

# Sistemas de Computación 2

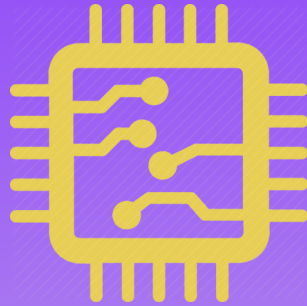
« Redes »

Introducción

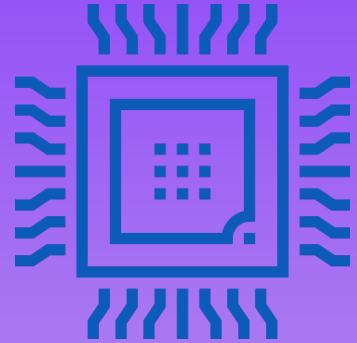
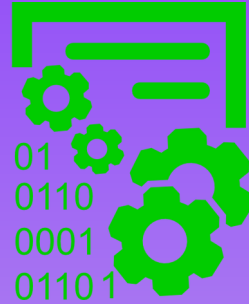


# Hardware, Software y Firmware

**Hardware:** Circuitos que controlan la electricidad.



**Software:** Electricidad.



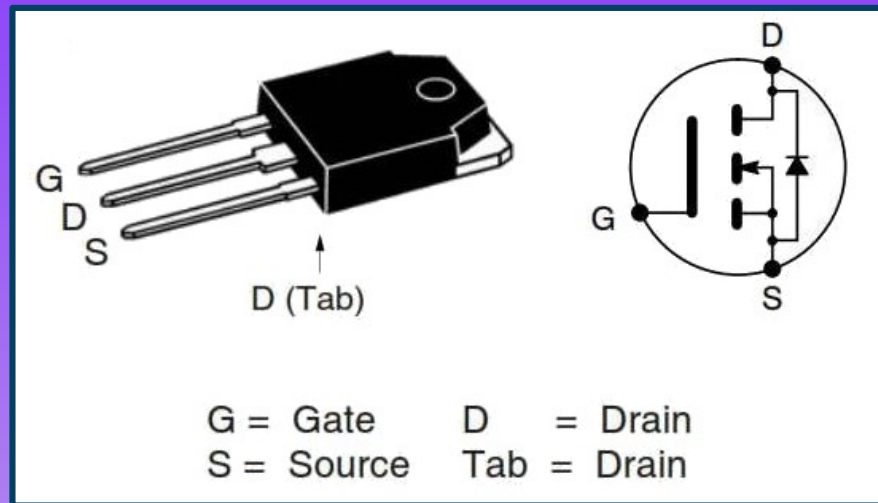
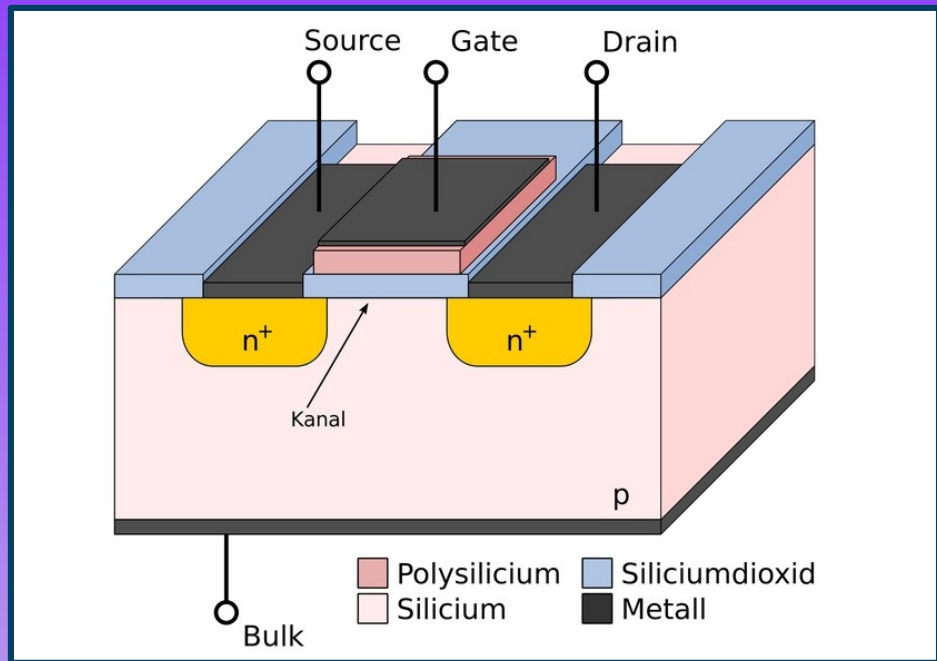
**Firmware:** conjunto de instrucciones que fijan la lógica primaria que ejerce el control de los circuitos de alguna clase de artefacto. Funciona como el nexo entre las instrucciones que llegan al dispositivo desde el exterior y sus diversas partes electrónicas. Forma parte del **hardware** ya que se encuentra integrado a la electrónica, pero también está considerado como parte del **software** al estar desarrollado bajo algún lenguaje de programación.

# El Transistor

La computadora está compuesta por conmutadores de electricidad que permiten la conexión y desconexión de circuitos por medio de impulsos eléctricos, esto lo consigue mediante el **transistor**.

**Transistor**: Elemento que conmuta los circuitos de la computadora con una señal eléctrica. El transistor utilizado en la computadora es el **MOSFET** (Metal-oxide-semiconductor Field-effect transistor) Transistor de efecto de campo metal-óxido-semiconductor, debido a que se pueden construir de tamaños más pequeños que los bipolares y su velocidad de conmutación (apertura y cierre) es muy rápida, unos nanosegundos, utiliza baja potencia para llevar a cabo su propósito y la disipación de la energía en términos de pérdida es muy pequeña, lo que hace que sea un componente importante en las modernas computadoras y dispositivos electrónicos como los teléfonos celulares, relojes digitales, juguetes, robots y calculadoras.

# El Transistor

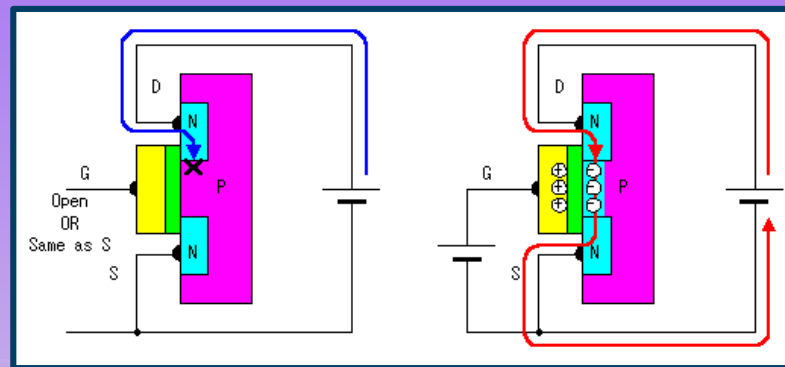


🌐 Transistor, ¿Cómo funciona?

🌐 ¿Qué es un MOSFET? ¿Cómo funciona?

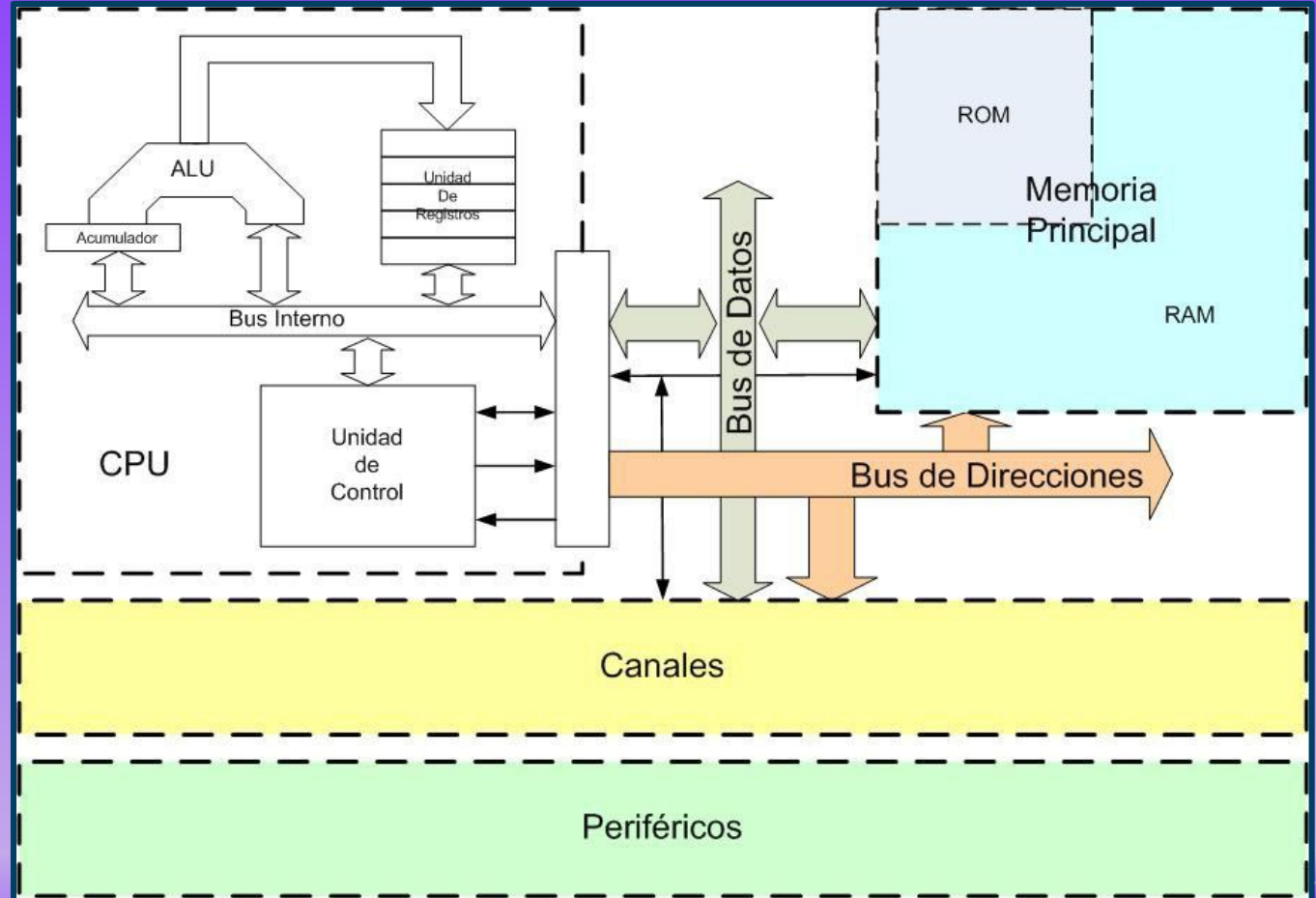
🌐 Funcionamiento del transistor MOSFET

🌐 Funcionamiento de los Transistores MOSFET y JFET



# Arquitectura de Von Neumann

Es un modelo conceptual que muestra cómo funciona una computadora y la forma en la que se interconectan los componentes de hardware. Este modelo está basado en el explicado por el físico y matemático John Von Neumann en el año 1945.



# Arquitectura de Von Neumman

El CPU (Central Processing Unit) o Unidad Central de Proceso está compuesta por lo siguiente:

- Unidad de Control
- Unidad Aritmético Lógica
- Unidad de Registros y Acumulador
- Bus Interno de Comunicación

Tipos de registros

- IR – Instruction Register
- PC – Program Counter
- MDR - Memory Data Register
- MAR - Memory Address Register
- IODR - IO Data Register - Uno por canal al exterior
- IOAR - IO Address Register

Buses

Estados de la Computadora:

- 0- Estado de tensión cero o de No circulación de corriente .
- 1- Estado de +5volt de tensión o de circulación de corriente.
- Z- Estado de Alta Impedancia, impide que se produzca un corto circuito en el bus Estado en que no se envía ni se recibe electricidad, "1" o "0"

Cuando los buses son cables, dos registros no pueden acceder simultáneamente en forma de escritura. Únicamente uno envía (escribe) y otro recibe (lee), los restantes se encuentran en estado Z o de Alta Impedancia. El Bus de control permite determinar que compuerta o puerto se abre de los diferentes que poseen tanto la CPU como los diferentes canales de Entrada/Salida.

# Arquitectura de Von Neumman

Dirección I/O: Todo periférico debe tener una dirección de entrada / salida para poder comunicarse con la CPU.

Las tres técnicas más extendidas de gestión de I/O son:

- Polling o espera activa: La CPU se encarga de la transferencia de información consultando continuamente el estado del dispositivo periférico. Simple e ineficiente.
- Uso de interrupciones: La CPU se encarga de la transferencia de información pero el dispositivo periférico le notifica los cambios de estado mediante una interrupción.
- DMA (Direct Memory Access): El controlador DMA se encarga de toda la transferencia de información (puede ser un bloque, y puede requerir de conversión). Al finalizar el controlador DMA utiliza una interrupción para notificarlo a la CPU. Con esta técnica, la CPU programa al controlador DMA para realizar el trabajo y queda libre (para realizar otras tareas). Es la técnica más eficiente.

Memoria Principal

- RAM (Random Access Memory) o Memoria de Acceso Aleatorio. Se puede acceder directamente a cualquier dirección de memoria. Es volátil. Es rápida.
- ROM (Read Only Memory) o Memoria de Solo Lectura. También es de acceso aleatorio. No es volátil. Es lenta.

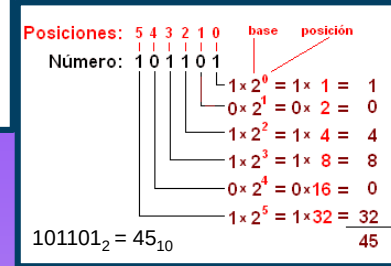
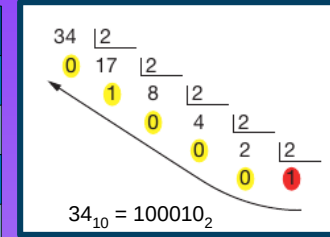
Periféricos

- Entrada
- Salida
- Almacenamiento

# Sistema Binario

- Se basa en el Álgebra de Boole.
- La menor unidad posible es el bit (Binary digIT), un 0 o un 1.
- Un nibble está compuesto por 4 bits.
- El código BCD (Decimal Codificado en Binario) permite representar los dígitos decimales en binario.
- El código ASCII (Código Americano Standard para el Intercambio de Información) permite la codificación de todos los caracteres alfanuméricos. Existe el original formado por 7 bits y el extendido por 8 bits, los cuales constituyen un BYTE (BinarY ocTET)

Decimal	Binario	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



Decimal	Múltiplos	Factor b	Equivalente
256	Byte	$2^0$	8 bits
$10^3$	Kilo	$2^{10}$	1024 B
$10^6$	Mega	$2^{20}$	1024 KB
$10^9$	Giga	$2^{30}$	1024 MB
$10^{12}$	Tera	$2^{40}$	1024 GB
$10^{15}$	Peta	$2^{50}$	1024 TB
$10^{18}$	Exa	$2^{60}$	1024 PB
$10^{21}$	Zetta	$2^{70}$	1024 EB
$10^{24}$	Yotta	$2^{80}$	1024 ZB



# ASCII

Caracteres de control ASCII			Caracteres ASCII imprimibles									ASCII extendido												
DEC	HEX	Simbolo ASCII	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	
00	00h	NULL	(carácter nulo)	32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
01	01h	SOH	(inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	ü	161	A1h	í	193	C1h	ł	225	E1h	ó
02	02h	STX	(inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	é	162	A2h	ó	194	C2h	ł	226	E2h	ô
03	03h	ETX	(fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	ú	195	C3h	ł	227	E3h	ö
04	04h	EOT	(fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ÿ	196	C4h	ł	228	E4h	ÿ
05	05h	ENQ	(enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	å	165	A5h	ÿ	197	C5h	ł	229	E5h	ÿ
06	06h	ACK	(acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	â	166	A6h	ÿ	198	C6h	ł	230	E6h	ÿ
07	07h	BEL	(timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	°	199	C7h	ł	231	E7h	ÿ
08	08h	BS	(retroceso)	40	28h	(	72	48h	H	104	68h	h	136	88h	è	168	A8h	¿	200	C8h	ł	232	E8h	ÿ
09	09h	HT	(tab horizontal)	41	29h	)	73	49h	I	105	69h	i	137	89h	ë	169	A9h		201	C9h	ł	233	E9h	ÿ
10	0Ah	LF	(salto de linea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	è	170	AAh		202	CAh	ł	234	EAh	ÿ
11	0Bh	VT	(tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ï	171	ABh	½	203	CBh	ł	235	EBh	ÿ
12	0Ch	FF	(form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	ì	172	ACH	¾	204	CCh	ł	236	ECh	ÿ
13	0Dh	CR	(retorno de carro)	45	2Dh	.	77	4Dh	M	109	6Dh	m	141	8Dh	í	173	ADh		205	CDh	ł	237	EDh	ÿ
14	0Eh	SO	(shift Out)	46	2Eh	.	78	4Eh	N	110	6Eh	n	142	8Eh	ÿ	174	ACh	«	206	CEh	ł	238	EEh	ÿ
15	0Fh	SI	(shift In)	47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	ÿ	175	ACh	»	207	CFh	ł	239	EFh	ÿ
16	10h	DLE	(data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	ÿ	176	B0h		208	D0h	ł	240	F0h	ÿ
17	11h	DC1	(device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h		209	D1h	ł	241	F1h	±
18	12h	DC2	(device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h		210	D2h	ł	242	F2h	
19	13h	DC3	(device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h	ø	179	B3h		211	D3h	ł	243	F3h	¼
20	14h	DC4	(device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h	ö	180	B4h		212	D4h	ł	244	F4h	
21	15h	NAK	(negative acknowle.)	53	35h	5	85	55h	U	117	75h	u	149	95h	ò	181	B5h		213	D5h	ł	245	F5h	
22	16h	SYN	(synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h	ú	182	B6h		214	D6h	ł	246	F6h	
23	17h	ETB	(end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h	ÿ	183	B7h		215	D7h	ł	247	F7h	
24	18h	CAN	(cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h		216	D8h	ł	248	F8h	
25	19h	EM	(end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h	ÿ	185	B9h		217	D9h	ł	249	F9h	
26	1Ah	SUB	(substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	ÿ	186	BAh		218	DAh	ł	250	FAh	
27	1Bh	ESC	(escape)	59	3Bh	;	91	5Bh	[	123	7Bh	{	155	9Bh	ø	187	BBh		219	DBh	ł	251	FBh	
28	1Ch	FS	(file separator)	60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch	£	188	BCh		220	DCh	ł	252	FCh	
29	1Dh	GS	(group separator)	61	3Dh	=	93	5Dh	]	125	7Dh	}	157	9Dh	Ø	189	BDh		221	DDh	ł	253	FDh	
30	1Eh	RS	(record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	x	190	BEh		222	DEh	ł	254	FEh	
31	1Fh	US	(unit separator)	63	3Fh	?	95	5Fh	_				159	9Fh	f	191	BFh		223	DFh	ł	255	FFh	

elCodigoASCII.com.ar

# UNICODE

Es un estándar de codificación de caracteres diseñado para facilitar el tratamiento informático, transmisión y visualización de textos de numerosos idiomas y disciplinas técnicas, además de textos clásicos de lenguas muertas. El término Unicode proviene de los tres objetivos perseguidos: universalidad, uniformidad y unicidad.

Unicode define cada carácter o símbolo mediante un nombre e identificador numérico, el code point ('punto de código'). Además incluye otras informaciones para el uso correcto de cada carácter, como sistema de escritura, categoría, direccionalidad, mayúsculas y otros atributos. Unicode trata los caracteres alfabéticos, ideográficos y símbolos de forma equivalente, lo que significa que se pueden mezclar en un mismo texto sin utilizar marcas o caracteres de control.

El Consorcio Unicode proporciona un único y extenso set de caracteres que pretende incluir todos los caracteres necesarios para cualquier sistema de escritura del mundo, incluyendo sistemas ancestrales (como el cuneiforme, el gótico y los jeroglíficos egipcios).

Los puntos de código de Unicode se identifican por un número entero. Según su arquitectura, un ordenador utilizará unidades de 8, 16 o 32 bits para representar dichos enteros. Las formas de codificación de Unicode reglamentan la forma en que los puntos de código se transformarán en unidades tratables por el computador.

# UTF

Unicode define tres formas de codificación bajo el nombre UTF (Unicode transformation format: formato de transformación Unicode):

- UTF-8: codificación orientada a byte con símbolos de longitud variable. Utiliza 1 byte para representar caracteres en el set ASCII7, dos bytes para caracteres en otros bloques alfabéticos y tres bytes para el resto del BMP. Para los caracteres complementarios se utilizan 4 bytes.
- UTF-16: codificación de 16 bits de longitud variable optimizada para la representación del plano básico multilingüe (BMP). Utiliza 2 bytes para cualquier carácter en el BMP y 4 bytes para los caracteres complementarios.
- UTF-32: codificación de 32 bits de longitud fija, y la más sencilla de las tres, emplea 4 bytes para todos los caracteres.

[!\[\]\(4729e517bc6a7cd81c8025b9646574fb\_img.jpg\) El código ASCII](#)

[!\[\]\(cbe80b694ebd74fcfe136a095b608235\_img.jpg\) Tabla de Caracteres UNICODE](#)

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001
あ	UTF-8	11100011 10000001 10000010
あ	UTF-16	00110000 01000010
あ	UTF-32	00000000 00000000 00110000 01000010

# Bibliografía & Licencia

- ◆ Textos tomados, corregidos y modificados de diferentes páginas de Internet, tutoriales y documentos.
- ◆ Este documento se encuentra bajo Licencia Creative Commons Attribution – NonCommercial - ShareAlike 4.0 International (CC BY-NC-SA 4.0), por la cual se permite su exhibición, distribución, copia y posibilita hacer obras derivadas a partir de la misma, siempre y cuando se cite la autoría del Prof. Matías E. García y sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.
- ◆ Autor:

*Matías E. García*

*Prof. & Tec. en Informática Aplicada*

[www.profmatiasgarcia.com.ar](http://www.profmatiasgarcia.com.ar)

[info@profmatiasgarcia.com.ar](mailto:info@profmatiasgarcia.com.ar)



[www.profmatiasgarcia.com.ar](http://www.profmatiasgarcia.com.ar)