

1. HABILIDADES DEL PENSAMIENTO.

1.1 HABILIDADES BÁSICAS DEL PENSAMIENTO.

Las habilidades básicas del pensamiento se refieren a los procesos que permiten obtener información precisa y ordenada de las características de un objeto de observación y así poder tomar decisiones, hacer conclusiones, o resolver problemas.



Las principales HBP son:

- **Percepción / Observación.**

Es el proceso mental que consiste en fijar la atención en una persona, objeto, evento o situación para recopilar y registrar la información a partir de nuestros sentidos e identificar sus características; Gracias a esta habilidad, somos capaces de extraer datos de nuestro entorno para identificar formas, texturas, colores, número y cualidades de los objetos que nos rodean.

- **Observación concreta.**

Aquella que se percibe frente al objeto, hecho o situación.

- **Observación abstracta.**

Aquella que se realiza mediante la experiencia previa, damos las características de un hecho o situación sin necesidad de tenerlo de forma concreta.

EJEMPLO:

Cuando miro el cabello de la persona que está hablando conmigo, puedo notar sus características; es decir, le damos valor a las variables; por ejemplo:

Color: Negro, Forma: Lacio, Textura: Grueso, Tamaño: Corto, Aroma: Vainilla.

- **Descripción.**

Es el proceso mediante el cual se informan de manera clara, precisa y ordenadas las características del objeto en observación. Para organizar las características, podemos seleccionarlas de manera que respondan a una serie de preguntas, tales como: ¿Qué es?, ¿Qué tiene?, ¿Cómo es?, ¿Qué función cumple?, ¿Qué pasó?

EJEMPLO:

Realizar la descripción asociada con las características siguientes: Tipo de alimento: Pastel, Color: Rosa y blanco, Temperatura: Frío, Tamaño: Grande, Relleno: Duraznos, Cubierta: Crema pastelera.

R. El pastel grande es de color rosa con blanco, está cubierto de crema pastelera y relleno de duraznos, además está frío.

- **Comparación.**

El proceso de comparar consiste en el establecimiento de diferencias y semejanzas entre personas, objetos, eventos o situaciones, el establecimiento de semejanzas permite generalizar, el de diferencias el particularizar y como consecuencia de ambos comparar.



EJEMPLO: ¿En qué se parecen y en qué se diferencian los siguientes objetos?



Ambos objetos son instrumentos musicales, poseen cuerdas y tienen un mástil, también comparten los colores negro y marrón en su diseño. Se diferencian en que la primera es una guitarra, su mástil es más corto, tiene 6 cuerdas delgadas y sus notas son más agudas, el segundo es un bajo, su mástil es más largo, posee 4 cuerdas más gruesas que se utilizan para entonar sonidos más graves.

- **Relación.**

Es el proceso que consiste en observar y comparar a los objetos, hechos o situaciones y unirlos mediante un nexo o proposición (**Mayor que, menor que, igual a, diferente a**). Estas relaciones pueden expresar **equivalencias, similitudes o diferencias**.

Se definen mediante el cuestionamiento: ¿Qué puedes afirmar o negar de los objetos que deseas relacionar en función de cada variable?

EJEMPLO: Escribir una relación a partir de los objetos o situaciones y la variable indicada.

* Violeta tiene 18 años y Luvia 21* Variable: Edad

R. Violeta es menor que Luvia.

Gato y camello Variable: Tamaño.

R. El camello tiene mayor tamaño que el gato

* A Marce le gustan más las matemáticas y a Eduardo la física*

Variable: Asignatura favorita.

R. A Marce y Eduardo le gustan asignaturas diferentes.

Israel ha trabajado 3 horas en el súper y Cirilo 180 minutos

Variable: Tiempo trabajando en el súper

R. Israel y Cirilo han trabajado el mismo tiempo en el súper.

- **Clasificación.**

La clasificación es un proceso mental que permite agrupar personas, objetos, eventos o situaciones con base en sus semejanzas y diferencias.

R.



EJEMPLO: Clasificar diferentes clases de zapatos.



Zapatos negros, zapatos blancos, zapatos altos, zapatos de suela gruesa, zapatos con agujetas, zapatos deportivos, zapatos ortopédicos, etc.

RETO: Observa los objetos que se representan en las siguientes figuras y completa la tabla de variables y características:

OBJETO	VARIABLES	CARACTERÍSTICAS
	TIPO DE OBJETO	OSO DE PELUCHE
		MARRÓN
	TEXTURA	
	TAMAÑO	
	COMPLEXIÓN	
	TIPO DE OBJETO	
	COLOR	
		CREMOSA
		PLÁSTICO Y METAL
	TAMAÑO	
		RECTANGULAR
	TAMAÑO	
	COLOR	
		CERRADO

	DISEÑO	
--	--------	--

RETO: Describe los siguientes objetos.

OBJETO	DESCRIPCIÓN
	
	

RETO: Encuentra las diferencias entre las siguientes imágenes.



RETO: Escribe las semejanzas y diferencias entre ambos objetos.



RETO: Escribe la relación a partir de los objetos o situaciones y la variable indicada.

1. Monte Everest y Popocatepetl. Variable: Altura.
2. PlatziConf 2017 y PlatziConf 2018. Variable: Número de asistentes.
3. Carolina pesa 45kg y Francisco pesa 68kg. Variable: Peso.

RETO: Escribe al menos 5 clases de:

1. Autos.
2. Árboles.
3. Casas.

1.2 INFERENCIAS LÓGICAS

1.2.1 CONCEPTOS BÁSICOS.

Lógica es la disciplina que estudia los métodos y principios que se usan para distinguir el razonamiento bueno (correcto) del malo (incorrecto).

Inferencia es el proceso de razonamiento, compuesto por proposiciones, por el cual se deriva o extrae una conclusión de una o varias premisas.

El término Inferencia es considerado como sinónimo de: “predicción”, “derivación” o “deducción”.

Enunciado. Entidad lingüística conformada por palabras.

Proposición. Información contenida en un enunciado que es verdadera o falsa.

Argumento. Conjunto de proposiciones que sirven de premisas que conducen a una conclusión.

Premisa. Proposición aseverada o supuesta que sirve de apoyo o razón para aceptar la conclusión de un argumento.

Conclusión. Es la proposición aseverada con base en otras proposiciones (premisas) del argumento.

1.2.2 LÓGICA PROPOSICIONAL.

La lógica proposicional es el nivel más básico de la lógica, se encarga de analizar las relaciones entre proposiciones, así como la verdad o la falsedad de estas.

➤ Elementos de la lógica proposicional

- **Variables.** Las variables proposicionales son los símbolos que sustituyen a las proposiciones. Se llaman de ese modo porque su significado cambia en las diferentes argumentaciones o expresiones donde se utilicen.

Las letras más comunes para asignar las variables son p, q, r, s, t.

- **Conectores.** Alteran, relacionan o conectan enunciados simples haciéndolos complejos. Los más frecuentes son la negación (\neg), la conjunción (\wedge) la disyunción (\vee), el condicional (\rightarrow) y el bicondicional (\leftrightarrow).
- **Auxiliares.** Cuando son muchos los enunciados complejos en un solo reglón, se utilizan los símbolos auxiliares. No tienen ningún significado lógico, pero se usan con el objetivo de clarificar la comprensión de los enunciados. Los símbolos auxiliares son los paréntesis (...) y los corchetes [...].

1.2.3 INFERENCIA LÓGICA.

- **Inferencia**

Es un razonamiento, compuesto por proposiciones, por el cual se deriva o extrae una conclusión de una o varias premisas.

El término Inferencia es considerado como sinónimo de: “predicción”, “derivación” o “deducción”.

Tipos de inferencia.

a) Según el número de premisas.

- ✦ **Inferencia inmediata.** Es una forma de razonamiento que presenta una sola premisa, de la cual derivamos una conclusión.

P: Si Lucía siempre es puntual

C: entonces, es falso que llegue tarde.

- ✦ **Inferencia mediata.** Es una forma de razonamiento compuesto por dos o más premisas de las cuales se deriva la conclusión.

P1. Todos los cuerpos se atraen.

P2. La Tierra y Venus son cuerpos.

C. La Tierra y Venus se atraen.

b) Según la forma de razonamiento.

- ✦ **Inferencia deductiva.** Es una forma de razonamiento cuya conclusión se deriva del contenido directo de las premisas enunciadas, haciendo referencia expresa de los términos enunciados.

P1. Ningún mamífero nace del huevo.

P2. Toda ave nace del huevo.

C. Ningún ave es mamífero.

- ✦ **Inferencia inductiva.** Es un razonamiento cuyas premisas representan casos particulares de las cuales se deriva una conclusión que resulta un principio general.

P1. El tucán tiene pico.

P2. La guacamaya tiene pico.

P3. La codorniz tiene pico.

C. Toda ave tiene pico.

1.2.4 PROPOSICIONES LÓGICAS

La lógica proposicional es la parte de la lógica que estudia la formación de proposiciones complejas a partir de proposiciones simples.

A continuación hay una tabla que despliega todas las conectivas lógicas que ocupan a la lógica proposicional, incluyendo ejemplos de su uso en el lenguaje natural y los símbolos que se utilizan para representarlas.

Conectiva	Expresión en el lenguaje natural	Ejemplo	Símbolo en este artículo	Símbolos alternativos
Negación	no	No está lloviendo.	\neg	\sim
Conjunción	y	Está lloviendo y está nublado.	\wedge	$\&$
Disyunción	o	Está lloviendo o está soleado.	\vee	
Condicional material	si... entonces	Si está soleado, entonces es de día.	\rightarrow	\supset
Bicondicional	si y sólo si	Está nublado si y sólo si hay nubes visibles.	\leftrightarrow	\equiv

Considérese el siguiente argumento:

- P1. Mañana es miércoles o mañana es jueves.
- P2. Mañana no es jueves.
- C. Por lo tanto, mañana es miércoles.

• DOBLE NEGACIÓN (DN)

$$\neg \neg p \leftrightarrow p$$

El esquema representa, “p doblemente negada equivale a p”. Siguiendo el esquema de una inferencia por pasos, la representaríamos así:

$\neg \neg p$	“No ocurre que Ana no es una estudiante”
<hr/>	
p	“Ana es una estudiante”

La regla ‘doble negación’, simplemente.

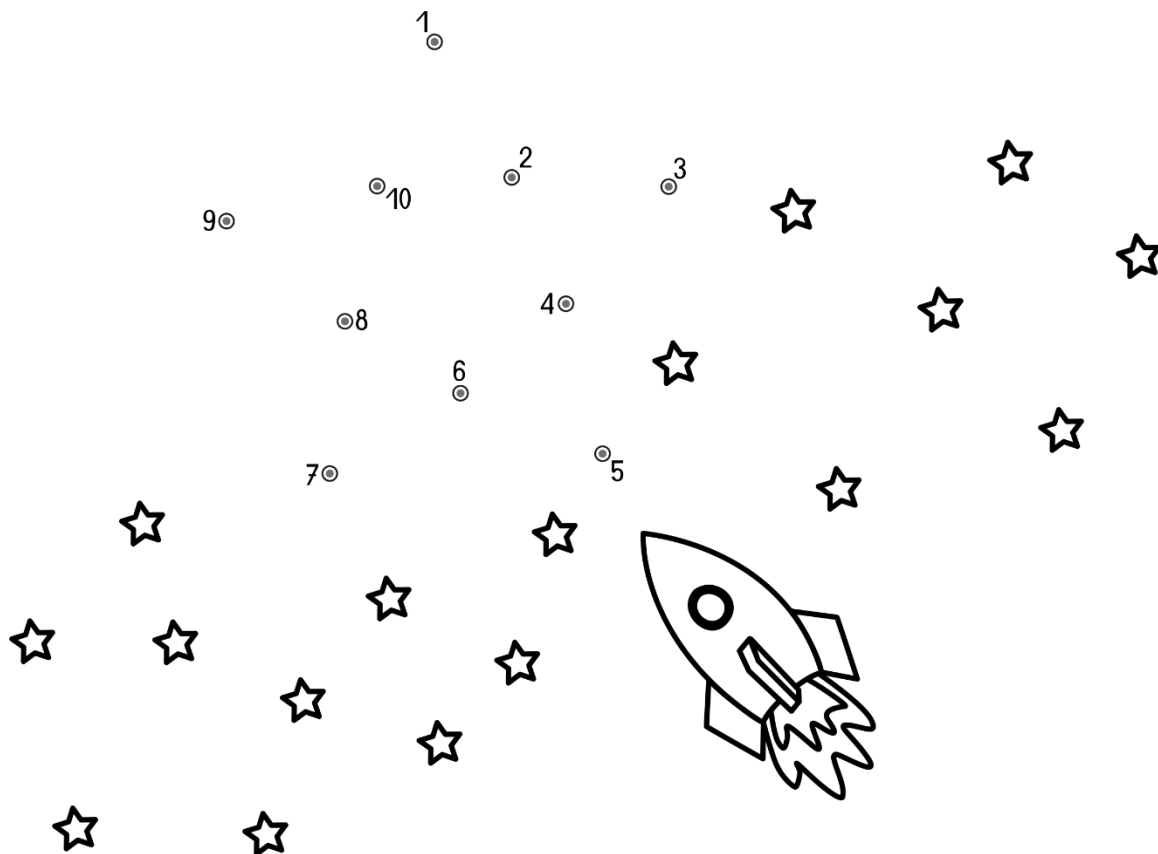
1.3 HABILIDADES LÓGICO MATEMÁTICAS.

El pensamiento lógico matemático se divide en 5 tipos de pensamiento:

1. Numérico
2. Espacial.
3. Medida.
4. Aleatorio o probabilístico.
5. Variaciones.

Para desarrollar este tipo de pensamiento es necesaria la imaginación y concentración.

1.3.1 ASOCIACIÓN DE PUNTOS.

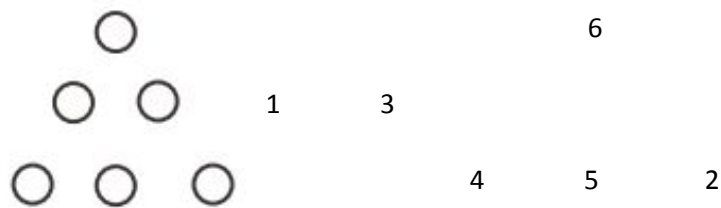


RETO: UNIR LOS PUNTOS EN LAS SIGUIENTE FIGURA

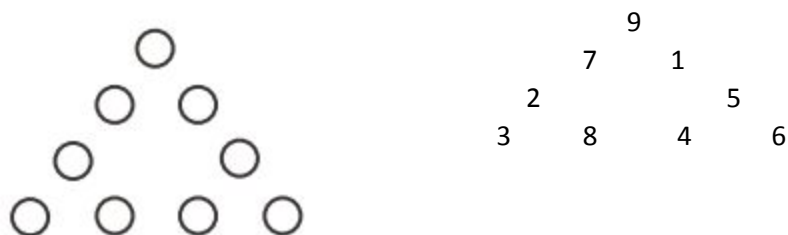


1.3.2 TRIÁNGULOS MATEMÁTICOS

Escribir los números del 1 al 6 sin que se repitan de tal forma que la suma por cada lado del triángulo sea igual.



RETO: Escribir los números del 1 al 9 sin que se repitan de tal forma que la suma por cada lado del triángulo sea igual.



1.3.3 ACERTIJOS

1. ¿Cuántas veces puede restarse el número 1 del número 1.111?

R. 1, puesto que en las ocasiones consecutivas estaríamos restándolo al número 1.110, 1.109, 1.108...

2. María tiene un hermano llamado Juan. Juan tiene tantos hermanos como hermanas. María tiene el doble de hermanos que de hermanas. ¿Cuántos chicos y chicas hay en la familia?

R. M++ J***

3. Una viejecita llevaba huevos al mercado cuando se le cayó la cesta. - ¿Cuántos huevos llevabas? - le preguntaron, - No lo sé, recuerdo que al contarlos en grupos de 2, 3, 4 y 5, sobraban 1, 2, 3 y 4 respectivamente. ¿Cuántos huevos tenía la viejecita?

R. 59 (Tabla de primos y eliminación por criterios de divisibilidad).

4. En una carrera, un corredor adelanta al que va segundo. ¿En qué posición se coloca?

R. En segundo lugar.

RETO: Resolver los siguientes problemas:

1. Dos personas viajan en coche. La menor es hija de la mayor, pero la mayor no es su padre.
¿Quién es?

R. Su madre.

2. El padre de Juan le dice a su hijo que le va a otorgar dos monedas de curso legal. "Entre las dos suman tres euros, pero una de ellas no es de un euro". ¿Cuáles son las monedas?

R. Una de dos euros y otra de un euro. El padre de Juan le dice a su hijo que una de ellas no es de un euro... pero la otra sí puede serlo.

3. Si un ladrillo pesa un kilo más medio ladrillo, entonces... ¿Cuánto pesa un ladrillo y medio?

R. 3 Kilos

1.3.4 SUDOKU

Complete los cuadros con números del 1 al 9 de modo tal que al sumarse en sentido horizontal, vertical y diagonal corresponda al mismo resultado.

6	4	2	3	5	7	8	1	9
1	5	8	2	6	9	4	7	3
9	3	7	1	4	8	5	6	2
8	1	4	7	9	3	2	5	6
7	6	9	8	2	5	3	4	1
5	2	3	6	1	4	9	8	7
2	7	5	4	3	6	1	9	8
3	9	6	5	8	1	7	2	4
4	8	1	9	7	2	6	3	5
4	9	2						
3	5	7						
8	1	6						

RETO: Resolver los siguientes sudokus.

4	9	8	3	1	5	6	7	2
---	---	---	---	---	---	---	---	---

1	3	6	2	4	7	8	9	5
2	5	7	6	8	9	3	4	1
9	1	3	8	2	6	7	5	4
6	7	5	4	9	1	2	8	3
8	4	2	5	7	3	9	1	6
5	6	1	7	3	8	4	2	9
7	2	4	9	6	5	1	3	8
3	8	9	1	4	2	5	6	7
2	9	4						
7	5	3						
6	1	8						

1.3.5 OBSERVACIÓN ATENTA

Determine en cada una de las figuras el número de caras planas.



R. 5, 6, 7

RETO: Determine en cada una de las figuras el número de caras planas.

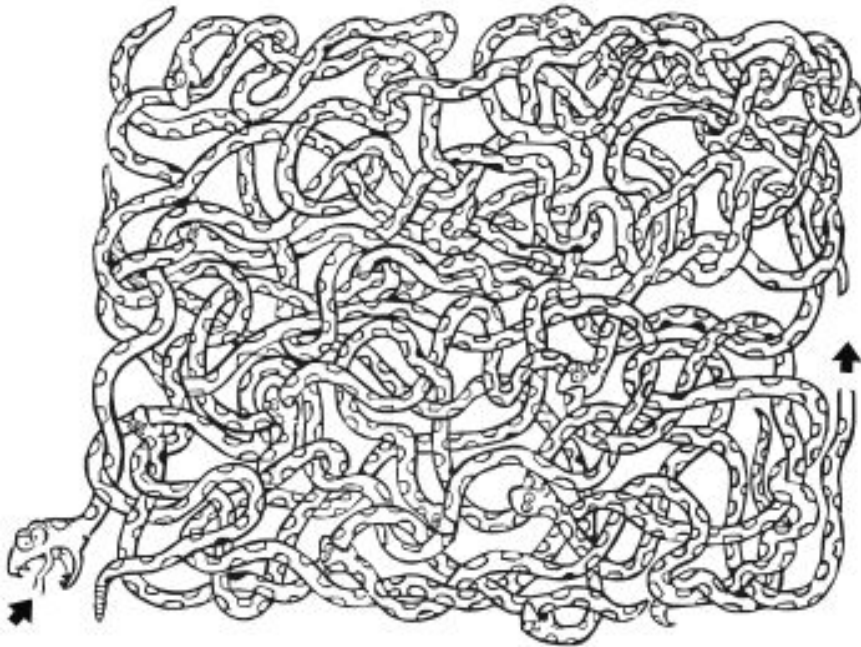


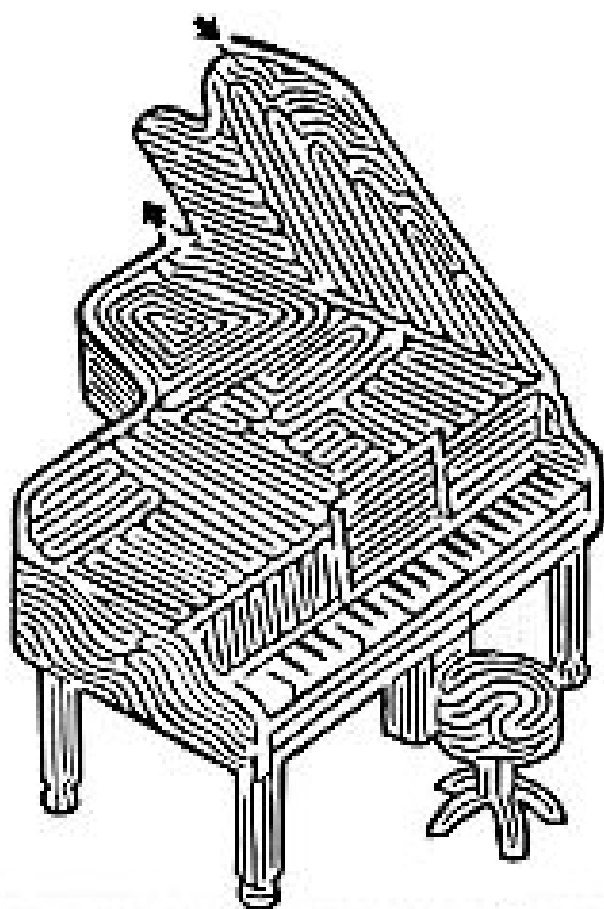
R. 4, 8, 48

CUBO DE RUBIK PRÁCTICA.

1.3.6 LABERINTOS

RETO: Determine el camino correcto para cruzar el laberinto.




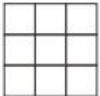


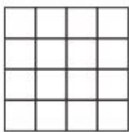
1.3.7 CONTEO DE FIGURAS

¿Cuántos cuadrados hay en cada figura?

a)  = 1

b)  = 5

c)  = $\begin{matrix} 1 \times 1 = 9 & 3 \times 3 = 1 \\ 2 \times 2 = 4 \end{matrix}$ **R. 14**

d)  = $\begin{matrix} 1 \times 1 = 16 \\ 2 \times 2 = 9 \\ 3 \times 3 = 4 \\ 4 \times 4 = 1 \end{matrix}$ **R. 30**

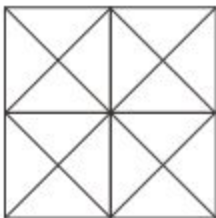
RETO: Determine cuántos rectángulos hay en cada figura.

a)  = 3

b)  = **R. $4+3+2+1= 10$**

c)  = **R. $7+6+5+4+3+2+1$**

RETO: Determine cuántos triángulos y cuántos rectángulos hay en la figura.



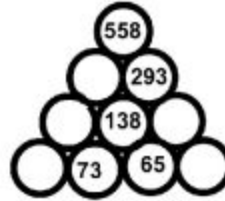
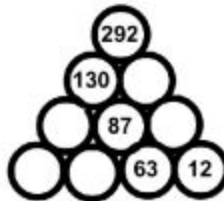
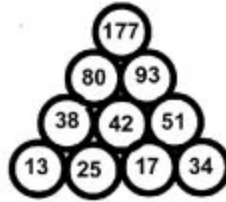
TRIÁNGULOS: $16+16+8+4=44$

RECTÁNGULOS: $4+4= 8E$

1.3.8 COMPLETAR VALORES

Completar los valores que faltan, sabiendo que el valor en cada círculo corresponde a la suma de los valores en los dos círculos debajo.

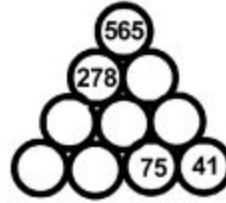
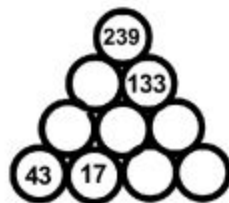
EJEMPLO:



A) 162, 43, 75, 19, 24.

B) 265, 127, 155, 54, 90.

RETO: Completar los valores que faltan, sabiendo que el valor en cada círculo corresponde a la suma de los valores en los dos círculos debajo.



A) 106, 60, 46, 87, 29, 58 B) 287, 107, 171, 116, 11, 96

1.3.9 LETRAS EN DESORDEN

Descubre los nombres de países que están escritos por las letras en desorden.

EJEMPLO:

tagriarln: Inglaterra

urpe: Perú

omlaiboc: Colombia

RETO: Descubre los nombres de países y apellidos que están escritos por las letras en desorden.

xmioec: México

iaalit: Italia

aiusr: Rusia

ianhc: China

anadac: Canadá

1.4 TEORÍA DE CONJUNTOS

La teoría de conjuntos se entiende como un contenido del área de matemáticas pero sus utilidades van mucho más allá del desarrollo del pensamiento lógico matemático. Comprender la teoría de conjuntos nos permite utilizar los conjuntos como herramienta para **analizar, clasificar y ordenar** los conocimientos adquiridos desarrollando la compleja red conceptual en que almacenamos nuestro aprendizaje.

1.4.1 CONCEPTOS BÁSICOS.

- ✦ **CONJUNTO.** Un conjunto es la agrupación, clase, o colección de objetos o elementos que pertenecen y responden a la misma categoría.

Así existen cuatro formas de las cuales podemos definir los conjuntos:

- **Extensión o enumeración:** Sus elementos son encerrados entre llaves y separados por comas. Cada conjunto describe un listado de todos sus elementos. Además, sus elementos no se repiten.

$A = \{A, E, I, O, U\}$

$B = \{0, 2, 4, 6, 8, 10\}$

$C = \{P, L, A, T, Z, I\}$

- **Comprensión:** Sus elementos se determinan a través de una condición que se establece entre llaves.

$A = \{x/x \text{ es una vocal}\}$

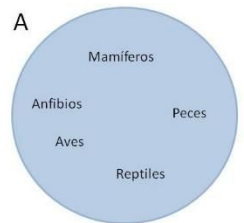
$B = \{x/x \text{ es un número par menor que } 11\}$

$C = \{x/x \text{ es una letra de la palabra Platzi}\}$

- **Diagramas de Venn:** Regiones cerradas que nos permiten visualizar las relaciones entre los conjuntos.

- **Descripción verbal:** Se trata de un enunciado que describe una característica común a todos los elementos del conjunto.

“A contiene a todos los animales que tienen esqueleto interno”



- ✦ **ELEMENTO.** Es cada uno de los objetos por los cuales está conformado un conjunto. $x \in A$ (El elemento x Pertenece al conjunto A). El símbolo \notin se utiliza cuando un elemento no pertenece al conjunto.

- ✦ **SUBCONJUNTO.** Sean los conjuntos $A = \{0, 1, 2, 3, 5, 8\}$ y $B = \{1, 2, 5\}$

En este caso decimos que B está contenido en A, o que B es subconjunto de A.

En general si A y B son dos conjuntos cualesquiera, decimos que B es un subconjunto de A si todo elemento de B lo es de A también.

∴ si B es un subconjunto de A se escribe $B \subseteq A$.

Si B no es subconjunto de A se indicará con una diagonal $B \not\subseteq A$.

1.4.2 TIPOS DE CONJUNTOS

✦ **CONJUNTO UNIVERSO (U, Ω).**

Se denomina así al conjunto formado por todos los elementos del tema de referencia.

$U = \{x/x \text{ es un animal}\}$

$A = \{x/x \text{ es un mamífero}\}$

$B = \{x/x \text{ es un reptil}\}$

CONJUNTO VACÍO ($\{ \}, \emptyset$).

Se denomina así al conjunto que no tiene ningún elemento. A pesar de no tener elementos se le considera como conjunto.

Ejemplos: Conjunto de los meses del año que terminan en a.

Conjunto de números impares múltiplos de 2.

✦ **CONJUNTO UNITARIO.**

Es el conjunto que tiene un solo elemento.

Ejemplo: Conjunto de los meses del año que tiene menos de treinta días, solamente febrero pertenece a dicho conjunto.

✦ **CONJUNTOS DISJUNTOS.**

Se llaman conjuntos disjuntos aquellos que no tienen ningún elemento que pertenezca a ambos al mismo tiempo.

Ejemplo: Los dos conjuntos siguientes:

$\{x/x \text{ es un número natural}\}$

$\{x/x \text{ es un día de la semana}\}$

Son disjuntos ya que no tienen ningún elemento común.

✦ **CONJUNTOS IGUALES.**

Los conjuntos son iguales, si tienen los mismos elementos, por ejemplo:

El conjunto $\{a, b, c\}$ también puede escribirse:

$\{a, c, b\}, \{b, a, c\}, \{b, c, a\}, \{c, a, b\}, \{c, b, a\}$

En teoría de conjuntos se acostumbra no repetir a los elementos por ejemplo:

El conjunto $\{b, b, b, d, d\}$ simplemente será $\{b, d\}$.

✦ **CONJUNTO INFINITO.**

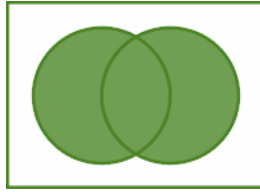
Es aquel conjunto cuya cantidad de elemento no se puede contar; es decir, es aquel conjunto en que sus elementos no se pueden nombrar o enumerar. Son

ejemplos de conjuntos infinitos, los conjuntos numéricos: los naturales (\mathbb{N}), los enteros (\mathbb{Z}), los reales (\mathbb{R}), los racionales (\mathbb{Q}), los imaginarios (\mathbb{I}), los complejos (\mathbb{C})

1.4.2 OPERACIONES CON CONJUNTOS.

✧ UNIÓN

La unión de dos conjuntos A y B es el conjunto $A \cup B$ que contiene cada elemento que está por lo menos en uno de ellos.



✧ INTERSECCIÓN

La intersección de dos conjuntos A y B es el conjunto $A \cap B$ que contiene todos los elementos comunes de A y B .



✧ DIFERENCIA

La diferencia entre dos conjuntos A y B es el conjunto $A \setminus B$ que contiene todos los elementos de A que no pertenecen a B .



✧ COMPLEMENTO

El complemento de un conjunto A es el conjunto A^C que contiene todos los elementos (respecto de algún conjunto referencial) que no pertenecen a A .



EJEMPLOS:

Tomando en cuenta los conjuntos:

$A = \{0, 1, 2, 3, 4, 5\}$ $B = \{2, 4, 6\}$ $C = \{ \}$

2. ARITMÉTICA EN LA LÓGICA.

2.1 INTERPRETACIÓN DE LOS NÚMEROS Y SIGNOS

La **recta numérica** es una recta en la que a cada uno de sus puntos le podemos asignar el valor de un número real.



UTILIZAR ROTAFOLIO O TABLETA PARA UBICAR PUNTOS EN LA RECTA NUMÉRICA

Ley de los signos

MULTIPLICACIÓN	DIVISIÓN	SUMA Y RESTA
$(+) (+) = +$ $(+) (-) = -$ $(-) (+) = -$ $(-) (-) = +$	$+ / + = +$ $+ / - = -$ $- / + = -$ $- / - = +$	<p>+ y + Se suman las cantidades.</p> <p>+ y - se restan las cantidades y se pone el signo del número más grande.</p> <p>- Y - Se suman las cantidades y se pone el signo -</p>

EJEMPLOS:

$(+3) \times (+6) = +18$	$(+7) \div (-1) = -7$
$4) = +8$	$(+10) \div (-2) = -5$
$(-6) = -18$	$(+18) + (+20) = +38$
$+4) = -8$	$(-6) + (-14) = -20$
$(+36) \div (+6) = +6$	$(+10) + (-2) = +8$
$(-150) \div (-10) = +15$	$(-5) = -29$

RETO: Resolver las siguientes operaciones:

$$(-18) + (+14) = -4$$

$$(+7) \times (-3) = -21$$

$$(+8) \times (+2) = +16$$

$$(-9) \div (-3) = +3$$

$$(+7) + (-13) = -6$$

$$(+9) \times (-2) = -18$$

$$(-10) \times (-2) = +20$$

$$(+12) \div (+2) = +6$$

$$(+9) + (-21) = -12$$

$$(-8) \times (+2) = -16$$

$$(-2) \times (-5) = +10$$

$$(-20) \div (+2) = -10$$

$$(-5) + (-25) = -30$$

$$(+8) \div (+2) = +4$$

$$(+18) \times (+2) = +36$$

$$(-16) \div (+8) = -2$$

$$(-14) + (-28) = -42$$

$$(-10) \div (-2) = +5$$

2.2 RAZONES Y PROPORCIONES

2.2.1 RAZÓN.

Una razón es una comparación entre dos o más cantidades. Puede expresarse mediante una fracción. Si las cantidades a comparar son a y b , la razón entre ellas se escribe como:

$$a : b, a / b \text{ ó } \frac{a}{b} \text{ y se lee "a es a b"}$$

Las razones se pueden amplificar y/o simplificar, y se mantiene la razón.

$$\frac{a}{b} \begin{array}{l} \rightarrow \text{antecedente} \\ \rightarrow \text{consecuente} \end{array}$$

EJEMPLOS:

1. Se realiza una encuesta a un grupo de estudiantes sobre "si practica algún deporte". Luego de un análisis de las respuestas se concluye que 4 de cada 10 estudiantes practica algún deporte.

R. La razón entre los estudiantes que practican deporte y el total de estudiantes es 4:10, que es equivalente a 2:5.

2. El radio de la Luna es $\frac{3}{11}$ del radio de la Tierra, y el radio del Sol es igual a 108 radios terrestres. Hallar la razón entre los radios de la Luna y del Sol.

$$\text{R. } \frac{L}{T} = \frac{3}{11} \text{ y } \frac{S}{T} = \frac{1}{108}$$

$$\frac{L}{S} = \frac{\frac{3}{11}}{108} = \frac{1}{396}$$

$$\frac{S}{T} = \frac{1}{396}$$

3. En una sala de clases hay 10 mujeres y 18 hombres. ¿Qué relación numérica existe entre el número de mujeres y el número de hombres?

R. La relación entre el número de mujeres y el número de hombres es de "10 es a 18", otra forma de leerlo es "10 de 18"

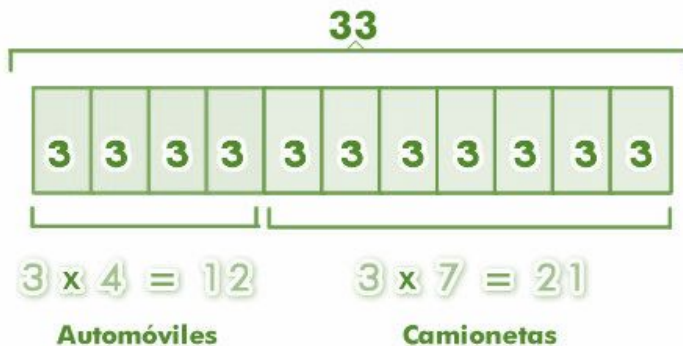
4. Si hay 33 vehículos entre automóviles y camionetas y la razón entre ellos es 4:7 ¿cuántos automóviles hay?

En este caso se está comparando la cantidad de automóviles con el de camionetas. Para conocer la cantidad de automóviles que hay podemos seguir los siguientes pasos:

1° se considera el total de vehículos: 33

2° Se divide 33 por la suma entre el numerador y el denominador de nuestra razón (4+7= 11). Con esto se obtienen 11 partes con 3 unidades cada una (ya que 33:11 = 3).

3° Se consideran 4 partes para los automóviles y 7 para las camionetas.



R. Hay 12 automóviles.

RETO: Resuelve los siguientes problemas:

1. Si la razón entre dos números es 2:3 y ambos suman 10 ¿Cuáles son los números?

R. 4 y 6

2. Martín tiene cinco fichas rojas por cada dos azules. Si tiene 21 fichas en total, entre rojas y azules, ¿Cuántas fichas tiene de cada color?

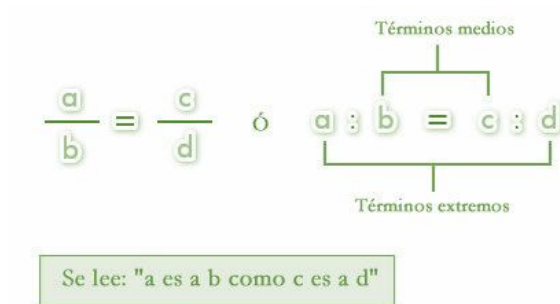
R. 15 rojas y 6 azules

3. A un taller de guitarra asisten 30 estudiantes. Si por cada 8 niñas hay 7 niños, ¿cuántos niños y niñas conforman el taller?

R. 16 niñas y 14 niños

2.2.1 PROPORCIÓN.

Una proporción es la igualdad de dos razones.



En toda proporción, el producto de los términos medios es igual al producto de los términos extremos (Teorema fundamental de las proporciones). Es decir:

$$\frac{a}{b} = \frac{c}{d} \quad \text{entonces} \quad a \cdot d = b \cdot c$$

EJEMPLOS:

1. En una tienda se venden dulces nacionales e importados, a razón de 3:2 Si sabemos que al día se vende 255 dulces nacionales, ¿Cuántos dulces importados se venden al día?

$$\frac{3}{2} = \frac{255}{x} \Rightarrow x = \frac{(2)(255)}{3} = \frac{510}{3} = 170$$

R. 3:2 = 255:170

2. Dos grúas mueven 50 contenedores en hora y media. ¿Cuántas grúas se necesitan para mover los 50 contenedores en media hora?

$$2:1.5 = ?:.5$$

$$2 \times 1.5 = 3$$

$$3 / .5 = 6 \text{ grúas son necesarias.}$$

$$2:1.5 = 6:.5 \text{ (dos grúas es a una hora y media, como seis grúas son a media hora)}$$

RETO: Resolver los siguientes problemas.

En una fiesta se invitaron a niños y niñas. Si sabemos que acudieron en una proporción de 6 niñas por cada 4 niños, y en la fiesta hay 32 niños ¿Cuántas niñas fueron?

$$6:4 = x:32$$

$$32 \times 6 = 192$$

$$192 / 4 = 48 \text{ niñas fueron a la fiesta.}$$

R. $6:4 = 48:32$ (6 es a 4 como 48 es a 32)

2. Si 4 alumnos realizan un trabajo en equipo en 45 minutos ¿Cuánto tiempo tardarán si el equipo está formado por 6, 8, 10 y 12 estudiantes?

Tendremos las siguientes proporciones:

a) $4:45 = 6:?$

b) $4:45 = 8:?$

c) $4:45 = 10:?$

d) $4:45 = 12:?$

$$4 \times 45 = 180$$

a) $180 / 6 = 30$ minutos

b) $180 / 8 = 22.5$ minutos

c) $180 / 10 = 18$ minutos

d) $180 / 12 = 15$ minutos

Por lo que las proporciones serán:

a) $4:45 = 6:30$

b) $4:45 = 8:22.5$

c) $4:45 = 10:18$

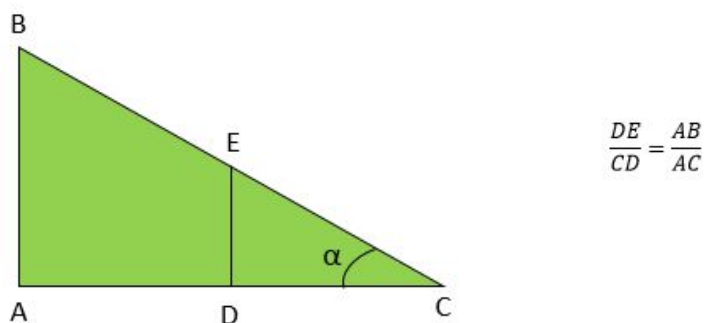
d) $4:45 = 12:15$

2.3 TEOREMA DE THALES



Antes de dar su enunciado, recordemos algunas nociones de semejanza de triángulos. Esencialmente, dos triángulos son semejantes si sus ángulos son congruentes (tienen la misma medida). Esto da lugar al hecho de que, si dos triángulos son semejantes, sus lados correspondientes (u homólogos) son proporcionales.

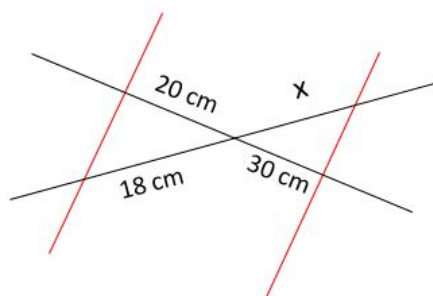
El primer teorema de Tales enuncia que si en un triángulo dado se traza una recta paralela a cualquiera de sus lados, el nuevo triángulo que se obtiene será semejante al triángulo inicial.



En la figura anterior, los triángulos ABC y DEC son semejantes.

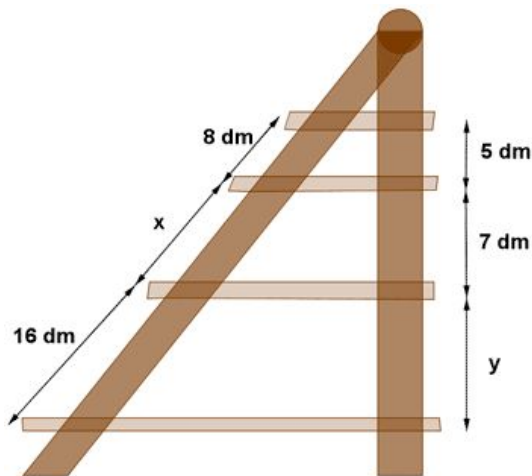
EJEMPLOS:

1. Hallar el valor de x.



$$\frac{x}{18} = \frac{30}{20} \leftrightarrow x = 27cm$$

2. Las maderas de las repisas de la figura son paralelas. Calcula las longitudes de las maderas representadas como x e y.

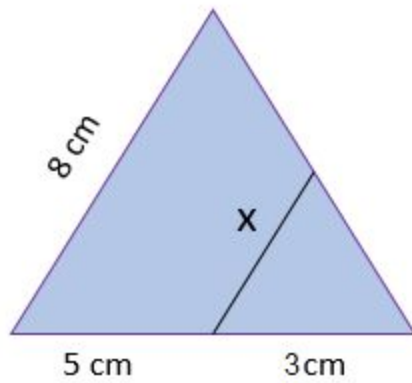


$$\frac{8}{5} = \frac{x}{7} \rightarrow x = \frac{8 \cdot 7}{5} = \frac{56}{5} = 11,2 \text{ dm}$$

$$\frac{8}{5} = \frac{16}{y} \rightarrow y = \frac{16 \cdot 5}{8} = \frac{80}{8} = 10 \text{ dm}$$

RETO: Resolver los siguientes problemas.

1. Hallar el valor de x.

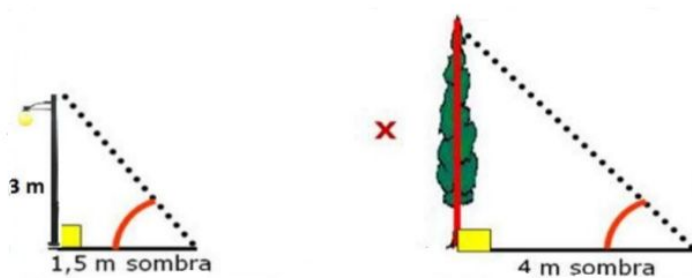


R. Aquí tenemos dos triángulos, uno de estos formado por un segmento paralelo a uno de los lados del otro (precisamente el lado de longitud x). Por el primer teorema de Tales se tiene que:

$$\frac{x}{8} = \frac{3}{3+5} \leftrightarrow x = 3 \text{ cm}$$

proyecta una sombra de 4m?

2. Un poste vertical de 3m proyecta una sombra de 1.5m, ¿Qué altura tendrá un árbol que a la misma hora



$$\frac{3}{x} = \frac{1.5}{4} \rightarrow x = \frac{(4)(3)}{1.5} = 8m$$

3. SUCESIONES

NUMÉRICAS Y SERIES

Las series numéricas son sucesiones ordenadas de números que guardan un vínculo entre sí.

Como los números son infinitos, la cantidad de series numéricas que pueden crearse también lo son. Supongamos que alguien desea detallar una serie numérica de números pares: dicha serie nunca tendrá final.

Las series numéricas, de todos modos, suelen acotarse a ciertos parámetros o instrucciones.

Las series numéricas pueden ser ascendentes o descendentes.

3.1 SUCESIONES / SERIES NUMÉRICAS

Encuentra el número que sigue las siguientes secuencias.

- a) 1, 3, 5, 7,... R. 9
- b) 10, 11, 40, 12, 13, 50,... R. 14
- c) 2000, 1000, 500, 250,... R. 125
- d) 40, 5, 8, 20, 4,... R. 5
- 76543, 2109, 876,... R. 54

RETO: Encuentra el número que sigue las siguientes secuencias.

- a) 3, 12, 3, 10, 3, 8, 3,... R. 6
- b) 5, 10, 25, 50, 125, 250,... R. 625
- c) 66, 63, 77, 73, 88, 83,... R. 99
- d) 9, 16, 25, 36, 49, 64,... R. 81. Porque 9, (+7) 16, (+9) 25, (+11) 36, (+13) 49, (+15) 64,....
- e) 12, 13, 30, 14, 15, 29, ... R. 16

3.2 SERIES DE LETRAS

EJEMPLOS: Hallar el siguiente elemento de la serie.

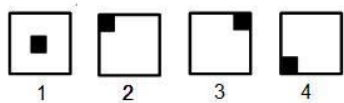
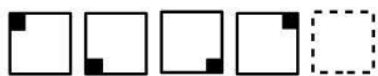
- a) a, m, b, c, m, d, e, f, m, g, h, i, j,... m
- b) a, z, b, b, y, c, c, c, x, d, d, d,... d
- c) G - F - H - L - K - M - P - O - Q - ... U
- d) A - Z - C - V - F - S - J -... Q

RETO: Hallar el siguiente elemento de la serie.

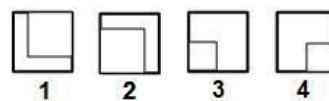
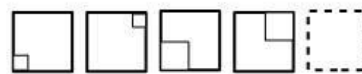
- a) a, z, b, z, c, y, d, y, e, x, f, ... x
- b) a,b,c,d,b,c,d,e,c,d, ... e
- c) a,a,c,c,e,e,g,g,i,i, ... k
- d) a, b, b, c, d, d, d, e, f, f, f, f, ... g
- e) A - B - D - G - K - ... O

3.3 SERIES DE FIGURAS

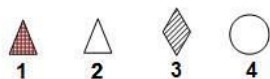
EJEMPLOS: Hallar el siguiente elemento de la serie.



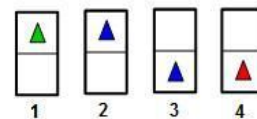
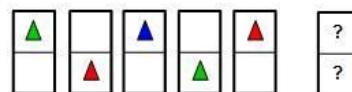
R.2



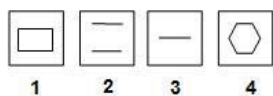
R. 2



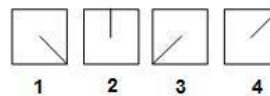
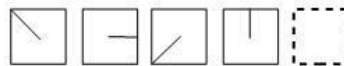
R. 2



R.3

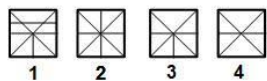


R.3

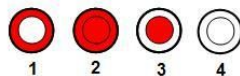


R.1

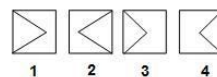
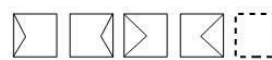
RETOS: Hallar el siguiente elemento de la serie.



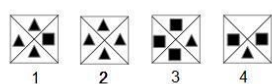
R.2



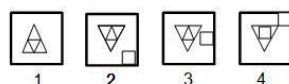
R.1



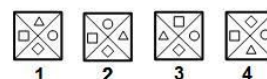
R.1



R.2



R.2



R.1

3.4 SERIES ALFANUMÉRICAS

EJEMPLOS: Hallar el siguiente elemento de la serie.

¿Qué número y qué letra continúan la serie?: a, z, 1, b, y, 10, c, _, _,
R. x, 100

a, 50, e, 40, i, _, _,
R. 30, o

9a, 98z, 987c, 9876x, _, 987654v,
R. 98765e

1ab, ..., 100ef, 1000GH, 10000ij, 100000KL
R. 10CD

A, 8, C, 6, F, 4, J, ..., ...,
R. 2, N

RETOS: Hallar el siguiente elemento de la serie.

¿Qué número y qué letra continúan la serie?: a, a, a, 3, b, b, 2, _, _,
R. c, 1

12, L, 6, F, _, _,
R. 3,

4, a, b, c, d, 3, e, f, g, _, _,
R. 2, h

a, 15, B, 35, c, 55, _, 75, e,
R. D

1, Ñ, 2, K, 4, G, 8, ..., ...,
R. C, 16




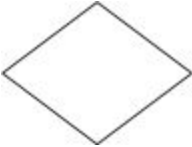
4. DIAGRAMAS DE FLUJO

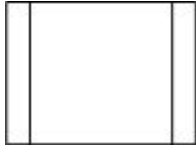





Un diagrama de flujo es la representación gráfica de la secuencia de rutinas simples, describe un **proceso, sistema o algoritmo** informático.

4.1 TOMA DE DECISIONES

El diagrama de flujo ofrece una **descripción visual** de las actividades implicadas en un proceso. Muestra la relación secuencial entre ellas, facilitando la **rápida comprensión** de cada actividad y su relación con las demás, el **flujo de la información** y los materiales, las **ramas en el proceso**, la **existencia de bucles** repetitivos, el **número de pasos del proceso**, facilita también la **selección de la decisión**.

4.2 SIMBOLOGÍA

SÍMBOLO	NOMBRE	USO
	Flecha de flujo.	
{ }	Comentario o anotación	
	Inicio o finalización	Use esta forma para el primer y último paso de un proceso.
	Proceso	Esta forma representa un paso típico del proceso. Es la forma más usada en casi todos los procesos.
	Decisión	Esta forma indica un punto en el que el resultado de una decisión indica el siguiente paso. Puede haber varios resultados, pero con frecuencia solo hay dos: sí y no.

	Subproceso	Use esta forma para un conjunto de pasos que se combinan para crear un subproceso que está definido en otro lugar, a menudo en otra página del mismo documento.
	Documento/Impresora	Esta forma representa un paso que da como resultado un documento o una impresión.
	Salida en Pantalla	Instrucción de presentación de mensajes o resultados en pantalla, es decir <u>muestra</u> el resultado del diagrama de flujo en la pantalla.
	Datos o Entrada/Salida	Esta forma indica que hay información que está entrando desde afuera en el proceso o saliendo de él. Esta forma también se puede usar para representar los materiales y a veces se denomina forma de entrada y salida.
	Referencia en página	Este pequeño círculo indica que el paso siguiente (o anterior) se encuentra en otra parte del dibujo. Resulta particularmente útil para diagramas de flujo grandes en los que, de lo contrario, se debería usar un conector largo, que puede ser difícil de seguir.
	Referencia a otra página	Cuando se coloca esta forma en la página de dibujo, se abre un cuadro de diálogo en el que se puede crear un conjunto de hipervínculos entre dos páginas de un diagrama de flujo o entre una forma de subproceso y una página de diagrama de flujo independiente que muestra los pasos en dicho subproceso.

4.3 ¿CÓMO HACER UN DIAGRAMA DE FLUJO?

1. **Define tu propósito y alcance..** ¿Qué deseas lograr? ¿Estás considerando las cosas correctas con un punto inicial y final apropiado para alcanzar ese propósito? Realiza una investigación lo suficientemente detallada, pero lo suficientemente simple a la hora de crear tus diagramas para comunicarte con tu audiencia.
2. **Identifica las tareas en orden cronológico.** Esto puede involucrar las conversaciones con los participantes, la observación de un proceso o la revisión de cualquier documentación existente. Puedes escribir los pasos en forma de notas o comenzar con un diagrama en versión borrador.
3. **Organízalos por tipo y figura correspondiente,** como procesos, decisiones, datos, entradas o salidas.
4. **Crea tu diagrama,** ya sea dibujándolo a mano o usando un programa.
5. **Confirma tu diagrama de flujo,** verificando todos los pasos con las personas que participan en el proceso. Observa el proceso para asegurarte de no dejar de lado nada que sea importante para tu propósito.

4.4 ALGORITMOS EN DIAGRAMAS

Como una representación visual del flujo de datos, los diagramas de flujo son útiles para escribir un programa o algoritmo y explicárselo a otros o colaborar con otros en el mismo. Puedes usar un diagrama de flujo para explicar detalladamente la lógica detrás de un programa antes de empezar a codificar el proceso automatizado. Puede ayudar a organizar una perspectiva general y ofrecer una guía cuando llega el momento de codificar.

Los diagramas de flujo pueden:

- Demostrar cómo el código está organizado.
- Visualizar la ejecución de un código dentro de un programa.
- Mostrar la estructura de un sitio web o aplicación.

- Comprender cómo los usuarios navegan por un sitio web o programa.

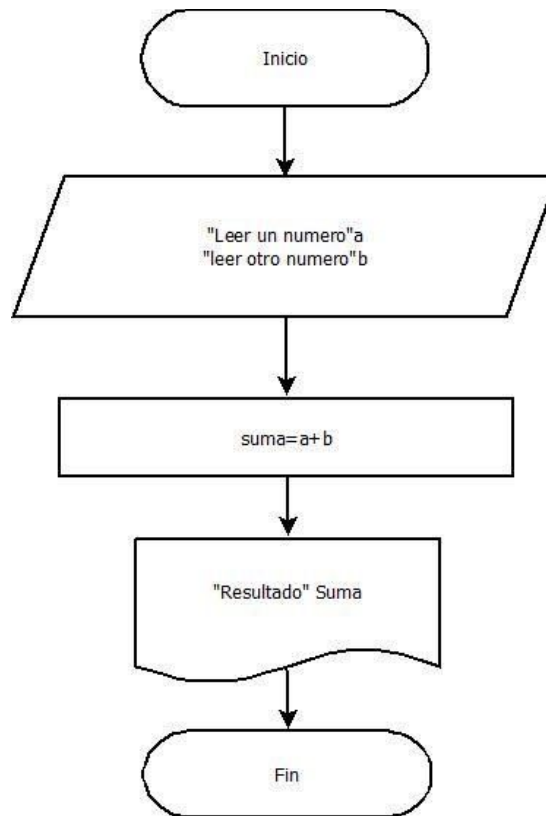
Los diagramas relacionados que se emplean en el software informático incluyen:

- Lenguaje unificado de modelado (UML): este es el lenguaje de propósito general usado en la ingeniería de software para el modelado.
- Diagramas Nassi-Shneiderman (NSD): usados para la programación informática estructurada. Llevan el nombre de sus creadores: Isaac Nassi y Ben Shneiderman, quienes los desarrollaron en 1972 en la Universidad Estatal de Nueva York en Stony Brook. También se denominan "estructogramas".
- Diagramas DRAKON: DRAKON es un lenguaje de programación visual de algoritmos empleado para crear diagramas de flujo.

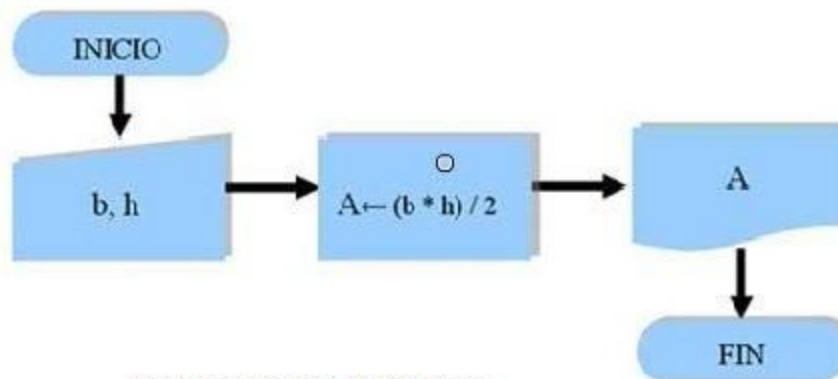
Más allá de la programación informática, los diagramas de flujo pueden tener muchos usos en diversos campos.

En cualquier campo pueden:

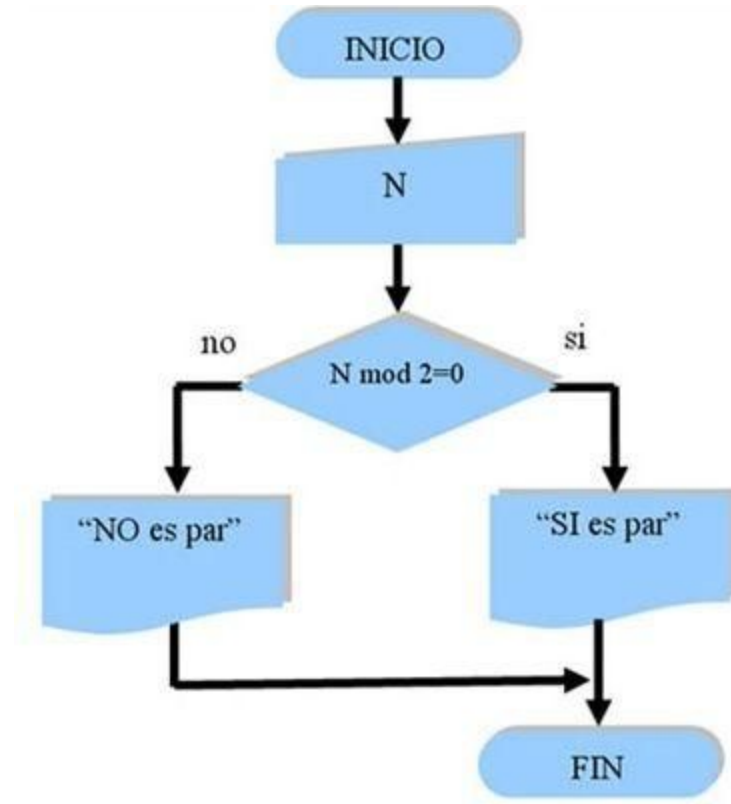
- Documentar y analizar un proceso.
- Estandarizar un proceso para obtener eficiencia y calidad.
- Comunicar un proceso para capacitar a otros sectores de la organización o lograr el entendimiento de su parte.
- Identificar cuellos de botellas, redundancias y pasos innecesarios en un proceso y mejorarlo.



Bueno vamos hacer uno que nos muestre el resultado del área de un triángulo en pantalla.



Vamos hacer uno en el que nos diga si el número es par o impar:



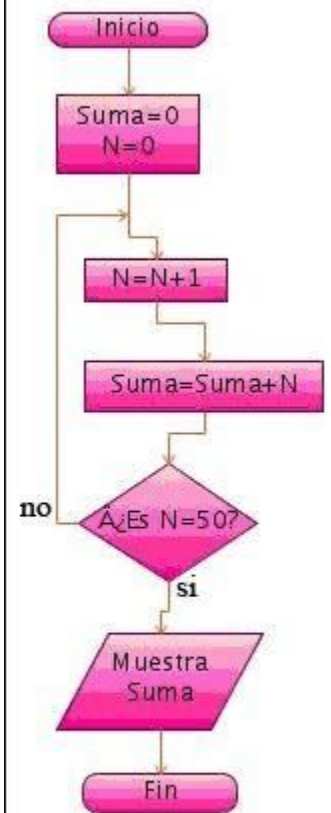
La palabra mod significa dividir, por lo tanto mod 2 es dividir entre 2. Como ya debes saber si divido un número entre 2 y el resto es 0 el número es par, en caso contrario sería impar. Bien pues hay esta la decisión.

¿Al dividirlo entre 2 el resto es 0? Hay 2 posibilidades. Si lo es, se ve en pantalla "Si es par", si no lo es, se ve en pantalla "No es par". Eso es la toma de decisiones. Toma una salida en función del resultado de la entrada.

Un diagrama de flujo para mostrar la suma de los 50 primeros números.

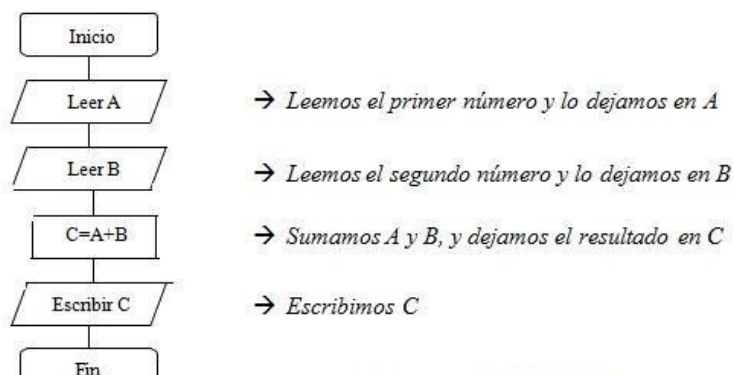
Lo primero es poner a cero la suma y dar el primer número a sumar que será el 0.

Fíjate que el diagrama acaba cuando N, que es el número en cada momento, es 50. Mientras no sea 50 el programa vuelve a la tercera secuencia que será sumarle un número al anterior $N = N + 1$. Intenta comprenderlo y ver lo que hace. Puedes realizar mentalmente el diagrama para el número 0 y verás cómo lo acabas entendiendo.

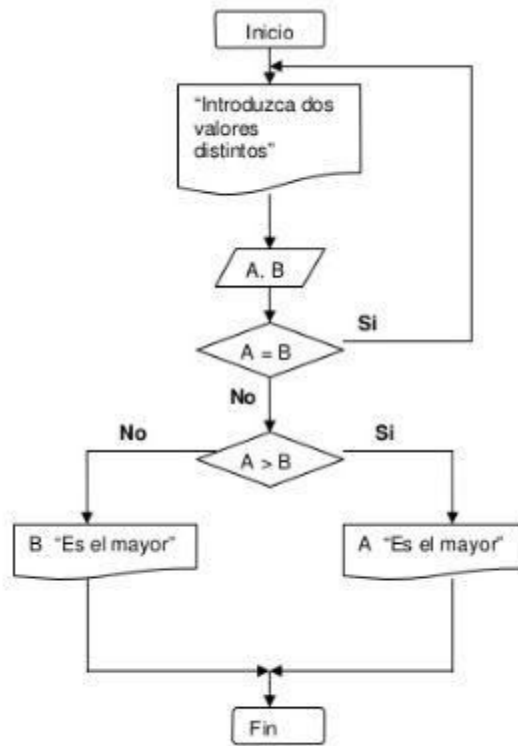


RETOS:

Hacer el diagrama de flujo para sumar dos números leídos por teclado y escribir el resultado.



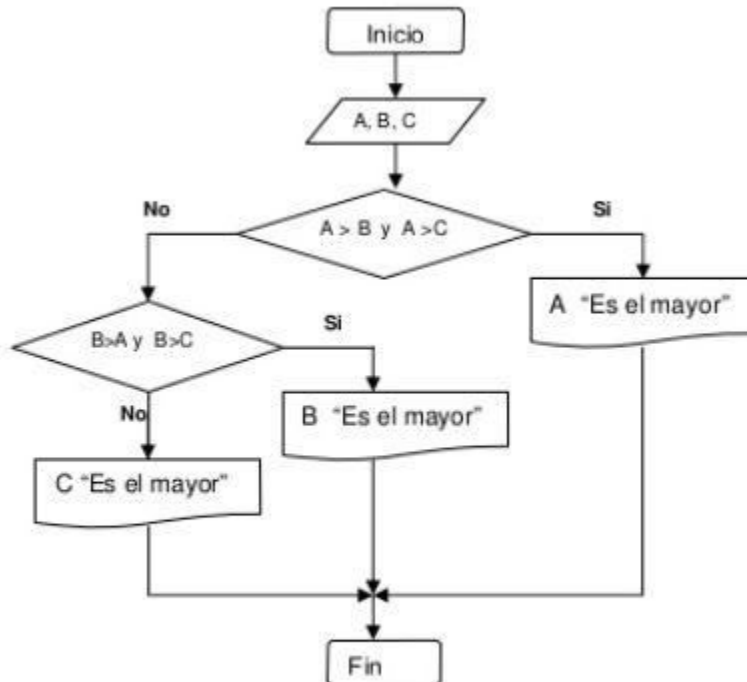
2. Hacer un diagrama de flujo que permita leer 2 números diferentes y nos diga cuál es el mayor de los 2 números.



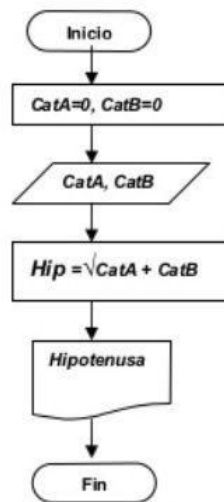
El pseudocódigo para este diagrama sería:

1. Inicio
2. Inicializar variables: $A = 0$, $B = 0$
3. Solicitar la introducción de dos valores distintos
4. Leer los dos valores
5. Asignarlos a las variables A y B
6. Si $A = B$ Entonces vuelve a 3 porque los valores deben ser distintos
7. Si $A > B$ Entonces
Escribir A, "Es el mayor"
8. De lo contrario: Escribir B, "Es el mayor"
9. Fin_Si
10. Fin

3. Crear un diagrama de flujo de procesos en el que se almacenen 3 números en 3 variables A, B y C. El diagrama debe decidir cuál es el mayor y cuál es el menor



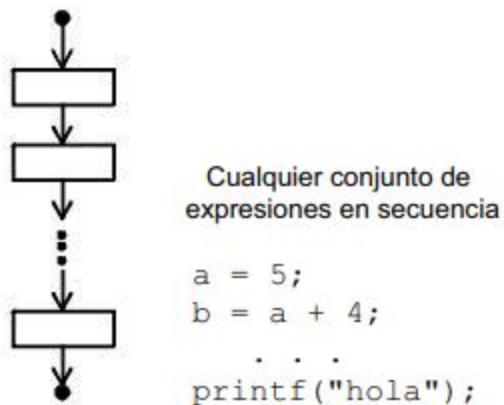
Realizar el diagrama de flujo para que nos calcule la hipotenusa de un triángulo rectángulo, conocidos sus dos catetos.



4.5 ESTRUCTURAS DE CONTROL: PALABRAS RESERVADAS

4.5.1 ESTRUCTURA DE CONTROL SECUENCIAL

La estructura de control secuencial representa la organización de sentencias o instrucciones consecutivas que se van ejecutando una después de la otra. Este diagrama comienza con el punto de entrada y la línea de flujo (→) apunta al primer rectángulo de acción que indica la primera sentencia a ejecutar y continua con el flujo que apunta a otro rectángulo de acción y así podríamos seguir colocando n rectángulos de acción y termina con el punto de salida. Todas las estructuras de control tienen la característica de tener un único punto de entrada y un único punto de salida.



```

1 //suma.c
2 #include <stdio.h>
3
4 main()
5 {
6     int entero1, entero2, suma;
7
8     printf("Introduce un entero: ");
9     scanf("%d", &entero1);
10    printf("Introduce otro entero: ");
11    scanf("%d", &entero2);
12
13    suma = entero1 + entero2;
14
15    printf("\t%d + %d = %d\n", entero1, entero2, suma);
16
17    return 0;
18 }

```

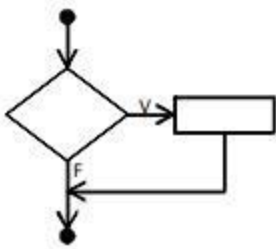
```

Introduce un entero: 45
Introduce otro entero: 13
45 + 13 = 58

```

4.5.2 ESTRUCTURA DE CONTROL DE SELECCIÓN SIMPLE

La estructura de control de selección simple representa una estructura que si se cumple alguna condición se realizan algunas acciones, mientras que si no se cumple, no se realizan dichas acciones.



La línea de flujo inicial señala directamente sobre el símbolo de decisión (!), el cual presenta como salida dos flujos. Una línea de flujo etiquetado con V e indica que la condición es verdadera y realiza las acciones representadas por el rectángulo de acción y otra línea de flujo etiquetado con F que indica que la condición es falsa y se dirige directamente al punto de salida. También debe observar, que en el caso verdadero después de realizar las acciones él se dirige al punto de salida de la estructura.

La selección simple tiene siguiente sintaxis general:


```

if (expresión) {
    expresiones;
}

```

En este caso si la condición (`entero > 0`) es verdadera se imprime en la salida estándar una cadena que expresa que el valor es positivo. Si la condición es falsa no hace nada el programa, regresa cero y termina. También debe observar, cómo las expresiones pertenecientes al `if` están entre inicio y fin de bloque (`{`). En el caso de este programa es una sola expresión, por lo cual se podría omitir las llaves, sin embargo es recomendable colocar las llaves, sea una o más instrucciones.

Programa que imprime si es un número positivo.

```

2  #include <stdio.h>
3
4  main ()
5  {
6      int entero;
7
8      printf("Introduce un entero: ");
9      scanf("%d", &entero);
10
11     if( entero > 0 ){
12         printf("\t%d es un entero positivo\n", entero);
13     }
14
15     return 0;
16 }
1 //positivo.c

```

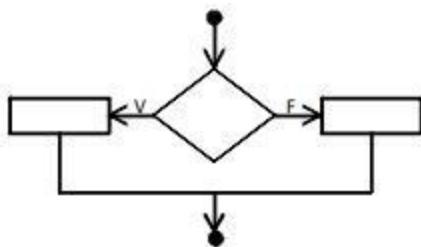
En la línea 11 del listado 2.2 se introduce un nuevo tipo de operador que es el mayor que (`>`) y pertenece a los operadores de relación e igualdad que nos proporciona el lenguaje C. En la tabla 2.1 se muestra una lista completa de este tipo de operadores. Unos de los errores más comunes al aplicar estos operadores es intercambiar su orden o colocar espacio entre los caracteres que lo forman. También comúnmente se confunde el operador de asignación con el de igualdad, por esta razón se remarca que el operador de igualdad son dos caracteres de igual (`==`) y el operador de asignación es solo un carácter de igual (`=`).

Operador	Significado	Ejemplo
<code>></code>	mayor que	<code>a>b</code>
<code><</code>	menor que	<code>a<b</code>
<code>>=</code>	mayor o igual que	<code>a>=b</code>
<code><=</code>	menor o igual que	<code>a<=b</code>
<code>==</code>	igual a	<code>a==b</code>
<code>!=</code>	diferente a	<code>a!=b</code>

4.5.2 ESTRUCTURA DE CONTROL DE SELECCIÓN DOBLE

La estructura de control de selección doble es una estructura que introduce una condición, si es verdadera, realiza un bloque de acciones; y si es falsa, hace otro bloque de acciones.

Desde este símbolo de decisión se tiene dos posibles caminos: uno etiquetado con V, que señala que si la condición es verdadera realizará las acciones representadas por el rectángulo de acción; y otro etiquetado con F que indica que si la condición es falsa realizará las acciones representadas por el rectángulo de acción. Después de realizar unas u otras acciones los dos lados de la estructura se unen nuevamente en el punto de salida de la estructura.



Un ejemplo de cómo se utiliza la estructura de selección doble, se puede observar en el listado 2.3. En este programa se solicita un entero de teclado y se envía a la salida estándar si el entero está en el rango [100, 500]. En las líneas 9 y 10 se solicita, lee y guarda un valor entero en la variable valor. De la línea 12 a 16 se tiene la estructura de control de selección doble con la sintaxis general

```

if(expresión){
    expresiones1;
} else {
    expresiones2;
}

```

En este caso expresiones1 representa las acciones a realizar si la condición es verdadera (expresión) y expresiones2 representa las acciones a realizar si la condición es falsa. En el caso de este programa la condición verifica si el valor leído de de la entrada estándar es mayor o igual que 100 y menor o igual que 500. Si es verdadera, imprime en la salida estándar que el valor está en el rango; y si es falsa, imprime que el valor no está en el rango. Aquí es importante resaltar que únicamente realiza un bloque de acciones y se va al final de la estructura. Finalmente el programa regresa cero y termina. En la figura 2.8 puede observar dos ejemplos de la ejecución de este programa: uno que realiza las acciones del caso verdadero, y en el ejemplo realiza las acciones de caso falso.

```

1 //rango.c
2 #include <stdio.h>
3
4 main()
5 {
6     int valor;
7
8
9     printf("Introduce un entero en [100,500]: ");
10    scanf("%d", &valor);
11
12    if(valor >= 100 && valor <=500){
13        printf("\t%d esta en [100,500]\n", valor);
14    } else{
15        printf("\t%d No esta en [100,500]\n", valor);
16    }
17
18    return 0;
19 }

```

En el programa del listado 2.3 se utilizó el operador de conjunción que es parte de los operadores lógicos con los que cuenta el lenguaje C y en la tabla 2.2 se puede observar la lista de este tipo de operadores.

Operador	Significado	Ejemplo
&&	conjunción (y)	a && b
	disyunción (o)	a b
!	negación (no)	!a

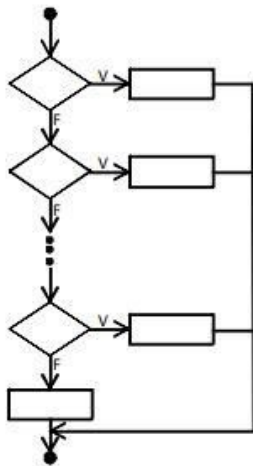
OPERADORES LÓGICOS

```
Introduce un entero en [100,500]: 45
45 No esta en [100,500]
```

```
Introduce un entero en [100,500]: 234
234 esta en [100,500]
```

4.5.3 ESTRUCTURA DE CONTROL DE SELECCIÓN MÚLTIPLE

La estructura de control de selección múltiple, es una estructura que evalúa una expresión que puede tomar varios y dependiendo del valor que tome realiza un conjunto de acciones.



El lado izquierdo muestra un conjunto de símbolos de decisión que señalan los distintos valores que puede tomar la expresión. Si la expresión toma el valor correspondiente al primer símbolo de decisión entonces toma la línea de flujo etiquetada con V y realiza las acciones correspondientes, representadas por el rectángulo de acción; si

no sigue la línea de flujo etiquetada con F y ahora compara con otro valor que está representado por otro símbolo de decisión, y así sucesivamente continua la lógica de esta estructura. En el diagrama se puede observar que cada símbolo de decisión corresponde a un caso diferente. Asimismo se puede observar que después de realizar el rectángulo de acción correspondiente, se dirige al punto de salida de la estructura. También se puede observar un rectángulo de acción después de todos los símbolos de decisión, que se realiza únicamente, si ningún caso fue verdadero, y finalmente termina la estructura.

4. Este programa lee un entero, si el entero está en el rango [1, 7] imprime el día de la semana correspondiente (Lunes = 1, martes = 2, ...); si no está en el rango, imprime, que ese valor no corresponde a ningún día de la semana. La estructura de selección múltiple en este listado está codificada de la línea 11 a la línea 20, y tiene una sintaxis general de la siguiente forma

```
switch(expresión){  
    case constante1: expresiones1; break;  
    case constante2: expresiones2; break;  
    . . .  
    case constanteN: expresionesN; break;  
    default:      otras_expresiones; break;  
}
```

- La palabra reservada para esta estructura es switch. La (expresión) puede tomar valores correspondientes a las constantes (constante1, constante2, ...). Si toma alguno de estos valores realiza las acciones correspondiente a ese case hasta encontrar la palabra reservada break que envía la ejecución al final de la estructura.

```

1 //dia.c
2 #include <stdio.h>
3
4 main()
5 {
6     int dia;
7
8     printf("Introduce un valor [Lunes = 1, Martes = 2, ...]: ");
9     scanf("%d", &dia);
10
11     switch(dia){
12         case 1: printf("\tCorresponde al Lunes\n"); break;
13         case 2: printf("\tCorresponde al Martes\n"); break;
14         case 3: printf("\tCorresponde al Miercoles\n"); break;
15         case 4: printf("\tCorresponde al Jueves\n"); break;
16         case 5: printf("\tCorresponde al Viernes\n"); break;
17
18         case 6: printf("\tCorresponde al Sabado\n"); break;
19         case 7: printf("\tCorresponde al Domingo\n"); break;
20         default: printf("\tNo corresponde ningun dia\n"); break;
21     }
22     return 0;
23 }

```

```

Introduce un valor [Lunes = 1, Martes = 2, ...]: 2
Corresponde al Martes

```

```

Introduce un valor [Lunes = 1, Martes = 2, ...]: 5
Corresponde al Viernes

```

```

Introduce un valor [Lunes = 1, Martes = 2, ...]: 10
No corresponde ningun dia

```

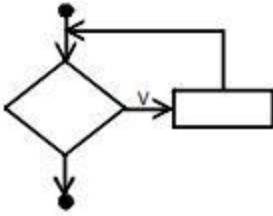
4.5.4 ESTRUCTURA DE CONTROL DE REPETICIÓN MIENTRAS

La estructura de control de repetición mientras es un ciclo que repite un bloque de instrucciones mientras la condición es verdadera.

La característica que la distingue de otros ciclos es que verifica la condición antes de ejecutar las acciones.

En la figura 2.11 se puede observar la lógica de este ciclo. Del punto de entrada la línea de flujo se dirige al símbolo de decisión, si la condición es verdadera continua por la línea etiquetada con V, que se dirige a un rectángulo de acción. Después de terminar esas

acciones representadas por el rectángulo de acción, la línea de flujo regresa al símbolo de decisión.



El ciclo termina cuando la condición es falsa y está representada por la línea etiquetada con F.

Para observar en la práctica cómo funciona el ciclo mientras se analizará el código del listado 2.5. Este programa lee calificaciones de la entrada estándar, las cuenta y las acumula; para después obtener el promedio. En el listado 2.5 la estructura de repetición mientras comienza en la línea 11 y termina la línea 16, tiene la sintaxis general de la siguiente forma:

```
while (expresión) {  
    expresiones;  
}
```

En el caso de este programa la expresión es una condición ($\text{calif} \neq -1$). La condición compara, si la calificación calif es diferente (\neq) de -1 . Si es verdadera, entonces realiza el bloque de instrucciones de la estructura limitado por las llaves. Si la condición es falsa, la estructura termina y continúa programa en la línea 17. Las instrucciones que pertenecen al bloque de esta estructura son: línea 12, acumulación de las calificaciones $\text{suma} += \text{calif}$; línea 13; cuenta el número de calificaciones leídas $\text{conta}++$; líneas 14 y 15, solicita y lee una nueva calificación respectivamente. En la línea 18, se calcula el promedio y en la línea 20, se imprime dicho promedio.

En el programa del listado 2.5 se introducen algunos operadores como: El operador abreviado de suma y asignación $+=$. Este operador se comporta como $a += b$ es equivalente a $a = a + b$. En la tabla 2.3 se puede observar los operadores abreviados de asignación y aritméticos. Otro operador introducido en este programa es el operador de incremento $++$, este operador aumenta en uno el valor de la variable. En la línea 18, se utiliza el operador de conversión explícita (Tipo). Este operador hace una conversión al tipo indicado, que en este caso el Tipo es float. En este programa la operación de suma entre conta daría como resultado un entero, ya que ambos operandos son enteros. Esta es precisamente la razón de colocar (float), ya que con

esto el compilador hace una copia temporal de suma de tipo float y como no puede operar con tipos diferentes automáticamente se hace una copia temporal.

Operador	Ejemplo	Significado
<code>*=</code>	<code>a *= b</code>	<code>a = a * b</code>
<code>/=</code>	<code>a /= b</code>	<code>a = a / b</code>
<code>%=</code>	<code>a %= b</code>	<code>a = a % b</code>
<code>+=</code>	<code>a += b</code>	<code>a = a + b</code>
<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code>

OPERADORES ABREVIADOS ARITMÉTICOS Y ASIGNACIÓN.

```

1 //promedio.c
2 #include <stdio.h>
3
4 main()
5 {
6     int calif, conta = 0, suma = 0;
7     float promedio;
8     printf("Introduce la Calificacion (termina con -1): ");
9     scanf("%d", &calif);
10
11     while(calif != -1){
12         suma += calif;
13         conta++;
14         printf("Introduce la Calificacion (termina con -1): ");
15         scanf("%d", &calif);
16     }
17
18     promedio = (float) suma/conta;
19
20     printf("\tEl promedio es: %6.1f", promedio);
21
22     return 0;
23 }

```

```

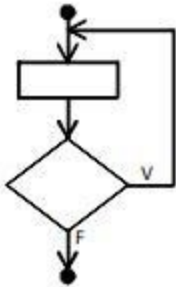
Introduce la Calificacion (termina con -1): 3
Introduce la Calificacion (termina con -1): 6
Introduce la Calificacion (termina con -1): 8
Introduce la Calificacion (termina con -1): -1
El promedio es: 5.7

```

4.5.5 ESTRUCTURA DE CONTROL DE REPETICIÓN HACER/MIENTRAS

La estructura de control de repetición hacer/mientras es otro de los ciclos que proporciona el lenguaje C. De manera semejante al ciclo anterior, esta estructura repite un bloque de instrucciones mientras una condición sea verdadera. La característica que distingue a este ciclo, es que primero realiza las acciones y después verifica la condición. Es decir, que en esta estructura las acciones se realizan al menos una vez.

La línea de flujo del punto de entrada se dirige directamente al rectángulo de acción y después de realizar las acciones la línea de flujo se dirige al símbolo de decisión. Si la decisión es verdadera, la línea de flujo etiquetado con V regresa la ejecución antes del rectángulo de acción. Si la condición es falsa, el camino que sigue es la línea de flujo etiquetado con F y se al punto de salida de la estructura.



Este programa lee de teclado cinco calificaciones, calcula el promedio y lo imprime. En este programa, el ciclo hacer/mientras esta codificado de la línea 9 a la 15 y la sintaxis general de la estructura es:

```
do{  
    expresiones;  
} while (expresión);
```

Como se puede observar las expresiones de esta estructura están antes de verificar el valor de la expresión asociada a la palabra reservada while. Por esta razón, las acciones se realizan al menos una vez y después se verifica su valor de verdad. Si valor de verdad de la expresión es verdadero, entonces se repiten nuevamente expresiones, y así sucesivamente continua el ciclo. El ciclo terminará cuándo el valor de verdad de expresión es falso. En el caso de este programa las expresiones asociadas a la estructura son: Líneas 10 y 11, solicita y lee una calificación, respectivamente. La línea 12, acumula la calificación y línea 13, cuenta el número de calificaciones. En la línea 15, La condición $\text{conta} \leq 5$, define que el ciclo se ejecutará seis veces. El ciclo termina cuándo conta tiene el valor de seis, debido a que la condición se hace falsa. La línea 17, calcula el promedio y la línea 19 lo imprime.

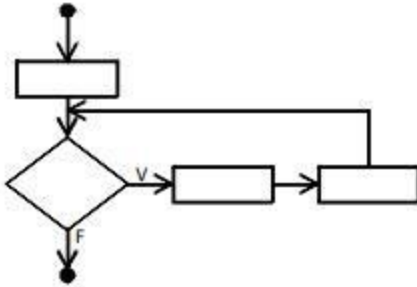
```

1 //promedioConta.c
2 #include <stdio.h>
3
4 main()
5 {
6     int calif, conta = 0, suma = 0;
7     float promedio;
8
9     do{
10         printf("Introduce la Calificacion: ");
11         scanf("%d", &calif);
12         suma += calif;
13         conta++;
14
15     }while(conta <= 5);
16
17     promedio = (float) suma/conta;
18
19     printf("\tEl promedio es: %6.1f", promedio);
20
21     return 0;
22 }

```

4.5.6 ESTRUCTURA DE CONTROL DE REPETICIÓN PARA/HASTA

se utiliza para repetir un conjunto de acciones, hasta que se deje de cumplir una condición. La característica que distingue este ciclo es que está diseñado para repetir un bloque de instrucciones de un valor inicial hasta un valor final.



Este programa lee un entero, calcula e imprime el factorial. En las líneas 11 a 13, este ciclo está codificado y tiene una sintaxis general de la siguiente forma:

```
for (expresión1; expresión2; expresión3;){  
    expresiones;  
}
```

Como se puede observar la palabra reservada para este ciclo es for y su sintaxis contiene tres expresiones: la expresión1 regularmente representa un contador en el valor inicial y se ejecuta únicamente una vez al iniciar la estructura. La expresión2 regularmente representa la condición, que verifica si el contador ha alcanzado el valor final. La expresión3 regularmente representa un incremento o decremento del contador para que vaya acercando al valor final y se ejecuta, después realizar las expresiones del ciclo.

RETO:

Utilizando como referencia los ejemplos anteriores y la estructura de control de repetición para/hasta, Realizar un diagrama de flujo y las líneas de código para un programa que imprima el factorial de un entero leído del teclado.

```
1 //factorial.c
2 #include <stdio.h>
3
4 main()
5 {
6     long conta, num, factorial = 1;
7
8     printf("Introduce un entero positivo o cero: ");
9     scanf("%d", &num);
10
11     for(conta=1; conta<=num; conta++){
12         factorial *= conta;
13     }
14
15     printf("\t%d! = %d", num, factorial);
16
17     return 0;
18 }
```

R.