionado un dominio correcto para el algoritmo. Cuando una situación no prevista se presenta, hay tres opciones:
 - Ignorarla porque es improbable y quizás nunca ocurra.
 - Restringir el dominio del algoritmo para excluirla.
 - Corregir el algoritmo.

- **Errores de Lógica:** son aquellos errores que se detectan después de que se ha definido en forma adecuada el dominio de un algoritmo, en la etapa de prueba o verificación. se deben principalmente a las siguientes causas:
 - Etapas incorrectas
 - Secuencia incorrecta de etapas.

3. FORMAS DE EXPRESAR UN ALGORITMO

Un mismo algoritmo puede ser expresado de distintas formas.

- **Lenguaje común.** En el lenguaje normal que hablamos y escribimos; útil para comunicar un algoritmo a otra persona o en una fase de análisis previo de un sistema computacional.
- **Diagramas de flujo.** Es un lenguaje gráfico; útil para visualizar en forma rápida la secuencia lógica de pasos a seguir para un algoritmo y de gran ayuda para la traducción del mismo a un programa de computación
- **Pseudocódigo:** Es una técnica para expresar en lenguaje natural la lógica de un programa, es decir, su flujo de control. El pseudocódigo no es un lenguaje de programación sino un modo de plantear un proceso de forma que su traducción a un lenguaje de alto nivel sea sencillo para el programador.
- **Lenguajes de Programación:** Es la forma obligada de expresión de un algoritmo para que pueda ser leído, ejecutado y almacenado por el computador.

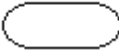

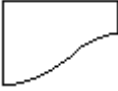

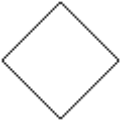

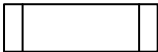

3.1. Diagramas de Flujo

Un diagrama de flujo es la representación gráfica o visual de un algoritmo. Se usan en el planeamiento, desarrollo y estructuración de un algoritmo. Mediante los diagramas de flujo el algoritmo se puede comunicar y documentar (porque enseña y describe el proceso).

Formalmente, un diagrama de flujo es un diagrama formado por símbolos (cajas, bloques, figuras) y flechas o líneas de flujo que conectan los símbolos entre si. Los símbolos denotan los pasos esenciales del algoritmo y las flechas indican la secuencia. Se dibujan de tal manera que la dirección del flujo sea hacia abajo o de izquierda a derecha.

3.2. Simbología Básica de los Diagramas de Flujo

Los símbolos que se describen a continuación, para la representación gráfica de los diagramas de flujo, son de uso universal.

SÍMBOLO	SIGNIFICADO
	Indica principio o fin de un algoritmo.
	Indica entrada de datos.
	Indica la salida de la información.
	Se usa generalmente para sentencias o enunciados de asignación (acción de asignar) y para la realización de un proceso matemático.
	Permite evaluar una expresión relacional ó lógica que puede tomar un valor verdadero o falso. En función de este resultado el flujo del algoritmo seguirá una determinada dirección.
	Se usa cuando el diagrama es largo y se requiere más de una hoja de papel o para evitar líneas que se crucen.
	Simbolizan módulos o segmentos lógicos
	Indica la dirección del algoritmo en cada momento mediante una flecha.

3.3. Utilidad de los Diagramas de Flujo

El diagrama de flujo refleja los pasos sucesivos que el computador debe dar para llegar a la solución de un problema. Las principales razones por las cuales es generalmente aconsejable el trazado de un diagrama de flujo son las siguientes:

- Oportunidad de verificar la lógica de la solución.
- Sirve de guía al programador para la codificación del programa.
- Permite fácilmente modificar un programa.
- Es útil para la discusión grupal.
 - Sirve para documentar el programa.

ACTIVIDAD Nº 1: ALGORITMOS

1- Dados los enunciados de los siguientes problemas y su correspondiente algoritmo de solución, decir si las soluciones propuestas cumplen con las propiedades que debe tener un algoritmo; en el caso de no hacerlo indicar cuales son los errores.

a) Problema: Abordar un taxi

1. Situarse al borde de la acera de cara al lado apuesto de la calle.
2. Buscar la señal de taxi.
3. Al ver la señal, silbar y hacer ademanes para llamar su atención.
4. Fin

b) Problema: Cambiar la rueda pinchada de un automóvil teniendo un gato mecánico, una rueda de reemplazo y una llave inglesa.

1. Aflojar los tornillos de la rueda pinchada con la llave inglesa
2. Ubicar el gato mecánico en su sitio
3. Levantar el gato hasta que la rueda pinchada pueda girar
4. Quitar los tornillos y la rueda pinchada.
5. Poner la rueda de repuesto y los tornillos.
6. Fin

c) Problema: Preparar una tortilla de papas de tres huevos.

1. Pelar las papas
2. Cortarlas en cuadraditos
3. Freír las papas
4. Batir los huevos en un recipiente
5. Verter las papas en un recip. y echar una pizca de sal
6. Freír la tortilla
7. Fin

d) Problema: Levantarse a las mañanas para ir al trabajo.

1. Apagar el despertador
2. Salir de la cama
3. Quitarse el pijama
4. Higienizarse
5. Vestirse
6. Si no hay tiempo, tomar el portafolio / la cartera e ir al paso 9
7. Desayunar
8. Buscar el portafolio o la cartera y e ir al paso 7
9. Salir de la casa
10. Fin

e) Problema: Cambiar un foco quemado de una lámpara en el techo, teniendo un foco nuevo

1. Buscar la escalera
2. Colocar la escalera debajo de la lámpara quemada
3. Elegir un foco nuevo similar al anterior
4. Subir por la escalera
5. Girar el foco quemado en sentido contrario al de las agujas del reloj
6. Ubicar el nuevo foco en el mismo lugar
7. Enroscar en el sentido de las agujas del reloj hasta que quede apretado
8. Bajar de la escalera
9. Fin

f) Problema: Obtener una determinada página de un libro.

1. Abrir el libro en la pagina 1.
2. Si la página que está examinando es la página 15 deténgase.
3. En caso contrario, observe la siguiente, pasando una hoja, si es necesario.
4. Repetir los pasos 2 y 3 tan a menudo como se requiera
5. Fin

2- Escriba la secuencia lógica para los siguientes algoritmos de solución

Problema: Preparar una taza de café con leche. Considerar que se cuenta con todos los elementos necesarios

1. Buscar la taza
2. Calentar la leche
3. Encender la hornalla
4. Endulzar
5. Agregar leche al café
6. Hacer el café
7. Fin

Problema: "Puesta en escena de una obra de teatro"

1. Redacción del guión.
2. Ensayo final con el vestuario terminado.
3. Iniciación del programa de publicidad.
4. Estreno de la obra en el teatro de la ciudad.
5. Terminación de los ensayos sin vestuario.
6. Selección de los actores.
7. Crítica de la obra en los periódicos.
8. Iniciación de los ensayos sin vestuario
9. Terminación de la confección del vestuario
10. Terminación de la escenografía
11. Venta de boletos

Problema: " Lanzar a la venta un nuevo producto"

1. Pruebas de control de calidad
2. Diseño de publicidad
3. Investigación de mercado
4. Diseño de especificaciones
5. Estimación de costos
6. Empaquetado y al mercado
7. Diseño final
8. Producción y Ensamblado
9. Determinación del sistema de distribución
10. Diseño de efecto en el consumidor
11. Análisis costo - beneficio
12. Determinación del precio
13. Lanzamiento de la propaganda

4. ESTRUCTURAS DE CONTROL BÁSICAS

Al ser un algoritmo una secuencia de pasos ordenados, estos deben seguir una trayectoria para su ejecución desde el primer paso hasta el último. Esta trayectoria se denomina **flujo de control** que indica el orden en el cual deben ejecutarse los pasos elementales.

Para organizar el flujo de control de un algoritmo se utilizan **estructuras de control**, estas son construcciones algorítmicas lineales, de selección e iteración. Las dos últimas alteran el flujo de control lineal del algoritmo.

Las estructuras de control básicas para organizar el flujo de control en un algoritmo, son las siguientes:

- Estructura secuencial
- Estructura de selección
- Estructura de iteración

4.1. Estructura Secuencial

La estructura de control más simple está representada por una sucesión de acciones que se ejecutan de arriba hacia abajo sin bifurcaciones, es decir, una acción a continuación de otra.

Ejemplo: escriba un algoritmo que describa la forma en que una persona se levanta todas las mañanas para ir al trabajo.

Paso 1: Salir de la cama

Paso 2: Quitarse el pijama

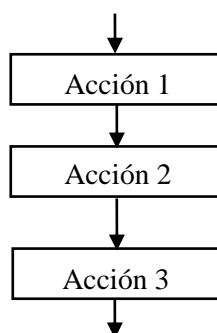
Paso 3: Ducharse

Paso 4: Vestirse

Paso 5: Desayunar

Paso 6: Arrancar el auto para ir al trabajo

A continuación se presenta gráficamente esta estructura.



4.2. Estructura de Selección

En un algoritmo representativo de un problema real es prácticamente imposible que las instrucciones sean secuenciales puras. Es necesario tomar decisiones en función de los datos del problema.

A través de la selección se incorpora, a la especificación del algoritmo, la capacidad de decisión. De esta forma será posible seleccionar una de **dos alternativas** de acción posibles durante la ejecución del algoritmo.

La selección se expresa con el siguiente pseudocódigo:

Si (condición)

entonces

acción o acciones a realizar si la condición es verdadera (1)

sino

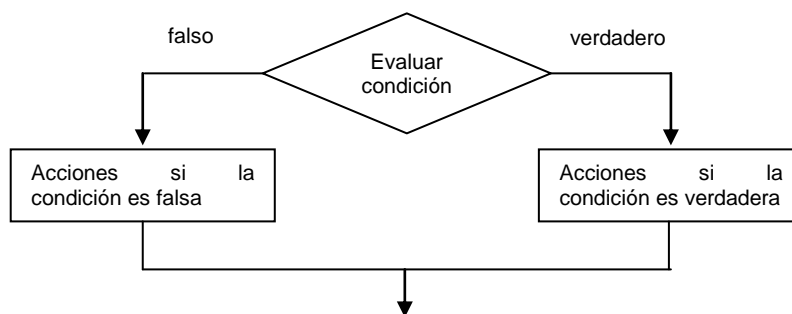
acción o acciones a realizar si la condición es verdadera (2)

FinSi

donde **condición** es una expresión que al ser evaluada puede tomar solamente uno de los dos valores posibles: verdadero o falso.

En el caso que la condición a evaluar resulte verdadera se ejecutarán las acciones (1) y **no** se ejecutarán las acciones (2). Si la condición a evaluar resulta falsa se ejecutarán las acciones (2) y **no** las acciones (1).

A continuación se presenta gráficamente esta estructura.



Puede ocurrir que no se tengan que representar acciones cuando la condición es falsa. En este caso se utilizará la siguiente notación:

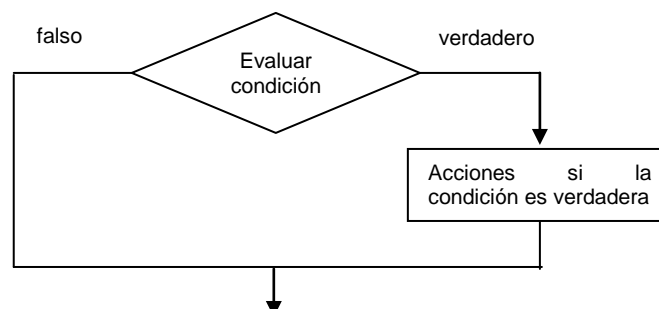
Si (condición)

entonces

acción o acciones a realizar si la condición es verdadera

FinSi

A continuación se presenta gráficamente esta estructura.



Si al evaluar un **decisión** toma más de dos valores, se utilizará la siguiente notación:

Si (variable de decisión)

Valor 1: Acción 1

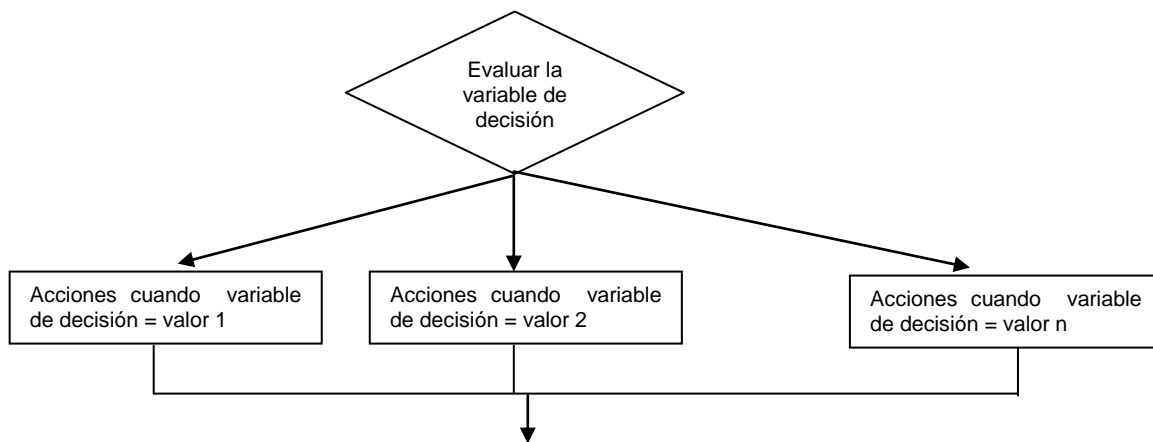
Valor 2: Acción 2

.....

Valor N: Acción N

[Otro: Acción N+1]

A continuación se presenta gráficamente esta estructura.



ACTIVIDAD Nº 2: ALGORITMOS Y ESTRUCTURA DE CONTROL

1- Dados los enunciados de los siguientes problemas y su correspondiente algoritmo de solución, utilizando estructuras de control de selección, indicar los datos que necesita utilizar, los resultados a obtener y si se llega al resultado deseado.

a) Problema Un amigo le ha pedido que compre en el kiosco \$2 de caramelos de menta. Si no hay, debe comprar \$1 de cualquier tipo de caramelos.

- Solución 1**
1. Ubicarse en el kiosco
 2. **Si** (tiene caramelos de menta)
Entonces
 3. Comprar \$2 de caramelos menta
 4. **Sino**
Comprar \$1 de otro tipo de caramelos
 - 5 **FinSi**
 - 6 Fin

- Solución 2**
1. Ubicarse en el kiosco
 2. **Si** (tiene caramelos de menta)
Entonces
 3. Comprar \$2 de caramelos menta
 4. **Fin Si**
 5. Comprar \$1 de otro tipo de caramelos
 6. Fin

b) Hacer una llamada telefónica desde un teléfono público a moneda. Considere que se cuenta con la moneda y siempre se encuentra un teléfono.

Solución 1

1. Obtener una moneda
2. Encontrar un teléfono
3. **Si** (el teléfono tiene tono)
Entonces
 4. Marcar el número
 5. Hablar
6. **Sino**
Buscar otro teléfono
7. **FinSi**
8. Fin

Solución 2

1. Obtener una moneda
2. Encontrar un teléfono
3. **Si** (no tiene tono el teléfono)
Entonces
 4. Buscar otro teléfono
5. **FinSi**
6. Marcar el número
7. Hablar
8. Fin

c) Se busca escribir un algoritmo para colgar un cuadro.

Solución 1

1. Clavar el clavo
2. **Si** (no quedo firme el clavo)
Entonces
 3. Golpear el clavo un poco más
4. **FinSi**
5. Colgar el cuadro
6. Fin

Solución 2

1. Clavar el clavo
2. **Si** (no quedo firme el clavo)
Entonces
 3. Golpear una vez más
4. **Sino**
Colgar el cuadro
5. **FinSi**
6. Fin

d) Obtener la página X de un libro

1. Tomar el libro
2. Obtener el N° de la pagina deseada y llamarla X
3. Abrir el libro en la pagina 1
4. **Si** (el N° de pagina es igual a X)
Entonces
 5. Vaya al paso 8
6. **Sino**
Pasar una hoja si es necesario
7. **FinSi**
8. Fin

2- Escriba los algoritmos de solución para los siguientes problemas.

- a) Preparar un té. Considere que si no dispone de un saquito de té debe preparar un mate cocido, y que de seguro existe el saquito de mate cocido. Tenga en cuenta que la preparación de las dos infusiones tienen muchos pasos en común.
- b) Una persona desea ingresar a un recital a beneficio. Para obtener la entrada al mismo, deberá presentarse en la ventanilla del club con un bolsón de pañales o bien con una caja de leche. En el caso de que la persona no lleve los pañales o la leche, podrá comprar la entrada en la ventanilla de dicho club.
- c) Juan programó jugar al básquet con sus amigos el jueves de la semana que viene, en caso de no poder hacerlo irá a visitar a su novia. El partido se realizará en la cancha del parque Aguirre si no llueve, mientras que si esta lloviendo a la hora programada, jugarán en el estadio techado del colegio siempre y cuando no este ocupado.
- d) Una persona va por la ruta y se le pincha una rueda, para cambiar la misma deberá contar con tres elementos (gato mecánico, rueda de auxilio y llave inglesa), los cuales no sabe si se encuentran en el baúl del auto. Si no posee alguno de estos elementos necesarios llamará a la grúa de auxilio.

4.3. Estructura de Iteración

Una extensión natural de una estructura secuencial consiste en repetir N veces un bloque de acciones. El fin de la repetición dependerá de un valor predefinido o del cumplimiento de una determinada condición.

Existen dos formas de expresar esta estructura la *Repetición* y la *Iteración* en este curso solo utilizaremos la iteración.

Iteración

Puede ocurrir que se desee ejecutar un conjunto de acciones de un algoritmo desconociendo el número exacto de veces que se ejecutan. Para estos casos existen estructuras de control iterativas condicionales, es decir, las acciones se ejecutan dependiendo de la evaluación de una condición.

Por lo tanto, dentro de una estructura iterativa, además de una serie de pasos elementales que se repiten; es necesario contar con un mecanismo que lo detenga.

Podemos definir una estructura iterativa como la estructura de control que permite al algoritmo ejecutar en forma repetitiva un conjunto de acciones utilizando una condición para indicar su finalización.

Estas estructuras se clasifican en **pre-condicionales y pos-condicionales**.

Las estructuras pre-condicionales evalúan la condición y, si es verdadera, se ejecuta el conjunto de acciones; esto hace que dicho conjunto se puede ejecutar 0, 1 o más veces.

La notación para esta estructura es la siguiente:

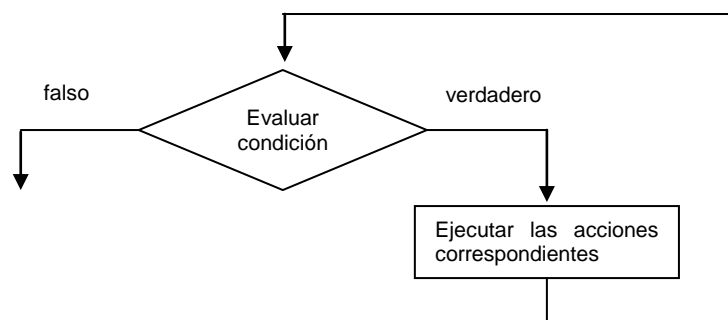
Mientras (condición)

Acción o acciones a realizar en caso de que la condición sea verdadera.

FinMientras

La condición es una expresión que sólo puede tener uno de dos valores posibles: verdadero o falso.

A continuación se presenta gráficamente esta estructura.



Las estructuras pos-condicionales, primero se ejecuta el conjunto de acciones, luego se evalúa la condición y, si es falsa, se ejecuta nuevamente el bloque de acciones. A diferencia de la estructura anterior iterativa anterior, el conjunto de acciones se debe ejecutar 1 o más veces. Nótese que, en este caso, el bloque de acción se ejecuta antes de evaluar la condición, por lo tanto se lleva a cabo al menos una vez.

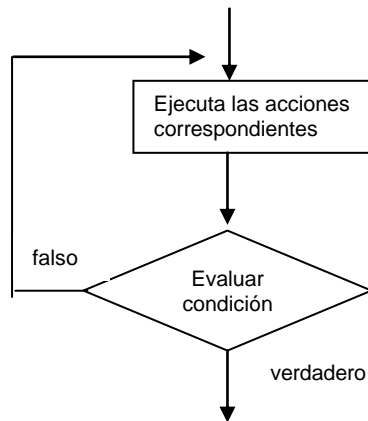
La notación a utilizar para esta estructura es la siguiente:

Repetir

Acción o acciones a realizar en caso de que la condición sea falsa

Hasta (condición)

A continuación se presenta gráficamente esta estructura.



ACTIVIDAD Nº 3: ALGORITMOS y ESTRUCTURA DE CONTROL

1- Dados los enunciados de los siguientes problemas y su correspondiente algoritmo de solución, en lo que se utilizan estructuras de selección y de iteración, analizar cada una de las soluciones propuestas e indicar, datos de entrada, de salida y si se obtiene con los mismos el resultado deseado.

a) Se busca escribir un algoritmo para colgar un cuadro.

Solución 1

- 1 Clavar el clavo
- 2 **Si** (no quedo firme el clavo)
- Entonces**
- 3 Golpear el clavo un poco más
4. **FinSi**
- 4 Colgar el cuadro
- 5 Fin

Solución 2

- 1 Clavar el clavo
- 2 **Mientras** (no quede firme el clavo)
3. Golpear el clavo una vez más
4. **Fin Mientras**
- 5 Colgar el cuadro
- 6 Fin

b) Se necesita hornear una pizza en un horno defectuoso. El horno se apaga cada cierto tiempo y debe ser encendido nuevamente. Como condición sabemos que tenemos una cantidad ilimitada de fósforos y dispone de la prepizza y todos los elementos necesarios

Solución 1

- 1 Encender el horno
- 2 Poner la pizza en el horno
- 3 **Si** (el horno se apagó)
- Entonces**
- 4 Encender nuevamente el horno
5. **FinSi**
- 6 **Si** (no está lista la pizza)
- Entonces**
- 7 Esperar 1 minuto
8. **FinSi**
- 9 Apagar el horno
- 10 Sacar la pizza
- 11 Fin

Solución 2

- 1 Encender el horno
- 2 Poner la pizza en el horno
- 3 **Mientras** (no este lista la pizza)
- 4 Esperar 1 minuto
- 5 **Si** (el horno se apagó)
- Entonces**
- 6 Encender nuevamente el horno
7. **FinSi**
8. **FinMientras**
- 9 Apagar el horno
- 10 Sacar la pizza
- 11 Fin

c) El Sr. Lombardi ha recibido su pago en un único cheque. Para cobrarlo deberá presentarse en el banco con su DNI, y este le pagará siempre y cuando la cuenta tenga fondos.

Solución 1:

- 1 Entrar al banco.
- 2 Colocarse en la caja correspondiente
- 3 **Mientras** (no le llega el turno)
- 4 Esperar 1 minuto
- 5 **FinMientras**
- 6 **Si** (tiene el cheque)
 Entonces
- 7 **Si** (tiene el DNI)
 Entonces
 Cobrar
- 8
9. **FinSi**
 Sino
 Maldecir
- 10
- 11 **FinSi**
- 12 Salir del banco
- 13 Fin

Solución 2:

- 1 Entrar al banco.
- 2 Colocarse en la caja correspondiente
- 3 **Repetir**
- 4 Esperar 1 minuto
- 5 **Hasta** (que le llegue su turno)
- 6 **Si** (tiene el cheque)
 Entonces
- 7 **Si** (tiene el DNI)
 Entonces
 Presentar el cheque al cajero
- 8
- 9 **Si** (tiene fondos en la cuenta)
 Entonces
 Cobrar
- 10
- 11 **Sino**
 Maldecir
- 12 **FinSi**
- 13 **FinSi**
- 14 **FinSi**
- 15 Salir del banco
- 16 Fin

d) Juan se encuentra sintonizando la radio para escuchar el programa "R & P". Luego de escuchar el programa, debe apagar la radio. Considere que el programa esta siendo actualmente transmitido.

Solución 1

- 1 Encender la radio.
- 2 Sintonizar una estación de radio
- 3 **Si** (están transmitiendo "R & P")
 Entonces
- 4 Escuchar el programa
- 5 **FinSi**
- 6 Apagar la radio.
- 7 Fin

Solución 2:

- 1 Encender la radio.
- 2 **Repetir**
- 3 Sintonizar una estación de radio
- 4 **Hasta** (esta transmitiendo "R & P")
- 5 Escuchar el programa
- 6 Apagar la radio.
- 7 Fin

Solución 3:

- 1 Encender la radio.
- 2 **Mientras** (no estén transmitiendo "R & P")
- 3 Sintonizar una estación de radio
- 4 **FinMientras**
- 5 Escuchar el programa.
- 6 Apagar la radio.
- 7 Fin

e) Con el fin de mejorar el arbolado de la Av. Belgrano, la municipalidad desea plantar 300 ligustros con una distancia de 1 metro entre ellos, . Para realizar dicha tarea resulta necesario descargar el ligustro, cavar un pozo y plantarlo.

Solución 1

- 1 Detener el Camión
- 2 **Repetir**
- 3 Descargar el ligustro
- 4 Cavar un pozo
- 5 Plantar el Ligustro
- 6 Contar que se ha plantado un ligustro más
- 7 **Hasta** (Se han plantado 300 ligustros)
- 8 Encender el Camión
- 9 Fin

Solución 2:

- 1 Detener el Camión
- 2 **Mientras** (no se hayan plantado los 300 ligustros)
- 3 Cavar un pozo
- 4 Descargar el ligustro
- 5 Plantar el Ligustro
- 6 Avanza un metro
- 7 **FinMientras**
- 8 Encender el Camión
- 9 Fin

Solución 3:

- 1 Detener el Camión
- 2 **Mientras** (no se hayan plantado los 300 ligustros)
- 3 Cavar un pozo
- 4 Descargar el ligustro
- 5 Plantar el Ligustro
- 6 Avanzar un metro
- 7 Contar que se ha plantado un ligustro más
- 8 **FinMientras**
- 9 Encender el Camión
- 10 Fin

2- Escriba los algoritmos de solución para los siguientes problemas, teniendo en cuenta el uso de las estructuras de control correspondientes.

- a) Una pileta debe ser vaciada utilizando un balde. Considerando que no conoce cuántos baldes necesitará sacar para vaciar la pileta.
- b) Trasladar 70 cajas de 30 kilos cada una, desde la sala A a la Sala B. Considere que sólo llevará una caja a la vez porque el contenido es muy frágil. Para realizar el trabajo debe ponerse un traje especial y quitárselo luego de haber realizado el trabajo.
- c) Ud. desea ordenar una bolsa con 54 fotografías viejas de manera que todas queden al derecho; esto es, con la imagen hacia Ud. y cabeza arriba.
- d) Usted desea comprar la revista "Crucigramas" que cada mes tiene reservada en el puesto de revistas que se encuentra en la esquina de su casa, al otro lado de la calle. Verifique que no pasen autos antes de cruzar.

5. MANEJO DE DATOS

5.1. Datos Simples

Los programas que implementan los algoritmos necesitan alguna manera de representar los objetos del mundo real. Para ello los algoritmos operan sobre datos y estructuras de datos de distinta naturaleza, tales como números, letras, símbolos, etc.

Por lo tanto un **dato** es la expresión general que describe los objetos con los cuales opera una computadora. Los tipos de datos están ligados a un conjunto de operaciones que permiten su creación y manipulación.

Los tipos de datos se caracterizan por:

- Un rango de valores posibles
- Un conjunto de operaciones realizables sobre ese tipo
- Su representación

Los tipos de datos simples son:

- Numérico
- Lógico
- Carácter

5.1.1. Tipo de Dato Numérico

El **tipo de dato numérico** es el conjunto de los valores numéricos que pueden representarse de dos formas:

- Enteros
- Reales

Enteros

El tipo entero consiste de un subconjunto finito de los números enteros y su tamaño puede variar según el valor que tenga:

...-3, -2, -1, 0, 1, 2, 3...

Dado que una computadora tiene memoria finita, la cantidad de valores enteros que se pueden representar sobre ella son finitos, por esto se deduce que existe un número entero máximo y otro mínimo.

La cantidad de valores que se pueden representar depende de la cantidad de memoria (bits) que se utilicen para representar un entero.

Ejemplo:

Hay sistemas de representación numérica que utilizan 16 dígitos binarios (bits) para almacenar en memoria cada número entero, permitiendo un rango de valores enteros entre -215 y +215. Otros sistemas utilizan 32 bits, por lo que el rango es entre -231 y +231.

En general, para cada computadora existe el entero maxint , tal que todo número entero n puede ser representado si

$$-\text{maxint} \leq n \leq \text{maxint}$$

donde **maxint** identifica al número entero más grande que se puede representar con la cantidad de dígitos binarios disponibles.

Reales

El tipo de dato real es una clase de dato numérico que permite representar números decimales. Los valores fraccionarios forman una serie ordenada, desde un valor mínimo negativo, hasta un valor máximo determinado por la norma IEEE 754, de 1985, pero los valores no están distribuidos de manera uniforme en ese intervalo, como sucede con los enteros.

Se debe tener en cuenta que el tipo de dato real tiene una representación finita de los números reales; dicha representación tiene una precisión fija, es decir, un número fijo de dígitos significativos. Esta condición es la que establece una diferencia con la representación matemática de los números reales. En este caso se tienen infinitos números diferentes, en tanto que la cantidad de representaciones del tipo de dato real está limitada por el espacio en memoria disponible.

La representación para números reales se denomina **coma flotante**. Esta es una generalización de la conocida notación científica, que consiste en definir cada número como una mantisa (o fracción decimal), que da los dígitos contenidos en el número; y un exponente (o característica), que determina el lugar del punto decimal con respecto a estos dígitos.

Ejemplo:

Sea el número 8941295000000000. Su representación científica en cualquier calculadora es $8,941295 \times 10^{16}$.

La mantisa del número real representado en la computadora en este caso es : 0,8941295 y el exponente 17. Nótese que, a diferencia de la representación anterior, la coma se ubica inmediatamente a la izquierda del primer dígito significativo y, por lo tanto, la magnitud del exponente se incrementa en 1.

Si el número es 0,000000000356798, su representación utilizando notación científica es $3,56798 \times 10^{-11}$.

La mantisa es 0,356798 y el exponente es -10 .

Se observa que un exponente positivo lleva a desplazar la coma decimal tantos lugares hacia la derecha como lo indica su magnitud, mientras un exponente negativo implica un desplazamiento hacia la izquierda.

5.1.2. Tipo de Dato Carácter

Un tipo de dato carácter proporciona objetos de la clase de datos que contiene solo un elemento como su valor. Este conjunto de elementos está establecido y normalizado por un estándar llamado ASCII (American Standard Code for Information Interchange), el cual establece cuales son los elementos y el orden de precedencia entre los mismos. Los elementos son las letras, número y símbolos especiales disponibles en el teclado de la computadora y algunos otros elementos gráficos. Cabe acotar que el código ASCII no fue único, pero es el más utilizado internacionalmente.

Son elementos carácter:

- Letras minúsculas: 'a', 'b', 'c', ..., 'y', 'z'
- Letras mayúsculas: 'A', 'B', 'C', ..., 'Y', 'Z'.
- Dígitos: '0', '1', '2', '3', ..., '8', '9'.
- Caracteres especiales tales como: '!', '@', '#', '\$', '%', ...

Se debe tener en cuenta que no es lo mismo el valor entero 1 que el símbolo carácter '1'. Un valor del tipo de dato carácter es **solo uno** de los símbolos mencionados.

Operaciones sobre Datos Carácter

Los operadores relacionales descriptos en el tipo de dato numérico, pueden utilizarse también sobre los valores del tipo de dato carácter. Esto es, dos valores de tipo carácter se pueden comparar por =, <, >, <=, >=; el resultado de cualquiera de ellos es un valor de tipo de dato lógico.

La comparación de dos caracteres es posible dado que el código ASCII define para cada símbolo un valor en su escala. De esta manera, al comparar dos símbolos, para determinar el resultado se utiliza el valor dado por el código.

Dentro del código ASCII los valores de los dígitos son menores que los valores de las letras mayúsculas, y estos a su vez menores que los de las letras minúsculas. Los valores dentro del código para los símbolos especiales, o bien son menores que los dígitos, o bien mayores que las letras minúsculas.

Ejemplo:

(á = Á)	da como resultado <i>falso</i>
('c' < 'Z')	da como resultado <i>falso</i>
('c' < 'z')	da como resultado <i>verdadero</i>
('X' > '5')	da como resultado <i>verdadero</i>
(' ' < 'H')	da como resultado <i>verdadero</i>
('4' = 4)	observación: ' ' representa un espacio en blanco no puede evaluarse pues los operandos son de tipos distintos

5.1.3. Tipo de Dato String o Cadena de Caracteres.

En la mayoría de los lenguajes de programación existe un tipo estándar que es la cadena de caracteres(string).

De acuerdo a definiciones previas, un carácter es una representación determinada de una letra, número o símbolo que se guarda internamente de acuerdo a su representación determinada por el código ASCII. Cuando se trabaja con el tipo de dato string, se tienen n caracteres tratados como una única variable, donde n proviene de la definición del string.

Un tipo de dato string es una sucesión de caracteres que se almacenan en un área contigua de memoria y que puede ser leído o escrito.

El tipo de dato string debe indicar el tamaño máximo que se desea manejar, y se define como:

Type nombre-string = string [longitud]

Donde *nombre-string* es el identificador del tipo, y *longitud* es el número máximo de caracteres que puede contener.

Una vez definido el tipo se pueden declarar variables de ese tipo.

Ejemplos (en lenguaje Pascal)

```
Type cadena1 = string[10]
```

```
    cadena2 = string[25]
```

```
Var c1, c2: cadena1
```

```
    denominación: cadena2
```

Ejemplos de asignación en variables string

```
c1 = 'pepe'
```

```
c2 = '678@#$abc'
```

```
denominación = 'remera deportiva'
```

5.1.4. Tipo de Dato Lógico

El tipo de dato lógico, también llamado booleano, en honor al matemático británico George Boole (quien desarrolló el Álgebra lógica o de Boole), es un dato que puede tomar un valor entre un conjunto formado por dos posibles:

- Verdadero (true)
- Falso (false)

Se utiliza en casos donde se representan dos alternativas a una condición. Por ejemplo, si se debe determinar si un valor es primo; la respuesta será verdadera (true) si el número es divisible solamente por si mismo y la unidad; en caso contrario, si tiene algún otro divisor, la respuesta será falsa (false).

5.2. Constantes y Variables

Los algoritmos contienen ciertos valores que no deben cambiar durante la ejecución del mismo. Tales valores se llaman Constantes. De igual forma existen otros valores que cambian durante la ejecución del mismo, estas son las Variables.

Una **constante** es una partida de datos(objetos) que permanece sin cambios durante todo el desarrollo del algoritmo o durante la ejecución del programa.

Una **variable** es un objeto o partida de datos cuyo valor puede cambiar durante el desarrollo del algoritmo o ejecución del programa.

5.3. Expresión: Definición

“Una expresión es un conjunto de variables y/o constantes unidas por operadores”.

Dependiendo del tipo de operadores con que se este trabajando y el resultado obtenido podremos hablar de *Expresiones Aritméticas* o *Expresiones Lógicas*.

Operadores Aritméticos

+	Suma
-	Resta
*	Multiplicación
/	División
^	Potencia

Una Expresión Aritmética es aquella que cuando se la evalúa siempre se obtiene un resultado numérico.

Ejemplos de expresiones aritméticas

$$\begin{aligned}2 * 5 + 7 &= 17 \\ 3 * A / 4 + \text{Cont} \wedge 2 \\ \text{Suma} &= \text{Suma} + N\end{aligned}$$

Operadores Relacionales

<	Menor
>	Mayor
<=	Menor igual
> =	Mayor igual
=	Igual a
<>	Distinto a

Operadores Lógicos

Los operadores lógicos o boléanos básicos son:

- Negación (Not)
- Conjunción (And)
- Disyunción (Or)

El resultado de estas operaciones es el correspondiente a las conocidas tablas de verdad.

Los operadores relacionales y lógicos mencionados anteriormente, que permiten comparar dos valores, dan como resultado un valor lógico.

Ejemplo:

(8 < 4)	da como resultado <i>falso</i>
(2.5 * 4 = 10)	da como resultado <i>verdadero</i>
(8 > 3)	da como resultado <i>verdadero</i>
not (8 > 3)	da como resultado <i>falso</i>
(7 > 2.4), (9 = 3)	son expresiones <i>verdaderas</i> y <i>falsa</i> respectivamente
(7 > 2.4) and (9 = 3)	una expresión que da un resultado <i>falso</i>
(7 > 2.4) or (9 = 3)	es una expresión que da un resultado <i>verdadero</i>

Para cada operador lógico se define un símbolo, los cuales se muestran en la tabla 2.1.

Operación	Operador	Simbolización
Conjunción	Y / AND	^
Disyunción	O / OR	v
Negación	NO / NOT	~

Tabla 2.1. Conectores Lógicos

Tablas de verdad

Para poder analizar cualquier proposición compuesta y decir qué valor de verdad tiene, es usual hacerlo a través de lo que se conoce como tabla de verdad, la cual se define de la siguiente manera:

La tabla de verdad de una proposición es, como su nombre lo indica, una tabla donde se muestran todas las combinaciones posibles de los valores de verdad de dicha proposición.

Conjunción

Dadas dos proposiciones cualquiera p y q, la proposición molecular $p \wedge q$ representa la conjunción de p y q.

La conjunción de dos proposiciones es cierta únicamente en el caso en que ambas proposiciones lo sean.

p	q	p ^ q
V	V	V
V	F	F
F	V	F
F	F	F

Disyunción

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Negación

Dada una proposición p, su negación $\sim p$, permitirá obtener el valor de verdad opuesto.

El valor de verdad de la negación de una proposición verdadera es falso y el valor de verdad de la negación de una proposición falsa es verdadero.

p	$\sim p$
V	F
F	V

Ejemplos de Expresiones Lógicas

- $(8 < 4)$ da como resultado falso
- $(2.5 * 4 = 10)$ da como resultado verdadero
- $(8 > 3)$ da como resultado verdadero
- NOT $(8 > 3)$ da como resultado falso
- $(7 > 2.4)$ AND $(9=3)$ es una expresión que da un resultado falso
- $(7 > 2.4)$ OR $(9=3)$ es una expresión que da un resultado verdadero

Orden de Evaluación en la expresiones

Operador	Prioridad
NOT	Más alta (se evalúa primero)
*, /, AND	↓
+, -, OR	↓
<, >, <=, >=, =, <>	Más baja (se evalúa al último)
Si existen paréntesis, las expresiones de su interior se evalúan primero	


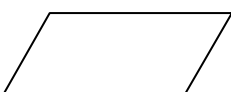
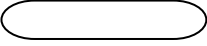
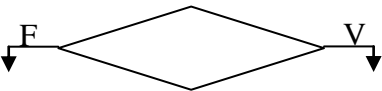


ACTIVIDAD N° 4: DATOS y EXPRESIONES

1. Dadas las siguientes expresiones, identifique variables, constantes, tipos de operadores, calcule el resultado e indique de que tipo es el mismo.

Asignación	Expresión	Variables Ctes, Operadores	Resultado	Tipo del resultado
A=1, B=5, C=0	$B - A * (B + C) * 2 + 1$		
	$A + C * 3 - 2 / 12$		
	$((C + 3 * (5 + 2) / 10) * A) \uparrow (-1)$		
	$(A + 10 / 5 * A - (B \uparrow 2)) * (-1)$		
A=1, B= -2, C=3	$A * B + (B - C) * 3 / (B - 3)$		
	$A = (A + (B - C) * 3)$		
	$A + B - C / 3$		
	$C > (A + (B - C) * 3)$			
	$(A - B - C / 3 + 2) \uparrow (-2)$		
N1=3, N2=6	$(2 + 3 * N1) <> 6 + (N2 - 1)$		
	$(1 + (2/2) * N2) < 4 * N1$		
	$2 - 6/2 * (N1 - N2) = 0$		
	$N2 >= N1 + (3 / N1)$		
R=2, S=6 T= verdadero U= falso	$S \uparrow 2 <= R + S$		
	$(S+S) * (-1) >= (R+1) \uparrow (-1)$		
	$T \text{ and } (R <= S) \text{ or } (\text{not } U)$		
	$R <= S / 3 \text{ and } T$		
	$\text{Not } U \text{ and } (S < 1) \text{ or } T$		
	$T \text{ and } (R <= 2)$		
	$T \text{ or } U \text{ and } (R < S)$		
	$(R * 2 > S - 3) \text{ or } (R > S - 1) \text{ and } (S < 5)$		

ACTIVIDAD Nº 5: ALGORITMOS Y DIAGRAMAS DE FLUJO

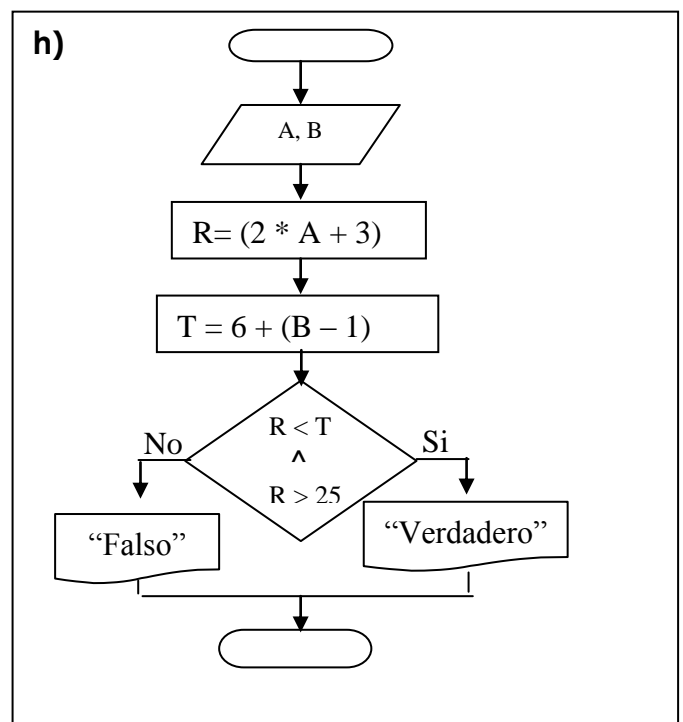
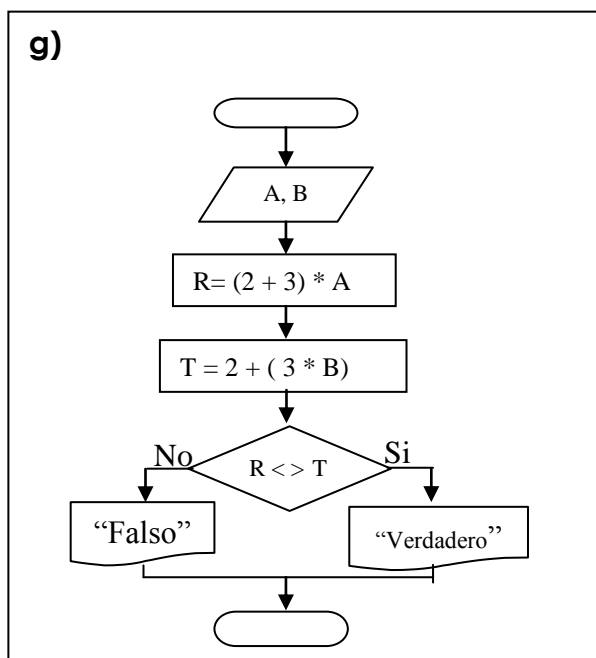
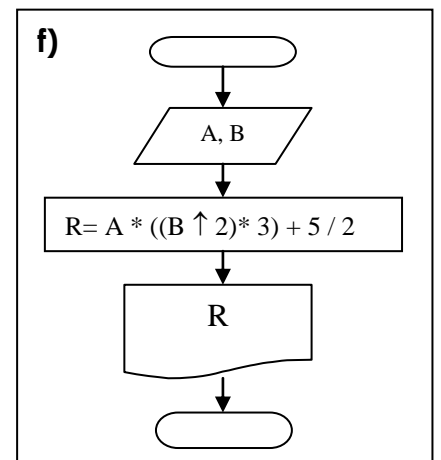
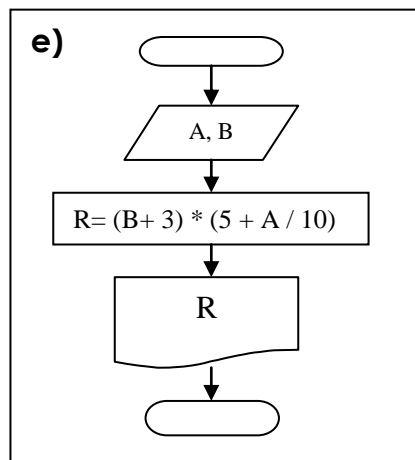
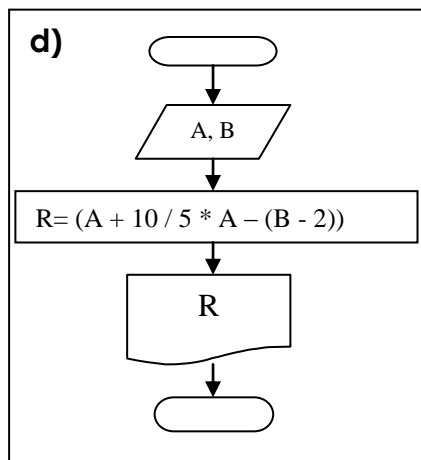
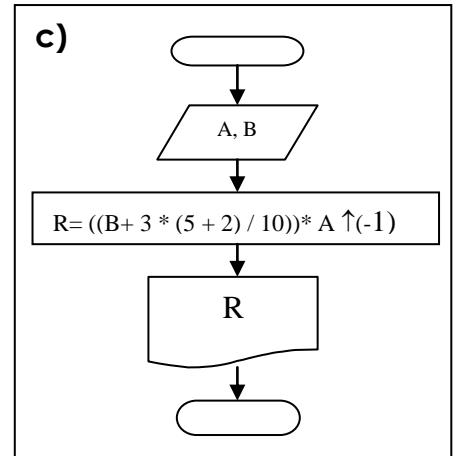
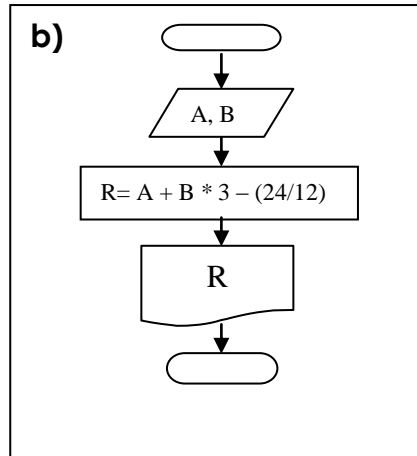
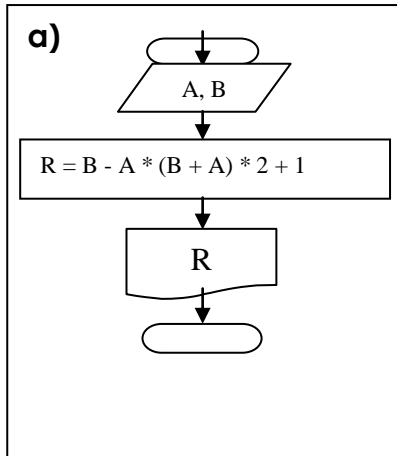
1. Unir con una flecha cada símbolo con su definición correspondiente

Símbolo	
a)	
b)	
c)	
d)	
e)	
f)	

Función	
①	CONECTOR O SÍMBOLO DE CONEXIÓN: se usa cuando el diagrama es largo y se requiere más de una hoja de papel o para evitar líneas que se crucen.
②	SALIDA/ IMPRESION: Indica la salida impresa de la información.
③	PROCESAMIENTO: Se usa generalmente para sentencias de asignación (acción de asignar).
④	ENTRADA/LECTURA: Se utiliza para la introducción de datos desde un periférico a la memoria del computador y para la salida de resultados desde la memoria del computador a un periférico.
⑤	INICIO/FIN: Indica principio o fin de un diagrama.
⑥	DECISIÓN: se usa para evaluar expresiones de tipo lógicas o comparativas, seleccionando la ruta en función del resultado: (V) Verdadero o (F) Falso.

2. Verificar los siguientes algoritmos especificando los resultados obtenidos con los valores indicados para cada variables en la tabla 1, luego comparar con el valor esperado de dicha tabla e indicar si se ha obtenido el mismo resultado.

Diagrama de flujo	Identificadores		Valor Esperado	Valor Obtenido
	A	B		
a)	5	2	-17	
b)	3	4	14	
c)	2	-1	0.55	
d)	-1	4	5	
e)	-2	-4	0	
f)	3	5	227.5	
g)	2	7	"Verdadero"	
h)	4	5	"Falso"	



3. Dados los siguientes problemas y su correspondiente algoritmo de solución, se pide:

- a) **Determinar datos de entrada, resultados y dominio del problema.**
- b) **Expresar mediante diagramas de flujo los algoritmos propuestos.**
- c) **Verificar la solución propuesta.**

a) Leer un número, determinar y mostrar el anterior y el siguiente

1. Inicio
2. Leer NUM
3. $ANT = NUM - 1$
4. $SIG = NUM + 1$
5. Mostrar NUM, ANT y SIG
6. Fin

b) Leer la base y la altura de un triángulo, calcular y mostrar la superficie del mismo

1. Inicio
2. Leer BASE
3. Leer ALTURA
4. $SUP = (BASE * ALTURA) / 2$
5. Mostrar SUP
6. Fin

c) Leer dos números intercambiar sus valores y mostrar sus valores antes y después del intercambio.

1. Inicio
2. Leer X
3. Leer Y
4. Mostrar "Valor X:" X
5. Mostrar "Valor Y:" Y
6. $AUX = X$
7. $X = Y$
8. $Y = AUX$
9. Mostrar "Valor X:" X
10. Mostrar "Valor Y:" Y
11. Fin

d) Leer tres números y calcular y mostrar el resultado de la siguiente operación algebraica: $A * B + 32 / 8 + 7 * C$.

1. Inicio
2. Leer A, B y C
3. $AUX = A * B$
4. $D = 32 / 8$
5. $P = C * 7$
6. $RESULT = AUX + D + P$
7. Mostrar RESULT
8. Fin

e) Leer un número entero, determinar si el mismo es Par o Impar.

1. Inicio
2. Leer NUM
3. $PAR = Int(NUM/2) * 2$
4. **Si** ($PAR = NUM$)
 Entonces
5. Mostrar "El número es Par"
- Sino**
6. Mostrar "El número es Impar"
7. **FinSi**
8. Fin

f) Leer dos números enteros, mostrar aquellos que sean de tres dígitos.

1. Inicio
2. Leer X
3. Leer Y
4. **Si** (X > 99) y (X <= 999)
5. **Entonces**
6. **Mostrar X**
7. **FinSi**
8. **Si** (Y > 99) y (Y <= 999)
9. **Entonces**
10. **Mostrar Y**
11. **FinSi**
12. Fin

4. Dados los siguientes problemas:

- I. Realizar el algoritmo de solución correspondiente, utilizando diagrama de flujo.
- II. Determinar datos de entrada y salida
- III. Verificar la solución propuesta

- a) Leer un número, encontrar el valor absoluto, mostrar el número y su valor absoluto.
- b) Leer un número entero mostrar si es positivo, negativo o nulo.
- c) Leer el radio de una circunferencia, calcular el área y la longitud de la misma.
- d) Leer una medida expresada en metros, calcular y mostrar su equivalente en decímetros, centímetros y milímetros.
- e) Leer 3 números, calcular y mostrar el promedio de los mismos.
- f) Leer las edades de 2 niños en edad escolar, mostrar la edad del mayor de ellos. Se asume que las edades leídas son distintas.
- g) Leer tres números enteros (distintos) que representan las edades de 3 niños en edad escolar, se pide:
 - Mostrar la edad del niño que tenga entre 8 y 12 años inclusive.
 - Mostrar la menor edad
- h) Leer la duración de una llamada telefónica en minutos, mostrar su costo teniendo en cuenta que:
 - Toda llamada que dure 3 minutos o menos tiene un costo de \$5.
 - Cada minuto adicional cuesta \$3.
- i) Leer un número de 3 cifras, se pide mostrar:
 - El primer y el último dígito, solo si el dígito central es par.
 - La suma del primer más el último dígito, si el dígito central es impar.
 - El mensaje “Capicúa” si el número ingresado lo es.
- j) Leer 3 números de un dígito, se pide componer con ellos un número de 3 dígitos y mostrar el número obtenido.

Ejemplo: A= 4 B= 7 C= 9 Número obtenido NUM= 479

k) Leer un número entero, mostrarlo si es de dos dígitos e impar.

l) Leer dos números determinar si uno es múltiplo del otro.

m) Leer el sueldo de un emplead, se pide mostrar el sueldo incrementado de acuerdo a la siguiente tabla

Sueldo	Porcentaje de incremento
0 a 1000	15 %
1001 a 2000	10%
2001 en adelante	5%

n) Leer dos números, si el segundo ingresado es igual al doble del primero más uno, mostrar los dos números ingresados, caso contrario mostrar el mensaje “No cumplen la condición”.

ACTIVIDAD Nº 6: ALGORITMOS Y DIAGRAMAS DE FLUJO

1. Dados los siguientes problemas y su correspondiente algoritmo de solución, se pide:

- I. Determinar datos de entrada, resultados y dominio del problema.
- II. Expresar mediante diagramas de flujo los algoritmos propuestos.
- III. Verificar la solución propuesta.

a) Leer diez números enteros, se pide calcular y mostrar la suma de los mismos.

1. Inicio
2. $SUM = 0$.
3. $C = 1$
4. **Mientras** ($C \leq 10$)
5. Leer NUM
6. $SUM = SUM + NUM$
7. $C = C + 1$
8. **FinMientras**
9. Mostrar SUM
10. Fin

b) Dadas cincuenta monedas, de 25 o 50 ctvos, o de 1 peso, calcular y mostrar cuantas monedas hay de cada valor

1. Inicio
2. $Cant = 0$
3. $C25 = 0$
4. $C50 = 0$
5. $C1 = 0$
- 6 **Repetir**
6. Leer Moneda
7. **Si** (Moneda = 25)
8. **Entonces**
8. $C25 = C25 + 1$
9. **Sino**
9. **Si** (Moneda = 50)
10. **Entonces**
10. $C50 = C50 + 1$
11. **Sino**
11. **Si** (Moneda = 1)
13. **Entonces**
13. $C1 = C1 + 1$
14. **FinSi**
15. **FinSi**
16. **FinSi**
13. $Cant = Cant + 1$
14. **Hasta** ($Cant = 50$)
15. Mostrar C25, C50, C1
16. Fin

c) Leer cincuenta números enteros, se pide calcular y mostrar el promedio de los mismos.

1. Inicio
2. $SUM = 0$.
3. $C = 0$
4. **Mientras** ($C < 50$)
5. Leer NUM
6. $SUM = SUM + NUM$
7. $C = C + 1$
8. **FinMientras**
9. $PROM = SUM / C$
10. Mostrar PROM
11. Fin

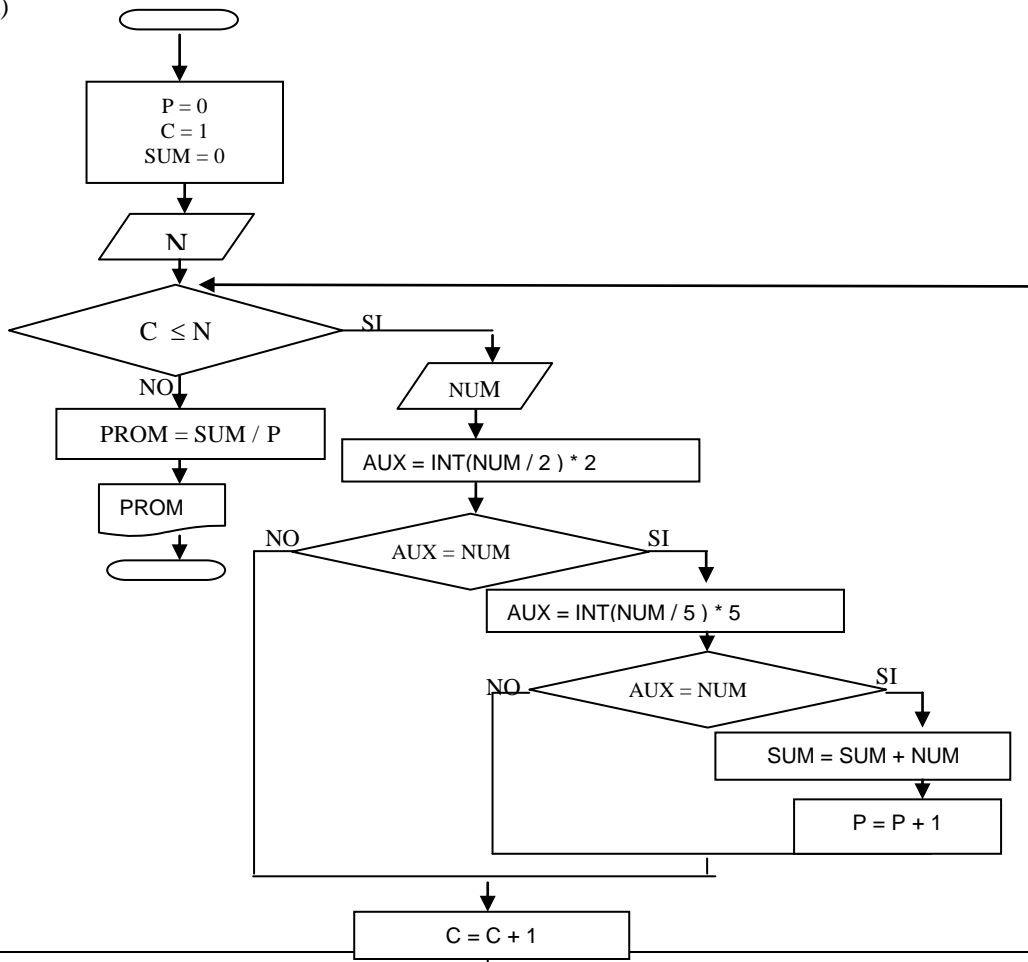
d) Leer N números enteros, calcular y mostrar la suma de los nros. positivos y la cantidad de los nros. negativos y nulos.

1. Inicio
2. SUMP = 0.
3. CNEG = 0.
4. CNU = 0.
5. C = 1
6. Leer N
7. **Mientras** (C <= N)
8. Leer NUM
9. **Si** (NUM > 0)
 - Entonces**
 - 10. SUMP = SUMP + NUM
 - Sino**
 - 11. **Si** (NUM < 0)
 - Entonces**
 - 12. CNEG = CNEG + 1
 - Sino**
 - 13. CNU = CNU + 1
 - 14. **FinSi**
 - 15. **FinSi**
 - 16. C = C + 1
 - 17. **FinMientras**
 - 18. Mostrar SUMP, CNEG, CNU
 - 19. Fin

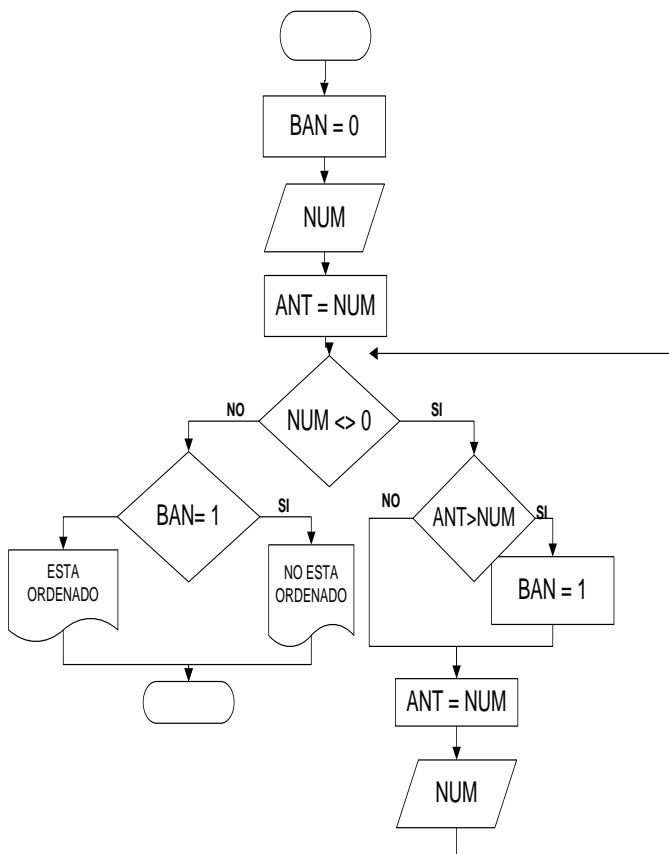
2. Dados los siguientes diagramas de flujo, se pide:

- I. Indicar el problema que resuelve
- II. Determinar datos de entrada y salida y estructuras de control utilizadas
- III. Verificar la solución propuesta con al menos 3 valores de prueba.

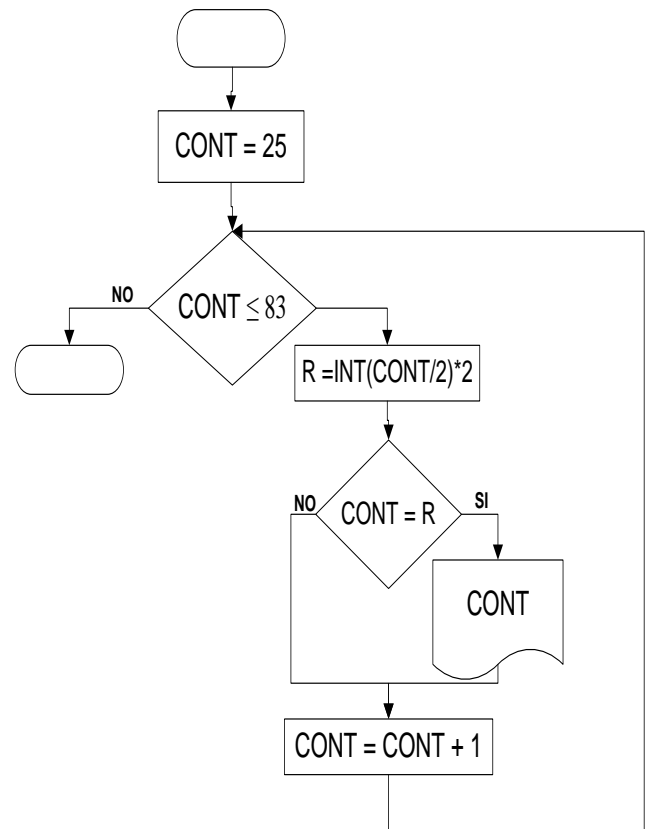
a)



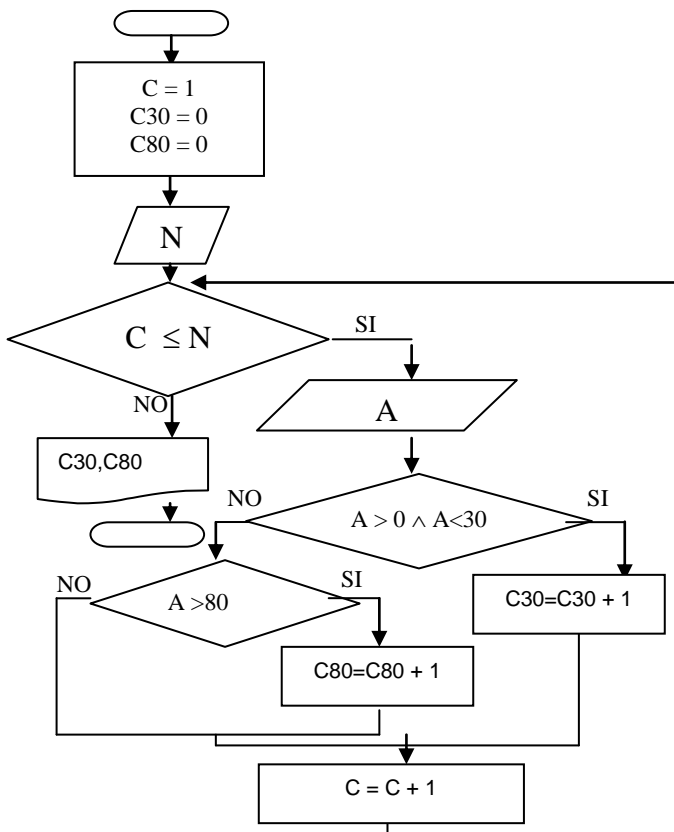
b)



c)



d)



3. Dados los siguientes problemas:

I. Realizar el algoritmo de solución correspondiente, utilizando diagrama de flujo.

II. Determinar datos de entrada y salida

III. Verificar la solución propuesta

- a) Leer 20 números, calcular e mostrar el promedio de los mismos.
- b) Leer N números, calcular e mostrar el promedio de los mismos.
- c) Leer una serie de números, cuyo final está dado por el ingreso del número 0 (cero), calcular e mostrar el promedio de los mismos.
- d) Leer 100 números, mostrar aquellos que sean mayores a 74
- e) Leer 60 números, mostrar aquellos que sean múltiplos de 7.
- f) Leer 100 números mostrar aquellos que sean múltiplos de tres y no de cinco.
- g) Mostrar los 50 primeros números naturales pares.
- h) Leer dos números enteros N1 y N2, calcular y mostrar la suma de los números comprendidos entre N1 y N2.
- i) Leer 100 números enteros, mostrar el mayor y el menor de los números ingresados.
- j) Leer un número NUM, mostrar todos los números desde 1 a NUM, si este es positivo; caso contrario, mostrar el mensaje “INCORRECTO” y el número ingresado.
- k) Leer N números, se pide imprimir el porcentaje de números positivos
- l) Leer N números positivos, mostrar:
 - Suma de números pares
 - Cantidad de números impares.
 - Cantidad de números pares de tres dígitos.
- m) Leer una serie de números de 3 dígitos cuyo final esta dado por el ingreso del número 999, mostrar:
 - Cantidad de números capicúas ingresados.
 - La suma de aquellos números cuyo dígito de la unidad sea par
 - Cantidad números cuyo dígito de la unidad sea impar
- n) Leer N números enteros. Al finalizar el ingreso de los mismos mostrar el mensaje “INGRESADO” si el número 25 estaba entre ellos.
- p) Leer 20 números, mostrar al final del ingreso si un número X leído se encontraba y en que posición.
- q) Leer una serie de números enteros cuyo final viene dado por el ingreso del número 0. Se pide mostrar la cantidad de números impares y el porcentaje de la serie.
- s) Leer N pares de números enteros, mostrar el mayor de cada par, o el mensaje “IGUALES”, en el caso de que lo sean.
- t) Leer N pares de datos correspondientes a los empleados de distintas sucursales de un banco nacional: ***código de empleado y código de sucursal***, mostrar:
 - Los códigos de empleados de la sucursal de la ciudad de Salta.
 - La cantidad de empleados de Tucumán.
 - El mayor código de empleado

Código sucursal	Descripción
1	Salta
2	Sgo. del Estero
3	Tucumán

u) Leer N cuaternas de datos correspondientes a los artículos de un negocio: ***Código del artículo, Cantidad de unidades disponibles, Costo por unidad, Código de marca***, mostrar:

- La cantidad de artículos que poseen cantidad de unidades disponibles menores que 100.
- Los códigos de los artículos cuyo código de marca sea igual a 355 y la cantidad de unidades disponibles sea mayor a 50.
- Los códigos de artículos cuyo costo por unidad no supere los 5 pesos

v) En el Hospital de Niños se lleva a cabo un control anual de las enfermedades infecciosas; para ello se tiene en cuenta los siguientes datos de N niños: **edad**, **sexo** (varón/ mujer) y **enfermedad** (1-hepatitis, 2-sarampión, 3-otra enfermedad infecciosa). Se pide calcular y mostrar:

- La cantidad de varones afectados con hepatitis que tengan entre 8 y 12 años.-
- El porcentaje de mujeres afectados de sarampión.
- La cantidad de niños menores de 4 años que ingresaron al hospital con otras enfermedades infecciosas.
- La menor y la mayor edad de las niñas que tuvieron sarampión.-

w) Leer N ternas de datos correspondientes a los empleados de una empresa. **Dni**, **sexo** (1- Femenino y 2- Masculino) y **DT** (días trabajados) y teniendo en cuenta que por cada día trabajado se abona \$10, mostrar:

- Cantidad de empleados mujeres que hayan trabajado menos de 20 días
- Promedio del sueldo de los empleados varones.
- Sueldo de cada empleado (varones y mujeres).
- DNI del empleado (varones y mujeres) con menor cantidad de días trabajados.

x) Leer N pares de números positivos, donde el primero es la base y el segundo el exponente, para cada par de número mostrar la potencia. (sin utilizar el operador de potencia)

y) Leer N pares de números, mostrar para cada par el resultado del producto (utilizando sumas sucesivas)

z) La AFIP realiza periódicamente controles a los ingresos anuales de sus N contribuyentes para cada uno de ellos se ingresa el **Código de contribuyente** y los correspondientes **Ingresos mensuales**. Calcular y mostrar

- El mayor de Ingreso mensual de cada contribuyente
- El **Condigo de contribuyente** que obtuvo el mayor ingreso anual.

Bibliografía Consultada

- ✓ De Giusti A., "Algoritmos, datos y programas. Con aplicaciones en Pascal, Delphi y Visual Da Vinci", Prentice Hall, Argentina. 2001.
- ✓ Joyanes Aguilar L., "Fundamentos de Programación. Algoritmos, Estructuras de datos y objetos", Tercera Edición, Mc Graw Hill, Madrid.2003.
- ✓ Maldonado, M. et al. "Cartilla de la Asignatura Fundamentos de la Programación". Universidad Nacional de Santiago del Estero. 2006.