

SESIÓN 2

COMPUTACIÓN Y MATEMÁTICAS.

I. CONTENIDOS:

1. Lógica proposicional.
2. Las tablas de verdad.
3. Los sistemas numéricos y el álgebra booleana.
4. Compuertas lógicas digitales.
5. Circuitos integrados.
6. El código ASCII.

II. OBJETIVOS:

Al término de la Sesión, el alumno:

- Conocerá las aportaciones de las matemáticas a la informática.
- Sintetizará los elementos conceptuales de la lógica proposicional.
- Conocerá las tablas de verdad de las fbf básicas.
- Analizará las relación entre el algebra booleana y las compuertas lógicas.
- Conocerá los conceptos básicos del los circuitos integrados.
- Comprenderá la función del código ASCII.

III. PROBLEMATIZACIÓN:

Comenta las preguntas con tu Asesor y selecciona las ideas más significativas.

- ¿Qué área de las matemáticas tiene mayor influencia en la informática?
- ¿Cómo utilizamos las matemáticas en la construcción de programas?
- ¿Qué estudia la lógica proposicional?

IV. TEXTO INFORMATIVO-FORMATIVO:

1.1. Lógica proposicional

La lógica es la disciplina que se encarga de los razonamientos, los cuales se basan en el pensamiento deductivo, a partir de una pista o conocimiento verdadero se busca llegar a otro conocimiento o certeza. ¿Cómo funciona? Primero los datos: Cada idea se expresa en forma de una oración simple o proposición, que se puede calificar como falsa o verdadera, a esta oración se le conoce como juicio. Para cada variable del problema le corresponde un juicio, de tal forma que se puede hablar de ellos en forma indistinta. Cada variable lógica se representa por un dato de entrada, al cual se le puede adjudicar una letra del alfabeto, normalmente las de inicio y en mayúsculas. Ejemplo: En el evento de identificar un número menor de 10.

Variable	Símbolo
Es un numero natural	A
Es un numero primo	B
Esta formado por solo un digito	C

En esta etapa es la formulación, esta tomando en cuenta todas las respuestas posibles y no solo la correcta, porque es posible que existan varias correctas, en este caso hay 10. Las funciones básicas son:

(No) negación. Consiste en negar el juicio, Ejemplo

	Juicio	Valoración	Negación	Valoración
Juicio	4 es numero par	Cierto	4 no es numero par	Falso

(Y) conjunción.

Es una función lógica que se utiliza para unir dos juicios, los cuales se tienen que cumplir para que el resultado se considere verdadero. Ejemplos

Función lógica: "El 9 es número impar y múltiplo de 3"

Representación: A Y B (lógica)

Función: "Brad Pitt es rubio y es actor"

Representación: A Y B (lógica)

En ambos casos los dos juicios son verdaderos por lo que la respuesta final es verdadero. Pero si por lo contrario afirmamos que: "9 es múltiplo de 3 y es primo" la respuesta es falso, ya que 9 si es múltiplo de 3 pero por lo mismo no es primo.

(°) Disyunción.

Este operador lógico se utiliza para elegir entre dos opciones, se considera la respuesta verdadera si tan solo uno de los juicios lo es. Ejemplo

Función: "8 es un número par o es múltiplo de 3" Representación: A ° B, (lógica)

La respuesta es verdadera porque si bien 8 no es múltiplo de 3 si es número par.

La única opción de tener falso es que ambos juicios sean falsos. Como en "5 es par o múltiplo de 3" ya que ni es par ni múltiplo de 3.

1.2. Tablas de verdad

Juicio	Negación
A	A'
F	V
V	F

Valor de variable	Negación
1	0
0	1

Tabla de verdad intersección

Juicio A	Juicio B	A Y B
F	V	F
V	V	V
F	F	F
V	F	F

Tabla de verdad disyunción

Juicio A	Juicio B	A ° B
F	V	V
V	V	V
F	F	F
V	F	V

1.3. Sistemas numéricos y el álgebra booleana

Para comunicarnos utilizamos un lenguaje, las computadoras también, para que sea práctico y rápido se utiliza el sistema binario. Para poder ingresar datos a un sistema informático, se debe transformar nuestro lenguaje a un lenguaje de 0 y 1. Al estado de ceros y unos se le conoce como sistema digital. No hay valores intermedios solo verdadero (1) y falso (0) Para convertir cualquier cantidad del sistema decimal al sistema binario, solo se divide entre 2 la cantidad y se van registrando los residuos de derecha a izquierda hasta que se tiene un cociente final y se anota. Los dígitos no deben ser mayores de 1. Ejemplo: Cambiar 123 a binario

$$\begin{array}{r}
 15 \\
 2 \overline{)30} \\
 \underline{20} \\
 10 \\
 \underline{10} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 3 \\
 2 \overline{)7} \\
 \underline{4} \\
 3 \\
 \underline{2} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 7 \\
 2 \overline{)15} \\
 \underline{10} \\
 5 \\
 \underline{4} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 30 \\
 2 \overline{)61} \\
 \underline{40} \\
 21 \\
 \underline{20} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 61 \\
 2 \overline{)123} \\
 \underline{60} \\
 3 \\
 \underline{2} \\
 1
 \end{array}$$

Se baja el último cociente y queda: $123 = 1111001$ en binario

Para comprobar seguir su valor en la tabla de números binarios

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2	1
504	256	128	64	32	16	8	4	2	1
			1	1	1	1	0	0	1

Sumamos el valor de cada uno según la casilla a la que pertenece y tenemos: $64+32+16+8+1=123$

Para otros sistemas con base diferente a 10 se siguen criterios semejantes, los más utilizados son el binario, el base 4, el decimal y el base 16.

1.4. Lógica Booleana

La lógica booleana se basa en el principio de falso y verdadero, asignando un valor de 0 y 1 a cada una de esas opciones, como base no puede haber valores intermedios porque nada puede ser falso y cierto a la vez. Boole introdujo una serie de símbolos nuevos basados en el sistema binario y en el lenguaje desarrollando las anteriores tablas de verdad en ceros y unos. Actualmente dichas tablas se comprueban en circuitos electrónicos mediante compuertas lógicas. Las funciones en algebra booleana

Negación

Valor de variable	Negación
1	0
0	1

Conjunción en algebra booleana

A	B	=	A Y B A*B
1	1	=	1
0	1	=	0
1	0	=	0
0	0	=	0

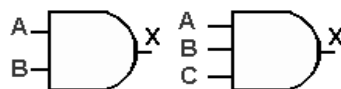
Disyunción en algebra booleana

A	B	=	A°B A+B
1	1	=	1
0	1	=	1
1	0	=	1
0	0	=	0

1.5. Compuertas lógicas digitales.

En un circuito electrónico, las entradas son interruptores, abiertos para los ceros y cerrados para los unos, mientras que la respuesta se visualiza en una lámpara o led, donde encendido es 1 y apagado es cero. Los circuitos integrados están construidos con transistores, diodos y resistencias, los cuales funcionan como las compuertas lógicas AND, OR Y NOT.

Compuerta AND: Se representa de la siguiente forma. La primera es la representación de una **compuerta AND** de 2 entradas y la segunda de una **compuerta AND** de 3 entradas.



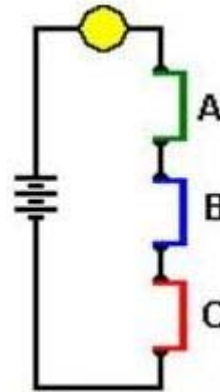
La **compuerta Y lógica** más conocida tiene dos entradas A y B, aunque puede tener muchas más (A,B,C, etc.) y sólo tiene una salida X. La **compuerta AND** de 2 entradas tiene la siguiente tabla de verdad.

A	B	X
1	1	1
0	1	0
1	0	0
0	0	0

Se puede ver claramente que la salida X solamente es "1" (1 lógico, nivel alto) cuando la entrada A como la entrada B están en "1". En otras palabras... **La salida X es igual a 1 cuando la entrada A y la entrada B son 1.** Una **compuerta AND** de 3 entradas se puede implementar con interruptores, como se muestra en el siguiente diagrama. La tabla de verdad se muestra al lado derecho donde: **A** = Abierto y **C** = Cerrado.

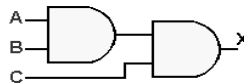
A	B	C	X
A	A	A	Apagado
A	A	C	Apagado
A	C	A	Apagado
A	C	C	Apagado
C	A	A	Apagado
C	A	C	Apagado
C	C	A	Apagado
C	C	C	Encendida

Diagrama eléctrico



Una compuerta AND de múltiples entradas puede ser creada conectando compuertas simples en series. Por ejemplo con dos sencillas en serie se puede construir una compuerta de 3 entradas

Compuerta OR o lógica



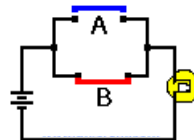
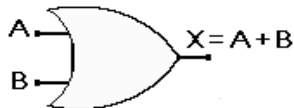
La salida X de la compuerta OR será "1" cuando la entrada "A" o la entrada "B" estén en "1".

En una compuerta OR, la salida será "1", cuando en cualquiera de sus entradas haya un "1".

La compuerta OR se representa con la siguiente función booleana: $X = A+B$ ó $X = B+A$

Compuerta OR de dos entradas.

La representación de la **compuerta "OR"** de 2 entradas y su tabla de verdad se muestran a continuación.



Su tabla de verdad en circuitos queda como:

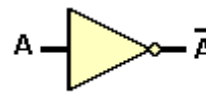
A	B	X=A+B
A	A	Apagado
A	C	Encendida
C	A	Encendida
C	C	Encendida

La **compuerta OR** también se puede implementar con interruptores como se muestra en la figura de arriba a la derecha, en donde se puede ver que: cerrando el interruptor A "O" el interruptor B se encenderá la luz. "1" = cerrado, "0" = abierto, "1" = luz encendida

La **compuerta NOT**

Esta compuerta entrega en su salida el inverso (opuesto) de la entrada. El símbolo y la tabla de verdad son los siguientes:

A	X=A'
A	C
C	A

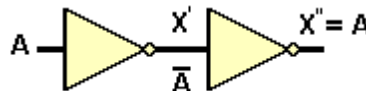


La salida de una **compuerta NOT** tiene el valor inverso al de su entrada. En el caso del gráfico anterior la salida $X = A$. Esto significa que:

- o Si a la entrada tenemos un "1" lógico, a la salida hará un "0" lógico y... Si a la entrada tenemos un "0" lógico a la salida habrá un "1" lógico.

Nota: El apóstrofe en la siguiente expresión significa "negado". Entonces: $X = A'$ es lo mismo que $X = \bar{A}$

Las **compuertas NOT** se pueden conectar en cascada, logrando después de dos compuertas, la entrada original.



Representación esquemática, física y electrónica de una compuerta lógica AND.

Solo utilizamos la primera para diseñar y la segunda para construir el circuito, el último esquema es la construcción interna.

Con las compuertas lógicas se maneja la unidad de información, el **bit**. El cual es 1 o 0. Si reunimos una palabra de 11 bits tenemos el **byte**. Un **byte** representa un carácter de un número, letra o símbolo específico. La lista de códigos para cada uno de los datos o letras se conoce como código **ASCII** (American Estándar Code for Information Interchange), cada código corresponde a un número en binario. *Ejemplo:* El código para la letra z es 111 1010 que corresponde al número 122 estos son 7 bits, los que faltan para completar 11 bits son la clave para la función ALT, así que si quiero que aparezca la letra z solo presiono ALT y el número 122 en el teclado y listo.

1.6. Código ASCII completo

0	32	64	U	96	ˆ	128	Û	160	à	192	L	224	α		
1	☐	33	!	65	À	97	a	129	ü	161	ı	225	β		
2	⊗	34	"	66	B	98	b	130	é	162	ó	194	τ	226	Γ
3	♥	35	#	67	C	99	c	131	â	163	ú	195	†	227	Π
4	♣	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	Σ
5	♠	37	%	69	E	101	e	133	à	165	Ñ	197	‡	229	σ
6	♣	38	&	70	F	102	f	134	ã	166	ã	198	‡	230	μ
7	•	39	'	71	G	103	g	135	ç	167	ç	199		231	τ
8	◼	40	(72	H	104	h	136	ê	168	ê	200	u	232	ϑ
9	◊	41)	73	I	105	i	137	ë	169	ƒ	201		233	θ
10	◊	42	*	74	J	106	j	138	è	170	ƒ	202		234	Ω
11	δ	43	+	75	K	107	k	139	ÿ	171	½	203		235	δ
12	♀	44	,	76	L	108	l	140	î	172	¼	204		236	ω
13	♂	45	-	77	M	109	m	141	ì	173	ı	205	=	237	ϕ
14	♂	46	.	78	N	110	n	142	ñ	174	«	206		238	€
15	*	47	/	79	O	111	o	143	ã	175	»	207	⊥	239	Π
16	▶	48	0	80	P	112	p	144	É	176	▒	208		240	≡
17	◀	49	1	81	Q	113	q	145	æ	177	▒	209	‡	241	±
18	‡	50	2	82	R	114	r	146	ff	178	▒	210		242	≥
19	!!	51	3	83	S	115	s	147	ô	179		211	u	243	≤
20	¶	52	4	84	T	116	t	148	ö	180	†	212	l	244	ƒ
21	§	53	5	85	U	117	u	149	ò	181	‡	213	ƒ	245	J
22	—	54	6	86	V	118	v	150	û	182		214		246	÷
23	‡	55	7	87	W	119	w	151	ù	183		215		247	≈
24	↑	56	8	88	X	120	x	152	ÿ	184	‡	216	‡	248	°
25	↓	57	9	89	Y	121	y	153	ö	185		217	J	249	·
26	→	58	:	90	Z	122	z	154	Ü	186		218	ƒ	250	·
27	←	59	:	91	[123	{	155	ç	187		219	▒	251	√
28	↵	60	<	92	\	124		156	£	188		220	▒	252	n
29	↶	61	=	93]	125	}	157	¥	189		221		253	z
30	▲	62	>	94	^	126	~	158	℞	190	J	222		254	■
31	▼	63	?	95	_	127	△	159	f	191	‡	223	▒	255	