



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
ESCUELA NACIONAL
COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL SUR
ACADEMIA DE MATEMÁTICAS**



**GUÍA PARA PREPARAR EL EXAMEN EXTRAORDINARIO
DE LA ASIGNATURA DE CIBERNÉTICA Y COMPUTACIÓN I**

**SEMINARIO INSTITUCIONAL
PARA DAR SEGUIMIENTO A LA APLICACIÓN DIDÁCTICA DE LOS PROGRAMAS
DE TALLER DE CÓMPUTO Y DE CIBERNÉTICA Y COMPUTACIÓN**

OCTUBRE 2005

INTRODUCCIÓN

La presente guía tiene como propósito orientarte en tu estudio para presentar con éxito el examen extraordinario de CIBERNÉTICA Y COMPUTACIÓN I.

En la guía encontrarás información acerca de los aprendizajes, estrategias de aprendizaje, temario, actividades de autoevaluación y bibliografía que, conforme al programa de estudio vigente, serán considerados en el examen extraordinario de la asignatura.

Para que obtengas mejores resultados durante tu estudio es conveniente que utilices la guía de la siguiente manera:

TEMARIO DE ESTUDIO

En este listado identifica los aprendizajes que consideras que ya dominas, así como aquellos que desconoces o te resultan particularmente difíciles. Elabora un plan tentativo de trabajo, con días y cantidad de horas que dedicarás al estudio y repaso de los temas, para que logres los aprendizajes.

ACTIVIDADES DE APRENDIZAJE

Realiza cada una de las actividades de aprendizaje sugeridas y elabora siempre un resumen o cuadro sinóptico destacando los aspectos más relevantes del aprendizaje. Esto te permitirá organizar tus conocimientos y ubicar los puntos principales de estudio y facilitará tus repasos. Ajusta tu plan inicial de trabajo de acuerdo a la importancia relativa a cada aprendizaje, según a tus aptitudes o dificultades para su estudio.

De manera particular deberás prestar atención a lo siguiente:

Para el logro de los aprendizajes es recomendable que realices todas las actividades contenidas en la guía, esto te permitirá avanzar en forma gradual con los aprendizajes planteados en el programa de la asignatura. En cuanto a la resolución de problemas, aplica la metodología de solución de problemas propuesta y realiza la prueba de escritorio de los diagramas de flujo o pseudocódigos, con la finalidad de que obtengas aprendizajes sólidos en la resolución de problemas con el apoyo de la computadora.

La guía contiene respuestas de algunas actividades de aprendizaje, las restantes contéstalas con base en tus conocimientos o bien con la consulta de la bibliografía recomendada. Además, contiene reactivos parecidos a los del examen, contéstalos y verifica tus aciertos con el listado de respuestas que se proporciona en la guía, para que detectes los aspectos que necesitas reforzar.

Para que practiques la Unidad IV del programa, deberás instalar el compilador Free Pascal que acompaña el CD y algunos programas para que ejercites. Instala el compilador Free Pascal para la versión de Windows.

Recuerda que los reactivos de la guía son sólo indicativos del tipo de reactivos que puede contener el examen, y los aprendizajes explorados no excluyen otros (considerados en el programa de estudio vigente) no abordados en esta muestra.

BIBLIOGRAFÍA

Consulta para cada unidad los libros sugeridos en la guía. Puedes utilizar cualquier otro libro con el cual te sientas a gusto, ya que la bibliografía recomendada puede ser complementada y ampliada con libros que tú ya tengas o hayas utilizado anteriormente al estudiar estos temas.

TEMARIO DE ESTUDIO

El programa de la asignatura de Cibernética y Computación I, corresponde al Área de Matemáticas del Plan de Estudios vigente y contiene cuatro unidades temáticas, las cuales se especifican a continuación:

Unidad I. La Cibernética.

Origen y objetivos de la cibernética.

Sistemas.

- Concepto.
- Tipos.
- Diseño.
- Aplicaciones.
- Control y comunicación.
 - ✓ Retroalimentación.
 - ✓ De información.

Modelos.

- Naturales y artificiales.
- Analógicos y digitales.
- Híbridos.
- Matemáticos.

Unidad II. Circuitos lógicos.

- Sistemas de numeración.
- Conversiones entre los sistemas de numeración
- Operaciones de adición, sustracción, división y multiplicación en el sistema binario.
- Operaciones de adición y sustracción en los sistemas octal y hexadecimal.

Ciclo de desarrollo de programas

- Planteamiento del problema.
- Análisis del problema.
- Elaboración de algoritmos.
- Codificación, edición y compilación.
- Ejecución y depuración.
- Documentación.
- Mantenimiento.

Expresiones y operadores

- Asignación.
- Operadores aritméticos.
- Operadores relacionales.
- Operadores lógicos.
- Precedencia de operadores y evaluación de expresiones.

Técnicas de desarrollo de algoritmos

- Diseño descendente.
- Refinación progresiva de solución.
- Seudocódigo y diagrama de flujo.
- Prueba de escritorio.

Unidad IV. Introducción a la programación y al lenguaje de programación Pascal.

Lenguajes de programación

Elementos del álgebra de Boole.

- Conjunción, disyunción y negación.
- Tablas de verdad de las funciones booleanas.

Compuertas y circuitos lógicos

- Componentes.
- Circuito en serie y en paralelo.
- Interruptores.
- Compuertas lógicas.
- Representación de las compuertas lógicas.
- Funciones booleanas.
- Representar la función booleana a partir de una tabla de verdad y/o circuito lógico.
- Construcción o simulación de circuitos lógicos.

Unidad III. Metodología de Solución de Problemas.

Definiciones y conceptos generales

- Problema.
- Elementos y relaciones del problema.
- Herramientas computacionales para la solución de problemas.
 - ✓ *Hardware.*
 - ✓ Sistema operativo.
 - ✓ Programas de aplicación.
 - ✓ Lenguajes de programación.

- Clasificación de los lenguajes.

- ✓ Lenguaje máquina.
- ✓ Lenguaje ensamblador.
- ✓ Lenguaje de alto nivel: imperativo, funcional, declarativo, orientado a objetos, entre otros.

- Características básicas de cada nivel.

- Sintaxis y semántica básicas.

Sistema Operativo

- Importancia.
- Características.
- Comandos.

Elementos del lenguaje

- Estructura de un programa.
- Variables y constantes.
- Identificadores.
- Palabras reservadas.
- Tipos de datos primitivos.
 - ✓ Numéricos.
 - ✓ Caracter y cadena.
 - ✓ Lógicos.

- Sentencias:

- ✓ Lectura, asignación, escritura.
- ✓ Estructura de la sentencia condicional simple (If-then).

Diagramas sintácticos básicos de las sentencias

Ambiente de trabajo

- Editor.
 - ✓ Comandos básicos.
 - ✓ Edición, compilación y generación de código.
 - ✓ Manejo de archivos.

UNIDAD I. LA CIBERNÉTICA

Introducción

La presente unidad te da la oportunidad tener información sobre la Cibernética, sus principales personajes, conocerla como una ciencia interdisciplinaria, reflexionar sobre el significado del concepto de sistemas, hacer analogías entre sistemas naturales y artificiales, con el fin de representarlos en modelos.

Propósito

Al finalizar la unidad el alumno obtendrá una visión general de la cibernética mediante el estudio y análisis de sistemas naturales y artificiales para el diseño de sistemas.

Aprendizajes:

- Describirás los antecedentes históricos del origen de la cibernética y personajes que contribuyeron a su desarrollo.
- Comprenderás la cibernética como ciencia interdisciplinaria.
- Compararás los sistemas naturales y artificiales con el fin de comprender sus similitudes y diferencias.
- Identificarás los elementos de los sistemas.
- Diseñarás un sistema.

Estrategia

- Investigación sobre el origen de la cibernética, sus principales personajes, y cuales fueron las principales disciplinas que intervinieron.
- Investigación sobre que es un sistema, las partes que constituyen un sistema, los diferentes sistemas que existen, un subsistema, la retroalimentación, la entropía, la homeostasis, un sistemas de control y un modelo.

Actividad de aprendizaje

Actividad 1. Con base en la investigación del tema que realizaste contesta las siguientes preguntas:

¿Qué es la Cibernética? _____

¿Cuál fue el origen de la Cibernética? _____

A que científico se le considera el padre de la Cibernética: _____

¿Por que? _____

¿De que disciplinas surge la Cibernética? _____

¿Qué científico mexicano participo en el desarrollo de la cibernética? _____

Da una definición de la Cibernética _____

Un sistema es: _____

La entrada el proceso y la salida son las partes que constituyen un _____

Un sistema _____ Interactúa con su medio ambiente

Un sistema _____ no recibe entradas ni recibe salidas

Los órganos del sistema respiratorio pulmones, boca, nariz, bronquios, laringe, traquea y pulmones ¿Cual es su función en el cuerpo humano?

Da otro ejemplo de subsistema del cuerpo humano _____

¿Qué es la retroalimentación? _____

La retroalimentación _____ es un sistema de control para mantener al sistema regulado o estable.

La retroalimentación _____ lo aleja de la estabilidad

Un sistema para regular su funcionamiento que lo hace pasar de un estado a otro y asegurar su estabilidad tiene un mecanismo de _____

El transmisor o fuente, Receptor y el Canal que lleva el comunicado desde el transmisor al receptor son partes de un sistema de _____

Los modelos se clasifican en dos _____

El control de una lavadora automática es un ejemplo de modelo _____

El control de la temperatura en los seres vivos es un ejemplo de modelo _____

Escribe tres similitudes entre los modelos naturales y artificiales _____

El reloj, el radio, son ejemplos de modelos _____

La computadora, el fax, impresora son ejemplo de modelos _____

El velocímetro mecánico de un coche es un ejemplo de modelo _____

Actividad 2. En base a la investigación que realizaste relaciona de manera correcta las columnas

1. Retroalimentación	() N. Wiener planteó que la organización debía concebirse como "una interdependencia de las distintas partes organizadas, pero una interdependencia que tiene grados. Ciertas interdependencias internas deben ser más importantes que otras, lo cual equivale a decir que la interdependencia interna no es completa"
2. Organización	() El segundo principio de la termodinámica establece el crecimiento de la entropía, es decir, la máxima probabilidad de los sistemas es su progresiva desorganización y, finalmente, su homogeneización con el ambiente. Los sistemas cerrados están irremediabilmente condenados a la desorganización. No obstante hay sistemas que, al menos temporalmente, revierten esta tendencia al

	aumentar sus estados de organización
3. Entrada o insumo o impulso (input)	() Este concepto está especialmente referido a los organismos vivos en tanto sistemas adaptables. Los procesos homeostáticos operan ante variaciones de las condiciones del ambiente, corresponden a las compensaciones internas al sistema que sustituyen, bloquean o complementan estos cambios con el objeto de mantener invariante la estructura sistémica, es decir, hacia la conservación de su forma. La mantención de formas dinámicas o trayectorias se denomina homeorrosis (sistemas cibernéticos).
4. Entropía	() Son los procesos mediante los cuales un sistema abierto recoge información sobre los efectos de sus decisiones internas en el medio, información que actúa sobre las decisiones (acciones) sucesivas. La retroalimentación puede ser negativa o positiva. Mediante los mecanismos de retroalimentación, los sistemas regulan sus comportamientos de acuerdo a sus efectos reales y no a programas de outputs fijos. En los sistemas complejos están combinados ambos tipos de corrientes (circularidad homeostasis).
5. Homeostasis	() Es la fuerza de arranque del sistema, que provee el material o la energía para la operación del sistema.

Respuestas: 2, 4, 5, 1, 3

Actividad de Aprendizaje

Actividad 3. En base a la investigación que realizaste, identifica con una N si el sistema es natural, con una A si es artificial y completa la siguiente Tabla.

Tipo de Sistema	Conjunto de elementos	Dinámicamente relacionados	Características (Formando una actividad, con un objetivo, operando y proveyendo)
Solar ()	Planetas, lunas, sol, cometas, asteroides,.....,	Orbitas de los cuantificadas,....,	Atmósfera, gravedad, temperatura
Operativo ()			
Numeración decimal ()	Números	Operaciones de adición, sustracción, multiplicación, división, propiedades de orden,....,	La base del sistema numérico es de diez elementos, posición,....,
Nervioso ()			
Transporte Colectivo (Metro) ()			
Agua Potable de la Ciudad de México ()			
Hardware de la máquina ()			
Básico de entrada y Salida (BIOS) ()			
Político ()			
Ecosistema ()			

AUTOEVALUACIÓN

Indicaciones: Lee con mucha atención la pregunta, selecciona la respuesta correcta y colócala dentro del paréntesis.

1.- La palabra Cibernética significa: ()

- a) La ciencia del control y de la comunicación utilizada en común por hombres, Máquinas para alcanzar un determinado fin.
- b) Un conjunto de elementos que interactúan para lograr un objetivo común.
- c) Cuando se producen las salidas del sistema vuelven a ingresar al sistema como información.
- d) Una palabra almacenada en la memoria que contiene datos
- e) La estabilidad de los sistemas naturales y artificiales.

2.- Relaciona las columnas según consideres la respuesta:

1. Cual de los siguientes precursores de la computación, desarrollo el equipo electromecánico de tarjetas perforadas	() Arturo Rosenblueth
2. Es uno de los personajes más importantes que formularon a la Cibernética como una ciencia	() Hermann Hollerith
3. Primer mexicano que sentó las bases de la Cibernética junto con Norbert Wiener	() Norbert Wiener

3.-Es un conjunto de elementos que interactúan entre si para lograr un objetivo común ()

- a) Bus de control
- b) canal de dirección
- c) un sistema
- d) un control
- e) la teoría

4.- El sistema que interactúa con su medio ambiente es: ()

- a) auto corrige
- b) Abierto
- c) Cerrado
- d) de procesamiento
- e) comunicativo

5.- El sistema que no recibe entradas ni produce salidas es: ()

- a) de computadora
- b) de comandos
- c) cerrado
- d) de procesamiento
- e) abierto

6.- Para regular su funcionamiento un sistema tiene: ()

- a) un mecanismo de control
- b) un mecanismo de conducir
- c) un sistema abstracto y determinado
- d) un mecanismo de proceso
- e) un mecanismo de comunicación

7.- Se produce cuando las salidas del sistema vuelven a ingresar al sistema como información y como mecanismo de control es la: ()

- a) organización
- b) comunicación
- c) evaluación
- d) dirección
- e) retroalimentación

8.- El control de la temperatura en el cuerpo de los seres vivos es un modelo:

- a) matemático
- b) sistema nervioso
- c) Sistema dinámico
- d) Sistema natural
- e) Sistema artificial

9.- Es un modelo de sistema artificial:

- a) el control del sistema nervioso
- b) el control del sistema circulatorio
- c) el control del sistema escolar
- d) el control del sistema respiratorio
- e) el control del sistema de temperatura de la plancha

Respuestas: 1. a, 2. [3,1,2], 3. c, 4. b, 5. c, 6. a, 7. c, 8. d, 9. c.

Bibliografía:

Wiener Norbert. *Cibernética ó El Control y Comunicación en Animales y Máquinas*, Barcelona, Tusquets editores, 1968.

Lectura complementaria:

Sabih, W. Ross. *Introducción a la cibernética*, Buenos Aires, Nueva Visión. 1977
Crosson, F. Y Sayre K. *Filosofía y Cibernética*, México, fondo de Cultura Económica, 1971

Ruyer Raymond. *La Cibernética y el Origen de la Información*, México, Fondo de Cultura Económica, 1984.

Electrónica.

<http://dblinux.sis.epn.edu.ec/elascano/componentes/historia/historia.html>

(www.yahoo.com.mx "sistemas")

<http://www.monografias.com/trabajos/teosis/teosis.shtml>

UNIDAD 2. CIRCUITOS LOGICOS

INTRODUCCIÓN

La Cibernética es la ciencia que se ocupa de los sistemas de control y de comunicación en las personas y en las máquinas, estudiando y aprovechando todos sus aspectos y mecanismos comunes de diferentes ciencias como la mecánica, electrónica, medicina, física, química y computación.

El programa de estudios actualizado marca a Cibernética como materia de quinto y sexto semestre.

PROPÓSITO DE LA UNIDAD

El alumno utilizara algunos elementos del Álgebra de Boole y Circuitos Integrados para el diseño, la construcción y simulación de circuitos lógicos mediante el desarrollo de practicas.

APRENDIZAJES

- 1.- **Conocerá y manejará** la metodología para transformar a los diferentes sistemas numéricos en particular el sistema binario.
- 2.- **Indicará** el papel de la cibernética en las otras áreas del conocimiento, **describirá** la relación de la cibernética con la computadora y **sistematizara** la información obtenida sobre la cibernética.
- 3.- **Representará** los circuitos serie paralelo, **describirá** una compuerta lógica AND, OR, **construirá** físicamente los circuitos serie paralelo mediante interruptores y elaborara las tablas de verdad de las compuertas AND y OR.
- 4.- **Identificará** los elementos básicos de un circuito integrado, **Conocerá** el procedimiento para la construcción de las tablas de verdad y el funcionamiento de la tableta de experimentación, implementara la polarización de la tableta de experimentación y de los circuitos integrados.
- 5.- **Implementará** en la tableta de experimentación el medio sumador y verificara la tabla de verdad del circuito.
- 6.- **Construirá y verificara** la tabla de verdad del sumador completo.

SECUENCIA DE ACTIVIDADES

- 1.- Investigará, conocerá y transformará cantidades a diferentes sistemas numéricos la solución se dará en una sesión plenaria.
- 2.- Investigación documental sobre el desarrollo de la cibernética y su relación con otras disciplinas, Exposición por equipos de la investigación realizada, Discusión colectiva de los conceptos de cibernética y presentación de diferentes dispositivos automáticos y no automáticos.
- 3.- Investigación documental de circuitos serie paralelo y de compuertas lógicas, exposición por equipo de los conceptos investigados, discusión grupal de los circuitos lógicos, construcción de los circuitos serie paralelo y de sus tablas de verdad.
- 4.- Implementación física de los circuitos serie y paralelo con interruptores eléctricos verificación de la tabla de verdad y analogía básica con las compuertas lógicas.
- 5.- Conocimiento y manejo de las compuertas AND; OR reducción de elementos y construcción de la tabla de verdad del circuito del medio sumador su implementación física y verificación de las tabla resultante.
- 6.-Implementación del sumador completo su representación física y la verificación de la tabla de verdad del circuito.

Cuestionario de sistemas de numeración.

¿Qué es un sistema de numeración ? _____

¿Qué es el sistema de numeración binario? _____

¿Cuáles son los elementos que constituyen el sistema binario? _____

¿Qué es el sistema de numeración octal? _____

¿Cuáles son los elementos que constituyen el sistema octal? _____

¿Qué es el sistema de numeración decimal? _____

¿Cuáles son los elementos que constituyen el sistema decimal? _____

¿Qué es el sistema de numeración hexadecimal? _____

¿Cuáles son los elementos que constituyen el sistema hexadecimal? _____

Completa la siguiente tabla

DECIMAL	BINARIO	OCTAL	HEXADECIMAL
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Describe el proceso mediante el cual puedes convertir un número del sistema decimal a cualquier otro sistema de numeración.

Efectúa las siguientes conversiones:

$$(1678)_{10} = (\quad)_2$$

$$(2456786)_{10} = (\quad)_8$$

$$(350718705)_{10} = (\quad)_{16}$$

Describe el proceso mediante el cual puedes convertir un número de cualquier sistema de numeración al sistema decimal.

Efectúa las siguientes conversiones:

$$(11000110)_2 = (\quad)_{10}$$

$$(20156)_8 = (\quad)_{10}$$

$$(4A0B5)_{16} = (\quad)_{10}$$

Efectúa las siguientes conversiones de una base directamente:

$$(1110100110101)_2 = (\quad)_8$$

$$(10100110101001111)_2 = (\quad)_{16}$$

$$(1000011010011111011110)_2 = (\quad)_8$$

$$(4501670)_8 = (\quad)_2$$

$$(300150121)_8 = (\quad)_{16}$$

$$(507601356)_8 = (\quad)_2$$

$$(1F09)_{16} = (\quad)_2$$

$$(9CA1953)_{16} = (\quad)_8$$

$$(535904)_{16} = (\quad)_2$$

Efectúa las siguientes operaciones:

$$110001101011_2 + 100111101_2 =$$

$$100010111111_2 - 11011111_2 =$$

$$11101100110_2 \times 1000111_2 =$$

$$1110011111_2 \div 1011_2 =$$

$$24567401_8 + 54327_8 =$$

$$3560124_8 - 3452_8 =$$

$$490A65C_{16} + 6573F_{16} =$$

$$53DC65A_{16} - 245E4_{16}$$

Cuestionario de circuitos lógicos.

¿Qué es una expresión algebraica? _____

¿Qué es una variable algebraica? _____

¿Cuáles son los operadores aritméticos básicos? _____

Escribe tres ejemplos de expresiones aritméticas:

¿Qué es una expresión lógica? _____

Escribe tres ejemplos de expresiones lógicas:

¿Qué es una variable lógica? _____

¿Cuáles son los operadores lógicos básicos? _____

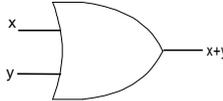
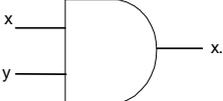
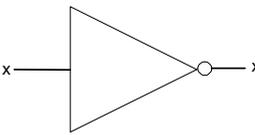
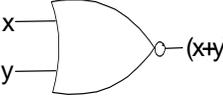
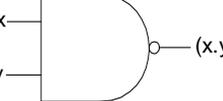
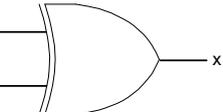
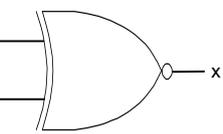
¿Qué es una compuerta lógica? _____

¿Qué es un circuito lógico? _____

¿Qué es un circuito integrado? _____

¿Qué es el álgebra booleana? _____

Compuertas lógicas digitales

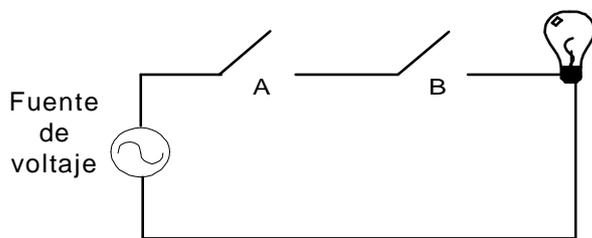
Compuerta	Símbolo	Función algebraica
OR		$F = x + y$
AND		$F = x \cdot y$
NOT		$F = x'$
NOR		$F = (x + y)'$
NAND		$F = (x \cdot y)'$
OR exclusiva		$F = xy' + x'y$ $F = x \oplus y$
NOR exclusiva		$F = xy + x'y'$ $F = (x \oplus y)'$

Práctica: Prueba de circuitos eléctricos.

Lista de materiales por equipos de cuatro alumnos:

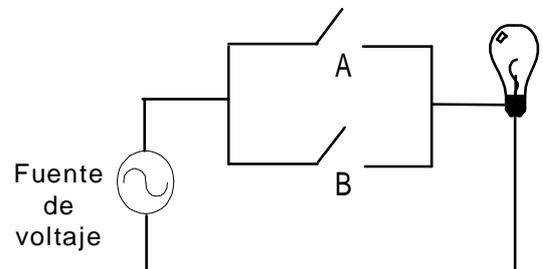
- Un panel eléctrico
- 4 interruptores eléctricos
- 2 soquets
- 2 focos
- 2 metros de cable

Enseguida se presentan dos circuitos eléctricos, los cuales deberás construir utilizando los materiales necesarios.



Representación de la compuerta AND con interruptores en serie.

Este circuito corresponde a la compuerta



Representación de la compuerta OR con interruptores en paralelo.

Este circuito corresponde a la compuerta

Circuitos integrados

7400	7402	7404	7408	7432	7486
NAND	NOR	NOT	AND	OR	XOR

Práctica: Prueba de circuitos.

Lista de materiales por equipos de cuatro alumnos:

Una tableta de experimentación simple.

Circuitos integrados de la tabla 1, dos de cada tipo.

Pila de 9 volt o fuente de alimentación con múltiples salidas a 3, 6, 9 y 12 volts.

Dos caimanes con alambres

Diez leds (diodos emisores de luz)

Diez resistencias de 300 Ω .

Alambre del 22 en colores rojo, negro, amarillo, verde y azul, un metro para cada color.

Pinza para cortar y pelar alambre.

Prueba de circuitos integrados:

- 1) Inserta dos alambres en cualquiera de las cuatro esquinas de la tableta de experimentación, alambre rojo para corriente y alambre negro para tierra, respetando los colores en la misma (rojo para corriente y azul para tierra).
- 2) Conecta con caimanes los alambres de la tableta a la fuente de alimentación máximo 6 volts) o a la pila, respetando el rojo para corriente y el azul para tierra .
- 3) Verifica la continuidad de la corriente en la tableta, par ello, inserta en la clavija adjunta al alambre de color rojo, el ánodo (pata más larga del led) a corriente (+), y en la clavija adjunta al alambre de color negro, inserta el cátodo (pata más corta del led) a tierra (-), repite lo mismo en ambos bordes de la tableta que vienen marcados con los colores rojo y azul, de ser necesario utiliza puentes para dar continuidad de corriente a la tableta.
- 4) Inserta el circuito 7408 en el canal de la tableta de experimentación.
- 5) Polariza el circuito, para ello, conecta con alambre de color negro la pata 7 a tierra y la pata 14 con alambre de color rojo a corriente.
- 6) Inserta las entradas x, y al circuito 7408, para ello, en la clavija adjunta a la pata 1, inserta el extremo de un alambre y en la clavija adjunta a la pata 2, inserta el extremo de otro alambre de diferente color al primero.
- 7) Inserta el ánodo de un led en la clavija adjunta a la entrada de la pata 1 e inserta el cátodo a tierra.
- 8) Inserta el ánodo de un led en la clavija adjunta a la entrada de la pata 2 e inserta el cátodo a tierra.

- 9) Inserta el ánodo de un led en la clavija adjunta a la salida (pata 3) e inserta el cátodo a tierra.
- 10) Coloca una resistencia a cada led, para ello, inserta a cada extremo de ella en la clavija adjunta al ánodo y el otro a tierra.
- 11) Inserta los otros extremos de los alambres de entrada a corriente y tierra indistintamente.

¿Qué entradas produce una salida de 1? _____

¿Qué entradas produce una salida de 0? _____

¿Qué señal manda el led cuando las entradas son uno? _____

¿Qué señal manda el led cuando las entradas son cero? _____

Construye la tabla de verdad del circuito 7408.

Nota: La prueba de los circuitos restantes es similar a la realizada, inténtalo, si tienes dudas pide ayuda a tu compañero u profesor y construye las tablas de verdad.

Tabla de verdad del circuito AND.

x	y	x.y
1	1	
1	0	
0	1	
0	0	

Tabla de verdad del circuito OR.

x	y	x+y
1	1	
1	0	
0	1	
0	0	

Tabla de verdad del circuito NAND

x	y	(x.y)'
1	1	
1	0	
0	1	
0	0	

Tabla de verdad del circuito NOR

x	y	(x+y)'
1	1	
1	0	
0	1	
0	0	

Tabla de verdad del circuito XOR

Tabla de verdad del circuito NOT

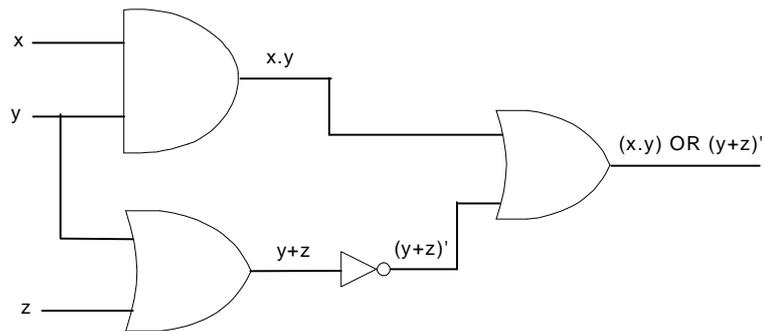
x	y	x + y
1	1	
1	0	
0	1	
0	0	

x	x'
1	
0	

Funciones booleanas

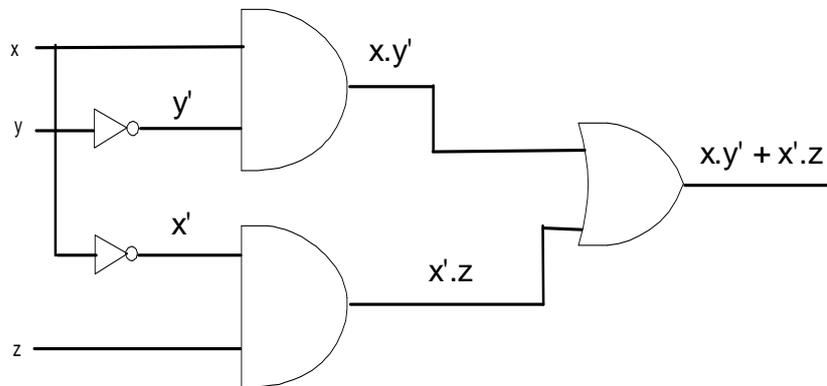
Una función booleana es una expresión formada por variables booleanas, los operadores binarios (AND, OR, NAND, NOR, XOR y NOR exclusiva), el operador unitario NOT, paréntesis y el signo de igual. Para un valor dado de las variables, la función puede ser 0 o bien 1. Los siguientes diagramas, ilustran funciones booleanas.

Función booleana $(x \cdot y) + (y + z)'$



x	y	z	x.y	y+z	(y+z)'	(x.y) + (y+z)'
1	1	1	1	1	0	1
1	1	0	1	1	0	1
1	0	1	0	1	0	0
1	0	0	0	0	1	1
0	1	1	0	1	0	0
0	1	0	0	1	0	0
0	0	1	0	1	0	0
0	0	0	0	0	1	1

Función booleana $xy' + x'z$



Ejercicio: Completa la tabla de verdad

x	y	z	x'	y'	x.y'	x'.z	xy' + x'.z
1	1	1					
1	1	0					
1	0	1					
1	0	0					
0	1	1					
0	1	0					
0	0	1					
0	0	0					

Ejercicio: Dada la tabla de verdad de la función booleana $(x + y) \cdot (x' + z)$ construye su diagrama.

x	y	z	x'	y'	x + y'	x' + z	$(x+y).(x'+z)$
1	1	1	0	0	1	1	1
1	1	0	0	0	1	0	0
1	0	1	0	1	1	1	1
1	0	0	0	1	1	0	0
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	0
0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1

Medio sumador y sumador completo

Las computadoras digitales realizan variados procesos de cómputo, entre los que destacan las operaciones aritméticas, de ellas, la suma de dos dígitos binarios es la más simple y cumple con las siguientes reglas:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

De lo anterior observamos que la suma en los tres primeros casos es de un dígito binario, mientras que en el cuarto caso, la suma, requiere de dos dígitos binarios para representarla. El dígito más significativo recibe el nombre de acarreo y el bit menos significativo el de suma. Un circuito que lleva a cabo la adición de dos bits se denomina **medio sumador**. Uno que lleva a cabo la suma de tres bits (dos bits significativos y el acarreo) se denomina un **sumador completo**.

Medio Sumador.

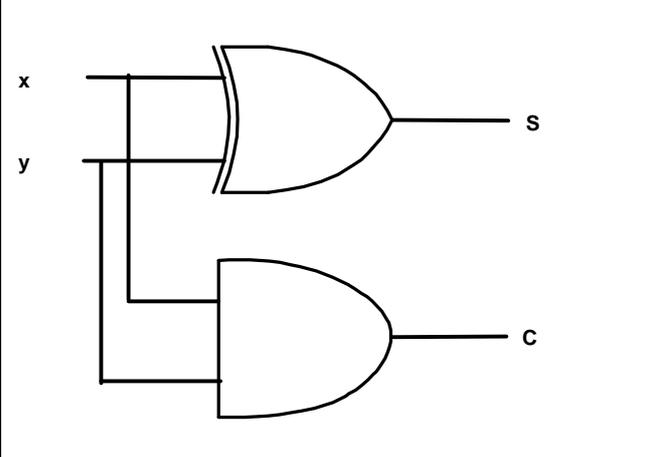
<p>Por lo anterior el medio sumador utiliza dos bits de entrada, el sumando y el adendo y regresa como salida dos funciones booleanas, una para la suma y la otra para el acarreo. En la tabla de verdad adjunta, x e y son las entradas al circuito, C es la función que representa al acarreo y S es el bit menos significativo de la suma.</p>	<p>Tabla de verdad: Medio Sumador</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>x</th> <th>y</th> <th>C</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	C	S	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	0
x	y	C	S																		
0	0	0	0																		
0	1	0	1																		
1	0	0	1																		
1	1	1	0																		

¿Cuál es el operador booleano que representa a la función acarreo? _____

¿Cuál es el operador booleano que representa a la función suma? _____

El circuito lógico del medio sumador, se presenta a continuación:

Diagrama lógico del medio sumador:

	Completa la tabla de verdad:																				
	<table border="1" data-bbox="895 600 1267 837"><thead><tr><th>x</th><th>y</th><th>C</th><th>S</th></tr></thead><tbody><tr><td>1</td><td>1</td><td></td><td></td></tr><tr><td>1</td><td>0</td><td></td><td></td></tr><tr><td>0</td><td>1</td><td></td><td></td></tr><tr><td>0</td><td>0</td><td></td><td></td></tr></tbody></table>	x	y	C	S	1	1			1	0			0	1			0	0		
x	y	C	S																		
1	1																				
1	0																				
0	1																				
0	0																				

Práctica: Prueba del medio sumador.

Material requerido:

Pila de 9 volts o fuente de alimentación de salidas múltiples 3, 6, 9 y 12 volts.

Tableta de experimentación (protoboard).

Circuitos integrados 7408 y 7486.

Diodos emisores de luz (leds).

Resistencias a 270 - 310 Ω .

Alambre del 22 en varios colores.

Procedimiento para la prueba del medio sumador

- 1) Inserta los circuitos 7486 y 7408 en la tableta de experimentación.
- 2) Polariza a ambos circuitos, recuerda que la patas 7 y 14 de cada circuito se insertan a tierra y a corriente, respectivamente.
- 3) Inserta los extremos de dos alambres (entradas x, y) en las clavijas adjuntas a las patas 1 y 2 del circuito 7486.
- 4) Conecta las entradas x, y al circuito 7408, para ello, inserta en las clavijas adjuntas de las patas 1 y 2 del circuito 7486, los extremos de dos alambres y los otros dos extremos insértalos en las clavijas adjuntas a las patas 1 y 2 del circuito 7408.

- 5) Inserta el ánodo de un led en la clavija adjunta a la pata 1 del circuito 7486 y el cátodo a tierra.
- 6) Inserta el ánodo de un led en la clavija adjunta a la pata 2 del circuito 7486 y el cátodo a tierra.
- 7) Inserta el ánodo de un led en la clavija adjunta a la pata 3 del circuito 7486 y el cátodo a tierra.
- 8) Inserta el ánodo de un led en la clavija adjunta a la pata 3 del circuito 7408 y el cátodo a tierra.

¿Qué representa la salida del circuito 7486? _____

¿Qué representa la salida del circuito 7408? _____

¿En qué casos se tiene acarreo? _____

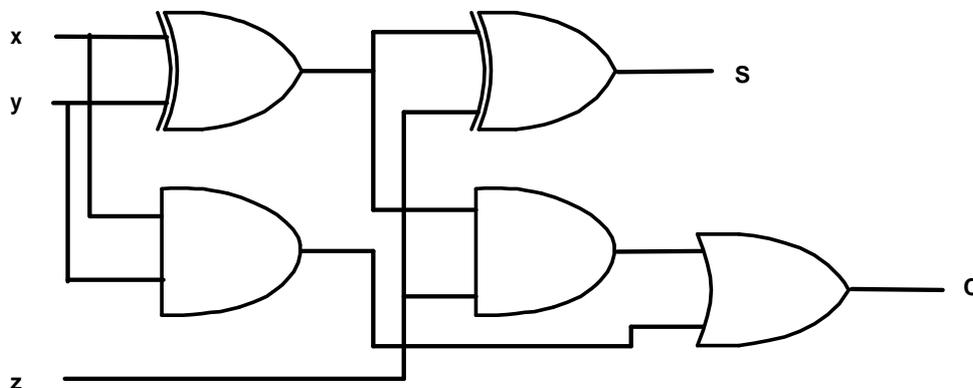
¿En qué casos no se tiene acarreo? _____

Práctica: Prueba del sumador completo.

Un sumador completo es un circuito lógico que procesa la suma aritmética. Consta de tres bits de entrada y dos bits de salida, dos de las entradas son bits significativos y la tercera es el bit del acarreo, una de las salidas corresponde a la suma y la otra al acarreo, representadas con S y C, respectivamente.

¿Qué se requiere para el diseño de un sumador completo? _____

El diseño lógico del sumador completo, lo podemos hacer con dos medios sumadores, tal como se ilustra a continuación:



Del circuito lógico anterior, observamos que la suma y el acarreo son dos funciones booleanas, escríbelas.

Función suma = _____

Función acarreo = _____

Completa la tabla de verdad de la función suma:

x	Y	z	$x \oplus y$	$(x \oplus y) \oplus z$
1	1	1		
1	1	0		
1	0	1		
1	0	0		
0	1	1		
0	1	0		
0	0	1		
0	0	0		

Completa la tabla de verdad de la función acarreo:

x	y	z	$x.y$	$x \oplus y$	$(x \oplus y) . z$	$(x.y) + (x \oplus y).z$
1	1	1				
1	1	0				
1	0	1				
1	0	0				
0	1	1				
0	1	0				
0	0	1				
0	0	0				

Tabla de verdad: Sumador Completo.

Funciones booleanas que representa a la suma y al acarreo.

Escribe la función booleana que representa al acarreo:

x	y	z	C	S
1	1	1	1	1
1	1	0	1	0
1	0	1	1	0
1	0	0	0	1
0	1	1	1	0
0	1	0	0	1
0	0	1	0	1
0	0	0	0	0

Escribe la función booleana asociada a la suma:

Práctica: Prueba del sumador.

Material requerido

Fuente de alimentación con salidas múltiples a 3, 4.5, 6, 9 y 12 volts.

Tableta de experimentación (protoboard).

Circuitos integrados 7408 (dos), 7486 (dos) y 7432.

Diodos emisores de luz (leds).

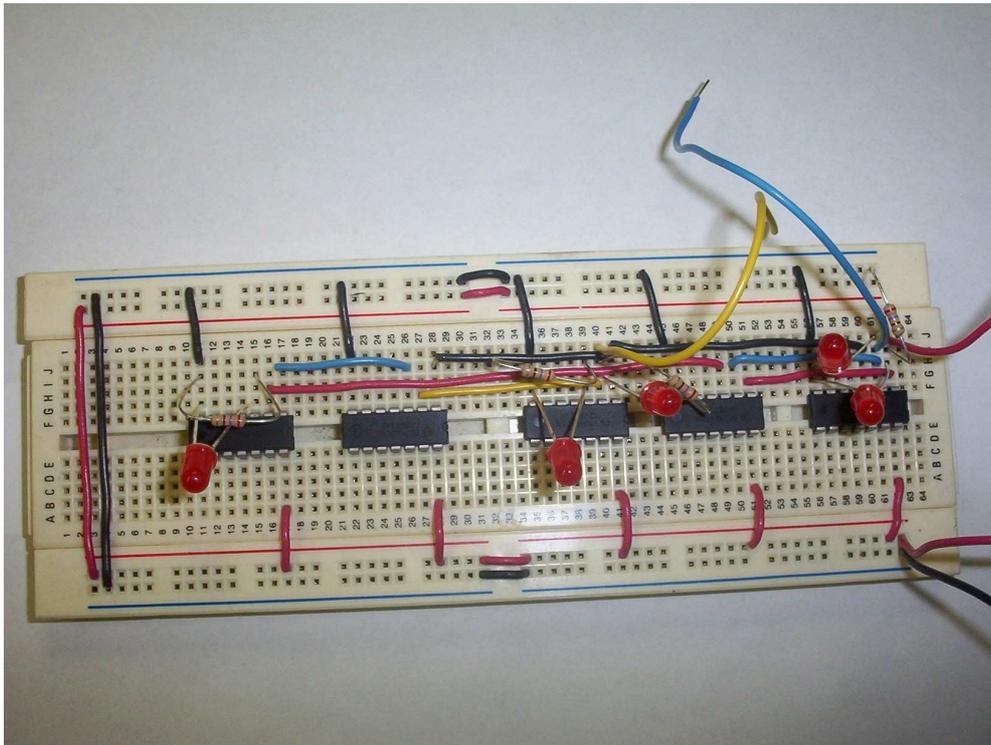
Resistencias a 270 - 310 Ω .

Alambre del 22 en varios colores.

Procedimiento para la prueba del sumador

1. Inserta los circuitos integrados 7486, 7408, 7486, 7408 y 7432 a la tableta de experimentación.
2. Polariza los cinco circuitos, recuerda que la pata 7 de cada circuito se conecta a tierra y la pata 14 a corriente.
3. Inserta el extremo de dos alambres (entradas x, y bits significativos) en las clavijas adjuntas a las patas 1 y 2 del primer circuito 7486.
4. Conecta las entradas x, y del primer circuito 7408), para ello, inserta el extremo de dos alambres, desde las clavijas adjuntas a las patas 1 y 2 del primer circuito 7486 a las clavijas adjuntas a las patas 1 y 2 del primer circuito 7408, respectivamente.
5. Conecta las entradas al segundo circuito 7486, para ello, inserta el extremo de un alambre desde la clavija adjunta a la salida del primer circuito 7486 a la clavija adjunta a la pata 1 del segundo circuito 7486 e inserta el extremo de otro alambre, desde la entrada z (acarreo) a la clavija adjunta a la pata 2 del segundo circuito 7486.
6. Conecta las entradas al segundo circuito 7408, para ello, inserta el extremo de un alambre, desde la clavija adjunta a la salida del primer circuito 7486 a la pata 1 del segundo circuito 7408 e inserta el extremo de otro alambre, desde la entrada z (acarreo) a la pata 2 del segundo circuito 7408.

7. Inserta las entradas al circuito 7432, para ello, conecta alambres desde las clavijas adjuntas a las salidas del primer y segundo circuito 7408 a las clavijas adjuntas a las patas 1 y 2 del circuito 7432, respectivamente.
8. Coloca un led en cada una de las entradas x, y, z, recuerda que el ánodo se inserta en los orificios adjuntos a las entradas y el cátodo a tierra.
9. Inserta el ánodo de un led en la clavija adjunta a la salida del segundo circuito 7486 y el cátodo a tierra.
10. Inserta el ánodo de un led en la clavija adjunta a la salida del circuito 7432 y el cátodo a tierra.
11. Conecta las entradas x, y, z a corriente o a tierra indistintamente.



Escribe las entradas para que la suma y el acarreo sean 1: _____

Escribe las entradas para que la suma sea 1 y el acarreo 0: _____

Escribe las entradas para que la suma sea 0 y el acarreo 1: _____

Escribe las entradas para que la suma y el acarreo sean 0: _____

Proyectos para ser desarrollados

Contador decimal.

Contador binario.

CONCLUSIONES

Dado que dentro de los propósitos generales del curso de Cibernética y Computación I, se encuentra el que corresponde a que el alumno utilizará algunos elementos del álgebra de Boole y circuitos lógicos, y que para esta unidad temática se amplía detallando su uso para el diseño, construcción o simulación de algunos autómatas mediante el desarrollo de prácticas.

Es claro que para alcanzar estos propósitos se diseñaron aprendizajes que si bien tienen un contenido algorítmico como el de las conversiones de números de un sistema de numeración a otro, la realización de operaciones en el sistema binario y operaciones booleanas, también contiene los que corresponden a las relaciones entre operadores booleanos con las compuertas lógicas y las funciones y sobre todo los que permiten y obligan a los alumnos a realizar prácticas de circuitos eléctricos y lógicos con sus correspondientes tablas de verdad.

Además se puede observar que los contenidos y metodología, permiten tanto al profesor como al alumnos cumplir cabalmente con los aprendizajes que se pretenden alcanzar y que proporcionarán a los alumnos obtener un nivel adecuado de conocimientos para afrontar las dificultades que pudieran presentarse en la metodología de solución de problemas y sobre todo poder aprovechar las inquietudes y creatividad de los alumnos en el diseño y elaboración de algunos autómatas.

Cabe también hacer mención que este material tiene como finalidad proporcionar los recursos didácticos que ayuden para el alcance de los aprendizajes.

Actividades de aprendizaje. Escribe en el paréntesis la letra que corresponda con la respuesta correcta.

1. El número $(219)_{10}$, es en base 2 equivalente a: ()

- A) 1101101
- B) 1011011
- C) 1101101
- D) 11011011
- E) 100100011

2. El hexadecimal $9F_{16}$, es equivalente en base 8, al número: ()

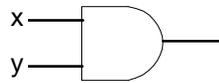
- A) 237
- B) 273
- C) 327
- D) 372
- E) 723

3. El producto binario siguiente, es: ()

$$\begin{array}{r} 111 \\ \times 111 \\ \hline \end{array}$$

- A) 101100
- B) 111000
- C) 110001
- D) 111111
- E) 101001

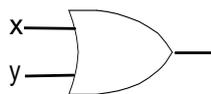
4. Relaciona ambas columnas, asociando la compuerta con su tabla de verdad.()



a

x	y	x + y
1	1	1
1	0	1
0	1	1
0	0	0

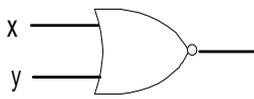
I



b

x	y	x . y
1	1	1
1	0	0
0	1	0
0	0	0

II



c

x	y	x ⊕ y
1	1	0
1	0	1
0	1	1
0	0	0

III



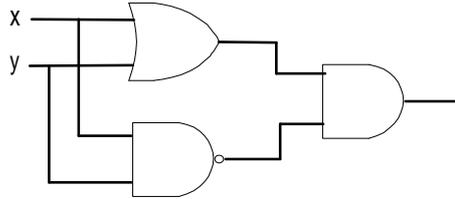
d

x	y	(x+y)'
1	1	0
1	0	0
0	1	0
0	0	1

IV

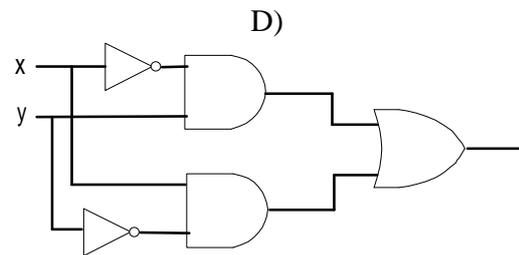
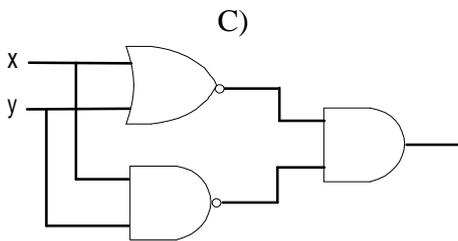
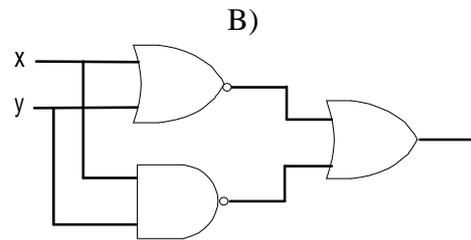
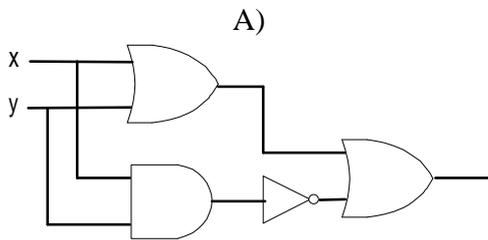
- A) a - IV, b - I, c - II, d - III.
- B) a - II, b - I, c - IV, d - III.
- C) a - III, b - IV, c - I, d - II.
- D) a - I, b - II, c - III, d - IV.
- E) a - II, b - III, c - IV, d - I.

5. La expresión lógica que corresponde al circuito de la figura, es: ()

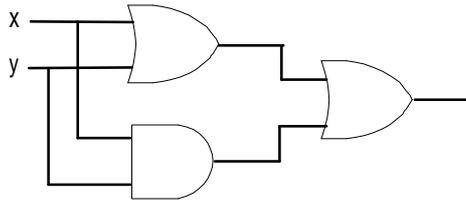


- A) $(xy) \cdot (x + y)$
- B) $(x + y) + (xy)'$
- C) $(x + y) + (xy)'$
- D) $(x + y)' + (xy)$
- E) $(x + y) \cdot (xy)'$

6. Para la expresión lógica $(x+y)' + (xy)'$, el circuito lógico correspondiente, es: ()



E)



7. El valor de la expresión $(x+y)+(xy)'$ es: ()

x	y	A	B	C	D	E
1	1	0	0	1	1	1
1	0	0	1	1	0	1
0	1	0	1	1	1	1
0	0	0	1	1	1	0

Respuestas

1	2	3	4	5	6	7
B	A	C	B	E	B	C

Bibliografía

M. Morris Mano, Diseño digital, PHH. México 1987.

Ronald J. Tocci, Sistemas digitales, Principios y aplicaciones, PHH. México 1993.

UNIDAD 3. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS

Introducción

En la presente unidad resolverás problemas que te permitirán formalizar el proceso de solución con las herramientas computacionales, para que te apropiés de la metodología de solución de problemas.

Propósito

Al finalizar la unidad, aplicarás la metodología de solución de problemas mediante la construcción de algoritmos para formalizar el proceso de solución.

Aprendizajes que lograrás en la unidad:

- Identificarás los elementos que intervienen en los problemas y las relaciones entre ellos para obtener los resultados.
- Describirás las características de las etapas que intervienen en la solución de problemas.
- Construirás el algoritmo, el diagrama de flujo y el pseudocódigo para los problemas planteados.
- Realizarás la prueba de escritorio de los algoritmos desarrollados.

Para alcanzar el propósito y los aprendizajes especificados, realizarás investigaciones sobre metodología de solución de problemas, misma que aplicarás en la solución de problemas. Los problemas están seleccionados de manera que primero se abordan las estructuras secuenciales (lectura, asignación y escritura); luego las estructuras selectivas (condicional simple, condicional doble y condicional múltiple); y finalmente las repetitivas (while, repeat y for).

Estrategias de enseñanza.

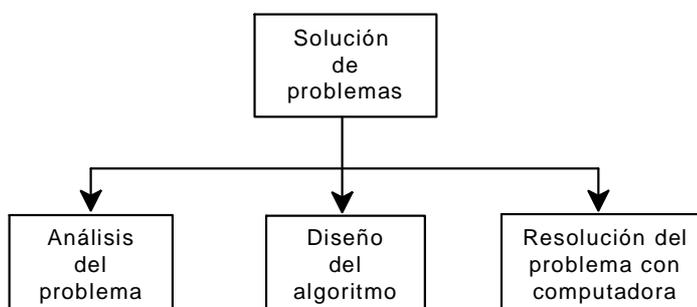
Investigación sobre el ciclo de desarrollo de programas.
Planteamiento y solución de de problemas.

Estrategias de aprendizaje.

Describirás las etapas en la solución de problemas.
Identificarás los elementos y las relaciones que intervienen en el problema.
Desarrollarás el diagrama de flujo asociado a los problemas planteados.
Elaborarás el pseudocódigo asociado a los problemas planteados.
Realizarás la prueba de escritorio del diagrama de flujo o pseudocódigo.

Metodología para la solución de problemas.

Para resolver problemas con la computadora y las herramientas de programación, es conveniente que te apoyes de la metodología de la solución de problemas que se propone en seguida:

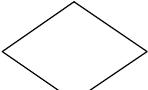


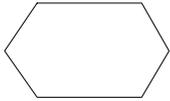
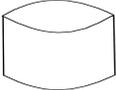
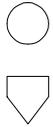
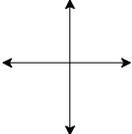
Actividad de aprendizaje. Describe los conceptos faltantes en la tabla.

Conceptos	Descripción
Análisis del problema	
Algoritmo y características	Es un conjunto de pasos lógicos que permiten la solución de problemas. El algoritmo es preciso (indica el orden de realización en cada paso), definido (al seguir el algoritmo varias veces se obtiene el mismo resultado) y finito (tiene un fin, número determinado de pasos).
Diseño del algoritmo	Es el proceso que convierte los resultados del análisis en un diseño modular (descendente) con refinamiento sucesivo (divide al problema en cada etapa y expresa a cada paso en forma más detallada), que permitan una posterior traducción a un lenguaje.
Diagramas de flujo	
Pseudocódigo	Es un lenguaje para la especificación de algoritmos, cuyas instrucciones se representan mediante palabras en el lenguaje natural, el español o inglés, para facilitar tanto la lectura como escritura de programas.
Prueba de escritorio	
Codificación	Es el proceso que consiste en la traducción de las instrucciones del diagrama de flujo o pseudocódigo al lenguaje de programación.

Edición	
Compilación	El proceso de compilación, acepta como entrada el programa fuente y checa la sintaxis de las instrucciones del mismo, de no existir errores de sintaxis, se genera el programa objeto escrito en el lenguaje de máquina, listo para su ejecución, en caso contrario dicho programa no es generado hasta que quede exento de errores.
Ejecución	El proceso de ejecución acepta como entrada el programa objeto que se generó en el proceso de compilación y realiza la ejecución del mismo.
Comprobación	Proceso que consiste en verificar que los resultados obtenidos en la corrida sean correctos.
Documentación	Proceso que consiste en la descripción de los pasos a dar en la solución del problema. La documentación puede ser interna y externa. La interna es mediante comentarios al programa para clarificarlos y la externa consiste en el manual de usuario.
Mantenimiento	

Actividad de aprendizaje. Escribe el significado y descripción de los símbolos gráficos faltantes en la tabla.

Símbolo	Significado	Descripción
	Inicio / Fin	Se utiliza para señalar el inicio y el fin de un conjunto de procesos relacionados con la computadora.
		
	Procesamiento por computadora	Se utiliza para indicar cualquier clase de procesamiento realizado por el sistema de cómputo.
		

	<p style="text-align: center;">Selección</p>	<p>Se utiliza para la seleccionar una opción de entre varias alternativas con el objeto de especificar la acción subsecuente en la solución del problema.</p>
		
	<p style="text-align: center;">Subprograma</p>	<p style="text-align: center;">Se utiliza para indicar un módulo dentro de un programa.</p>
		
	<p style="text-align: center;">Conectores</p>	<p>Se utilizan para unir partes en los diagramas de flujo, el círculo los conecta en la misma página, mientras que el pentágono, los conecta en páginas diferentes.</p>
		

Primera etapa. Problemas que involucran las estructuras secuenciales

Las sentencias de lectura, asignación y escritura forman las estructuras secuenciales, mismas que aplicaremos en la solución de los tres problemas: salario bruto, viaje y promedios.

Problema salario bruto. Encuentra el salario bruto semanal (sal_bruto) que la empresa “Terabyte”, pagará a uno de sus empleados, considerando el nombre del empleado (nomb), las horas trabajadas a la semana (hor_trab) y la tarifa por hora (tar_hor). Deberá visualizarse como salida el nombre del empleado, las horas trabajadas, la tarifa por hora y el salario bruto.

Análisis del problema salario bruto.

Estrategia de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo
nomb	cadena	<i>sal_brut</i> ←		
tar_hor	entero			

Diseño del algoritmo del problema salario bruto.

<p>Actividad de aprendizaje. Escribe en el diagrama de flujo, las sentencias faltantes del problema salario bruto.</p> <div style="text-align: center; margin: 20px 0;"> <pre> graph TD Inicio([Inicio]) --> Leer[/Leer
nomb, hor_trab,
tar_hor/] Leer --> Proc[] Proc --> Escribir[/Escribir
nomb, hor_trab,
tar_hor, sal_brut/] Escribir --> Fin([Fin]) </pre> </div>	<p>Actividad de aprendizaje. Escribe en las rayas en blanco, las sentencias del diagrama de flujo, faltantes en el pseudocódigo.</p> <p style="margin-left: 20px;">Inicio</p> <p style="margin-left: 40px;">leer nomb</p> <p style="margin-left: 40px;">_____</p> <p style="margin-left: 40px;">leer tar_hor</p> <p style="margin-left: 40px;">_____</p> <p style="margin-left: 40px;">escribir nomb</p> <p style="margin-left: 40px;">escribir hor_trab</p> <p style="margin-left: 40px;">escribir tar_hor</p> <p style="margin-left: 40px;">_____</p> <p style="margin-left: 20px;">Fin</p>
---	--

Prueba de escritorio del problema salario bruto.

Actividad de aprendizaje. Realiza la prueba de escritorio del problema salario bruto con los datos de la tabla y escríbelos en los renglones faltantes.

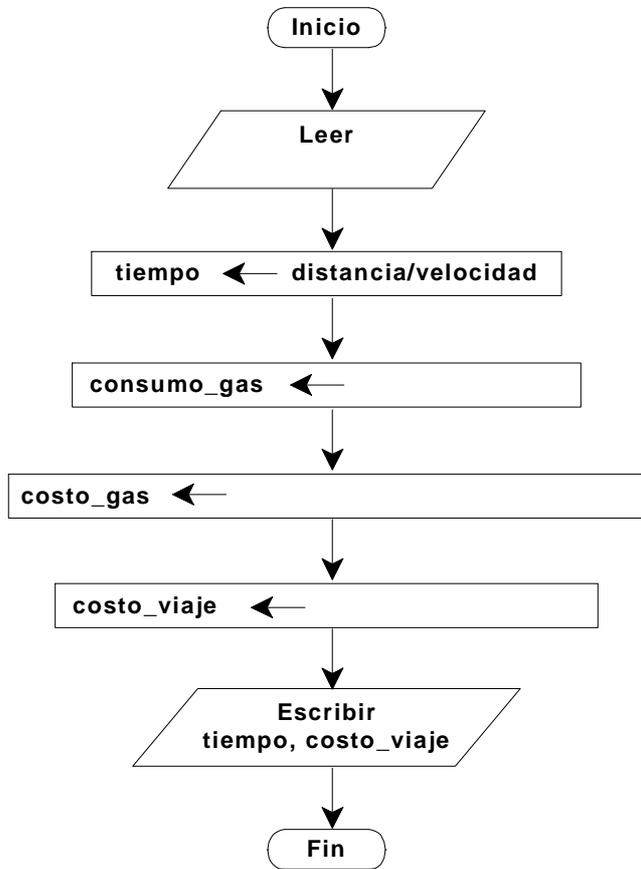
nomb	hor_trab	tar_hor	sal_brut
Raquel	40	50	
Anabel	40	45	1800
Abinaí	40	55	

Problema viaje. Encuentra el tiempo y costo de un viaje en automóvil que la familia Sánchez, realiza del Distrito Federal al Puerto de Acapulco (tiempo, costo_viaje), durante el recorrido pagará en total \$ 450 de cuotas en las casetas ubicadas en el trayecto. Además, se sabe que la distancia entre ambas ciudades es de aproximadamente 400 kilómetros (distancia), la velocidad promedio del auto es de 90 kilómetros por hora (velocidad). El rendimiento del automóvil (rend) y el costo por litro de gasolina (costo_litrogas) deberán ser introducidas con el teclado. Se entiende por rendimiento del automóvil a los kilómetros recorridos por litro de gasolina.

Actividad de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo
	entero	$tiempo \leftarrow \frac{distancia}{velocidad}$	tiempo	real
Costo_litrogas	real	$consumo_gas \leftarrow \frac{distancia}{rend}$		real
		$costo_gas \leftarrow$		
		$costo_viaje \leftarrow$		

Actividad de aprendizaje. Escribe en el diagrama de flujo, las sentencias faltantes del problema viaje.



Actividad de aprendizaje. Escribe en las rayas, las sentencias del diagrama de fluido, faltantes en el pseudocódigo.

Inicio

 $tiempo \leftarrow \frac{distancia}{velocidad}$
 $consumo_gas \leftarrow \frac{distancia}{rend}$
 $costo_gas$ _____
 $costo_viaje$ _____
 escribir tiempo
 escribir costo_viaje
 Fin

Actividad de aprendizaje. Realiza la prueba de escritorio del problema viaje con los datos de la tabla y escríbelos en los renglones faltantes.

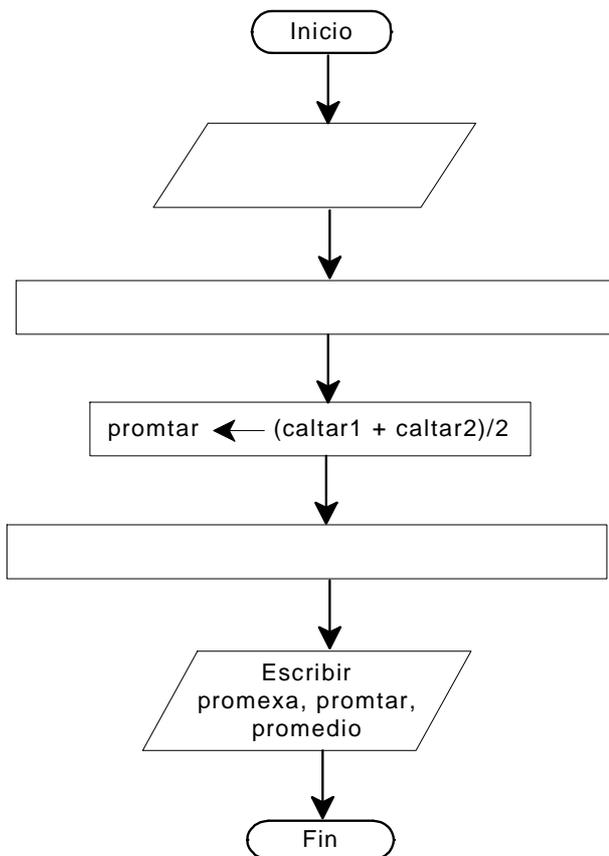
rend	costo_litrogas	tiempo	consumo_gas	costo_gas	costo_viaje
12	6.50	4.4	33.3	216.6	666.6
8	6.75				
10	6.80				

Problema promedio. Calcula el promedio académico de un alumno de Cibernética y Computación del CCH en una escala de diez (promedio), considerando el nombre del alumno (nombre), la calificación de tres exámenes (calexa1, calexa2, calexa3), la calificación de dos tareas (caltar1, caltar2) y los porcentajes del 70% y 30% para exámenes y tareas, respectivamente.

Actividad de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo
calexa1	entero	$promexa \leftarrow \frac{calexa1 + calexa2 + calexa3}{3}$	promedio	real
		$promtar \leftarrow$		
		$promedio \leftarrow$		
caltar2	entero			

Actividad de aprendizaje. Escribe en el diagrama de flujo, las sentencias faltantes del problema promedio.



Actividad de aprendizaje. Escribe en las rayas, las sentencias del diagrama de flujo, faltantes en el pseudocódigo.

Inicio

$$promexa \leftarrow \frac{calexa1 + calexa2 + calexa3}{3}$$

$promtar \leftarrow$

$promedio \leftarrow$

escribir promexa

Fin

Segunda etapa. Problemas que involucran estructuras selectivas.

En la solución de los problemas anteriores, aplicaste las estructuras secuenciales, mismas que retomarás para el tratamiento de las estructuras selectivas con la finalidad de que avances en la apropiación de la metodología de solución de problemas. La sentencia condicional simple, doble y múltiple, forman las estructuras selectivas, mismas que aplicarás en la solución de problemas, que dan entrada de manera estelar su aplicación y cuya función consiste en la elección de una alternativa en el proceso de solución, tal como se ilustran con la solución de los siguientes problemas:

Problema salario neto. Encuentra el salario semanal neto (*sal_net*) que la empresa “Terabyte”, pagará a uno de sus empleados considerando el nombre del empleado (*nombre*), las horas trabajadas a la semana (*hor_trab*), la tarifa por hora (*tar_hora*) y la tasa de impuestos de acuerdo a las siguientes categorías (*cat*), para la categoría A el impuesto es del 5% (*imp_a*), para la categoría B el impuesto es del 10% (*imp_b*) y para la categoría C el impuesto es del 15% (*imp_c*). Deberá visualizarse como salida el nombre del empleado, las horas trabajadas, la tarifa por hora, el salario bruto, la categoría y el salario neto.

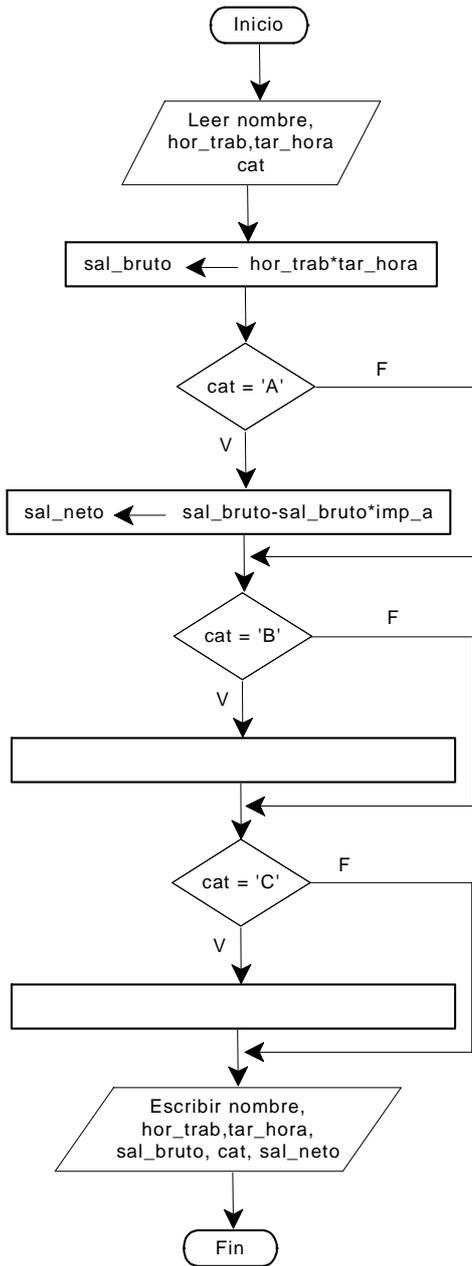
Análisis del problema salario neto.

Actividad de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo
nombre	cadena	<i>sal_brut</i> ←		
hor_trab	entero	<i>sal_net</i> ←		

Diseño del algoritmo del problema salario neto.

Actividad de aprendizaje. Escribe en el diagrama de flujo, las sentencias del problema salario neto.



Actividad de aprendizaje. Escribe en las rayas, las sentencias del diagrama de flujo, faltantes en el pseudocódigo.

Inicio

leer nombre
leer hor_trab

sal_bruto ← hor_trab * tar_hora
si (cat="A") o (cat='a') entonces
 sal_netto ← sal_bruto - sal_bruto * imp_a
fin_si

Si (cat='C') o (cat='c') entonces
 sal_netto ← sal_bruto - sal_bruto * imp_c
fin_si
escribir nombre
escribir hor_trab
escribir tar_hora

Fin

Prueba de escritorio del problema salario neto.

Actividad de aprendizaje. Realiza la prueba de escritorio del problema salario neto con los datos de la tabla y escríbelos en los renglones faltantes.

nomb	hor_trab	tar_hor	sal_brut	cat	sal_net
Raquel	40	50		A	
Anabel	40	45	1800	B	1620
Abinaí	40	55		C	

Problema deporte. Selecciona el deporte que es apropiado practicar, considerando como dato la temperatura (temp) en grados Fahrenheit con base en la siguiente tabla:

Deporte	Temperatura
Natación	temp mayor que 85
Tenis	temp mayor que 70, pero menor o igual a 85.
Golf	temp mayor que 32, pero menor que 70.
Esquí	temp mayor que 10, pero menor o igual a 32.
Marcha	temp menor o igual que 10

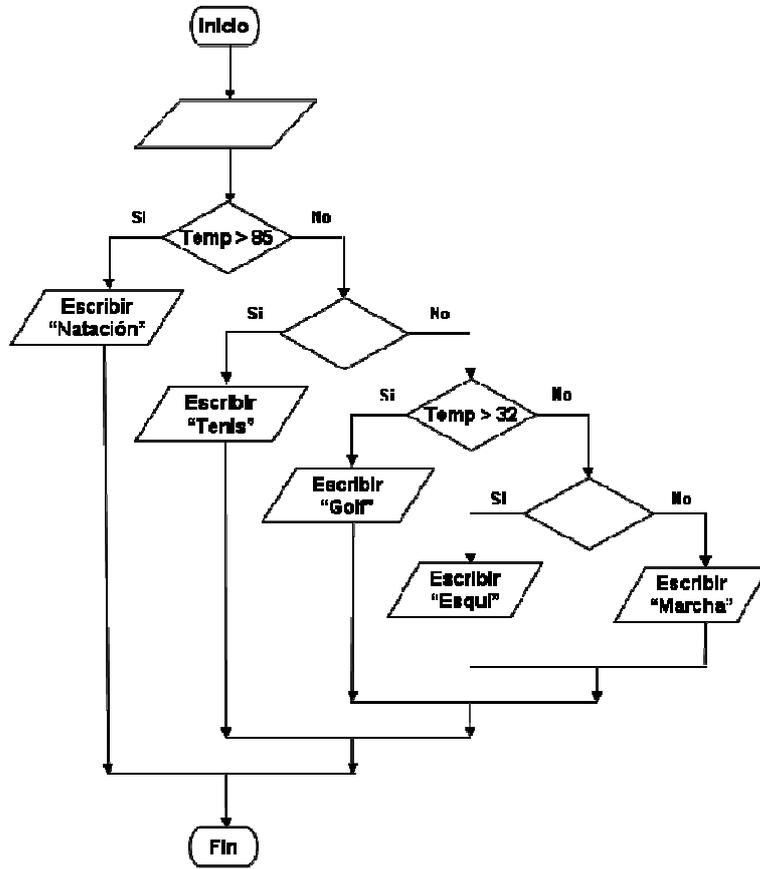
Análisis del problema deporte.

Actividad de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo

Diseño del algoritmo del problema deporte.

Actividad de aprendizaje. Escribe las sentencias faltantes en el diagrama de flujo del problema deporte.



Actividad de aprendizaje.

Escribe en las rayas, las sentencias del diagrama de flujo, faltantes en el pseudocódigo.

Inicio

Si (temp > 85) entonces
Escribir ("Natación")
Sino

Si (temp > 32) entonces
Escribir("Golf")
Sino

Fin_temp>10
Fin_temp>32
Fin_temp>70
Fin_temp>85

Fin

Prueba de escritorio del problema deporte.

Actividad de aprendizaje. Realiza la prueba de escritorio del problema deporte con los datos de la tabla y escríbelos en los renglones faltantes.

Temperatura	Deporte
temp mayor que 85	
temp mayor que 70, pero menor o igual a 85.	
temp mayor que 32, pero menor que 70.	
temp mayor que 10, pero menor o igual a 70.	
temp menor o igual que 10	

Problema días. Escribir en pantalla el nombre del día de la semana, en función de la variable entera día (día).

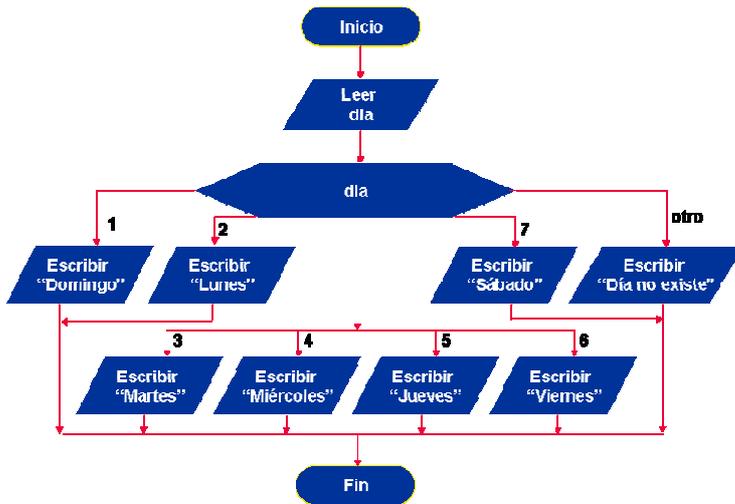
Análisis del problema días.

Actividad de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo

Diseño del algoritmo del problema días.

Actividad de aprendizaje. Escribe las sentencias en el diagrama de flujo del problema días.



Actividad de aprendizaje.

Escribe en las rayas, las sentencias del diagrama de flujo, faltantes en el pseudocódigo.

Inicio

Según temp hacer

1: Escribir ("Domingo")

2: Escribir ("Lunes")

7: Escribir ("Sábado")

Otro

Escribir ("Día no existe")

Fin_según

Fin

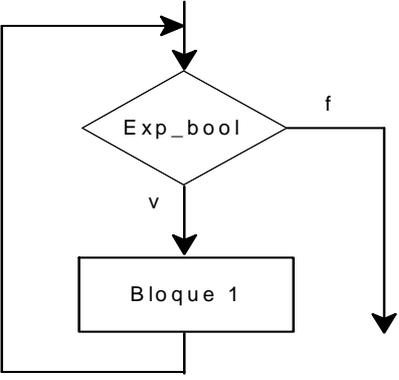
Prueba de escritorio del problema días.

Actividad de aprendizaje. Realiza la prueba de escritorio del problema días con los datos de la tabla y escríbelos en los renglones faltantes.

dia	Nombre del día de la semana							
1	Domingo							
2		Lunes						
3								
4								
5								
6								
7							Sábado	
otro								Día no existe

Tercera etapa. Problemas que involucran estructuras repetitivas.

En la solución del problema aplicaste las estructuras secuenciales y selectivas, mismas que retomarás para el tratamiento de las estructuras repetitivas con la finalidad para que avances en la apropiación de la metodología de solución de problemas. Las sentencias repeat, while y for forman las estructuras repetitivas, mismas que aplicarás en la solución de problemas que dan entrada de manera estelar su aplicación, cuya función consiste en la repetición de una o varias sentencias dependiendo de la condición de ciclo, tal como se muestra en su diagrama y pseudocódigo.

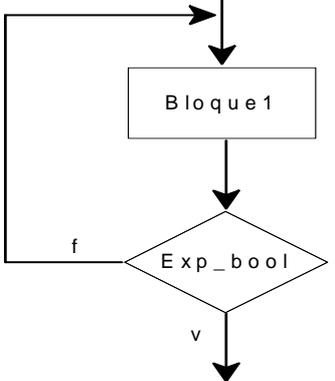
<p>Diagrama de flujo: sentencia while</p> 	<p>Descripción</p> <p>While es una sentencia de ciclo, utilizada para la repetición de una o varias instrucciones, dependiendo del valor lógico de la condición de ciclo (exp_bool), cuando el valor lógico es verdadero se ejecuta la sentencia(s) contenida(s) en bloque 1, en caso contrario, se obtiene la salida del ciclo while, tal como se observa en el diagrama de flujo.</p>
--	---

Pseudocódigo en español

Mientras expbool **hacer**
Bloque1
Fin_mientras

Pseudocódigo en inglés

While expbool do
Bloque1
End_while

<p>Diagrama de flujo: sentencia repeat</p> 	<p>Descripción</p> <p>Repeat es una sentencia de ciclo, utilizada para la repetición de una o varias instrucciones, dependiendo del valor lógico de la condición de ciclo (exp_bool), cuando el valor lógico es falso se ejecuta (n) la(s) sentencia(s) contenida(s) en bloque 1, en caso contrario, se obtiene la salida del ciclo repeat, tal como se observa en el diagrama de flujo.</p>
---	--

Pseudocódigo en español

Repite
Bloque1
Hasta que expbool = v

Pseudocódigo en inglés

Repeat
Bloque1
Until expbool = v

Diagrama de flujo de la sentencia for	Descripción
<pre> graph TD A[vi ← valor inicio] --> B[vf ← valor final] B --> C[vc ← vi] C --> D{vc > vf} D -- v --> E[] D -- f --> F[acciones] F --> G[vc ← vc + 1] G --> D </pre>	<p>For es una sentencia de ciclo, utilizada para la repetición de una o varias instrucciones, dependiendo del valor lógico de la condición de ciclo ($vc > vf$), cuando el valor lógico es falso se ejecuta (n) la(s) sentencia(s) contenida(s) en acciones, en caso contrario, se obtiene la salida del ciclo for, tal como se observa en el diagrama de flujo.</p> <p>Pseudocódigo en español</p> <p>Para vc desde vi hasta vf hacer Acciones Fin _ desde</p> <p>Pseudocódigo en inglés</p> <p>For $vc := vi$ to vf do Acciones End_for</p>

Problema salario neto. Encuentra el salario semanal y el salario total neto que la empresa “Terabyte”, pagará a cada uno de sus cinco empleados considerando el nombre del empleado, horas trabajadas a la semana, tarifa por hora y la tasa de impuestos de acuerdo a las siguientes categorías, para la categoría A el impuesto es del 5%, para la categoría B el impuesto es del 10% y para la categoría C el impuesto es del 15%. El problema planteado da entrada de manera estelar a las sentencias repetitivas. Para su comprender sus características, el problema se resuelve con la sentencia de ciclo while, intenta la solución con las sentencias repeat y for.

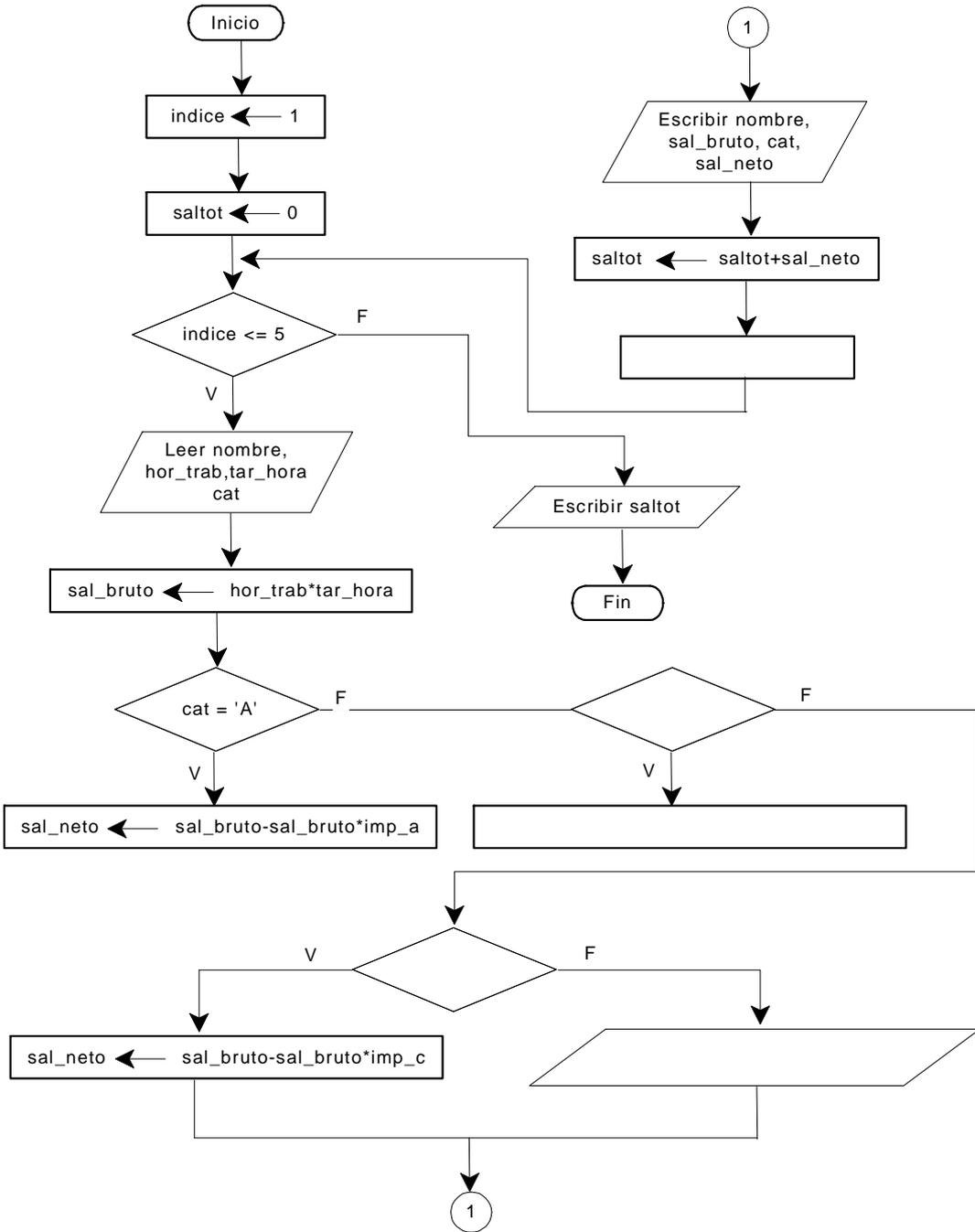
Análisis del problema salario neto.

Actividad de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo
nombre	cadena	<i>sal_brut</i> ←		
hor_trab	entero	<i>sal_neto</i> ←		

Diseño del algoritmo del problema salario neto con la sentencia while.

Actividad de aprendizaje. Escribe las sentencias faltantes en el diagrama de flujo del problema salario neto.



Actividad de aprendizaje. Escribe en las rayas, las sentencias del diagrama de flujo, faltantes en el pseudocódigo.

Inicio

Indice ← 1

saltot ← 0

Mientras (indice ≤) hacer

 Inicio

 leer (nom)

 leer (hortrab)

 leer (tarhor)

 leer (cat)

 salbrut ← hortrab*tarhor

 Si (cat="A") o (cat="a") entonces

 salnet ← salbrut-salbrut*imp_a

 Sino

 Escribir (nom)

 Escribir (salbrut)

 Escribir (cat)

 Escribir (salnet)

 saltot ← saltot+salnet

 indice ← indice+1

 Fin_mientras

 Escribir (saltot)

Fin

Prueba de escritorio del problema salario neto.

Actividad de aprendizaje. Realiza la prueba de escritorio del problema salario neto con los datos de la tabla y escríbelos en los renglones faltantes.

nomb	hor_trab	tar_hor	sal_brut	cat	sal_net	sal_tot	indice	indice≤5
Raquel	40	50	2000	A	1900	1900	1	V
Anabel	40	45	1800	B	1620	3520	2	V
Abinaí	40	55		C			3	
Isaura	50	30		C			4	
Edgar	30	40		B			5	
Mallely	60	25		A			6	

Actividad de aprendizaje.

Realiza las adecuaciones pertinentes a la solución del problema de salario neto, utilizando las sentencias de repeat y for, respectivamente.

Problema deporte. Escribir en pantalla el nombre del deporte en función de la variable temperatura (temp), para más de una corrida.

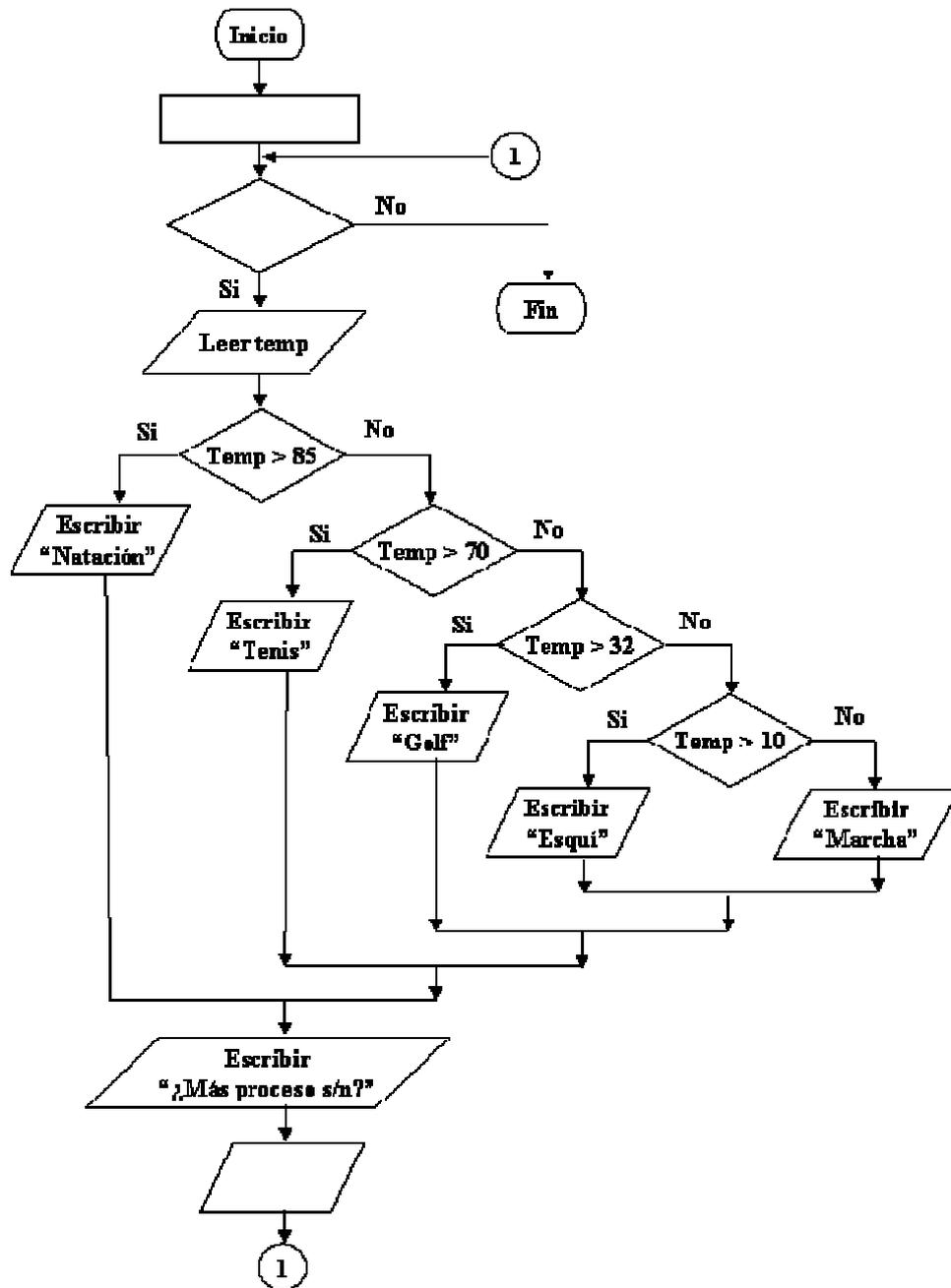
Análisis del problema deporte.

Actividad de aprendizaje. Escribe en la tabla los datos que completan el análisis del problema.

Datos entrada	Tipo	Relaciones	Datos salida	Tipo

Diseño del algoritmo del problema deporte.

Actividad de aprendizaje. Escribe las sentencias en el diagrama de flujo del problema deporte.



Actividad de aprendizaje. Escribe en las rayas, las sentencias del diagrama de flujo, faltantes en el pseudocódigo.

Inicio

```
_____
_____
inicio
  leer temperatura
  si (temp > 85) entonces
    escribir ("Natación")
  sino
    si (temp > 70) entonces
      escribir ("Tenis")
    sino
      si (temp > 32) entonces
        escribir ("Golf")
      sino
        si (temp > 10) entonces
          escribir ("Esquí")
        sino
          escribir ("Marcha")
      fin_10
    fin_32
  fin_70
fin_85
  escribir ("¿Más deporte s/n ?")
```

Fin_mientras

Fin

Prueba de escritorio del problema deporte.

Actividad de aprendizaje. Realiza la prueba de escritorio del problema deporte con los datos de la tabla y escríbelos en los renglones faltantes.

Temperatura	Sigue	Sigue='S') o sigue='s')	Deporte
temp mayor que 85	S	V	Natación
temp mayor que 70, pero menor o igual a 85.	S		
temp mayor que 32, pero menor que 70.	S		
temp mayor que 10, pero menor o igual a 70.	S		
temp menor o igual que 10	S		
	N		

Actividad de aprendizaje. Describe las características de las sentencias de ciclo:

While: _____

Repeat: _____

For: _____

Explica la diferencia entre las sentencias de ciclo: _____

Describe los siguientes conceptos:

Contador: _____

Acumulador: _____

Interruptor: _____

Problemas

Con la finalidad de orientarte para que te apropiés de la metodología de solución de problemas, planteamos otros para que los resuelvas, aplicando tus estrategias de solución.

Problemas propuestos

1. Calcular la suma de los n primeros números naturales con las estructuras repeat, while y for. Para hacerlo repetitivo utiliza las sentencias de ciclo repeat y while.
2. Calcula el factorial de un número n con las sentencias repeat, while y for. Para hacerlo repetitivo utiliza las sentencias de ciclo repeat, while.
3. El señor Pérez tiene la opción de invertir un capital a un interés compuesto anual. Encuentra la inversión después de n años. Para hacerlo repetitivo utiliza las sentencias de ciclo repeat, while.

Para la solución de los problemas planteados, utiliza la metodología de solución de problemas.

Reactivos muestra para el examen extraordinario.

1. Elige en forma correcta ambas columnas.

()

- | | |
|-------------------------|--|
| a. Algoritmo | I Lenguaje para la especificación de algoritmos. |
| b. Diagrama de flujo | II Proceso para seguir paso a paso la lógica del algoritmo. |
| c. Pseudocódigo | III Representación gráfica del algoritmo. |
| d. Prueba de escritorio | IV Conjunto finito de pasos lógicos para la solución del problema. |

- A) a - III, b - IV, c - II, d - I
B) a - I, b - II, c - IV, d - III
C) a - II, b - I, c - III, d - IV
D) a - IV, b - III, c - I, d - II.
E) a - I, b - II, c - III, d - IV.

2. Relaciona ambas columnas en forma correcta.

()

- | | |
|-------------|-----------------------------|
| a. 1ª etapa | I Codificación del programa |
| b. 2ª etapa | II Análisis del problema |
| c. 3ª etapa | III Ejecución del programa |
| d. 4ª etapa | IV Diseño del algoritmo |

- A) a - I, b - II, c - III, d - IV
B) a - II, b - IV, c - I, d - III
C) a - IV, b - III, c - II, d - I
D) a - III, b - I, c - IV, d - II
E) a - II, b - I, c - III, d - IV

3. La traducción de las sentencias de un diagrama de flujo o pseudocódigo, mediante un lenguaje de programación, corresponde a la etapa de:

()

- A) Análisis del problema.
B) Planteamiento del problema.
C) Codificación de programa.
D) Diseño del algoritmo.
E) Edición del programa.

4. El diseño descendente y refinamiento sucesivo, corresponde a:

()

- A) Diseño del algoritmo.
B) Planteamiento del problema.
C) Análisis del problema.
D) Codificación del programa.
E) Ejecución del programa.

5. Relaciona correctamente ambas columnas.

()

- | | |
|----------------|--|
| a. acumulador | I Es una variable que solo toma un valor de dos posibles. |
| b. contador | II Es una variable que incrementa su valor en una cantidad variable. |
| c. interruptor | III Es una variable que incrementa su valor en una cantidad constante. |

- A) a - II, b - III, c - I
B) a - I, b - II, c - III
C) a - III, b - II, c - I
D) a - I, b - III, c - II
E) a - II, b - I, c - III.

6. Al evaluar la expresión $\left(\left(\frac{b+c}{2}\right) * a + 10\right) * 3 * b$ div 6, con $a = 4$, $b = 5$ y $c = 1$, se obtiene: ()

- A) 0
- B) 6
- C) 55
- D) 60
- E) 330

7. Al evaluar la expresión lógica $\left(\left(\frac{b+c}{2}\right) \geq a\right) \text{ or } (2 * b < 5 * a * c)$, con $a = 5$, $b = 8$ y $c = 2$, ¿qué valores se obtienen en los operadores (\geq , or, $<$)? ()

- A) F, V, V
- B) V, F, V
- C) V, V, F
- D) F, F, F
- E) V, V, V.

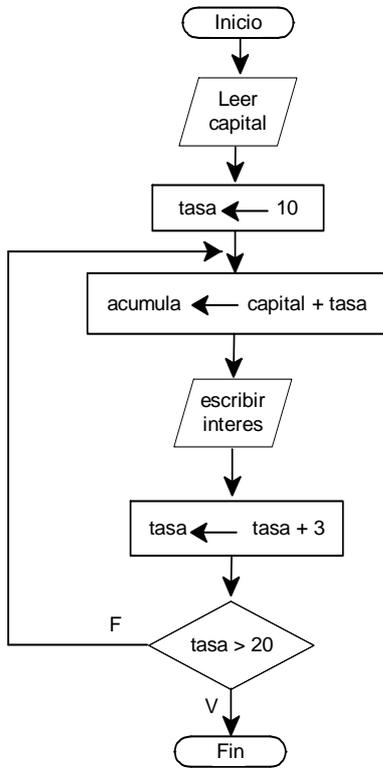
8. En el siguiente pseudocódigo, ¿cuántas veces se ejecuta el ciclo? ()

```

Inicio
  i ← 1
  j ← 1
  mientras ( i < 4 y j < 5) o (i + j = 8) hacer
    inicio_mientras
      i := i + 2
      j := j + 1
    fin_mientras
fin
  
```

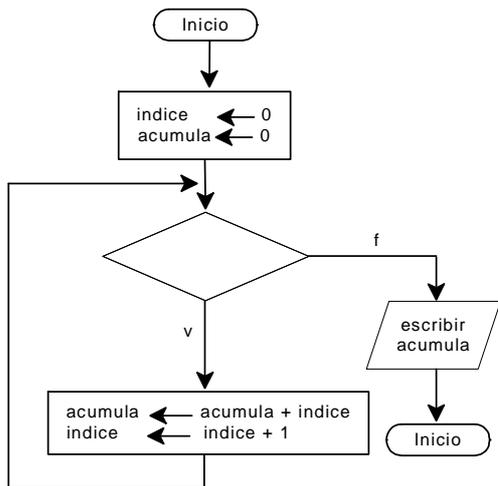
- | | |
|----|----------|
| A) | 1 |
| B) | 2 |
| C) | 3 |
| D) | 4 |
| E) | ∞ |

9. En el siguiente diagrama de flujo, ¿cuántas veces se ejecuta el ciclo? ()



- A) 2
- B) 3
- C) 4
- D) 5
- E) 6

10. En el diagrama de flujo, ¿cuál debe ser la condición de ciclo, para que la sentencia while se ejecute exactamente 5 veces? ()



- A) indice < 4.
- B) indice = 4.
- C) indice ≤ 4.
- D) indice ≠ 4.
- E) indice = 5.

11. En el pseudocódigo, considera que la lectura de límite es 5, ¿cuál es el valor de suma al salir del ciclo? ()

Inicio

```

num ← 0
suma ← 0
leer limite
repetir
    num ← num + 1
    suma ← suma + num
hasta que suma > limite

```

Fin

- | |
|------|
| A) 2 |
| B) 3 |
| C) 4 |
| D) 5 |
| E) 6 |

Respuesta a los reactivos:

Reactivo	1	2	3	4	5	6	7	8	9	10	11
Respuesta	D	B	C	A	A	C	E	C	B	C	E

Bibliografía.

- CAIRÓ, Osvaldo. *Metodología de la Programación. Algoritmos, Diagramas de Flujo y Programas*, México, Alfaomega, 2003.
- JOYANES, Luis. *Fundamentos de la Programación. Algoritmos, Estructura de Datos y Objetos*, México, Mc. Graw-Hill, 2003.
- FUENTES, Romero Gilberto, *Propuesta educativa para la materia de Cibernética y Computación I*, México, CCH, Plantel Sur, 2000.

UNIDAD IV: INTRODUCCIÓN A LA PROGRAMACIÓN Y AL LENGUAJE DE PROGRAMACIÓN PASCAL.

Introducción.

El desarrollo de esta unidad tiene como finalidad reforzar y profundizar los conceptos ya tratados de la metodología de resolución de problemas utilizando la computadora y el lenguaje de programación Pascal. De la unidad anterior retomo los problemas propuestos con sus algoritmos, diagrama de flujo y pseudocódigos mismos que traduciremos al lenguaje de programación Pascal. Continuando este proceso en la codificación, edición, compilación y ejecución de programas. La estrategia de aprendizaje, consiste en el planteamiento y solución de un problema que incluya las sentencias de escritura, lectura y de asignación. Es decir, que el alumno comprenda la importancia de la utilización del lenguaje de programación Pascal para resolver los problemas utilizando la computadora.

Propósito:

Al finalizar la unidad el alumno utilizará de manera básica el lenguaje de programación Pascal, mediante la codificación de algoritmos a fin de comprender los fundamentos de la programación, destacando las características con otros lenguajes.

Aprendizajes:

1. Describe los antecedentes y características de los lenguajes de programación.
2. Describe las características del sistema operativo.
3. Utiliza sentencias básicas del lenguaje de programación Pascal.
4. Utiliza la sintaxis y semántica básica del lenguaje de programación.
5. Utiliza el ambiente de trabajo.
6. Codifica algoritmos básicos en Pascal.

Estrategias de aprendizaje.

Esta unidad se desarrollará a través de un cuestionario sobre el tema, el alumno investigará el significado de estos conceptos, lo cual le permita comprender la parte teórica del lenguaje de programación Pascal. Por otra parte, practica simultáneamente de los conceptos que aparecen propuestos en los problemas donde utilizará las sentencias secuenciales (lectura, escritura y de asignación); y se abordará la sentencia de selección simple (If then).

Actividad de aprendizaje. Realiza lo que se te pide:

1. ¿Qué es un lenguaje de programación? _____

2. ¿En que niveles se clasifican los lenguajes de programación? Estos se clasifican en Lenguaje de nivel bajo (lenguaje máquina), lenguaje de nivel intermedio (ensamblador) y los lenguajes de alto nivel (lenguajes de programación).

3. Describe las características del lenguaje máquina _____

4. Describe las características del lenguaje ensamblador _____

5. Describe las características del lenguaje de alto nivel _____

6. ¿Qué es la sintaxis de un lenguaje de programación? _____

7. ¿Qué es la semántica de un lenguaje de programación? Es el significado que se da a las diversas construcciones sintácticas en los lenguajes de programación.

8. ¿Donde se utilizan los operadores aritméticos? Se utilizan para realizar cálculos aritméticos de acuerdo a la precedencia de los operadores de suma (+), resta (-), multiplicación (*), división (/), etc.

9. ¿Dónde se utilizan los operadores relacionales? _____

10. ¿Dónde se utilizan los operadores lógicos? _____

11. ¿Qué es una constante? Su valor permanece inalterable durante el proceso de ejecución del programa.

12. ¿Qué es una variable? Puede cambiar su valor durante el proceso de ejecución del programa.

13. ¿Qué es un comando? _____

14. ¿Qué acción ejecuta el comando ls? _____

15. ¿Qué acción se realiza con el comando fpc? _____

16. ¿Qué acción se realiza con el comando ./nombre del archivo?_____

17. ¿Qué es dato primitivo?_____

18. ¿Cuántos tipos de datos primitivos hay?_____

19. Describe las características de un dato primitivo del tipo numérico y da ejemplos:

20. Describe las características de un dato del tipo carácter_____

21. Describe las características de un dato primitivo del tipo lógico_____

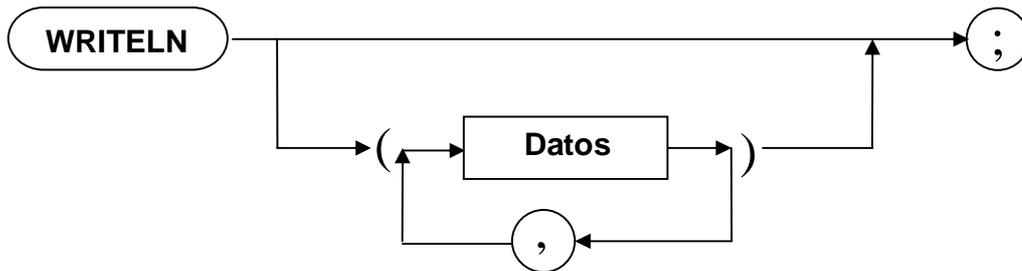
Actividad de aprendizaje. Escribe en los renglones en blanco lo que se indica:

Sentencia de	Descripción	Codificación
Lectura. Leer (lista de datos de entrada)		Readln(datos de entrada);
Asignación identificador ← expresión		Identificador := expresión;
Escritura. Escribir (lista de datos de salida)		Writeln(datos de salida)

Para una mejor comprensión por parte de los educandos sobre la sintaxis y semántica de las sentencias secuenciales de lectura, asignación y escritura, damos el diagrama sintáctico de para cada una de ellas.

Sentencia de escritura **WRITELN**.

Esta sentencia se utiliza para escribir en algún dispositivo de salida números, letras, títulos y símbolos del código ASCII. A continuación se presenta el diagrama sintáctico.

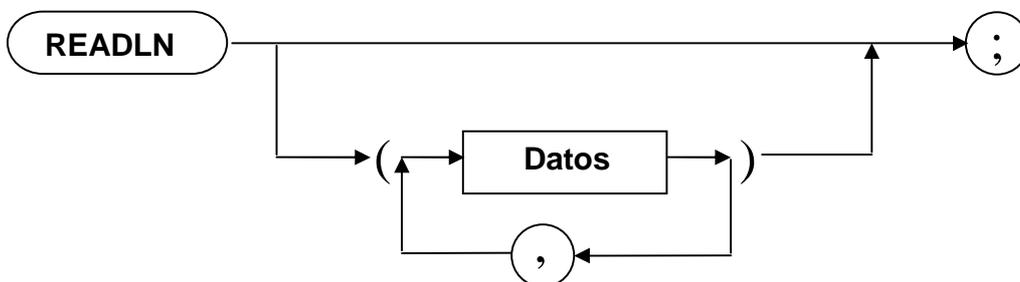


Este diagrama sintáctico de la sentencia WRITELN, indica la sintaxis y semántica que debe seguirse para la escritura de datos, para ello se recorre el diagrama como lo indican las flechas, en este recorrido, lo que está en el interior del símbolos óvalo y círculo se escriben igual y lo que está en el rectángulo representa a los datos de salida que serán visualizados, terminando con el carácter punto y coma (;).

Ejemplos: sentencia de escritura	Visualización
WRITELN;	Línea en blanco
WRITELN('HOLA AMIGA');	HOLA AMIGA
WRITELN(' ', 'DEJA 5 ESPACIOS');	DEJA 5 ESPACIOS
SUMA:=5000;	
WRITELN('SUMA = ',SUMA);	SUMA = 5000

Sentencia de lectura READLN

Esta sentencia permite la lectura de datos desde cualquier periférico de entrada, el teclado u disco, para un mejor conocimiento de ella, se da su diagrama sintáctico.



Este diagrama sintáctico de la sentencia READLN, indica la sintaxis y semántica que debe seguirse para la lectura de datos, para ello se recorre el diagrama como lo indican las flechas, en este recorrido, lo que está en el interior del símbolos óvalo y círculo se escriben igual y lo que

está en el rectángulo representa los datos de entrada al programa, termina la introducción de datos con el carácter de punto y coma (;).

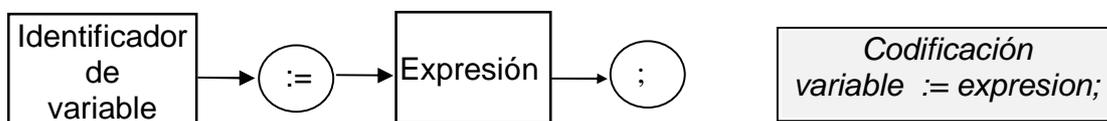
Ejemplo: Para lectura de datos, se tiene:

Ejemplos: sentencia de lectura	Función de la sentencia
READLN;	Pausa en la ejecución
READLN(distancia);	Pausa para introducir el valor de distancia
READLN(num1, num2);	Pausa para introducir los valores de num 1 y 2.
WRITE('Introduce costo: ');	Escritura de Introduce costo:
READLN(costo);	Pausa para introducir el valor de costo, inmediatamente, después del título anterior.

Sentencia de asignación :=

Esta sentencia permite asignar valores a las variables utilizadas en el programa, para una mejor comprensión de la sintaxis y semántica de la misma, se presenta su diagrama sintáctico.

Diagrama sintáctico de la sentencia de asignación :=



Este diagrama sintáctico, indica la sintaxis y semántica que debe seguirse para la asignación de valores a las variables especificadas en el programa, para ello se recorre el diagrama como lo indican las flechas, en este recorrido, lo que está en el interior del símbolo círculo se escriben igual y lo que está en el primer rectángulo representa a la variable que se le asignará el valor de expresión contenida en el segundo rectángulo, la asignación termina con el carácter de punto y coma (;).

En la asignación de valores a variables, el tipo de variable y el de expresión deberán ser compatibles (que sean del mismo tipo), de lo contrario se originará un error por la incompatibilidad de tipos, por otro lado no deberá ser considerada como una igualdad.

El programa operaciones ilustra la estructura de programas en Pascal y los elementos básicos de programación.

```

program operaciones;           {Comentario: zona para la identificación del programa}
var                               {Comentario: zona para la declaración de datos}
    num1, num2, suma, resta, producto, cociente: integer;
begin                               {Comentario: inicia zona para la declaración de sentencias}
    writeln('Programa operaciones aritméticas');
    write('Digita el número 1: ');
    readln(num1);
    write('Digita el número 2: ');
    readln(num2);
    suma:=num1+num2;
    resta:=num1-num2;
    producto:=num1*num2;
    cociente:=num1/num2;
    writeln('Suma = ', suma);
    writeln('Resta = ',resta);
    writeln('Producto = ', producto);
    writeln('Cociente = ', cociente);
    readln;
end.                               {Comentario: termina zona para la declaración de sentencias}

```

En la codificación anterior, **program**, **var**, **integer**, **begin** y **end** son **palabras reservadas**, utilizadas en el lenguaje de programación Pascal y se utilizan para describir el encabezado del programa, para la declaración de datos, para especificar el tipo de los datos de entrada y de salida, y para delimitar la sección de código del programa, respectivamente. La palabra operaciones se utiliza para identificar al programa, las palabras num1 y num2 son los identificadores para los datos de entrada y las palabras suma, resta, producto y cociente son los identificadores para los datos de salida.

Edición de un programa en Pascal.

Cabe señalar, que las instrucciones u órdenes para compilar y ejecutar programas en Pascal, puede variar según el tipo del compilador, en algunos casos compila y ejecuta con una sola orden, debido a que es un compilador de ambiente integrado y fuertemente amigable en el sentido de que es manipulado a través de menús y submenús; en otros casos las versiones del compilador necesitan instruir al sistema operativo, para que realice la fase de "montaje" o enlace (link), carga del programa objeto con las librerías del programa del compilador. El proceso de "montaje", produce un programa ejecutable. Cuando el programa ejecutable se ha creado, se puede ejecutar desde el sistema operativo con solo teclear el nombre del programa. Los ambientes mencionados, corresponden a los sistemas operativos Windows y Linux.

La edición de un programa, consiste en introducirlo con el teclado a la memoria RAM de la computadora por medio de un editor o procesador de textos. Para ello, puedes utilizar el compilador de Pascal en la versión para el sistema operativo Linux o el sistema operativo Windows. Los procedimientos de edición, compilación y ejecución del programa con el sistema operativo Linux, se indican en la siguiente tabla:

Edición	Compilación	Ejecución
Abre algún editor de línea del Sistema Operativo Linux, edita (escribe) el programa operaciones y grábalo con el nombre de archivo operarit.pas .	Para la compilación del programa operarit.pas , abre una terminal, escribe el comando fpc operarit.pas y oprime la tecla enter para aceptar el comando.	Para la ejecución del programa operarit.pas , escribe el comando: ./operarit y oprime la tecla enter para aceptar el comando. Digita los datos 100 y 50. Para aceptalos oprime la tecla Enter.

Los procedimientos de edición, compilación y ejecución del programa operaciones con el sistema operativo Windows es como se indica a continuación:

Abre el compilador de Free Pascal. pp/fp/File/New.

Edita el programa operaciones y grábalo con el nombre de archivo **operarit.pas**.

Compila el programa **operarit.pas**.

Ejecuta el programa **operarit.pas**.

El ambiente integrado del compilador Free Pascal en la versión del sistema operativo Windows, se presenta en la siguiente pantalla.

```

program salario_bruto;
uses crt;
var
  nomb: string[10];
  hortrab, tarhora, salbruto: integer;
begin
  clrscr;
  write ('Digita nombre : ');
  readln(nomb);
  write ('Horas trabajadas : ');
  readln(hortrab);
  write ('Tarifa por hora : ');
  readln(tarhora);
  salbruto := hortrab * tarhora;
  writeln('Nombre: ', nomb);
  writeln('Horas trabajadas: ', hortrab);
  writeln('Tarifa por hora: ', tarhora);
  writeln('Salario bruto: ', salbruto);
  readln;
end.

```

El IDE del compilador de Pascal, para editar, grabar, compilar y ejecutar programas.

Estrategias de aprendizaje. Para que te familiarices con el ambiente de trabajo del compilar de Free Pascal que tengas disponible, edita, graba, compila y ejecuta del programa operaciones, tal como se indicó con anterioridad. Ejecuta el programa varias veces con valores diferentes para los datos de entrada.

Con a finalidad de reforzar y ampliar el aprendizaje de los educandos sobre la aplicación de las sentencias lectura (readln), asignación (:=) y escritura (writeln), retomamos los problemas de salario bruto, viaje y promedio, los cuales fueron analizados en la unidad anterior. Los algoritmos resultantes, los codificaremos con el lenguaje de programación Pascal.

Problema salario bruto. Encuentra el salario bruto semanal (salbruto) que la empresa “Terabyte”, pagará a uno de sus empleados, considerando el nombre del empleado (nomb), horas trabajadas a la semana (hortrab) y la tarifa por hora (tarhor). Deberá visualizarse como salida el nombre del empleado, las horas trabajadas, la tarifa por hora y el salario bruto.

Actividad de aprendizaje. Escribe en los espacios en blanco el tipo de los identificadores que se especifican.

Identificador	Tipo
nomb	
hortrab	
tarhora	
Salbruto	

El pseudocódigo y la codificación del problema de salario bruto, se presentan en la siguiente tabla. Escribe las líneas faltantes en ambos procedimientos.

Seudocódigo del problema salario bruto	Codificación del problema salario bruto
<p>Actividad de aprendizaje. Escribe en las rayas en blanco, las sentencias faltantes en el pseudocódigo.</p> <p>Inicio</p> <p> Escribir('nombre del empleado: ')</p> <p> leer (nomb)</p> <p> Escribir ('horas trabajadas: ')</p> <p> _____</p> <p> Escribir('tarifa por hora: ')</p> <p> _____</p> <p> salbruto ← hortrab*tarhora</p> <p> escribir nomb</p> <p> escribir hortrab</p> <p> escribir tarhor</p> <p> _____</p> <p>Fin</p>	<p>Actividad de aprendizaje. Escribe en las rayas en blanco, las sentencias faltantes en la codificación.</p> <pre> Program salario_bruto; Uses crt; Var nomb:string[20]; hortrab, tarhora: integer; salbruto: integer; Begin Write('nombre del empleado: '); Readln(nomb); Write('horas trabajadas: '); Readln(hor_trab); _____ _____ salbruto:= hortrab*tarhora Writeln('Nombre empleado: ', nomb); Writeln('horas trabajadas: ', hortrab); Writeln('tarifa por hora: ', tarhora); _____ readln; end.</pre>

Estrategias de aprendizaje. Utiliza el ambiente de trabajo del compilador de Free Pascal que tengas disponible. Edita, graba el programa salario bruto con el nombre de archivo **salbruto.pas**, compila y ejecuta del programa **salbruto.pas** con los datos de entrada que se indican en la tabla y completa los datos faltantes en la misma.

Nombre	Horas trabajadas	Tarifa por hora	Salario bruto
José	20	30	
Carolina	55	35	
Pedro	30	46	

Problema viaje. Encuentra el tiempo y el costo de un viaje en automóvil que la familia Sánchez, realiza del Distrito Federal al Puerto de Acapulco (tiempo, costo), durante el recorrido pagará en total \$ 450 de cuotas en las casetas ubicadas en el trayecto. Además, se sabe que la distancia entre ambas ciudades es de aproximadamente 400 kilómetros (distancia), la velocidad promedio del auto es de 90 kilómetros por hora (velocidad). El rendimiento del automóvil (rend) y el costo por litro de gasolina (costo_litrogas) deberán ser introducidas con el teclado. Se entiende por rendimiento del automóvil a los kilómetros recorridos por litro de gasolina.

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes en el pseudocódigo.

```

Inicio
  Escribir('Rendimiento: ')
  Leer(rend);
  Escribir('Costo litro gasolina: ')
  Leer(costo_litrogas);
  tiempo ← _____
  consumo_gas ← _____
  costo ← consumo_gas*costo_litrogas
  Escribir('Tiempo viaje: ',tiempo:5:2)
  Escribir('total gas: ',consumo_gas:5:2)
  Escribir('costo = ',costo_viaje:5:2)
Fin

```

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes en la codificación.

```

program viaje;
uses crt;
const
  dist = 400;
  vm = 90;
var
  rend:integer;
  costo_litrogas, consumo_gas, costo:real;
begin
  clrscr;
  write('Rendimiento: ');
  _____
  write('Costo litro gasolina: ');
  _____

```

```

tiempo:=distancia/velocidad;
consumo_gas:=distancia/rend;
costo:= _____
writeln('total gas: ',consumo_gas:5:2);
writeln('costo = ',costo:5:2);
readln;
end.

```

Actividad de aprendizaje. Realiza la edición y graba el programa con el nombre de archivo viaje.pas. Compila y ejecuta el programa viaje.pas con los datos de la tabla y completa los datos faltantes en la misma.

rend	costo_litrogas	tiempo	consumo_gas	costo
12	6.50			
8	6.75			
10	6.80			

Problema promedio. Calcula el promedio de un alumno de Cibernética y Computación del CCH en una escala de diez (promedio), considerando el nombre del alumno (nombre), la calificación de tres exámenes (calexa1, calexa2, calexa3), la calificación de tres tareas (caltar1, caltar2, caltar3) y los porcentajes del 70% y 30% para exámenes y tareas, respectivamente.

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes en el pseudocódigo.

```

Inicio
  Escribir('Digita examen1: ')
  Leer(calexa1)
  Escribir('Digita examen2: ')
  Leer(calexa2)
  Escribir('Digita examen3: ')
  Leer(calexa3)
  Escribir('Digita tarea1: ')
  Leer(caltar1)
  Escribir('Digita tarea2: ')
  Leer(caltar2)
  Escribir('Digita tarea3: ')
  Leer(caltar3)
  _____
  _____
  _____
  Escribir('Promedio = ',promedio)
Fin

```

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes en la codificación.

```

program promedio_alumno;
uses crt;
var
  calexa1,calexa2,calexa3: integer;
  caltar1,caltar2,caltar3:integer;
  promexa,promtar,promedio: real;
begin
  clrscr;
  write('Digita examen1: ');
  readln(calexa1);
  write('Digita examen2: ');
  readln(calexa2);
  _____
  _____
  write('Digita tarea1: ');
  readln(caltar1);
  _____
  _____
  write('Digita tarea3: ');
  readln(caltar3);
  _____
  _____
  _____
  writeln('Promedio = ',promedio:2:2);
  readln;
end.

```

Actividad de aprendizaje. Edita y graba el programa promedio_alumno con el nombre de archivo promedio.pas, compílalo y ejecútalo con los datos de la tabla y completa los datos faltantes en la misma.

Calexa1	Calexa2	Calexa3	Caltar1	Caltar2	Caltar3	promexa	promtar	promedio
7	5	9	8	7	10			
8	9	6	9	8	7			
9	6	7	10	10	9			

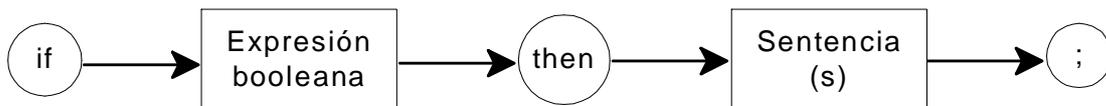
Sentencia condicional IF THEN.

Los problemas resueltos con el apoyo de la computadora y el lenguaje de programación Pascal, hemos utilizado las sentencias de lectura, asignación y escritura, mismas que aplicaremos en la solución de problemas que involucran la sentencia condicional. La sentencia condicional IF THEN, se utiliza para elegir una alternativa en la ejecución del programa que resuelve el problema. Para comprender sus características se te da el pseudocódigo y codificación, y se te pide que expliques su descripción.

Actividad de aprendizaje. Escribe en el cuadro la descripción de la sentencia condicional.

Sentencia	Descripción	Codificación
Condicional SI exp_bool Entonces sentencia		IF exp_bool THEN sentencia

Diagrama sintáctico de la sentencia simple **if expresión booleana then**



El diagrama sintáctico de la sentencia simple **if then**, indica la sintaxis y semántica que deberá seguirse para elegir una alternativa en función de la evaluación de la expresión booleana, para ello, se recorre el diagrama como lo indican las flechas, en este recorrido lo que está en el interior de los círculos se escriben igual, lo que está en el interior del primer rectángulo es la expresión booleana a ser evaluada, si la evaluación resulta verdadera, se ejecutan las acciones contenidas en el segundo rectángulo sentencia (s), en caso contrario ninguna acción se ejecuta y el programa continúa con la ejecución de las siguientes sentencias. Cuando la sentencia es compuesta, deberá delimitarse entre las palabras reservadas **Begin** y **end**.

Para la comprensión de la sentencia condicional por parte de los educandos en la solución de problemas, retomaremos los problemas deporte y salario neto, los cuales, fueron analizados en la unidad anterior y procederemos a la codificación de los algoritmos con el lenguaje de programación Pascal.

Problema deporte. Selecciona el deporte que es apropiado practicar, considerando como dato la temperatura (temp) en grados Fahrenheit con base en la siguiente tabla:

Deporte	Temperatura
Natación	temp mayor que 85
Tenis	temp mayor que 70, pero menor o igual a 85.
Golf	temp mayor que 32, pero menor que 70.
Esquí	temp mayor que 10, pero menor o igual a 32.
Marcha	temp menor o igual que 10

Estrategia de aprendizaje. Escribe el dato de entrada y su tipo

Dato de entrada	Tipo

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes del pseudocódigo.

Inicio

Escribir ('Digita temperatura')

Si (temp <= 10) entonces

 Escribir ('deprte preferido Marcha')

 Fin_si

Si (temp > 10) y (temp <= 32) entonces

 Escribir ('Deporte preferido Esquí')

 Fin_si

Si (temp > 32) y (temp <= 70) entonces

 Escribir ('Deporte preferido Golf')

 Fin_si

Si (temp > 70) y (temp <= 85) entonces

 Escribir ('Deporte preferido Tenis')

 Fin_si

Si (temp > 85) entonces

 Escribir ('Deporte preferido Natación')

 Fin_si

Fin

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes en la codificación.

```
Program deporte;
Uses crt;
Var
    temp:real;
begin
    writeln('digita la temperatura actual:');
    readln(temp);
    if (temp<=10) then
        writeln('deporte preferido Marcha');
    if (temp>10) and (temp<=32) then
        writeln('Deporte preferido Esqui');
    _____
    _____
    _____
    _____

    if (temp > 80) then
        writeln('Deporte preferido Natación');
    readln;
end.
```

Estrategia de aprendizaje. Completa las sentencias faltantes en el programa deporte y sálvalo con el nombre de programa deporte.pas. Edítalo, compílalo y ejecútalo, para varias corridas con valores diferentes para la temperatura.

Problema salario neto. Encuentra el salario semanal neto (sal_net) que la empresa “Terabyte”, pagará a uno de sus empleados considerando el nombre del empleado (nombre), las horas trabajadas a la semana (hor_trab), la tarifa por hora (tar_hora) y la tasa de impuestos de acuerdo a las siguientes categorías (cat), para la categoría A el impuesto es del 5% (imp_a), para la categoría B el impuesto es del 10% (imp_b) y para la categoría C el impuesto es del 15% (imp_c). Deberá visualizarse como salida el nombre del empleado, las horas trabajadas, la tarifa por hora, el salario bruto, la categoría y el salario neto.

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes en el pseudocódigo.

Inicio

leer nombre
leer hor_trab

sal_bruto ← hor_trab*tar_hora
si (cat="A") o (cat='a') entonces
 sal_netto ← sal_bruto-sal_bruto*imp_a
fin_si

Si (cat='C') o (cat='c') entonces
 sal_netto ← sal_bruto-sal_bruto*imp_c
fin_si
escribir nombre
escribir hor_trab
escribir tar_hora

Fin

Actividad de aprendizaje. Escribe en las rayas, las sentencias faltantes en el pseudocódigo.

Program salneto2;

Uses crt;

Var

Nombre:string[10];
Horastrab, tarifahora: integer;
Salbruto, salneto: real;
Cat:char;

Begin

writeln('nombre del empleado es:');
readln(nombre);

salbruto:=horastrab*tarifahora;
If (cat='a') or (cat='A') then
 salneto:=salbruto *(0.95);

```

_____
_____
_____
_____
writeln('el nombre del empleado es:',nombre);
writeln('las horas trabajadas en la semana son:',horastrab);
_____
_____
_____
readln;
end.

```

Estrategia de aprendizaje. Edita y graba el programa con el nombre de salneto2.pas. Compílalo y ejecútalo con los siguientes datos que aparecen en la tabla:

Nombre	Horas trabajadas	Tarifa por hora	Categoría
José	20	30	A
Carolina	55	35	B
Pedro	30	46	C

Estrategia de aprendizaje. Elabora el programa que resuelve el problema. Edítalo y grábalo con el nombre de archivo **promedios.pas**. Compílali y ejecútalo para varias corridas con diferentes datos de prueba.

Problema promedio. Calcula el promedio académico de un alumno de Cibernética y Computación del CCH en una escala de diez, considerando el nombre del alumno, la calificación de tres exámenes, la calificación de dos tareas y los porcentajes del 70% y 30% para los exámenes y tareas respectivamente. Cuando el promedio sea al menos seis y menor de 7.5, deberá aparecer la leyenda 'alumno regular'; cuando sea al menos de 7.5 y menor o igual a nueve, deberá aparecer la leyenda 'alumno destacado'; cuando sea mayor que nueve y menor o igual a 10, deberá aparecer la leyenda 'alumno sobresaliente', en caso contrario, deberá aparecer la leyenda 'alumno no aprobado'.

Reactivos muestra.

Indicaciones: Escribe en el paréntesis la respuesta correcta.

1. Relaciona la columna lenguaje y descripción en forma correcta. ()

Lenguaje	Descripción
a. Lenguaje de programación	I. Las instrucciones son directamente entendibles por la computadora
b. Lenguaje de alto nivel	II. Las instrucciones a la computadora son escritas con palabras similares a los lenguajes humanos.
c. Lenguaje máquina	III. Estos lenguajes son generalmente dependientes de la máquina.
d. Lenguaje de bajo nivel	IV. Son lenguajes utilizados para escribir programas de computadoras que pueden ser entendidos por ellas.

- A) a-IV, b-III, c-II, d-I
- B) a-III, b-II, c-I, d-IV
- C) a-I, b-II, c-III, d-IV
- D) a-IV, b-II, c-I, d-III
- E) a-II, b-IV, c-III, d-I

2. Relaciona los conceptos en la siguiente tabla: ()

Conceptos	Descripción
a. Sistema operativo	I. Para escribir programas que permiten la comunicación usuario/máquina.
b. Lenguaje de programación	II. Facilitan el uso de la computadora.
c. Programa de aplicación	III. Permite la comunicación entre el usuario y la computadora, además administra los recursos de la misma.
d. Programa de utilidad	IV. Son programas que realizan tareas concretas.

- A) a-IV, b-III, c-II, d-I
- B) a-III, b-I, c-IV, d-II
- C) a-I, b-II, c-III, d-IV
- D) a-II, b-III, c-IV, d-I
- E) a-IV, b-II, c-III, d-I

3. ¿Cuáles son los comandos del ambiente de trabajo para la ejecución de un programa?

Paso	Programas o Comandos
a. Compilación	I. ./nombre del programa
b. Edición	II. Kate, Vi o Xedit
c. Ejecución	III. Fpc programa.pas
d. Búsqueda de la trayectoria del programa	IV. home/usuario/programa

- A) a-IV, b-III, c-II, d-I
- B) a-I, b-III, c-II, d-IV
- C) a-III, b-II, c-I, d-IV
- D) a-II, b-III, c-IV, d-I
- E) a-I, b-II, c-III, d-IV

4. Encuentra el error en la siguiente codificación: ()

```

Program suma;
Uses crt;
Var
    x,y:real;
    Sum:integer;
Begin
Write('Digite el valor de x:');
Readln(x);
Write('Digite el valor de y:');
Readln(y);
Sum:=x+y;
Writeln('el valor de la suma es:');
Readln;
End.

```

- A) sum es tipo entero
- B) sum es del tipo char
- C) sum es del tipo boolean
- D) sum es del real
- E) sum es del tipo longint

5. En la codificación del programa salneto completa los espacios en blanco: ()

```

Program salbruto1;
Uses crt;
Var
    Nombre:string[10];
    Horastrab,tarifahora:_____
    Salbruto:real;
Begin
Writeln('nombre del empleado:');
Readln(nombre);

```

```
Writeln('Digite las horas trabajadas:');
```

```
Writeln('Digite la tarifa por hora:');
```

```
writeln('el nombre del empleado es:', nombre);  
writeln('las horas trabajadas en la semana son:', horastrab);  
writeln('la tarifa por hora es:', tarifahora);  
writeln('el salario bruto es:', salbruto:5:2);
```

```
end.
```

A)integer, readln(horastrab), readln(tarifahora), salbruto:=horastrab*tarifahora, readln.

B) real, readln(horastrab), readln(tarifahora), salbruto:=horastrab*tarifahora, readln.

C)integer, readln(horastrab), readln(tarifahora), salbruto:=horastrab+tarifahora, readln.

D)integer, readln(horast), readln(tarifahora), salbruto:=horastrab*tarifahora, readln.

E)integer, readln(horastrab), readln(tarhora), salbruto:=horastrab*tarifahora, readln.

Respuestas a los reactivos:

Reactivo	1	2	3	4	5
Respuesta	D	B	C	A	A

Bibliografía.

Joyanes, Luis. Fundamentos de programación, México, Mc Graw-Hill, 1997.

López, Leobardo. Programación Estructurada en turbo Pascal 7, México, Alfaomega, 1998.

Tucker, Allen B. Lenguaje de Programación, México, Mc Graw-Hill, 1988.

AUTORES

Unidad 1: La Cibernética

José Chacón Castro y Asunción Reynoso Díaz:

Unidad 2: Circuitos lógicos

Helios Becerril Montes y María Mónica Fuentes Romero

Unidad 3: Metodología de solución de problemas

Gilberto Fuentes Romero

Unidad 4: Introducción a la programación y al lenguaje de programación Pascal

José Mateos Cortés y Jesús Ramírez Vega

Coordinación:

Gilberto Fuentes Romero y Asunción Reynoso Díaz