

Tecnologías aplicadas a la enseñanza de las ciencias computacionales y la electrónica



Ana Eugenia Romo González

Tecnologías aplicadas a la enseñanza de las ciencias computacionales y la electrónica

Tecnologías aplicadas a la enseñanza de las ciencias computacionales y la electrónica

Ana Eugenia Romo González

Tecnologías aplicadas a la enseñanza de las ciencias computacionales y la electrónica /
Autora: Ana Eugenia Romo González —Guadalajara, Jalisco. 2022.

102 p. 23 cm.

Primera edición

ISBN: 978-84-19152-68-8

D. R. Copyright © 2022.

Edición y corrección: Astra Ediciones S. A. de C. V..

Todos los derechos reservados conforme a la ley. Las características de esta edición, así como su contenido no podrán ser reproducidas o transmitirse bajo ninguna forma o por ningún medio, electrónico ni mecánico, incluyendo fotocopiadora y grabación, ni por ningún sistema de almacenamiento y recuperación de información sin permiso por escrito del propietario del Derecho de Autor.

IMPRESO EN MÉXICO / PRINTED IN MEXICO

Contenido

Capítulo I

Representación de sistemas numéricos para el área digital	9
Unidades temáticas	10
1.1 Sistemas numéricos	11
1.2 Conversiones entre sistemas numéricos: binario a Hexadecimal.....	15
1.3 Operaciones de un sistema numérico.....	16
Tarea 1. Investigación.....	25

Capítulo II

Aprendizaje autónomo del álgebra Booleana	31
2.1 Lógica proposicional	32
2.2 Cálculo de predicados	36
Práctica 1. Lógica proposicional A.....	44
Práctica 2. Lógica proposicional B.....	45
Práctica 3. Cálculo de Predicados A.....	47
Práctica 4. Cálculo de Predicados B. Formalización.....	48
Práctica 5. Cálculo de predicados C. Evaluación	49

Capítulo III

Software de simulación aplicado a la electrónica digital	51
3.1. Funciones básicas en Multisim.....	52
3.2. Instrumentos virtuales	55
3.3. Construcción del circuito	60
Práctica de laboratorio 1	61
Práctica de laboratorio 1	63
Práctica de laboratorio 2	65
Práctica de laboratorio 3	67

Capítulo IV

Entornos virtuales para la simulación de sistemas digitales	69
4.1. Funciones básicas	70
4.2. Instrumentos virtuales	76
4.3. Construcción de circuitos.....	80
Práctica de laboratorio 1	81

Práctica de laboratorio 2	82
Práctica de laboratorio 4	86
Práctica de laboratorio 5	88
Capítulo V	
Taxonomía de tipos de datos en lenguaje C	89
Referencias Bibliográficas	999

Capítulo I

Representación de sistemas numéricos para el área digital

Objetivo de Aprendizaje	Resolver problemas de conversiones entre sistemas numéricos binario y hexadecimal para representar y manejar información computacional.
-------------------------	---

Unidades temáticas

1.1 Sistemas numéricos.

Binario y hexadecimal.

1.2 Conversiones entre sistemas numéricos.

Proceso de conversión.

Realizar conversiones.

1.3 Operaciones de un sistema numérico.

Proceso de realizar operaciones de suma y resta en los sistemas numéricos (binario y hexadecimal).

Realizar operaciones de suma y resta en sistemas numéricos binario y hexadecimal.

Antecedentes: Sistema de computadora digital. En un sistema de cómputo los datos y las instrucciones de un programa deben almacenarse en la memoria principal (de acuerdo con los modelos de las figuras 1 y 2) y se representan en sistema binario.

Figura 1. Modelo de John Von Neumann

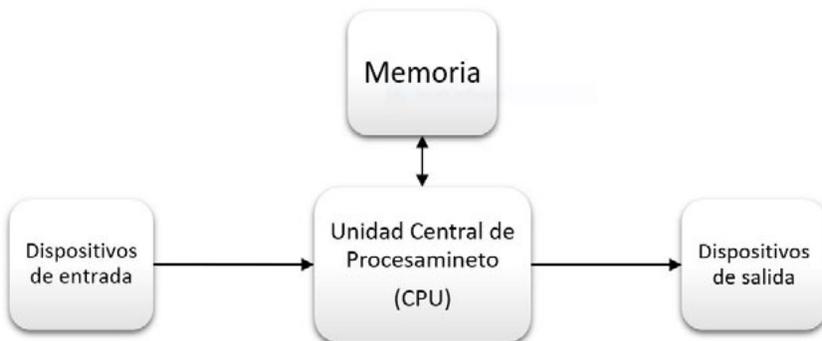
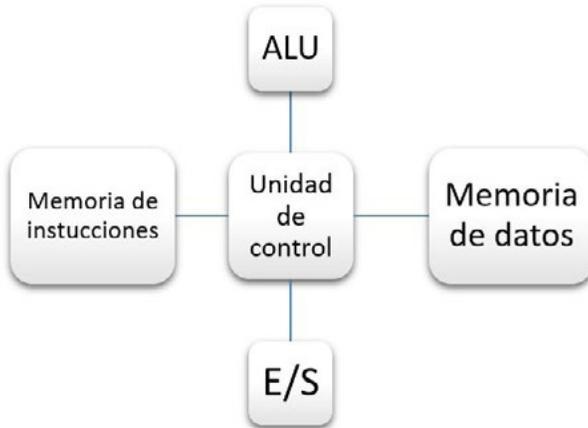


Figura 2. Modelo Harvard



El modelo permite establecer la representación de los datos e instrucciones y el modo de operación de la unidad de control.

1.1 Sistemas numéricos

Los sistemas numéricos más utilizados en el área digital son:

1. Decimal.
2. Binario.
3. Hexadecimal.

La expresión general (Nashelsky, 1989) para representar cualquier número de base **B** es:

$$\text{Fórmula 1. } N = d_n B^n + \dots + d_3 B^3 + d_2 B^2 + d_1 B^1 + d_0 B^0$$

Donde, N es el número a representar, d_n el dígito en cada posición, y B la raíz o base del sistema.

1.1.1 Sistema decimal (base 10)

Por ejemplo de acuerdo con la Fórmula 1: el número 1531 en el sistema decimal se escribe como:

Ya que:

10^4	10^3	10^2	10^1	10^0
10,000	1000	100	10	1

$$1531 = 1 \times 10^3 + 5 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$$

Donde, $B = 10$, $d_3 = 1$, $d_2 = 5$, $d_1 = 3$, $d_0 = 1$

1.1.2 Sistema binario

El número binario está formado por solo dos dígitos básicos, 0 y 1, por lo que su especificación simplificada es:

$$N = \dots + 8d_3 + 4d_2 + 2d_1 + d_0$$

Donde, d_3, d_2, d_1, d_0 pueden ser 0 o 1.

Tabla de potencias de 2.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Tomando en cuenta la tabla de potencias de 2 ¿Cuál es el número más grande que se puede representar con 8?

Por ejemplo 1001_2 es:

$$\begin{aligned} & \quad \quad \quad 1 \quad \quad 0 \quad \quad 0 \quad \quad 1 \\ (1001)_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ & \quad \quad \quad 8+0+0+1 \\ & \quad \quad \quad (9)_{10} \end{aligned}$$

1.1.2.1 Conversión de binario a decimal

La conversión de binario a decimal requiere agregar el valor para cada posición del número que tiene un dígito binario con valor de 1 de acuerdo con la Fórmula 1.



Elabore: **Ejercicio 1a y Ejercicio 1b.**

1.1.2.2 Conversión de decimal a binario

Se emplea el método de divisiones sucesivas entre 2 (Raíz). Por ejemplo la conversión del número $(28)_{10} = (11100)_2$.

$1/2$	$3/2$	$7/2$	$14/2$	$28/2$	□ Raíz
↓	↓	↓	↓	↓	□
0	1	3	7	14	Cociente
↓	↓	↓	↓	↓	□
1	1	1	0	0	Residuo
(MSD)		(LSD)			*Most/Least significant digit.

- *MSD: Dígito más significativo.
- *LSD: Dígito menos significativo.



Elabore: **Ejercicio 1c**

1.1.3 Sistema hexadecimal

El sistema base 16 requiere símbolos para representar los dígitos de 10, 11, 12, 13,14, y 15, además de los números del 0 al 9 mostrados en la tabla 1.

Tabla 1. Tabla de conversión decimal/hexadecimal.

Decimal	HexaD	Decimal	HexaD	Decimal	HexaD	Decimal	HexaD
0	0	4	4	8	8	12	C
1	1	5	5	9	9	13	D
2	2	6	6	10	A	14	E
3	3	7	7	11	B	15	F

Fuente: Tocci, Widmer, & Moss, (2007).

Por ejemplo: El número $(25)_{16}$ es el decimal $(37)_{10}$.

$$(25)_{16} = 2 \times 16^1 + 5 \times 16^0$$

$$2 \times 16 + 5 \times 1 = 32 + 5 = (37)_{10}$$

$$(2E)_{16} = 2 \times 16^1 + E \times 16^0$$

$$2 \times 16 + 14 \times 1 = 32 + 14 = (46)_{10}$$



Elabore: **Ejercicio 1d**

1.1.3.1 Conversión de decimal a entero hexadecimal

Se emplea el método de divisiones sucesivas entre 16 (Raíz). Por ejemplo la conversión del número $(150)_{10} = (96)_{16}$.

□	$9/16$	$150/16$	□
□	↓	↓	Raíz
□	0	9	Cociente
□	+	+	□
□	9	6	Residuo

(MSD) (LSD)

*Most/Least significant digit.

La conversión del número $(348)_{10} = (15C)_{16}$.

$$\begin{array}{r}
 1/16 \quad 21/16 \quad 348/16 \\
 \downarrow \quad \downarrow \quad \downarrow \\
 0 \quad 1 \quad 21 \\
 + \quad + \quad + \\
 1 \quad 5 \quad 12 = C \\
 \text{(MSD)} \quad \text{(LSD)}
 \end{array}
 \begin{array}{l}
 \square \\
 \text{Raíz} \\
 \square \\
 \text{Cociente} \\
 \square \\
 \text{Residuo}
 \end{array}$$


Elabore: Ejercicio 1e

1.2 Conversiones entre sistemas numéricos: binario a Hexadecimal

Se realiza mediante agrupamiento de dígitos binarios de cuatro esto debido a que la base 16 requiere 4 dígitos para su representación como se muestra en la Tabla 2.

Tabla 2. Relación de binario hexadecimal.

Binario	HexaD
0000	0
0001	1
0010	2
0011	3

Binario	HexaD
0100	4
0101	5
0110	6
0111	7

Binario	HexaD
1000	8
1001	9
1010	A
1011	B

Binario	HexaD
1100	C
1101	D
1110	E
1111	F

Por lo que la conversión del número $(110000110110)_2 = (C36)_{16}$ ya que:

1100 0011 0110

C 3 6



Elabore: **Ejercicio 1f**

1.3 Operaciones de un sistema numérico

Las operaciones aritméticas básicas de suma, resta, multiplicación y división también pueden realizarse en los sistemas binario y hexadecimal.

1.3.1 Suma binaria

Debido a que el sistema binario tiene como base el número 2 (0 y 1), solo pueden existir cuatro posibles combinaciones para los resultados (mientras que en decimal existen 100). Las combinaciones para la suma se presentan en la Tabla 3.

Tabla 3. Suma binaria.

Sumando	Incremento	Resultado
0	0	0
0	1	1
1	0	1
1	1	$0 + c (10)_2$

Fuente: Elaboración Propia a partir de Floyd, Salas, González & López, (2007).

En la tabla 3 la adición para $0+0 = 0$, $1 + 0 = 1$, $0 + 1 = 1$ pero $1 + 1 = (10)_2$, es decir, se utilizan dos posiciones para su representación por lo que se escribe el 0 (o bit menos significativo) y la c representa el acarreo de 1 (lo que se lleva o arrastra) que se suma a la siguiente posición, en la tabla 4 se presentan los ejemplos:

Tabla 4. Ejemplos de suma binaria.

Binario	Decimal	Binario	Decimal
100100 ₂ + 011000 ₂	36 ₁₀ +24 ₁₀	c 11 0111 ₂ + 1110 ₂	07 ₁₀ +14 ₁₀
1111002	60 ₁₀	10101 ₂	21 ₁₀
Binario	Decimal	Binario	Decimal
001101 ₂ + 100101 ₂	13 ₁₀ +37 ₁₀	110111011 ₂ + 100111011 ₂	443 ₁₀ +315 ₁₀
110010 ₂	50 ₁₀	1011110110 ₂	758 ₁₀

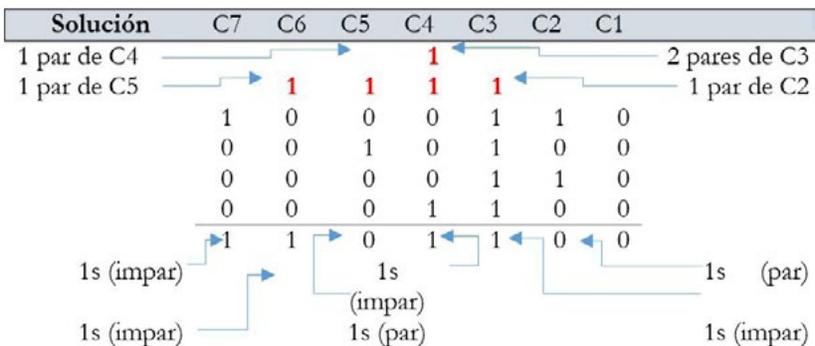
Fuente: Elaboración propia.

En una suma binaria con múltiples cifras se puede determinar si se tendrá o no acarreo de acuerdo con las siguientes reglas:

1. Si el número de 1s en la columna es impar, entonces el bit de suma será 1.
2. Si el número de 1s en la columna es par, entonces el bit de suma será 0.
3. Por cada par de 1s en la columna se suma un acarreo de 1 a la siguiente columna de bits.

Problema: Realizar la siguiente suma binaria.

$$\begin{array}{r}
 1\ 0\ 0\ 0\ 1\ 1\ 0 \\
 0\ 0\ 1\ 0\ 1\ 0\ 0 \\
 0\ 0\ 0\ 0\ 1\ 1\ 0 \\
 +\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \\
 \hline
 \end{array}$$





Elabore: **Ejercicio 1g**

1.3.2 Resta binaria

La resta binaria puede tener un mayor grado de complejidad debido a que se deben identificar el minuendo y el sustraendo para obtener su diferencia (Tabla 5).

Tabla 5. Resta binaria

Minuendo	Sustraendo	Resultado
0	0	0
0	1	1 + h
1	0	1
1	1	0

Fuente: Elaboración Propia a partir de Floyd, Salas, González & López, (2007).

En la Tabla 5 la substracción para $0 - 0 = 0$, $1 - 0 = 1$, $1 - 1 = 0$ pero $0 - 1 = 1 + h$, donde h indica préstamo a la derecha, por lo que en la resta binaria podemos considerar tres casos (con dos métodos):

- Resta “sin préstamo”.
- Resta “con préstamo”.
- Resta con complemento a 1 y complemento a 2 y resultados de números negativos.

Caso a: Resta “sin préstamo” (Tabla 6).

Tabla 6. Ejemplos de Resta Binaria “Sin préstamo”.

Binario	Decimal	Binario	Decimal
111_2 $- 010_2$	7_{10} $- 2_{10}$	1011_2 $- 0001_2$	11_{10} $- 1_{10}$
101_2	5_{10}	1010_2	10_{10}

Fuente: Elaboración propia.

Caso b: Resta “con préstamo” (Tabla 7).

Se debe “tomar prestado” del bit de la derecha (si puede hacerse el préstamo de forma sucesiva), en dos pasos:

1. Se resta al bit de la derecha uno (se tacha el bit 1 de la derecha y se reemplaza con 0).
2. Se considera el préstamo como 10₂ con lo que a dos se le resta uno, quedando 1 como resultado.

Tabla 7. Ejemplos de resta binaria.

Binario	Decimal	Caso en Binario	Decimal
$h \ 010$ $1 \ 10_2$ $- 101_2$	06_{10} $- 05_{10}$	1000_2 $- 0001_2$	8_{10} $- 1_{10}$
001_2	01_{10}		7_{10}
Resuelto en Binario	Caso en Binario	Decimal	Resuelto en Binario
$h \ 11$ $0 \ 1010$ 1000_2 $- 0001_2$	11000_2 $- 0001_2$	24_{10} $- 1_{10}$	$h \ 11$ $0 \ 1010$ 11000_2 $- 00001_2$
0111_2		23_{10}	10111_2

Fuente: Elaboración propia.

Caso c: Resta con complemento a 1 y complemento a 2.

Dado que los números negativos no pueden ser representados en binario clásico se utiliza el complemento a 1.

El complemento a uno (C_1^N) de un número binario es una operación matemática que nos permite la representación binaria de números negativos. Se representa como una función que se obtiene al cambiar cada uno de los dígitos del número binario N por su complementario, esto es, cambiar los unos por ceros y los ceros por unos.

Por ejemplo, el número binario = $(0110)_2 = (6)_{10}$

Con complemento a uno = $(1001)_2 = (-6)_{10}$

Donde el bit más significativo (el que está más a la izquierda representa el signo). Por lo que el número máximo que se puede representar con 4 bits es el 7 en decimal. Sin embargo, el complemento a uno no permite representar el 0, por lo que suele utilizarse el complemento a 2.

El complemento a dos de un número binario se puede definir como en la Fórmula 2:

$$\text{Fórmula 2: } C_2^N = C_1^N + 1$$

Es decir, el complemento a uno de un número binario más 1.

Número binario = $(0110)_2 = (6)_{10}$

Complemento a uno = $(1001)_2 = (-6)_{10}$

Más 1

$$\begin{array}{r} 1001 \\ + 0001 \\ \hline 1010 \end{array}$$

La **substracción** utilizando el complemento se realiza “sumando” el minuendo al complemento a 1 o a 2 del sustraendo (Tabla 8):

Tabla 8. Ejemplos de resta binaria con complementos.

Binario	C_1^N		C_2^N	
1101_2 -1011_2	1101_2 $+0100_2$		1101_2 $+0101_2$	
	1000_2	Acarreo cíclico	$+0010_2$	Acarreo ignorado
	$\rightarrow 1$		\leftarrow	
	0010	Respuesta	Respuesta	

Binario	C_1^N		C_2^N	
111001010_2 -110110101_2	111001010_2 $+001001010_2$		111001010_2 $+001001011_2$	
	100001010_2	Acarreo cíclico	$+000010101_2$	Acarreo ignorado
	$\rightarrow 1$		\leftarrow	
	000010101	Respuesta	Respuesta	

Fuente: Elaboración propia.



Elabore: **Ejercicio 1h**

1.3.3 Suma hexadecimal

En la adición hexadecimal, la suma puede ser cualquiera de los 15 dígitos que lo representan y el acarreo (c) siempre es 1, en la tabla 9 se muestran los valores y en la Tabla 10 los ejemplos.

Tabla 9. Suma Hexadecimal con acarreo.

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0+c
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0+c	1+c
3	3	4	5	6	7	8	9	A	B	C	D	E	F	0+c	1+c	2+c
4	4	5	6	7	8	9	A	B	C	D	E	F	0+c	1+c	2+c	3+c
5	5	6	7	8	9	A	B	C	D	E	F	0+c	1+c	2+c	3+c	4+c
6	6	7	8	9	A	B	C	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c
7	7	8	9	A	B	C	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c
8	8	9	A	B	C	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c
9	9	A	B	C	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c	8+c
A	A	B	C	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c	8+c	9+c
B	B	C	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c	8+c	9+c	A+c
C	C	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c	8+c	9+c	A+c	B+c
D	D	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c	8+c	9+c	A+c	B+c	C+c
E	E	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c	8+c	9+c	A+c	B+c	C+c	D+c
F	F	0+c	1+c	2+c	3+c	4+c	5+c	6+c	7+c	8+c	9+c	A+c	B+c	C+c	D+c	E+c

Fuente: Nashelsky, Chong & Gómez (1989).

Tabla 10. Ejemplos de suma hexadecimal.

Hexadecimal				Equivalencias	
6_{16}	B_{16}	A_{16}	$302B_{16}$	12,33110	0011000000101011 ₂
$+ 5_{16}$	$+ 8_{16}$	$+ 7_{16}$	$+ 8393_{16}$	33,68310	1000001110010011 ₂
B_{16}	13_{16}	11_{16}	$B3BE_{16}$	46,01410	1011001110111110 ₂
$3A + 6$	$= 40_{16}$	$3A + 7 =$	41_{16}		

Fuente: Elaboración propia.



Elabore: **Ejercicio 1i**

1.3.4 Resta hexadecimal

En la resta hexadecimal se utiliza en complemento a 15 (Tabla 11) y complemento a 16 por lo que puede realizarse empleando el siguiente algoritmo:

1. A cada dígito del sustraendo se le resta F (Complemento a 15).
2. Al resultado se le suma 1 (Complemento a 16).
3. Se suma al minuendo el Complemento a 16 del sustraendo (en caso de haber, el último acarreo se descarta).

Tabla 11. Complementos del hexadecimal.

Hexadecimal	Complemento a 15	Hexadecimal	Complemento a 15
0	F	8	7
1	E	9	6
2	D	A	5
3	C	B	4
4	B	C	3
5	A	D	2
6	9	E	1
7	8	F	0

En la Tabla 12 se presentan ejemplos de operaciones de resta a hexadecimal.

Tabla 12. Ejemplos de resta a hexadecimal.

Hexadecimal	C_{15}^N		C_{16}^N	
424_{16} $-2A6_{16}$	424_{16} $+D59_{16}$		424_{16} $+D5A_{16}$	
	$117D_{16}$	Acarreo cíclico	$+17E_{16}$	Acarreo ignorado
	$17E$	Respuesta		Respuesta
Hexadecimal	C_1^N		C_2^N	
$23DB816_{16}$ $-FCD2_{16}$	$23DB816_{16}$ $+032D_{16}$		$23DB816_{16}$ $+032E_{16}$	
	$23DBB43_{16}$	Acarreo cíclico	$23DBB44_{16}$	Sin acarreo
	$23DBB44$	Respuesta		Respuesta
Hexadecimal	C_1^N		C_2^N	
$711A_{16}$ $-502D_{16}$	$711A_{16}$ $+AFD2_{16}$		$711A_{16}$ $+AFD3_{16}$	
	$120EC_{16}$	Acarreo cíclico	$+20ED_{16}$	Acarreo ignorado
	$20ED$	Respuesta		Respuesta

Fuente: Elaboración propia.



Elabore: Ejercicio 1j y el Ejercicio 2

Tarea 1. Investigación

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante (Apellido P, M y N)		
Tarea 1	Definición y representación de Operaciones complementarias: a) Complemento a 1 y complemento a 2 en el sistema binario. b) Complemento a 9 y complemento a 10 en el sistema decimal. c) Resta de números decimales empleando complemento a 9 y a 10. d) Complemento a 15 y complemento a 16 en el sistema hexadecimal. e) Proporcione la fuente de la información.			

Ejercicio 1. Conversiones numéricas

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante (Apellido P, M y N)		
Ejercicio 1	Conversiones numéricas			

Ejercicio 1^a Complete la siguiente tabla en binario con representación a 4 bits para los números decimales 0 a 15.

Decimal	Binario	Decimal	Binario	Decimal	Binario	Decimal	Binario
0	0000	4		8		12	
1		5		9		13	
2		6		10		14	
3		7		11		15	

¿Cuál es el número máximo que puede representarse con 8 bits?

Ejercicio 1b		Ejercicio 1c		Ejercicio 1d		Ejercicio 1e	
Binario	Decimal	Decimal	Binario	HexaD.	Decimal	Decimal	HexaD.
110101		45		A30		45	
101001		31		F32		33	
100101		29		1F5		22	
101111		19		D2		19	
110101		64		E5		64	

Ejercicio 1f			
Binario	HexaD.	HexaD.	Binario
110110		6F6	_____
101100		CB9	_____
111110		F28	_____
101111		2A7	_____
111101		B3C	_____

Ejercicio 1g	Ejercicio 1h	Ejercicio 1i	Ejercicio 1j
Sumas	Restas	Sumas	Restas
110011_2 + 101000_2	1100011_2 - 1011101_2	$12EA5_{16}$ + $3729C_{16}$	329663_{16} - $113E0F_{16}$
1110011_2 + 0101111_2	100110_2 - 111010_2	403911_{16} + $6E1115_{16}$	$228CAB16_{16}$ - $1A6FDC18_{16}$
100110_2 + 111110_2	1101100011_2 - 101111_2	$EBD51A_{16}$ + $AED611_{16}$	$45C3C41F_{16}$ - $3761AE14_{16}$

Ejercicio 2. Desarrollo de pensamiento lógico-matemático

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante (Apellido P, M y N)		
Ejercicio 2	Pensamiento lógico matemático			

Determine en las columnas B y E si el número de la izquierda (columnas A y D) es mayor (>), menor (<) o igual (=) al número de la derecha (columnas C y F).

No.	A	B	C
1.	$2\frac{1}{2}$		$\frac{4}{2}$
2.	$\frac{1}{3}$		$\frac{1}{3}$
3.	0.38		0.4
4.	0.5		0.2
5.	$\frac{9}{5}$		$1\frac{2}{5}$
6.	$\frac{1}{2}$		0.5
7.	$\frac{8}{16}$		$\frac{1}{2}$
8.	$\frac{25}{8}$		3.125
9.	25%		$\frac{1}{4}$
10.	$\frac{2}{5}$		0.75
11.	$\frac{8}{11}$		0.24
12.	$\frac{201}{223}$		$\frac{50}{52}$
13.	0.5		$\frac{2}{5}$
14.	$1\frac{1}{2}$		$\frac{8}{5}$
15.	0.26		$\frac{23}{5}$

No.	D	E	F
16.	2.6		$\frac{52}{20}$
17.	$\frac{3}{6}$		0.5
18.	5.34		5.76
19.	107.4		107.7
20.	2.0		$\frac{16}{8}$
21.	$\frac{1}{4}$		$\frac{4}{10}$
22.	$\frac{1}{6}$		$\frac{2}{3}$
23.	$\frac{2}{14}$		25.8
24.	$\frac{22}{22}$		1.0
25.	$\frac{1}{3}$		$\frac{2}{6}$
26.	3.7		$\frac{6}{3}$
27.	$\frac{1}{4}$		0.4
28.	$\frac{250}{75}$		$3\frac{1}{3}$
29.	$\frac{2}{6}$		$\frac{1}{3}$
30.	2.4		4.2

Capítulo **II**

Aprendizaje autónomo del álgebra Booleana

Objetivo de Aprendizaje	Construir proposiciones y predicados para evaluarlos mediante tablas de verdad.
-------------------------	---

2.1 Lógica proposicional

La **computabilidad** es el término matemático que nombra los estudios sobre la teoría de la computación, se parte de las ideas filosóficas consistentes en encontrar una representación que describa un problema de forma adecuada. Para describir la realidad se debe utilizar un lenguaje y representarse a través de **símbolos**, por lo que además de emplear los símbolos como elementos formales se establecen asociaciones entre conceptos (Levine, 2001).

1.1.1 Proposiciones y conectores lógicos (AND, OR y NOT)

La lógica proposicional es un lenguaje formal para modelar fenómenos o procesos computacionales.

Una proposición es un elemento de inferencia en un proceso de razonamiento lógico, es una afirmación (Oración declarativa) que puede ser verdadera o falsa, por ejemplo:



Elabore: **Tarea 1**

1. Hoy hay examen.
2. Los estudiantes no van a misa.
3. México está en el continente americano.
4. La UTJ está en la colonia 1ro de mayo.
5. David es alto.
6. Sofía es inteligente.
7. Mañana es domingo.
8. Toda la mañana jugué Tetris.

Los argumentos en lógica proposicional involucran proposiciones atómicas que pueden combinarse mediante conexiones para formar proposiciones compuestas, por ejemplo:

1. Toda la mañana jugué Tetris **Y** hoy hay examen.
2. Mañana es domingo **Y** los estudiantes **NO** van a misa.

La rama de la gramática que estudia la disposición de las palabras en una oración se llama sintaxis. En lógica proposicional la sintaxis explica cómo colocar las proposiciones, los conectores y los signos de agrupación para crear fórmulas bien formadas (FBF) y posteriormente, mediante la semántica se determina si es falsa o verdadera.

Sintaxis

- a. Para representar las proposiciones se utilizan letras como: p, q o r.

p: Sofía es inteligente.

El conjunto de proposiciones = {p, q, r...}.

- b. Para representar los conectores se emplean con símbolos (Tabla 1):

Tabla 1. Conectores lógicos.

Tipo	Conector	Símbolo	Prioridad
Negación	no	\neg ~	1
Conjunción	y	\wedge	2
Disyunción	o	\vee	3
Condicional	si... entonces	\rightarrow	4
Bicondicional	si y sólo si	\leftrightarrow	5

El conjunto de conectores = { \wedge , \vee , \neg , \rightarrow , \leftrightarrow }.

- c. Para representar alcance de las FBF se utilizan **los signos de agrupación** que clarifican y determinan alcance.

Conjunto de signos de agrupación = {(), [], {} }.

Al evaluar una proposición, ésta puede ser falsa o verdadera.

Los ejemplos para la construcción de proposiciones con conectores se muestran en la tabla 2.

Tabla 2. Ejemplos de proposiciones con conectores.

Conector	Significado	FBF	Ejemplos:	Representación	Observación
Negación	Símbolos (\neg)(\sim) Se agrega a una proposición la palabra “no”.	$\neg p$.	p: El edificio A tiene dos plantas.		
Conjunción	Símbolo (\wedge) Unión de dos proposiciones con la palabra “y”.	$p \wedge q$.	p: La temperatura es mayor a 28o. q: Traigo un suéter.		
Disyunción	Símbolo (\vee) Unión de dos proposiciones con la palabra “o”.	$p \vee q$.	p: El celular no sirve. q: No hay WiFi		
Condicional	Símbolo (\rightarrow) Unión de dos proposiciones con las palabras “si... entonces...”	$p \rightarrow q$	p: Suena el timbre. q: Alguien llega.		
Bicondicional	Símbolo \leftrightarrow . Es la unión de dos proposiciones mediante las palabras “si y sólo si”.	$p \leftrightarrow q$	p: El aire se contamina. q: Los ciudadanos usan auto.		

Los conectores permiten estructurar FBF mediante proposiciones compuestas, por ejemplo:

p: Estudio Lenguaje C.

q: Aprendo correctamente.

r: Reprobaré. $(p \wedge q) \rightarrow \neg r$: Si estudio Lenguaje C y aprendo correctamente entonces no reprobaré.



2.1.2 Proposiciones y tablas de verdad (AND, OR y NOT)

Para determinar el valor de una proposición molecular (si es falsa o verdadera) se analiza y evalúa mediante su significado (semántica), sin embargo, cuando la proposición es compuesta su evaluación es más compleja por lo que se utilizan tablas de verdad para determinar su resultado a través de las posibles combinaciones de sus valores.

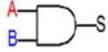
Semántica: El **principio de bivalencia** establece que un enunciado puede ser falso o verdadero, pero no ambas cosas a la vez y para su representación se utiliza una tabla de verdad.

El número de filas se determina mediante 2^n , donde n =número de variables en la fórmula que se está calculando. Para $n=1$, tenemos 21 posibles combinaciones de los valores de verdad, es decir, 0 es V o F.

p	p
V	1
F	0

La Tabla 3, muestra un resumen de los valores de verdad para los conectores lógicos.

Tabla 3. Tablas de verdad para los conectores lógicos.

Conector	Simbología	Tabla	Aplicación electrónica
Negación	$\neg p$		Compuerta invert (Inversor) o NOT  $S = \neg A$
Conjunción	$p \wedge q$		Compuerta AND (Producto Lógico)  $S = A * B$ La salida es "1" cuando todas las entradas son "1"
Disyunción	$p \vee q$		Compuerta OR (Suma Lógica)  $S = A + B$ La salida es "0" cuando todas las entradas son "0"
Condicional	$p \rightarrow q$		
Bicondicional	$p \leftrightarrow q$		



Elabore: **Tarea 2**



Elabore: **Practica 2**

2.2 Cálculo de predicados

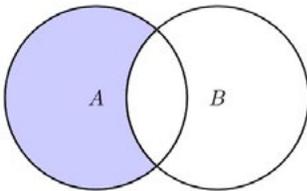
En la lógica de predicados las frases declarativas permiten establecer sentencias complejas que no siempre quedan correctamente determinadas empleando lógica proposicional, esto se debe a que cuando se emplean pocos símbolos se puede perder su significado.

Para esclarecer el significado de los datos, en lógica de predicados se estudian los objetos (sustantivos) y sus relaciones a través de la identificación de:

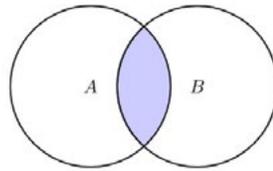
- a) El objeto (de quién se hace una afirmación).
- b) El predicado o relación (qué es lo que se afirma del objeto).

La estructura general de una Fórmula Bien Formada (FBF) con predicados utiliza *cuantificadores* que se derivan de la teoría de conjuntos, como (Ferrater y Leblanc, 2015):

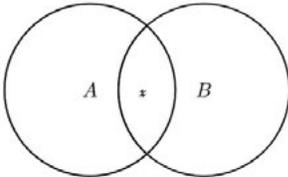
$$\forall x:P(x)\rightarrow R(x)$$



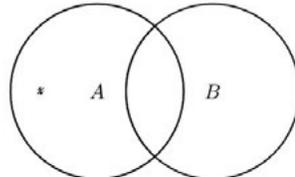
Todo A es B



Ningún A es B



Algún A es B



Algún A no es B

2.2.1 Describir la sintaxis de las proposiciones y predicados

El alfabeto de la lógica de predicados (Shoenfield, 2018) estará formado por los conjuntos simbólicos de las Tablas 4 a la 7.

Tabla 4. Conjuntos simbólicos del cálculo de predicados.

Conjuntos de Símbolos		Conjuntos de Letras	
Variables (VAR)	Constantes (CONS)	Función (FUNC):	Predicado (PRED):
Se emplean las últimas letras minúsculas del abecedario.	Se emplean las primeras letras minúsculas del abecedario.	Se representan con las letras f, g, h , para utilizar más de una se utilizan subíndices, ej.	Se representan con las letras mayúsculas $P, Q, R \in \text{PRED}$
Para utilizar más de una se utilizan subíndices, ej. $x, y, z, x_1, y_2, z_3 \in \text{VAR}$	Para utilizar más de una se utilizan subíndices, ej. $a, b, c, a_1, b_2, c_3 \in \text{CONS}$.	$f_1, g_2, h_5 \in \text{FUNC}$. Una Función con aridad* 2: $f_2 \in \text{FUNC}$ es una función con dos argumentos.	$P_4 \in \text{PRED}$ es un predicado con cuatro argumentos.
*La aridad es el número de argumentos que tiene una función o un predicado.			
Un Término es una cadena de símbolos utilizada para representar objetos, como una variable o contante.			
Para los términos t_1, t_2, \dots, t_n son términos y la función f^n se establece el término $f^n(t_1, t_2, \dots, t_n)$.			
Un átomo o literal puede ser falso o verdadero y se representa como una cadena de símbolos con términos y predicados como: $R^n(t_1, t_2, \dots, t_n)$.			

Fuente: (Shoenfield, 2018).

Tabla 5. Conectivos, cuantificadores y signos de puntuación del cálculo de predicados.

Enunciado declarativo (o cláusula - conectores)	Predicado(término)	Abreviación
La UTJ es una Universidad.	Universidad (UTJ)	U(UTJ)
Mercurio es un planeta.	Planeta (mercurio)	P(m)
El 1 es un número.	Número (1)	N(1)
El cielo es azul.	Azul (cielo)	A(c)
Sofía y David son estudiantes.	Estudiantes (Sofía, David)	E(s, d)
La sala esta oscura o la cocina esta luminosa.	Oscura (sala) \vee	
Luminosa (cocina)	O(s) \vee L(c)	

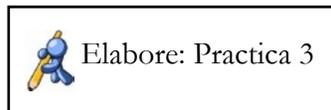
Fuente: Elaboración propia.

Ejemplos utilizando variables

Tabla 7. Uso de variables en el cálculo de predicados.

Enunciado declarativo (o cláusula - conectores)	Predicado(término)	Abreviación
Esta es una Universidad.	Universidad (x)	U (x)
Ese es un planeta.	Planeta (y)	P (y)
Es un número.	Número (x_1)	N (x_1)
Algo es azul.	Azul (y_1)	A (y_1)
Ella y el son estudiantes.	Estudiantes (z_1, z_2)	E (z_1, z_2)
Aquí esta oscuro o ahí esta luminoso.	Oscuro (x_3) \vee Lumino- so(y_3)	O (x_3) \vee L (y_3)

Fuente: Elaboración propia.



2.2.2 Evaluación de proposiciones y predicados con tablas de verdad

En lógica de predicados la semántica de la lógica de predicados se refiere al significado del objeto y de las propiedades o relaciones establecidas en el predicado por lo que se utilizan cuantificadores como el Universal y Existencial.

a) Cuantificador Universal:

Indica que algo es cierto para todos los individuos (todo, ningún, cada, cualquiera):

$(\forall x)P(x)$ o $AxP(x)$, se lee “para todo x, P”. Se interpreta como:

$$P(a_1) \wedge P(a_2)$$

Pueden tener las formas mostradas en la Tabla 8:

Tabla 8. Uso del cuantificador universal en el cálculo de predicados.

Interpretación	Representación
Para todas las x, x es P.	$(\forall x) P(x)$
No todas las x son P (o algunas x son P).	$\neg(\forall x) P(x)$
Todas las x no son P.	$(\forall x) \neg P(x)$
Para toda x, todas las P son Q.	$(\forall x) (P(x) \rightarrow Q(x))$
Para toda x, las P no son Q.	$(\forall x)(P(x) \rightarrow \neg Q(x))$

Fuente: (Shoenfield, 2018).

b) Cuantificador existencial

Indica que algo es cierto para algunos individuos (algún, alguno, hay un, cierto(s)).

$(\exists x)P(x)$ o $Ex P(x)$, se lee “existe un x tal que P“. Se interpreta como:

$$P(a_1) \vee P(a_2) \vee P(a_3) \vee \dots \vee P(a_n).$$

Pueden tener las formas mostradas en la tabla 9.

Tabla 9. Uso del cuantificador de existencia en el cálculo de predicados.

Interpretación	Representación
Algunas x son P (o hay P).	$(\exists x) P(x)$
Para todas las x, hay una y tal que P.	$(\forall x) (\exists y) P(x,y)$
Algunas x no son P.	$(\exists x) \neg P(x)$
Ninguna x es P.	$\neg(\exists x) P(x)$
Para algunas x, x son P y Q.	$(\exists x) (P(x) \wedge Q(x))$
Para algunas x, x no son P y Q.	$(\exists x) (\neg P(x) \wedge \neg Q(x))$

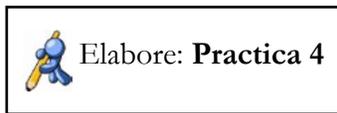
Fuente: Ferrater Mora & Leblanc (2015).

Los ejemplos de la formalización de oraciones se presentan en la tabla 10.

Tabla 10. Conjuntos simbólicos del cálculo de predicados.

Oración	Formalización
Itzel es metódica y ordenada.	$M(i) \wedge O(i)$
Si Itzel metódica, entonces también es ordenada.	$M(i) \rightarrow O(i)$
Nadie es metódica y además ordenada.	$\neg(\exists x) (M(x) \wedge O(x))$
Todas las metódicas son ordenadas.	$(\forall x) (M(x) \rightarrow O(x))$
Todos los informáticos son listos, David es informático, luego David es listo.	$(\forall x) (\text{Informático}(x) \rightarrow \text{Listo}(x)) \wedge \text{Informático}(\text{david}) \rightarrow \text{Listo}(\text{david})$
	$(\forall x (I(x) \rightarrow L(x)) \wedge I(l)) \rightarrow L(l)$

Fuente: Elaboración propia.



Equivalencias

Las equivalencias formales en el cálculo de predicados se presentan en la tabla 11.

Tabla 11. Ejemplos de equivalencias en el cálculo de predicados.

Ejemplo	Formalización
Toda la madera es café.	$(\forall x) (M(x) \rightarrow C(x)) \quad \exists$
No es cierto que alguna madera no es café.	$\neg(\exists x) (M(x) \wedge \neg C(x))$
Alguna madera es café.	$(\exists x) (M(x) \wedge C(x)) \quad \exists$
No es cierto que ninguna madera es café.	$\neg(\forall x) (M(x) \rightarrow \neg C(x))$
Alguna madera no es café.	$(\exists x) (M(x) \wedge \neg C(x)) \quad \exists$
No es cierto que toda madera es café.	$\neg(\forall x) (M(x) \rightarrow C(x))$

Fuente: Ferrater Mora & Leblanc (2015).

Validación

La validación e interpretación de predicados se presentan en las tablas 12 y 13.

Tabla 12. Validación de predicados.

Eliminar la generalización	Introducir la variable
$(\exists x) P(x)$	$(\forall x) P(x)$
$P(a)$ EG	$P(a)$ EP
$P(a)$	$P(a)$
$(\exists x) P(x)$ IG	$(\forall x) P(x)$ IP

Fuente: Ferrater Mora & Leblanc (2015).

Tabla 13. Interpretación de predicados.

Interpretación	Representación
Todo hombre es mortal. Todo mortal es débil.	1. $(\forall x) (H(x) \rightarrow M(x))$ 2. $(\forall x) (M(x) \rightarrow D(x))$
Todo hombre es débil.	$\therefore (\forall x) (M(x) \rightarrow D(x))$
	3. $H(a) \rightarrow M(a)$ EG(1)
	4. $M(a) \rightarrow D(a)$ EG(2)
	5. $H(a) \rightarrow D(a)$ S.H.(3,4)
	6. $(\forall x) (H(x) \rightarrow D(x))$ »

Fuente: Shoenfield (2018).



Elabore: Practica 5

Tarea 1. Investigación

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Ejercicio 1	Investigar la definición y las tablas para evaluar fórmulas complejas CON ejemplos: <ul style="list-style-type: none"> • Tautologías. • Contradicciones. • Contingencias. 			

Tarea 2. Investigación

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Ejercicio 1	Investigar en lógica de predicados. <p style="text-align: center;">Axiomas.</p> <p style="text-align: center;">Reglas de inferencia.</p> <p style="text-align: center;">Forma normal de Skolem, forma normal Prenex.</p>			

Práctica 1. Lógica proposicional A

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante (Apellido P, M y N)		
Práctica 1	Construya las proposiciones utilizando conectores (como en la tabla 2).			

No.	Descripción	Representación
1	La recta es una figura lineal y el polígono es una figura plana.	
2	Si es una figura plana de 3 lados iguales entonces es un triángulo equilátero.	
3	El río Amazonas no está en América del Norte.	
4	Si es una figura plana de 3 lados diferentes entonces es un triángulo escaleno.	
5	Si el sufragio es efectivo entonces no hay reelección.	
6	El pollito es amarillo y México está en el continente americano.	
7	La geometría estudia los planos representacionales y el cálculo cómo cambian las funciones continuas.	
8	El sábado voy al cine y el domingo voy a cenar.	
9	El pingüino no es un pez.	
10	Guadalajara es una ciudad y Jalisco es un estado.	

Práctica 2. Lógica proposicional B

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante (Apellido P, M y N)		
Práctica 2	Evaluación con tablas de verdad			

Simbolizar las proposiciones																		
No.	Proposición	Simbolización	Tabla de verdad															
1	No estudié inglés, pero aprendí francés.		<table border="1"> <thead> <tr> <th>p</th> <th>q</th> <th></th> </tr> </thead> <tbody> <tr> <td>V</td> <td>V</td> <td></td> </tr> <tr> <td>V</td> <td>F</td> <td></td> </tr> <tr> <td>F</td> <td>V</td> <td></td> </tr> <tr> <td>F</td> <td>F</td> <td></td> </tr> </tbody> </table>	p	q		V	V		V	F		F	V		F	F	
p	q																	
V	V																	
V	F																	
F	V																	
F	F																	
2	Ni estudié inglés, ni aprendí francés.																	
3	No es cierto que estudiaste inglés y aprendiste francés.																	
4	Estudie inglés aunque no aprendí francés.																	
5	No me gusta leer ni escribir.																	
6	O es incorrecto el color o es falsa la factura emitida.																	
7	Si no sabes inglés, no apruebas el cuatrimestre.																	

Simbolizar las proposiciones			
No.	Proposición	Simbolización	Tabla de verdad
8	Es primavera y o bien la noche dura 12 horas o el sol ilumina la tierra directamente en el ecuador.		
9	O es primavera y la noche dura 12 horas o el sol ilumina la tierra directamente en el ecuador.		
10	Las personas estudian doctorado solamente cuando concluyen la maestría.		

Práctica 3. Calculo de Predicados A

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 3	Definición de predicados y abreviaciones con base en enunciados declarativos.			

No.	Enunciado	Predicado (término)	Abreviación
1	Valeria es fotógrafa.		
2	David es arquitecto.		
3	Arely es médico y Sofía es ingeniero.		
4	El elefante y la Jirafa son cuadrúpedos.		
5	Holmes es un detective y Watson es famoso.		
6	Los estudiantes aprenden Java o los maestros enseñan C/C++.		
7	Los planetas orbitan en el sistema solar.		
8	Los arquitectos son ingeniosos.		
9	Los programadores son inteligentes		
10	Los diseñadores son creativos y los abogados no son honestos.		

Práctica 4. Cálculo de Predicados B. Formalización

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 4	Formalización de oraciones en el cálculo de predicados.			

No.	Oración	Formalización
1	Valeria es fotógrafa y creativa.	
2	Si Valeria es fotógrafa entonces también es creativa.	
3	Algunos robots hablan con fluidez.	
4	Todas las maestras inspiran confianza, María es maestra, entonces María inspira confianza.	
5	Algunas herramientas son manuales y utilizan cables.	
6	Los estudiantes que aprenden C/C++ consiguen empleo.	
7	Cada país tiene leyes y cultura.	
8	Cualquier persona es creativa.	
9	Ciertas aves son mamíferos y no vuelan.	
10	Plutón orbita en el sistema solar y no es un planeta.	

Práctica 5. Cálculo de predicados C. Evaluación

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 5	Evaluación de oraciones en el cálculo de predicados.			

No.	Oración	Interpretación	Evaluación
1 V(3)	Ningún nadador rema. Todos los que no son nadadores son corredores. Por lo tanto, todos los remadores son corredores.	<ul style="list-style-type: none"> • F=ser nadador • P=ser remador • E=ser corredor 	
2 V(3)	Ningún presidente duerme temprano. Algunas personas son presidentes. Por lo tanto, hay personas que no duermen temprano.	<ul style="list-style-type: none"> • C=ser presidente • R=dormir temprano • M=ser persona 	
3 V(4)	Algunos hombres son inteligentes. Ninguna mujer miente. Por lo tanto, algunos hombres que mienten son inteligentes.	<ul style="list-style-type: none"> • H=ser hombre • I=ser inteligente • M=ser mujer • T=ser mentiroso 	

Capítulo **III**

Software de simulación aplicado a la electrónica digital

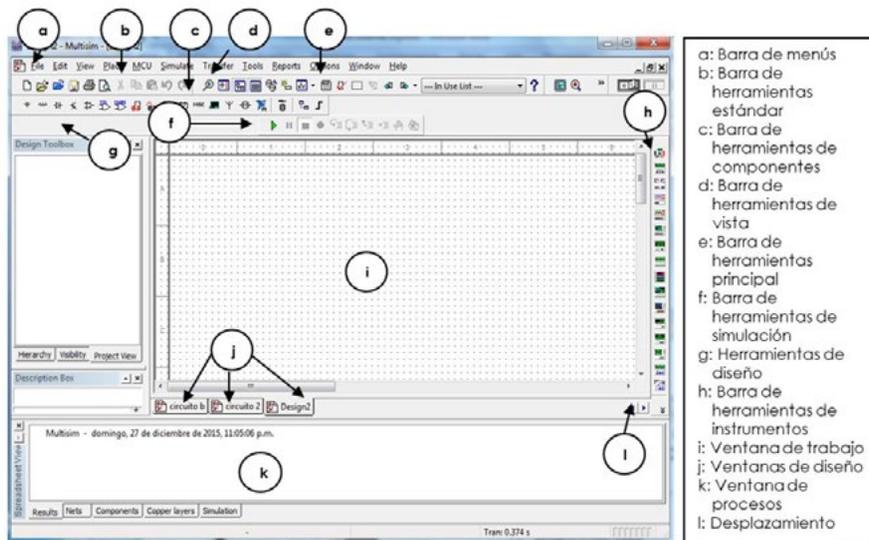
El *Software* de simulación es una herramienta valiosa que se aplica en diversas áreas del conocimiento, dado que uno de sus objetivos es el de predecir el comportamiento completo de un sistema previo a su implementación son de especial utilidad en la industria.

Uno de los *software* de simulación más utilizados en el área de sistemas digitales es Multisim© de National Instruments (NI) por lo que en este documento se presentan conceptos y ejercicios prácticos.

3.1. Funciones básicas en Multisim

La interfaz del *software* MULTISIM se puede observar en la figura 1 en la que se encuentran los elementos básicos:

Figura 1. Funciones básicas del simulador.

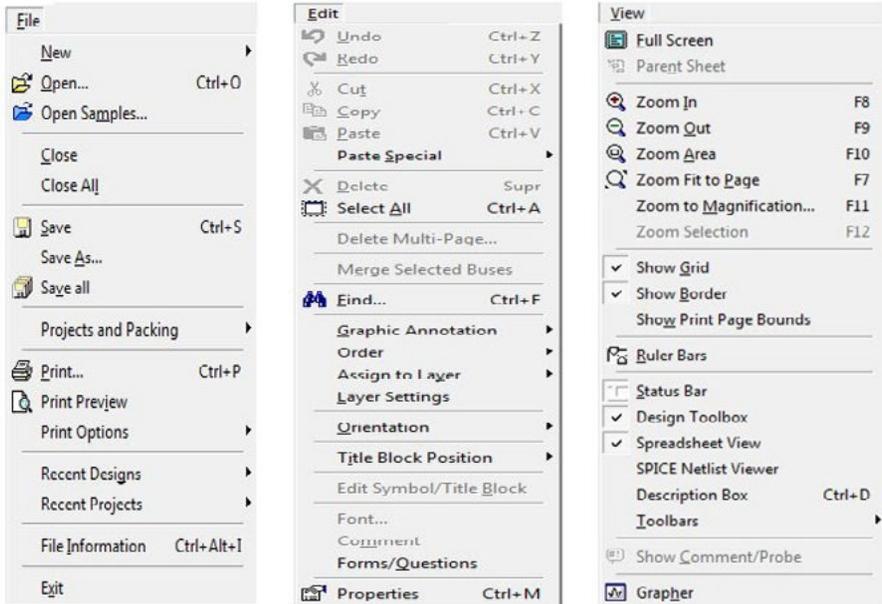


Fuente: Pantalla de Multisim, elaboración propia a partir de (NI, 2021).

3.1.1. Barra de menús

Contiene los comandos para las funciones de configuración del espacio de trabajo y el control de archivos (figura 2).

Figura 2. Comandos principales de la barra de menús.



Fuente: Pantalla de Multisim, elaboración propia a partir de (NI, 2021).

3.1.2. Barra de herramientas estándar

Contiene botones para las funciones más comunes (Figura 3):

Figura 3. Herramientas estándar.



Fuente: Pantalla en Multisim, elaboración propia a partir de (NI, 2021).

3.1.3. Barra de herramientas de componentes

Permite colocar en la ventana de trabajo los siguientes componentes o seleccionar de un conjunto (Figura 4):

Figura 4. Componentes

									
Fuente	Básico	Diodo	Transistor	Analógico	TTL	CMOS	Misceláneo digital	Mixto	Indicador

							
Potencia	Misceláneo	Periféricos avanzados	RF	Electromecánico	MCU	Bloque jerárquico	BUS

Fuente: Pantalla en Multisim, elaboración propia a partir de (NI, 2021).

3.1.4. Barra de herramientas de simulación

Contiene los botones para el control de la simulación (figura 5).

Figura 5. Herramientas para simulación.

			
Ejecutar/continuar	Pausa	Detener	Pausar hasta MCU

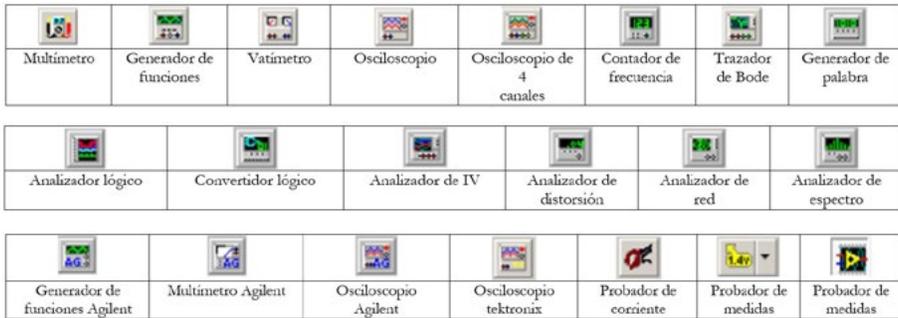
					
Paso por dentro	Paso por encima	Paso por fuera	Ejecutar a cursor	Alternar punto de interrupción	Remover los puntos de interrupción

Fuente: Pantalla en Multisim, elaboración propia a partir de (NI, 2021).

3.1.5. Barra de herramientas de instrumentos

Contiene los botones de los instrumentos de medición, verificación y análisis (figura 6).

Figura 6. Instrumentos de medición.



Fuente: Pantalla en Multisim, elaboración propia a partir de (NI, 2021).

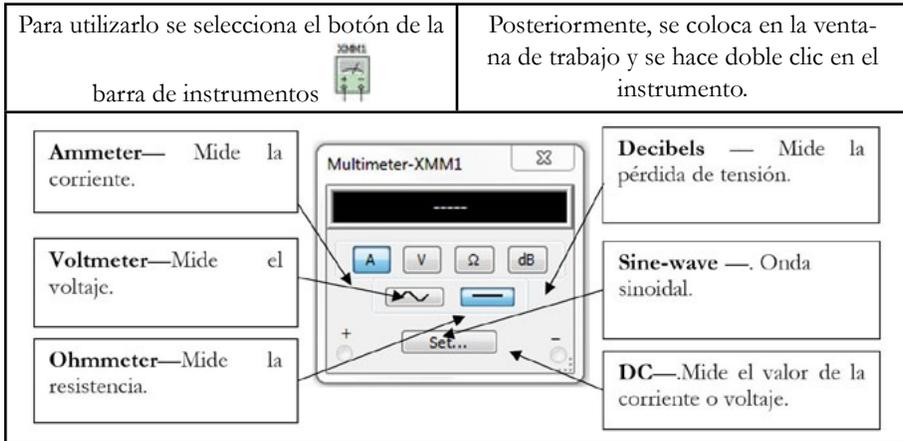
3.2. Instrumentos virtuales

Entre los principales instrumentos virtuales empleados en Multisim se encuentran el *multímetro*, el analizador de estados lógicos y el generador de palabras cuyo funcionamiento se describe en el presente apartado. Se explica tanto el procedimiento general para el acceso y configuración de los instrumentos como los pasos específicos para su uso.

3.2.1. Multímetro

El multímetro (polímetro o tester) permite medir las magnitudes eléctricas: voltaje o corriente alterna o continua (AC o DC), y resistencia. En Multisim se tiene un rango automático (National Instruments_a, 2021) por lo que se puede utilizar de forma directa y tanto los valores de la resistencia interna como la corriente están predefinidas (figura 7).

Figura 7. Características del multímetro.

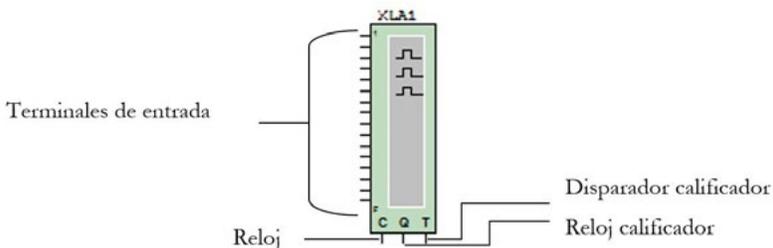


Fuente: Pantalla en Multisim, elaboración propia a partir de (NI_a, 2021).

3.2.2. Analizador de estados lógicos

El analizador de estados lógicos muestra los niveles de hasta 16 señales digitales en un circuito (National Instruments_b, 2021). Se utiliza para la adquisición rápida de datos de estados lógicos y análisis a través del tiempo que contribuye a analizar el diseño de grandes sistemas y llevar a cabo la solución de problemas (figura 8)

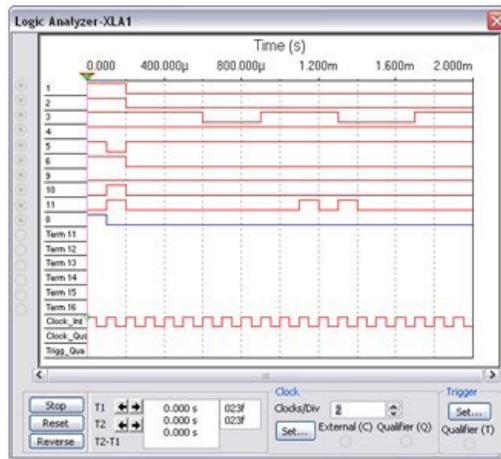
Figura 8. Conectores del Analizador de estados lógicos.



Fuente: Pantalla en Multisim, elaboración propia a partir de (NI_b, 2021).

Para abrir las características del analizador lógico se debe hacer doble clic para hacer ajustes y ver mediciones (figura 9).

Figura 9. Características generales del analizador de estados lógico.



Fuente: Elaboración propia del uso del analizador.

Los 16 círculos en el lado izquierdo de la Figura 9 corresponden a las terminales y los renglones horizontales muestran los cambios de estados con respecto al tiempo. Cuando la terminal está conectada a un nodo, su círculo se despliega con un punto negro y el nombre (número) del nodo.

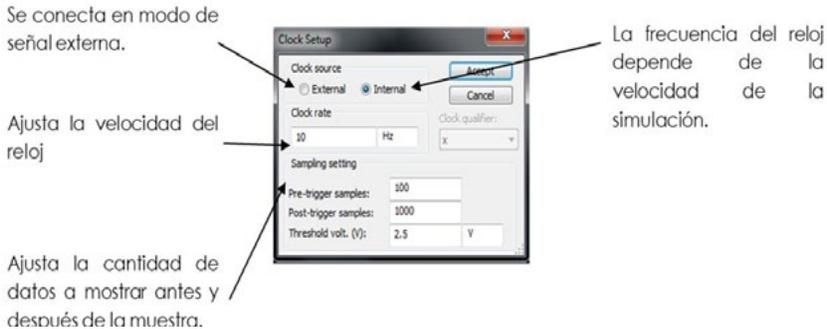
El analizador se puede detener y reiniciar, cuando se pausa la simulación mientras el circuito se está simulando, se detienen tanto el circuito como el analizador lógico y cuando se utiliza el botón Reiniciar (*reset*) solo podrán visualizarse los resultados hasta el momento en que el instrumento fue restablecido (NI_b, 2021). Para reiniciar el analizador, se emplea el botón *reset*, lo que borra toda la información y comienza a almacenar nuevos datos.

3.2.2.1. Configuración del reloj

El reloj informa al analizador lógico cuando leer una muestra de entrada y puede ser interno o externo. Para ajustar la configuración del reloj se deben realizar los siguientes pasos:

1. Seleccionar la opción de Configurar en la zona del reloj del analizador lógico con lo que se tendrá acceso al cuadro de diálogo de configuración del reloj de la Figura 10.

Figura 10. Pantalla para ajuste del reloj.



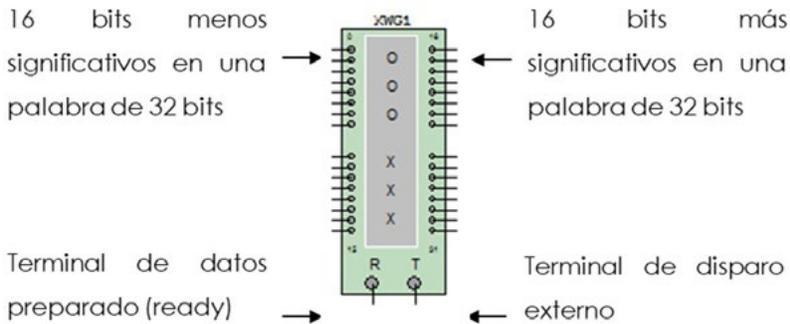
Fuente: Pantalla en Multisim, elaboración propia a partir de (NI_b, 2021).

2. Seleccione el modo de reloj externo o interno.
3. Establezca la velocidad del reloj interno. El calificador de reloj es una señal de entrada que filtra la señal de reloj. Si se establece en "x", el calificador se desactiva y la señal de reloj determina cuando se leen las muestras. Si se establece en "1" o "0", se leen las muestras solo cuando la señal de reloj cambia.
4. Establecer la cantidad de datos para mostrar antes (muestras pre - trigger) y después (Muestras post- trigger) de la muestra.
5. Haga clic en Aceptar.

3.2.3. Generador de palabras

Se utiliza para enviar palabras digitales o patrones de bits que proporcionan estímulos a los circuitos. Figura 11 muestra la distribución del generador de palabras.

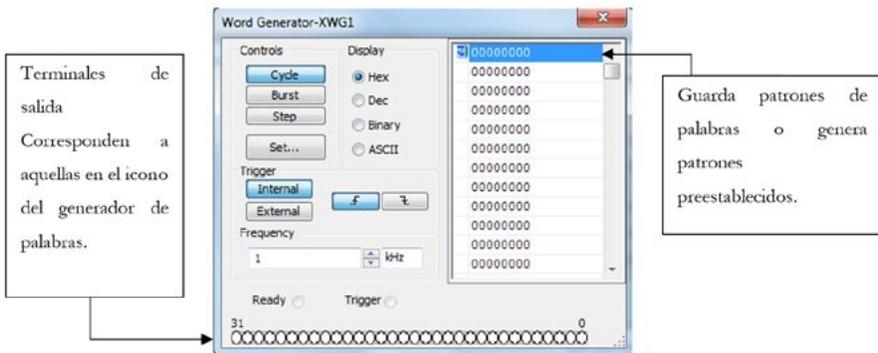
Figura 11. Conectores del generador de palabras.



Fuente: Pantalla en Multisim, elaboración propia a partir de (NI_b, 2021).

Al hacer doble clic en el icono se abre la interfaz del instrumento que es empleada para ajustarlo y ver las mediciones.

Figura 11. Interfaz del generador de palabras.



Fuente: Pantalla en Multisim, elaboración propia a partir de (NI_b, 2021).

3.2.3.1. Configuración del generador de palabras

Ingreso de palabras: En el lado derecho de la interfaz se muestran filas de números desde 00000000 hasta FFFFFFFF en hexadecimal o de 0 a 4294967295 en decimal en función del tipo de dato mostrado (Hex, Dec, Binay o ASCII) que puede ser seleccionado por el usuario (NI_b, 2021)

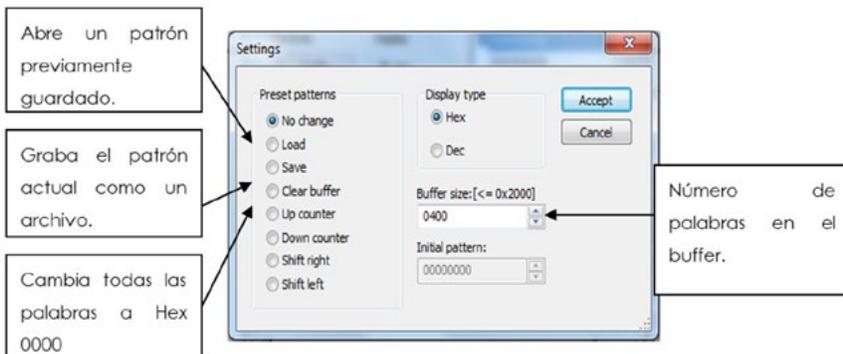
Controles

Para ingresar las palabras de 32 bits en un circuito, haga clic en los botones *Cycle*, *Burst* o *Step*.

- *Cycle*: Transmite un flujo continuo de palabras que se puede detener haciendo clic en el botón de Stop de la barra de herramientas de simulación.
- *Burst*: Transmite todas las palabras en secuencia.
- *Step*: Transmite una palabra a la vez.

Para insertar un punto de ruptura, seleccione la palabra en la lista donde se quiere que la entrada se detenga, clic derecho y seleccionar Definir punto de interrupción con lo que será marcada la "señal de alto". Para eliminarlo, seleccione el punto de interrupción y "Eliminar Breakpoint" en el menú emergente. Haga clic en el botón Set para mostrar el conjunto de opciones mostrados en la Figura 12.

Figura 12. Configuración para el generador de palabras .



Fuente: Pantalla en Multisim, elaboración propia a partir de (NI_b, 2021).

3.3. Construcción del circuito

El primer paso para la captura esquemática consiste en colocar los componentes apropiados. Puede utilizar los siguientes métodos para localizar un componente

1. Utilice la barra de herramientas de Componentes para navegar.
2. Busque un grupo de componentes / familia específica

El segundo paso consiste en conectar los componentes. Para alambroslos automáticamente:

1. Haga clic en un pin del primer componente para iniciar la conexión (el apuntador cambia a una cruz(X))
2. Mueva el ratón (junto al cursor aparece el alambre)
3. Haga clic en un pin del segundo componente para finalizar la conexión.



Realice: Prácticas de laboratorio 1- 4

Práctica de laboratorio 1

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 1	Implementación de compuertas lógicas			

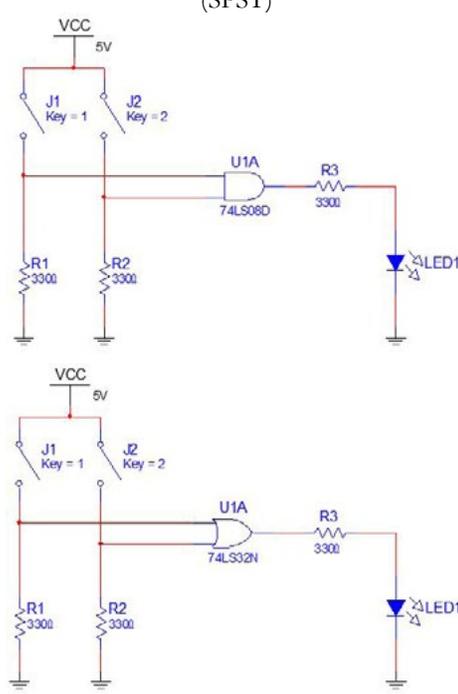
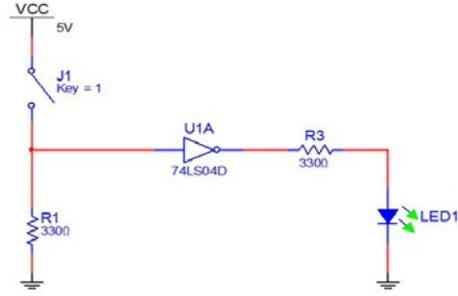
Materiales:	Cantidad	Descripción	Cantidad	Descripción
	1	IC 74ls04	1	LED
	1	IC 74ls08	1	Protoboard
	1	IC 74ls32	1	Fuente de voltaje.
	2	Interruptores	1	Multímetro
	3	Resistencias de 330 Ω		
Procedimiento:	1	<p>Arme el circuito del diagrama y compruebe la tabla de verdad de cada una de las compuertas.</p>		
	2	<p>Utilice compuertas NOT, AND y OR para implementar y probar una compuerta EXOR.</p>		

Informe

Proceso	Descripción: Complete las tablas.																						
1	<table border="1"> <thead> <tr> <th>J1</th> <th>J2</th> <th>AND</th> <th>OR</th> <th>NOT</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td rowspan="4"></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> <td></td> </tr> </tbody> </table>	J1	J2	AND	OR	NOT	0	0				0	1			1	0			1	1		
J1	J2	AND	OR	NOT																			
0	0																						
0	1																						
1	0																						
1	1																						
2	<p>Diagrama</p> <table border="1"> <thead> <tr> <th>J1</th> <th>J2</th> <th>EXOR</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	J1	J2	EXOR	0	0		0	1		1	0		1	1								
J1	J2	EXOR																					
0	0																						
0	1																						
1	0																						
1	1																						

Práctica de laboratorio 1

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante (Apellido P,M y N)		
Práctica 1	Simulación de compuertas lógicas.			

<p>Procedimiento:</p>	<p>1</p>	<p>Arme los siguientes circuitos en multisim (SPST)</p> 
		
	<p>2</p>	<p>Construya el circuito para implementar y probar una compuerta EXOR Utilice compuertas NOT, AND y OR.</p>
	<p>3</p>	<p>Verifique los valores de salida en tablas de verdad. Incluya un Texto con su nombre. Cargue un solo archivo del procedimiento 1 en la plataforma. Cargue un solo archivo del procedimiento 2 en la plataforma.</p>

Informe

Proceso	Descripción: Complete las tablas y genere el diagrama.																						
1	<table border="1"> <thead> <tr> <th>J1</th> <th>J2</th> <th>AND</th> <th>OR</th> <th>NOT</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td rowspan="4"></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> <td></td> </tr> </tbody> </table>	J1	J2	AND	OR	NOT	0	0				0	1			1	0			1	1		
J1	J2	AND	OR	NOT																			
0	0																						
0	1																						
1	0																						
1	1																						
2	<p>Diagrama</p> <table border="1"> <thead> <tr> <th>J1</th> <th>J2</th> <th>EXOR</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table>	J1	J2	EXOR	0	0		0	1		1	0		1	1								
J1	J2	EXOR																					
0	0																						
0	1																						
1	0																						
1	1																						

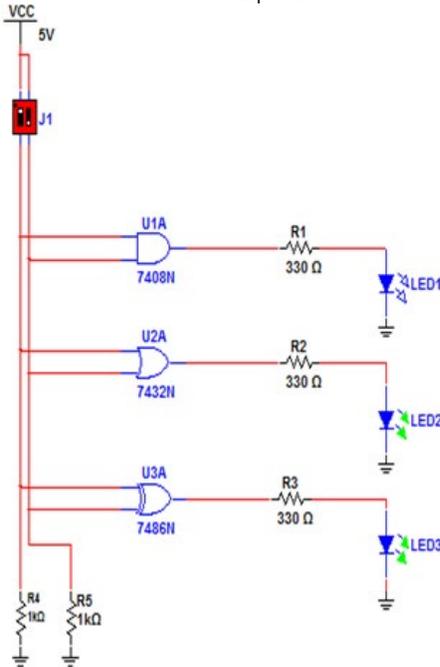
Práctica de laboratorio 2

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 2	Uso de analizador de estados lógicos.			

Procedimiento:

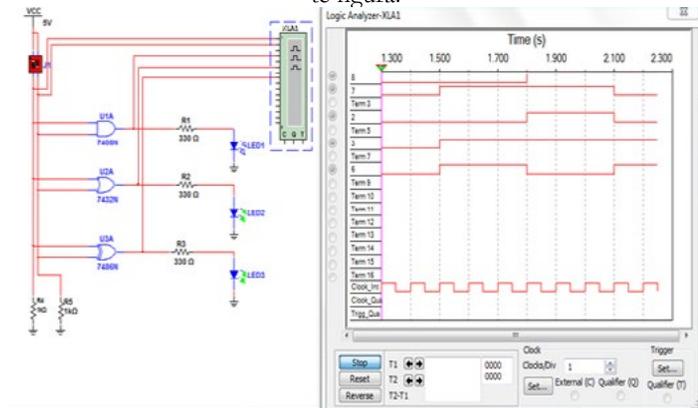
1

Arme el siguiente circuito y compruebe las tablas de verdad de las tres compuertas:



2

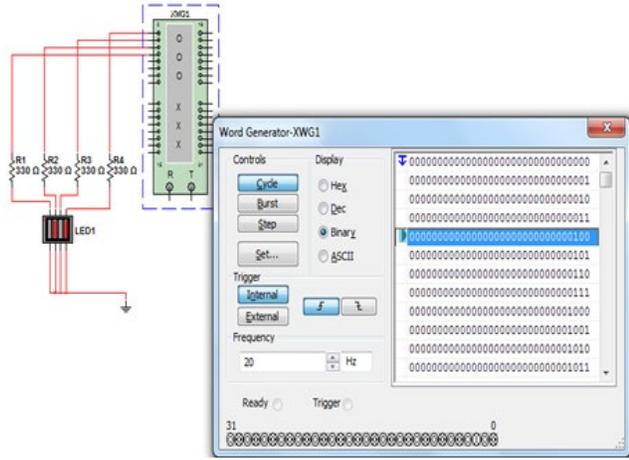
Agregue el analizador de estados lógicos como se muestra en la siguiente figura.



Procedimiento:	
3	Coloque el reloj interno a 10 Hz y mantenga 1 ciclo de reloj / división.
4	Ponga en cero ambos interruptores (las entradas de las compuertas).
5	Ejecute la simulación e introduzca en los interruptores las combinaciones 00, 01, 10 y 11.
6	Detenga la simulación y compruebe el resultado de las compuertas en la pantalla del analizador.
7	Con el botón Reverse cambie el color de fondo del analizador lógico.
8	Incluya un Texto con su nombre. Cargue un solo archivo en la plataforma.

Práctica de laboratorio 3

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 3	Uso de analizador de estados lógicos.			

Procedimiento:	
1	<p>Implemente un contador binario de 0000 a 1111 como se muestra en la siguiente figura. (BAR_LED_RED_FOUR).</p> 
2	<p>(Modifique ION a un valor apropiado para que los LEDS enciendan e introduzca una F=20hz).</p>
3	<p>Incluya un Texto con su nombre. Cargue un solo archivo en la plataforma.</p>

Capítulo **IV**

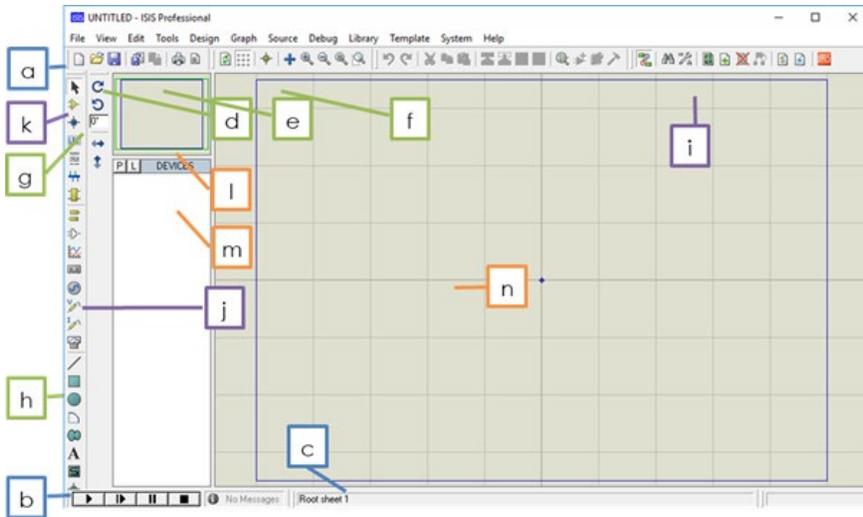
Entornos virtuales para la simulación de sistemas digitales

Proteus © de Labcenter Electronics. (LE) es una plataforma virtual para la simulación y construcción electrónica para las etapas de diseño de esquemas, programación de software, construcción de PCB, simulación, depuración de errores y construcción.

4.1. Funciones básicas

La interfaz del software PROTEUS se puede observar en la Figura 1 con los elementos básicos que contiene:

Figura 1. Funciones básicas del simulador



Fuente: Elaboración propia con designación basada en la pantalla del software Proteus (LE, 2021).

Tabla 1. Designación de Funciones básicas.

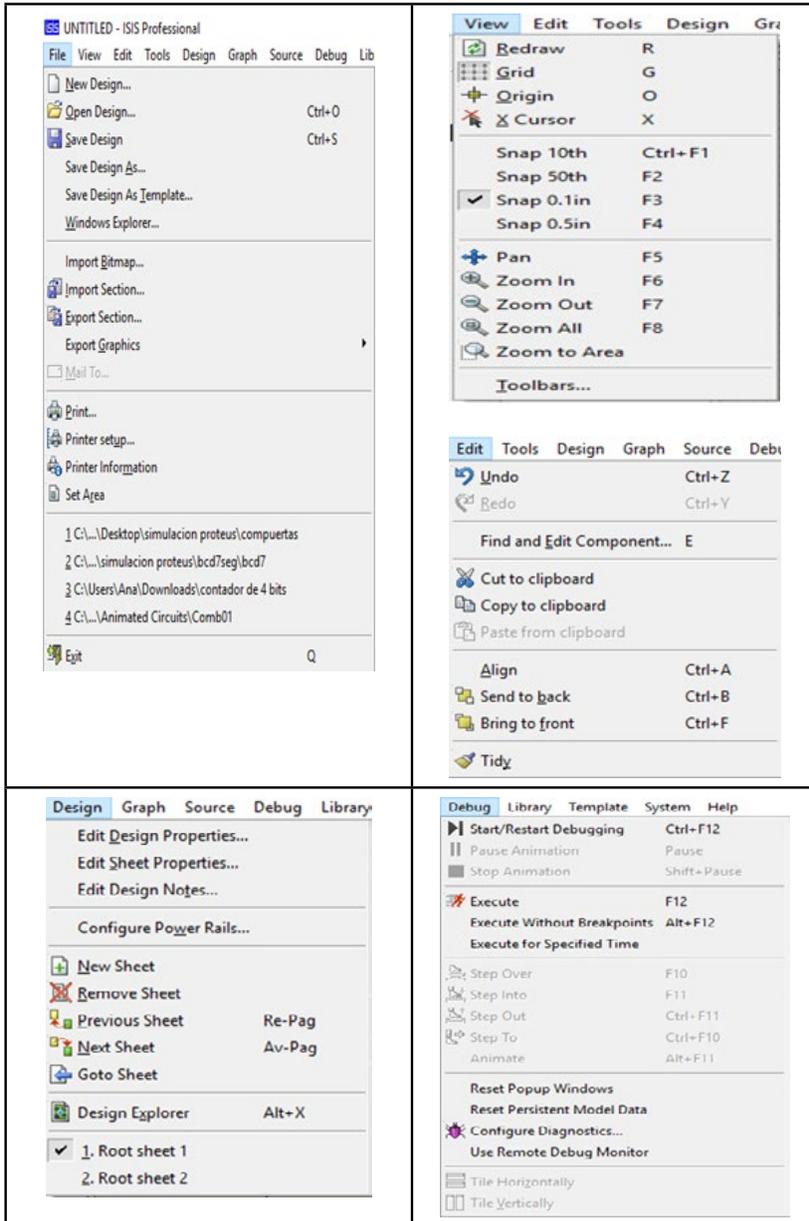
a	Barra de menú	h	Comandos de dibujo
b	Barra de simulación	i	Herramientas de diseño
c	Barra de estado	j	Herramientas de diseño electrónico
d	Comandos de archivo e impresión	k	Modos de trabajo
e	Comandos de edición	l	Ventana de edición (vista previa)
f	Comandos de visualización	m	Ventana para componentes y bibliotecas
g	Comandos de rotación y reflexión	n	Ventana de trabajo

Fuente: Elaboración propia en la pantalla del software Proteus (I.E, 2021).

4.1.1. Barra de menú

Contiene los comandos para las funciones de configuración del espacio de trabajo y el control de archivos (Figura 2).

Figura 2. Resumen de comandos principales de la barra de menús.



Fuente: Elaboración propia basada en la pantalla del software Proteus (LE, 2021).

4.1.2. Barra de comandos

Contiene botones para las funciones más comunes (figura 3).

Figura 3. Comandos de la barra de menús.



Fuente: Elaboración propia basada en la pantalla del software Proteus (LE, 2021).

4.1.3. Barra de herramientas de simulación

Contiene los botones para el control de la simulación (figura 4).

Figura 4. Herramientas para simulación.

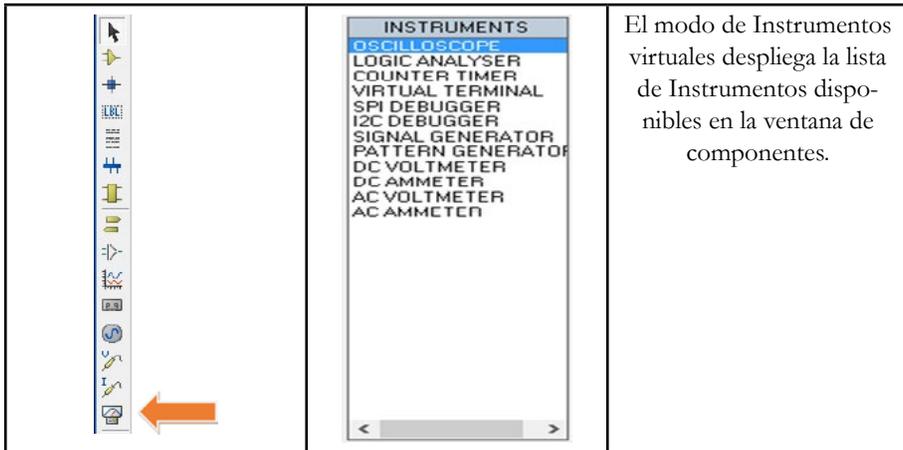


Fuente: Elaboración propia basada en la pantalla del software Proteus (LE, 2021).

4.1.4. Barra de herramientas de instrumentos

Para visualizar el conjunto de instrumentos que se pueden utilizar se activa en la barra de modos de trabajo el modo de instrumentos virtuales seleccionando el icono .

Figura 5. Selección de instrumentos virtuales.

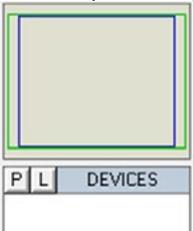


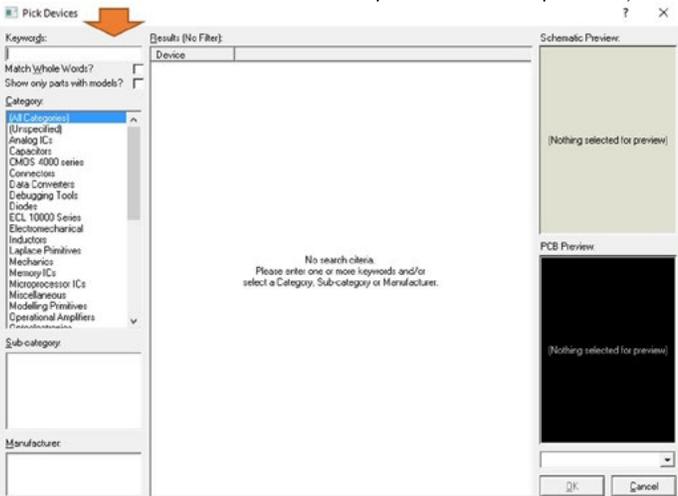
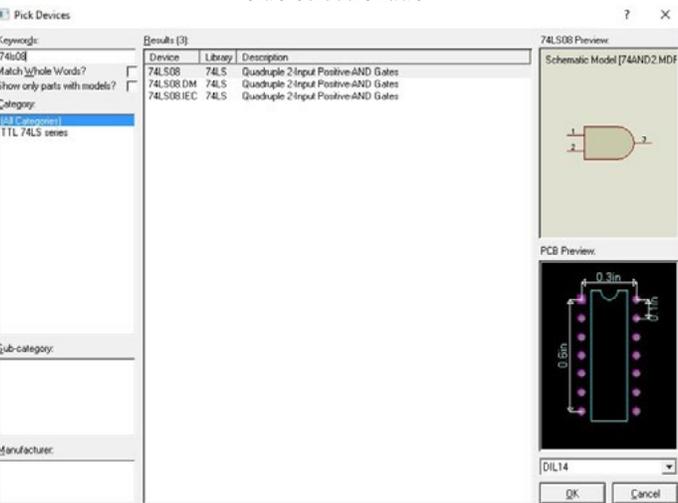
Fuente: Elaboración propia basada en la pantalla del software Proteus (LE, 2021).

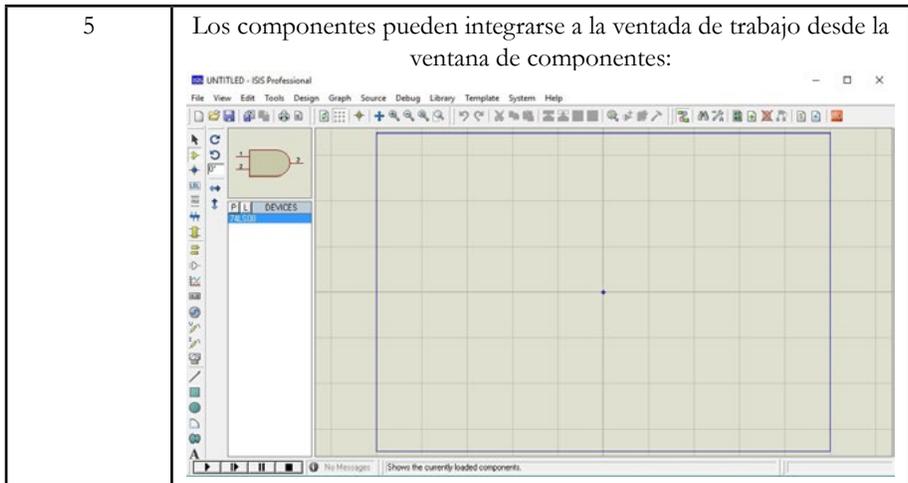
4.1.5. Colocación de componentes

Es posible integrar en la ventana de componente y bibliotecas los elementos que se requieren en el diseño mediante el procedimiento de la Tabla 2:

Tabla 2. Selección y colocación de componentes.

Paso 1	Seleccionar de la barra de modos de trabajo el icono de componentes: 
Paso 2	Seleccionar de la ventana de componentes y bibliotecas el icono  . 

<p>Paso 3</p>	<p>Escribir el número o nombre del componente en el espacio <i>Keywords</i>:</p> 
<p>Paso 4</p>	<p>El componente seleccionado se integra al listado una vez que haya sido seleccionado:</p> 



Fuente: Elaboración propia basada en la pantalla del software Proteus (I.E, 2021).

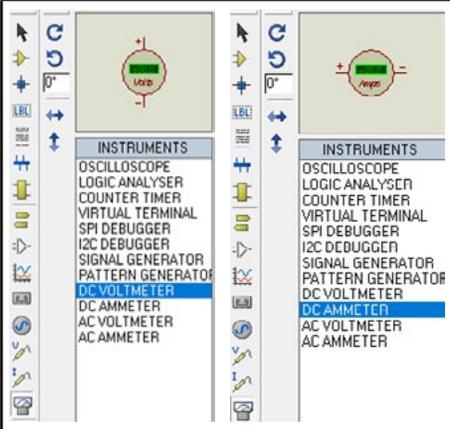
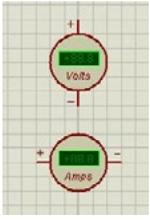
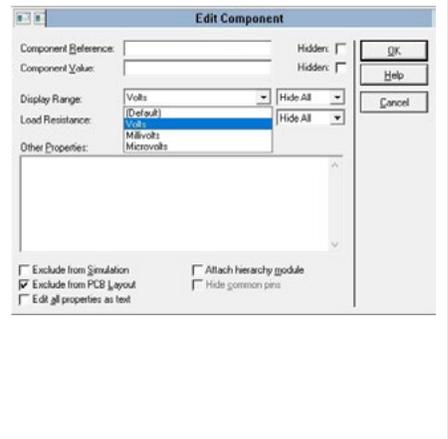
4.2. Instrumentos virtuales

Entre los principales instrumentos virtuales se encuentran el multímetro, el analizador de estados lógicos y el generador de patrones cuyo funcionamiento se describe en el presente apartado. Se explica tanto el procedimiento general para el acceso y configuración de los instrumentos como los pasos específicos para su uso.

4.2.1. Multímetro

El multímetro (polímetro o tester) permite medir las magnitudes eléctricas: voltaje o corriente alterna o continua (AC o DC), y resistencia. Se tiene un rango automático por lo que se puede utilizar de forma directa y tanto los valores de la resistencia interna como la corriente están predefinidos (Tabla 3).

Tabla 3. Características del multímetro.

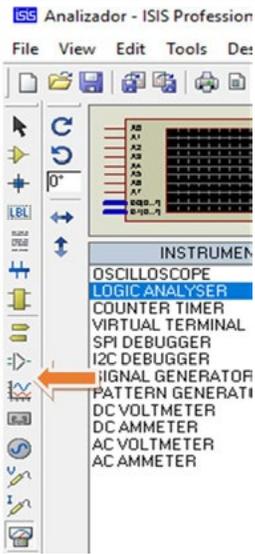
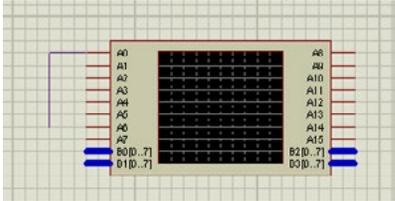
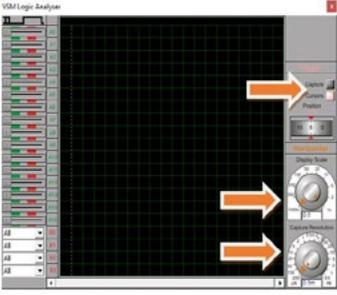
 <p>The image shows two side-by-side screenshots of the Proteus Instruments list. In both, the 'DC VOLT METER' and 'DC AMMETER' options are highlighted in blue. The top screenshot shows the 'DC VOLT METER' selected, and the bottom screenshot shows the 'DC AMMETER' selected. The list includes: OSCILLOSCOPE, LOGIC ANALYSER, COUNTER TIMER, VIRTUAL TERMINAL, SPI DEBUGGER, I2C DEBUGGER, SIGNAL GENERATOR, PATTERN GENERATOR, DC VOLT METER, DC AMMETER, AC VOLT METER, and AC AMMETER.</p>	<p>Para utilizarlo se selecciona el modo de instrumentos virtuales y en la lista de la ventana de componentes “DC VOL-TIMETER”, o “DC AMMETER” lo que permite colocarlos en la ventana de trabajo.</p>
<p>Al hacer doble clic en el instrumento se muestran sus características como el rango en volts, millivolts o microvolts o Amps, milliamps y microamps. Es indispensable considerar la polaridad del instrumento para su correcta conexión.</p>  <p>The diagram shows two meters on a grid. The top meter is a DC Voltmeter with a red '+' terminal at the top and a black '-' terminal at the bottom. The bottom meter is a DC Ammeter with a red '+' terminal on the left and a black '-' terminal on the right.</p>	 <p>The image shows the 'Edit Component' dialog box for a DC Voltmeter. The 'Component Reference' and 'Component Value' fields are empty. The 'Display Range' is set to 'Volts'. The 'Load Resistance' is set to 'DC'. The 'Other Properties' section includes checkboxes for 'Exclude from Simulation', 'Exclude from PCB Layout', 'Attach hierarchy module', and 'Hide gomon pins'. The 'Exclude from Simulation' and 'Exclude from PCB Layout' checkboxes are checked.</p>

Fuente: Elaboración propia basada en la pantalla del software Proteus (LE, 2021).

4.2.2. Analizador de estados

El analizador de buses y transmisión de datos permite verificar la información que viaja por hasta 40 canales de datos en lógica digital. Permite el seguimiento de buses de 8 x 1 bit y 4 x 8 bit (tabla 4).

Tabla 4. Características del analizador de estados.

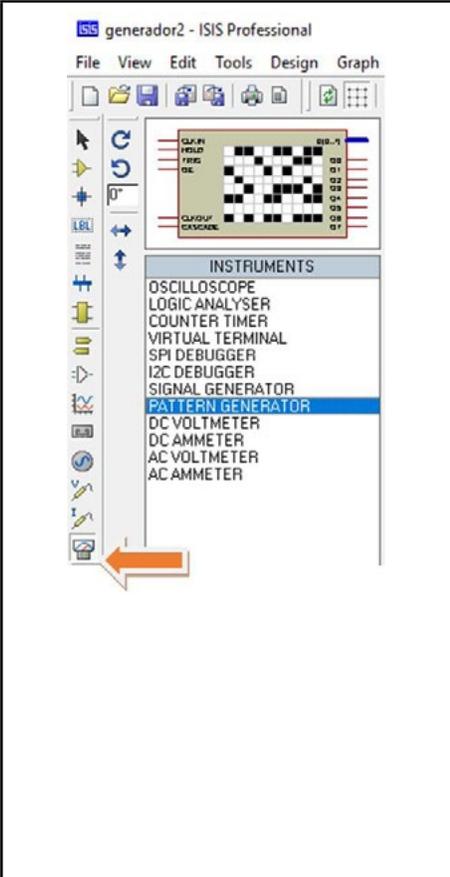
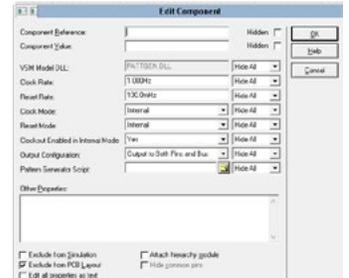
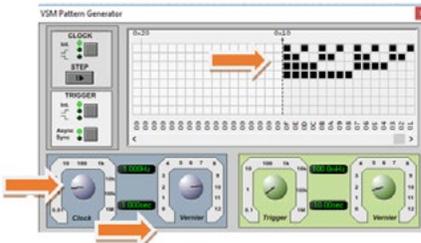
	<p>Para utilizarlo se selecciona el modo de instrumentos virtuales y en la lista de la ventana de componentes “LOGIC ANALYSER” lo que permite colocarlo en la ventana de trabajo.</p>  <p>Las salidas de datos que se desean analizar se vinculan a las entradas del analizador.</p>
<p>Al ejecutar la simulación</p>  <p>Aparece el visor</p>	
<p>Es importante no cerrar el visor al concluir el análisis sino que se debe detener o pausar la simulación.</p> <p>Los valores de la escala de <i>Display</i> y la resolución de captura pueden ajustarse directamente en el visor.</p> <p>Al presionar el botón de Captura  se pueden obtener las combinaciones de datos una vez que el botón ha cambiado a verde.</p>	

Fuente: Elaboración propia basada en la pantalla del software Proteus (LE, 2021).

4.2.3. Generador de patrones digitales

Permite crear un patrón de conjuntos de 8 bits de hasta 1kbyte y sirve como complemento del generador de señales analógicas. Funciona en modo gráfico e interactivo y permite el modo de visualización hexadecimal y decimal (tabla 5).

Tabla 5. Caracterización del generador de patrones

	<p>Para utilizarlo se selecciona el modo de instrumentos virtuales y en la lista de la ventana de componentes “PATTERN GENERATOR” lo que permite colocarlo en la ventana de trabajo.</p>  <p>Las salidas del generador se vinculan a las entradas del componente en el que se mostrarán los datos.</p> <p>Al hacer doble clic en el componente se puede ajustar sus parámetros en la ventana de edición.</p> 
<p>Al ejecutar la simulación</p>  <p>Aparece el visor</p> <p>Es importante no cerrar el visor al concluir el análisis sino que se debe detener o pausar la simulación.</p> <p>Los valores del reloj y el disparador pueden ajustarse de manera directa.</p> <p>El patrón a generar se establece presionando los valores en la cuadrícula.</p>	

Fuente: Elaboración propia basada en la pantalla del *software* Proteus (LE, 2021).

4.3. Construcción de circuitos

El primer paso para la captura esquemática consiste en colocar los componentes apropiados. Puede utilizar los siguientes métodos para localizar un componente.

1. Utilice la barra de herramientas de Componentes para navegar.
2. Busque un grupo de componentes / familia específica.

El segundo paso consiste en conectar los componentes. Para alambroslos automáticamente:

4. Haga clic en un pin del primer componente para iniciar la conexión
Mueva el ratón (junto al cursor aparece el alambre)
5. Haga clic en un pin del segundo componente para finalizar la conexión.



Realice: Prácticas de laboratorio 1-5

Práctica de laboratorio 1

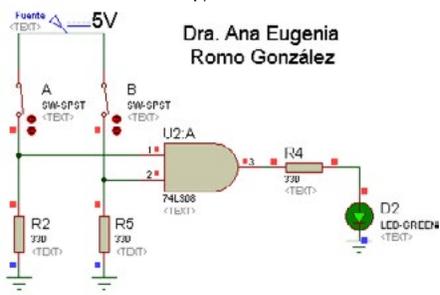
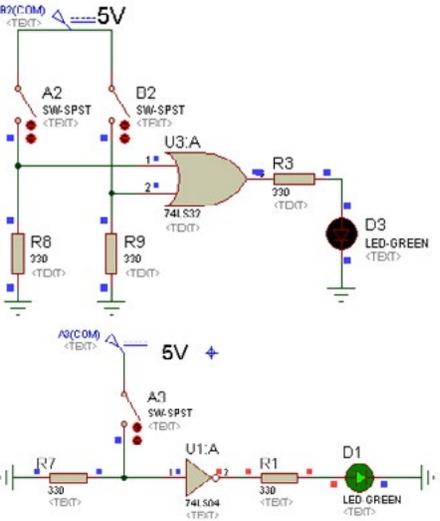
Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 1	Implementación de compuertas lógicas.			
Materiales:	Cantidad	Descripción	Cantidad	Descripción
	1	IC 74ls04	3	LED
	1	IC 74ls08	1	Protoboard
	1	IC 74ls32	1	Fuente de voltaje.
	2	Interruptores	1	Multímetro
	5	Resistencias de 330 Ω		
Procedimiento:	1	<p>Arme el circuito del diagrama y compruebe la tabla de verdad de cada una de las compuertas. Las resistencias deben ser DIGITAL (doble clic en el componente). Fuente: 5 Volts (<i>bottom</i>).</p> <p>Dra. Ana Eugenia Romo González UTJ. Mecatrónica</p>		
	2	<p>Utilice compuertas NOT, AND y OR para implementar y probar una compuerta EXOR.</p>		

Informe

Proceso	Descripción				
1	a	b	AND	OR	NOT (a)
	0	0			
	0	1			
	1	0			
	1	1			

Práctica de laboratorio 2

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 2	Simulación de compuertas lógicas.			

<p>Procedimiento:</p>	<p>1</p>	<p>Arme los siguientes circuitos en el simulador (SW-SPST), utilice las resistencias digitales.</p>  <p style="text-align: right;">Dra. Ana Eugenia Romo González</p>
		
	<p>2</p>	<p>Construya el circuito para implementar y probar una compuerta EXOR Utilice compuertas NOT, AND y OR.</p>
	<p>3</p>	<p>Verifique los valores de salida en tablas de verdad.</p> <p>Incluya un Texto con su nombre.</p> <p>Cargue un solo archivo del procedimiento 1 en la plataforma. Cargue un solo archivo del procedimiento 2 en la plataforma.</p>

Informe

Proceso	Descripción				
1	a	b	AND	OR	NOT (a)
	0	0			
	0	1			
	1	0			
	1	1			
2	a	b	EXOR		
	0	0			
	0	1			
	1	0			
	1	1			

Práctica de laboratorio 3

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 3	Simulación de circuito e informe de fallas.			

Procedimiento e Informe de fallas:	
1	<p>Arme el siguiente circuito empleando resistencias analógicas: (Battery – Cell – device)</p> <p>Dra. Ana Eugenia Romo Gonzalez UTJ. Mecatrónica</p>
2	Modifique la fuente para que entregue 10 volts.
3	Ejecute la simulación.
4	<p>Mida el voltaje y la corriente para:</p> <p>A1 = A2 = A3 = V1 = V2 =</p>
5	<p>Incluya un Texto con su nombre. Cargue un solo archivo del procedimiento 1 en la plataforma.</p>

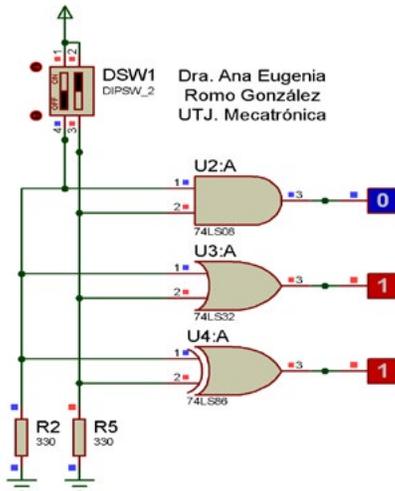
Práctica de laboratorio 4

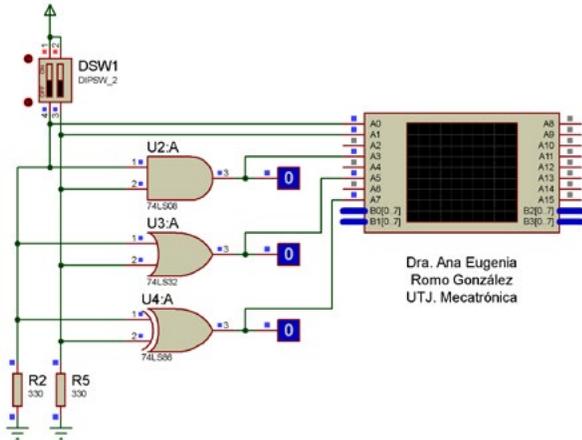
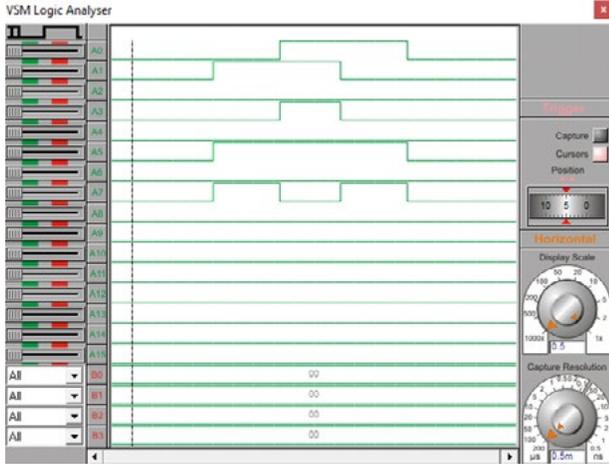
Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 4	Uso de analizador de estados lógicos.			

Procedimiento:

1

Arme el siguiente circuito y compruebe las tablas de verdad de las tres compuertas:



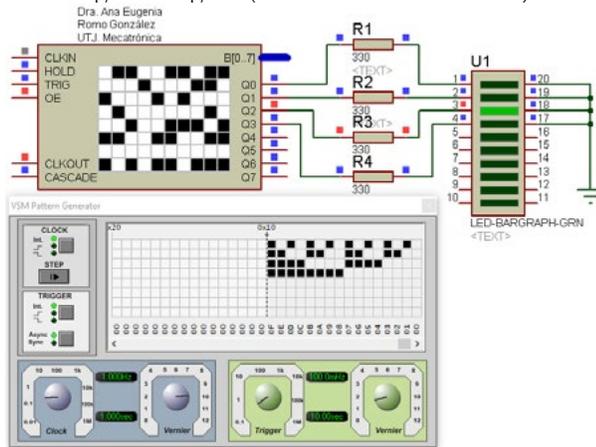
Procedimiento:	
2	<p>Agregue el analizador de estados lógicos como se muestra en la siguiente figura.</p>  <p style="text-align: right;">Dra. Ana Eugenia Romo González UTJ. Mecatrónica</p>
	
3	Coloque la escala del Display en 1000k y la resolución de captura en 200us.
4	Ponga en cero ambos interruptores (las entradas de las compuertas).
5	Presione el botón de captura (capture) y al cambiar a verde introduzca en los interruptores las combinaciones 00, 01, 10 y 11.
6	Detenga la simulación y compruebe el resultado de las compuertas en el analizador.
7	Con el botón derecho del ratón cambie el color de fondo del analizador de estados.
8	Incluya un texto con su nombre y cargue un solo archivo en la plataforma.

Práctica de laboratorio 5

Programa educativo		Grupo	Turno	Fecha
Asignatura				
Profesora	Dra. Ana Eugenia Romo González	Periodo		
No.	Descripción	Nombre del estudiante		
Práctica 5	Uso de generador de patrones.			

Procedimiento:

1 Implemente un contador binario de 0000 a 1111 como se muestra en la siguiente figura. (LED-BARGRAPH-GRN).



2 Edite el componente para ajustar los siguientes parámetros:
 Velocidad del reloj (*Clock Rate*) a 1.700 HZ
 Tasa de reinicio (*Reset Rate*) a 100.0mHz
 Salida de Reloj habilitada en modo interno (*Clockout Enabled in Internal Mode*) en Yes.

3 Incluya un Texto con su nombre.
 Cargue un solo archivo en la plataforma.

Capítulo **V**

Taxonomía de tipos de datos en lenguaje C

El contenido presentado forma parte del curso de Lenguaje de programación en el área de Mecatrónica.

Propósito

Elaborar programas en lenguaje C utilizando los puertos de comunicación para lograr controlar y monitorear una variable física de un proceso.

Competencias

Implementar sistemas de medición y control bajo los estándares establecidos, para el correcto funcionamiento de los procesos industriales.

The slide features a header with two logos: on the left, a circular logo for 'CENTRO DE INVESTIGACIÓN Y DESARROLLO' with a gear and circuit icon; on the right, the 'UTJ' logo for 'Universidad Tecnológica de Jalisco' with the tagline 'Innovación y Excelencia'. The main title is 'Tipos de datos en Lenguaje C' in a large, bold, dark blue font. Below the title, the author's name 'Dra. Ana Eugenia Romo González' and her field 'Lenguaje de programación' are listed in a smaller, dark blue font. At the bottom, a dark grey footer contains the author's name on the left and contact information on the right: 'Luis J. Jiménez 577, Col. 1° de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900, Guadalajara, Jalisco, México, www.utj.edu.mx'.

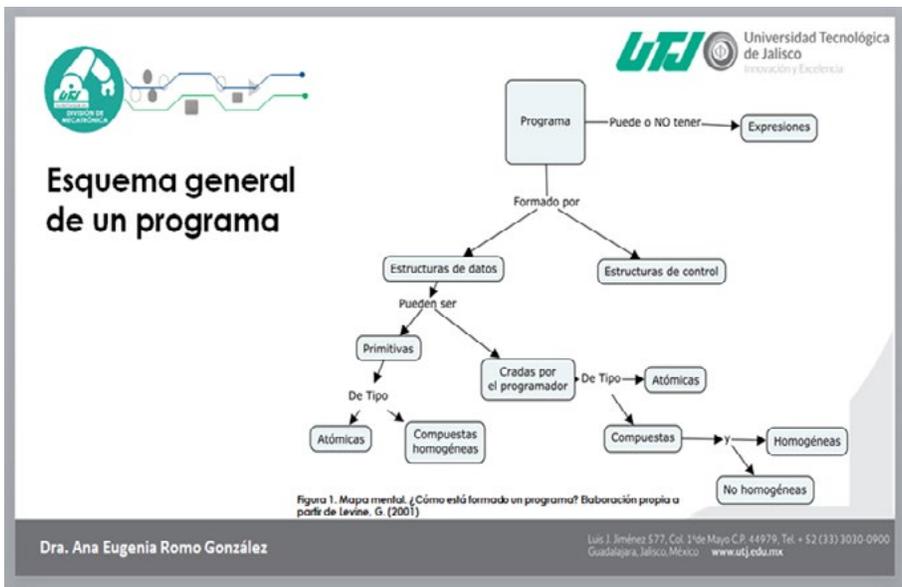
Contexto

Un programa está formado por:

- a) Estructuras de datos.
- b) Estructuras de control.

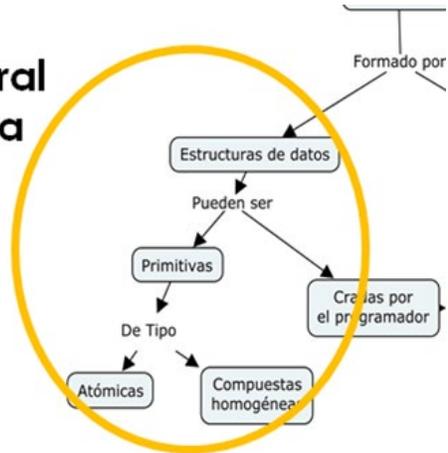


Puede o no tener expresiones.



Los tipos de datos en Lenguaje C se refieren a las estructuras de datos en el programa:

Esquema general de un programa



Con respecto a las estructuras de datos primitivas; en lenguaje C pueden clasificarse de la siguiente forma:

Tipos de datos primitivos atómicos en C

1. Carácter
2. Numérico
3. Lógicos (booleanos)
4. Apuntadores

Calificativo **unsigned** aplica a todos los tipos

Dra. Ana Eugenia Romo González

UTeJ Universidad Tecnológica de Jalisco
Innovación y Excelencia

Luis J. Jiménez 577, Cal. 1º de Mayo C.P. 44979, Tel. + 52 (33) 3030 0900
Guadalajara, Jalisco, México www.utj.edu.mx



Universidad Tecnológica de Jalisco
Innovación y Excelencia

Tipos de datos primitivos atómicos en C

1. Carácter



Se puede representar como numéricos (0 al 9), letras (a-z) y símbolo (_).

Tipo de dato	Rango	Tamaño de bits
char	-127 a 127	8 bits
unsigned char	0 a 255	
signed char	-127 a 127	

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1° de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx



Universidad Tecnológica de Jalisco
Innovación y Excelencia

Ejemplos de Declaración de variables y constantes

1. Carácter



Sintaxis para la declaración

Variables
Tipo identificador;

Constantes
const tipo identificador = valor;

Ejemplos
char dato;
unsigned char V1;
char valor_inicial;
const char temporal = 'a';

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1° de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx

Tipos de datos primitivos atómicos en C

2. Numéricos



Puede ser entero o real, dependiendo de los valores a almacenar.

Tipo		Tamaño de bits
Enteros	byte	1 byte (8 bits)
	short	2 bytes (16 bits)
	int	4 bytes (32 bits)
	long	8 bytes (64 bits)
Reales	float	4 bytes (32 bits)
	double	8 bytes (64 bits)

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1da Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx



Ejemplos de Declaración de variables y constantes

Sin `axis` para la declaración

Variables

Tipo identificador;

Constantes

`const` tipo identificador = valor;

Ejemplos

```
byte n1;
```

```
short elemento;
```

```
int resultado;
```

```
long suma;
```

```
float n2;
```

```
double x;
```



2. Numérico

Ejemplo de declaración de constante

```
const float PI = 3.1416;
```

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1da Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx

Tipos de datos primitivos atómicos en C

3. Lógicos (Booleanos)



Un tipo lógico o **booleano en C** es un entero que utiliza:

- cero para falso y
- no cero para verdadero

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1° de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx



Ejemplos de Declaración de variables y constantes

Sintaxis para la declaración

Variables

Tipo identificador;

Constantes

const tipo identificador = valor;



Ejemplos

```
int bandera = 0;
```

```
int valor = 1;
```

3. Lógicos (booleanos)

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1° de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx

Tipos de datos primitivos atómicos en C

4. Apuntadores



- El operador unario `&` proporciona la dirección de un objeto, por ejemplo: `p = &c;`;
- Solo aplica objetos en memoria.
- El operador `*` es el operador de indirección o desreferencia.

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1 de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx



Ejemplos de Declaración de variables y constantes

Sintaxis para la declaración

Variables

Tipo identificador;

Constantes

const tipo identificador = valor;

Ejemplos

```
char *ap;
```

```
char dato;
```

```
ap = &dato;
```



4. Apuntadores

Dra. Ana Eugenia Romo González

Luis J. Jiménez 577, Col. 1 de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900
Guadalajara, Jalisco, México www.utj.edu.mx

The slide features the UTJ logo and name in the top right corner. The main title is 'Revisión de la sesión'. Below it, the text reads: 'Los tipos de datos primitivos atómicos en C' followed by 'Son indispensables para construir programas porque permiten declarar variables y constantes, mismos que forman parte del esquema de un programa.' At the bottom, the presenter's name 'Dra. Ana Eugenia Romo González' is on the left, and contact information 'Luis J. Jiménez 577, Col. 1° de Mayo C.P. 44979, Tel. + 52 (33) 3030-0900 Guadalajara, Jalisco, México www.utj.edu.mx' is on the right.

Ejemplos de declaraciones en el compilador en línea https://www.onlinegdb.com/online_c++_compiler

The screenshot shows an online C++ compiler interface with a toolbar at the top containing buttons for Run, Debug, Step, Share, Save, Beautify, and Download. The language is set to C++. The code editor shows the following code:

```
1 //.....
2
3                               Online C++ Compiler.
4                               Code, Compile, Run and Debug C++ program online.
5
6   Dra. Ana Eugenia Romo - Ingeniería Mecatrónica
7   ...../
8
9   #include <stdio.h>
10
11  using namespace std;
12
13  int main()
14  {
15
16      // declaraciones de variables y constantes
17      int x;
18      float valor;
19      char temporal;
20      const float IVA = 3.1416;
21
22      return 0;
23  }
24
```


Referencias Bibliográficas

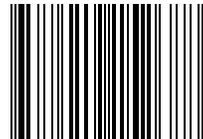
- Caoto (s/F). *Imagen de George Boole*. Recuperada de: https://c1.staticflickr.com/9/8050/8146153622_9d29b47980.jpg
- Ferrater Mora, J., & Leblanc, H. (2015). *Lógica matemática*.
- Floyd, T. L., Salas, R. N., González, L. M. O., & López, G. P. (2007). *Principios de circuitos eléctricos*. Pearson Educación.
- Labcenter Electronics. (2021). *Circuit Simulation*. Recuperado de: <https://www.labcenter.com/simulation/>
- Levine Gutiérrez, G. (2001). *Computación y programación moderna: perspectiva integral de la informática*. Pearson Educación de México.
- National Instruments Corp_a, (2021). *Multimeter*. Recuperado de: <https://zone.ni.com/reference/en-XX/help/375482B-01/multisim/multimeter/>
- Nashelsky, L., Chong, D. C., & Gómez, E. A. (1989). *Fundamentos de tecnología digital*. Limusa.
- Shoenfield, J. R. (2018). *Mathematical logic*. AK Peters/CRC Press.
- Tocci, R. J., Widmer, N. S., & Moss, G. L. (2007). *Digital systems*. Pearson
- _____, (2021). *Explore Instrumentos NI ELVIS*. Recuperado de: <https://www.ni.com/es-mx/innovations/white-papers/18/explore-ni-elvis-instruments.html#5>

Tecnologías aplicadas a la enseñanza de las ciencias computacionales y la electrónica

Se terminó de editar en junio de 2022 en las oficinas de Astra Ediciones
S. A. de C. V. Avenida Acueducto 829, Colonia Santa Margarita, C. P.
45140, Zapopan, Jalisco, México.
edicion@astraeditorial.com.mx



ISBN: 978-84-19152-68-8



9 788419 152688