

Clase 01 - Representaciones numéricas - Números Enteros

Profesor: **IIC2343 - Arquitectura de Computadores**
- Felipe Valenzuela González
Correo:
frvalenzuela@alumni.uc.cl

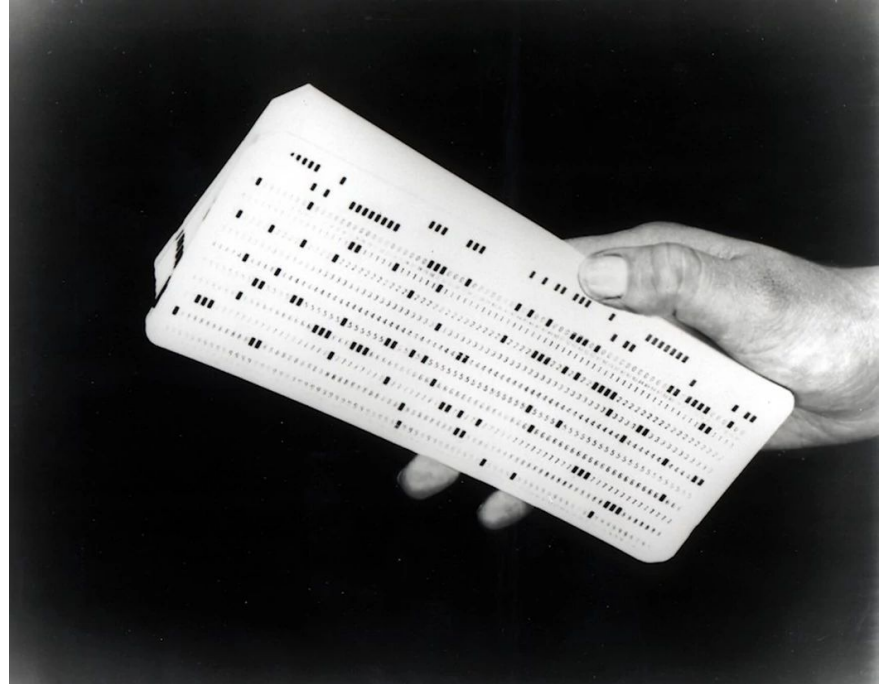
Resumen de la clase pasada

Introducción: ¿Qué es un computador?



Introducción: ¿Qué es un computador?

- En el contexto de este curso, nos centraremos en una definición concreta: **máquina programable que ejecuta programas.**



Introducción:

- Un computador lo definimos como una **máquina programable que ejecuta programas.**
- Para programar necesitamos:
 - Datos: números (enteros, reales) , texto, imágenes, etc
 - Operaciones: suma, resta, multiplicación, división, etc
 - Variables: simples, arreglos
 - Control de flujo: comparaciones, manejo de ciclos-
- La clase de hoy veremos con lo básico que sería los datos, específicamente cómo **representar datos en un computador!**

¿Dudas?

Objetivos de la clase

- Conocer distintas representaciones de números naturales y enteros
- Conocer los distintos esquemas de representación de datos basados en codificación numérica binaria
- Conocer cómo representar datos en un computador

**¿Cómo
representamos
números?**

Conteo

- Un forma simple
- Fácil de operar
- No escala bien



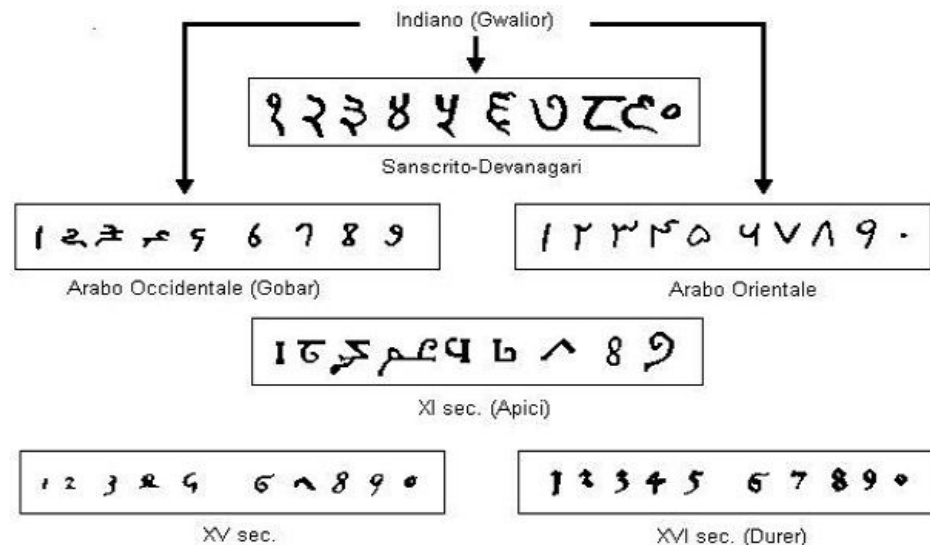
Números Romanos

- Un forma más sofisticada
- Escala bien
- Es difícil de operar



Sistema de Numeración indo-arábigo

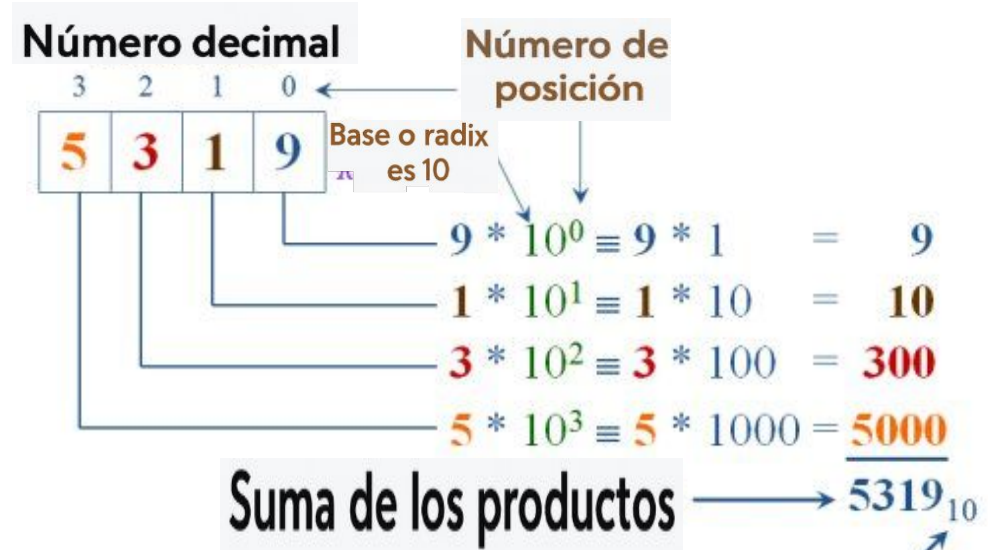
- Un forma aún más sofisticada
- Escala bien
- Es fácil de operar
- También llamado sistema posicional



Sistema posicional

Sistema posicional decimal

- Es el sistema de representación numérica que usamos
- Se llama posicional por la importancia del índice o posición del número
- Decimal viene de la **base** utilizada, que en este caso es diez



La base (radix) del sistema de numeración. Para base-10 no se muestra. Se muestra aquí como ejemplo.

Sistema posicional de base genérica

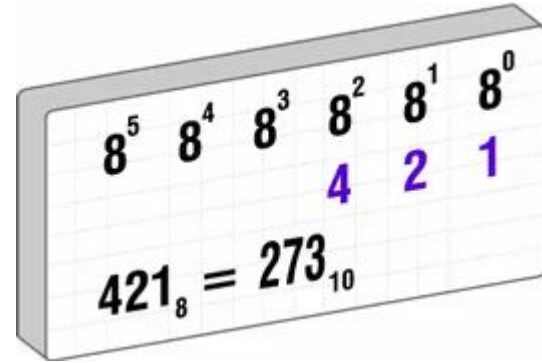
- s : símbolo
- k : = Posición del símbolo en la secuencia, siendo 0 la posición del extremo derecho.
- b : **base**
- n : cantidad de símbolos en la secuencia
- Notación típica: $(\text{---})_b$

$$\sum_{k=0}^{n-1} s_k \times b^k$$

Distintas bases numéricas

Base octal

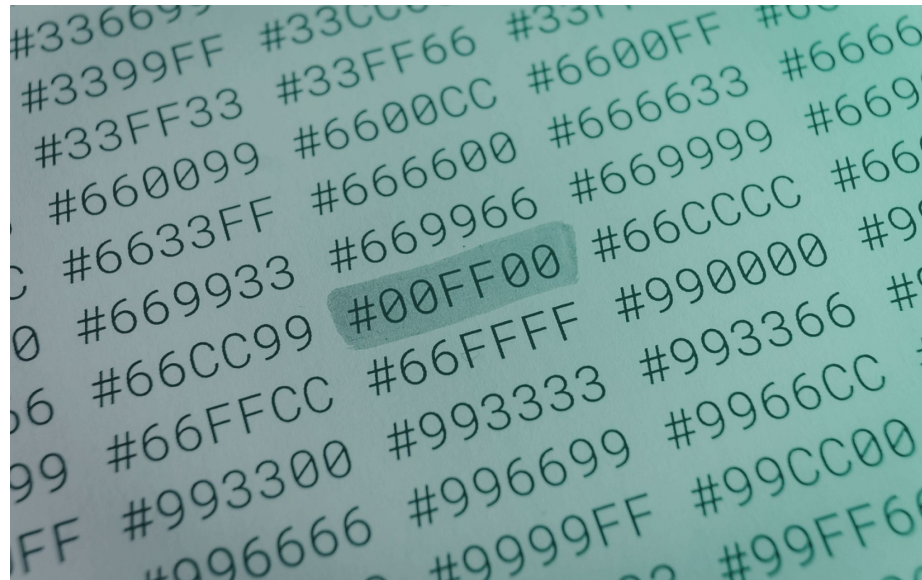
- También conocida como sistema posicional de base ocho
- Tiene un total de 8 símbolos
- Sus símbolos son
[0, 1, 2, 3, 4, 5, 6, 7]



Base hexadecimal

- También conocida como sistema posicional de base 16
- Tiene un total de 16 símbolos
- Sus símbolos son

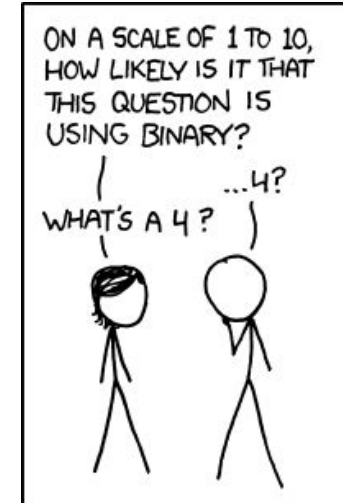
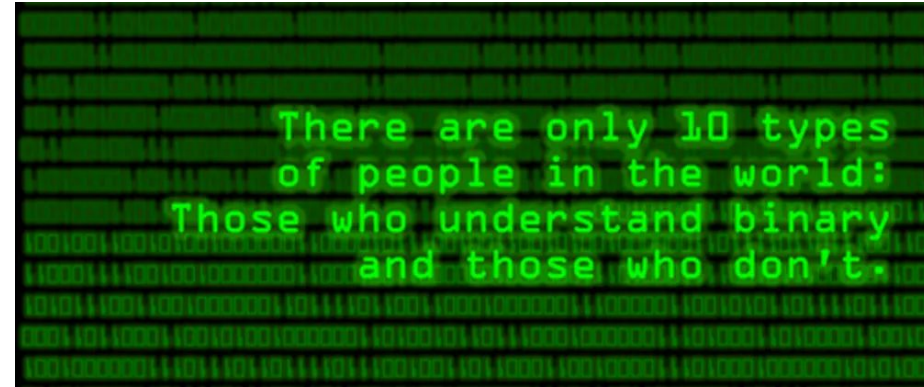
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F]



Base binaria

- También conocida como sistema posicional de base dos
- Tiene un total de 2 símbolos
- Sus símbolos son

[0, 1]



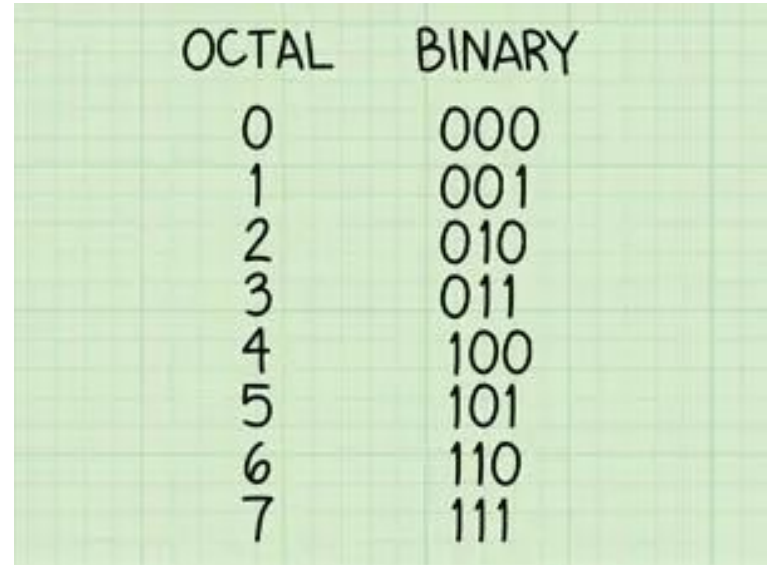
¿Dudas?

Método de conversión entre bases

- Existen diversos métodos de conversión entre bases
- Para este curso se requerirá como mínimo saber convertir desde una de las cuatro bases mencionadas hacia otra

Método de conversión: binaria hacia octal

- Ocuparemos un método aprovechando concatenación
- Agrupamos los términos numéricos para obtener el resultado

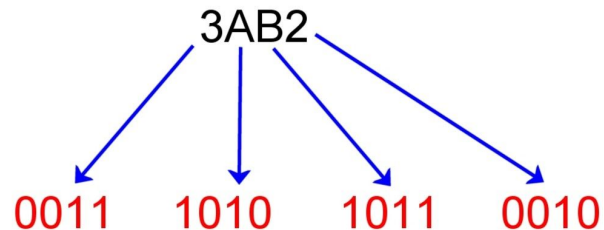


OCTAL	BINARY
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Método de conversión: binaria hacia hexa

- Ocuparemos un método aprovechando concatenación
- Agrupamos los términos numéricos para obtener el resultado

Converting Hex to Binary



$$3AB2_{16} = 11101010110010_2$$

Método de conversión: decimal hacia binario

- Se obtiene el resto entre el número en base decimal y el divisor 2.
- Se obtiene el resto entre el número en base decimal y el divisor 2.
- Para obtener el siguiente símbolo de la secuencia, realizar la misma operación con el resultado de la división entera del número

CÓMO CONVERTIR DECIMAL A BINARIO		
Método de divisiones sucesivas		
2	25	Restos
2	12	1
2	6	0
2	3	0
2	1	1
	0	1
Lea los restos de abajo hacia arriba		
11001		

¿Dudas?

Números negativos

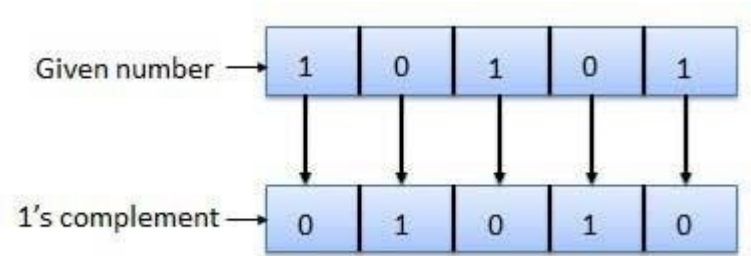
- Existen diversas formas de representar números negativos:
 - 1.- Agregar un símbolo (-)
 - 2.-Usar un símbolo

Números negativos en binario

- Existen diversas formas de representar números negativos:
 - 1.- Agregar un símbolo (-)
 - 2.- Usar un símbolo
 - 3.- Complemento 1**
 - 4.- Complemento 2**

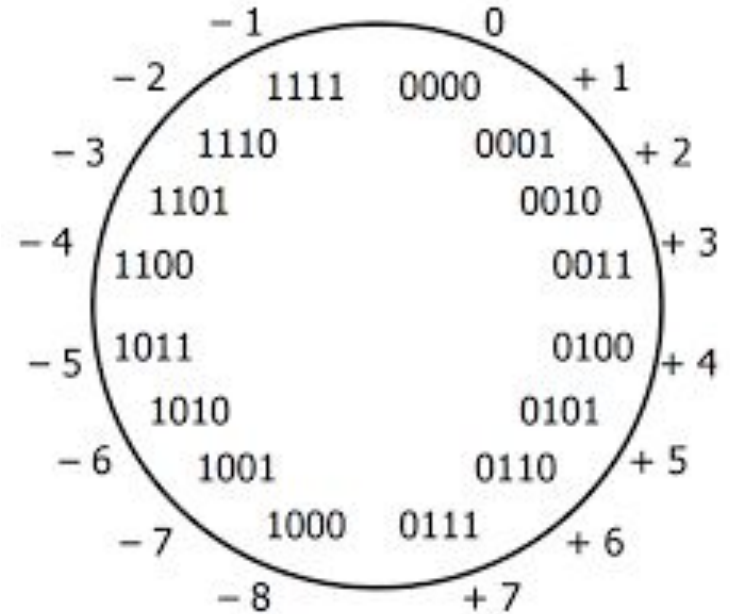
Complemento 1 (C1)

- Reemplazar todos los bits por su inverso
- El símbolo que representa cada uno de los valores sale natural
- Esta representación el cero queda asociadas a una concatenación de símbolos 1 (poco intuitivo)



Complemento 2 (C2)

- Sumar una unidad al complemento al C1
- Ahora el cero es intuitivo
- **Contra:** Tenemos una representación desbalanceada
- **Overflow:** Si una operación aritmética resulta en un valor no representable, nos dará un valor erróneo



Ejemplo de overflow



¿Dudas?

Máquina programable

- Si quisiéramos hacer una máquina programable tiene que ser:
 - Finita
 - Escalable
 - Simple
- Tomando todo esto en cuenta usar la escala binaria ayuda a soportar con la menor cantidad de estados.
- Consideraremos un solo símbolo en nuestra máquina un **Bit**
- Y llamaremos a la agrupación de Bits en en grupos de a 8, un **Byte**:

[0, 1, 0, 1, 1, 0, 1, 1]

Representación de datos

- Ahora con una forma de representar números podemos almacenar y **codificar** de distintas formas
- Un ejemplo es la interpretación que da la tabla **ASCII**

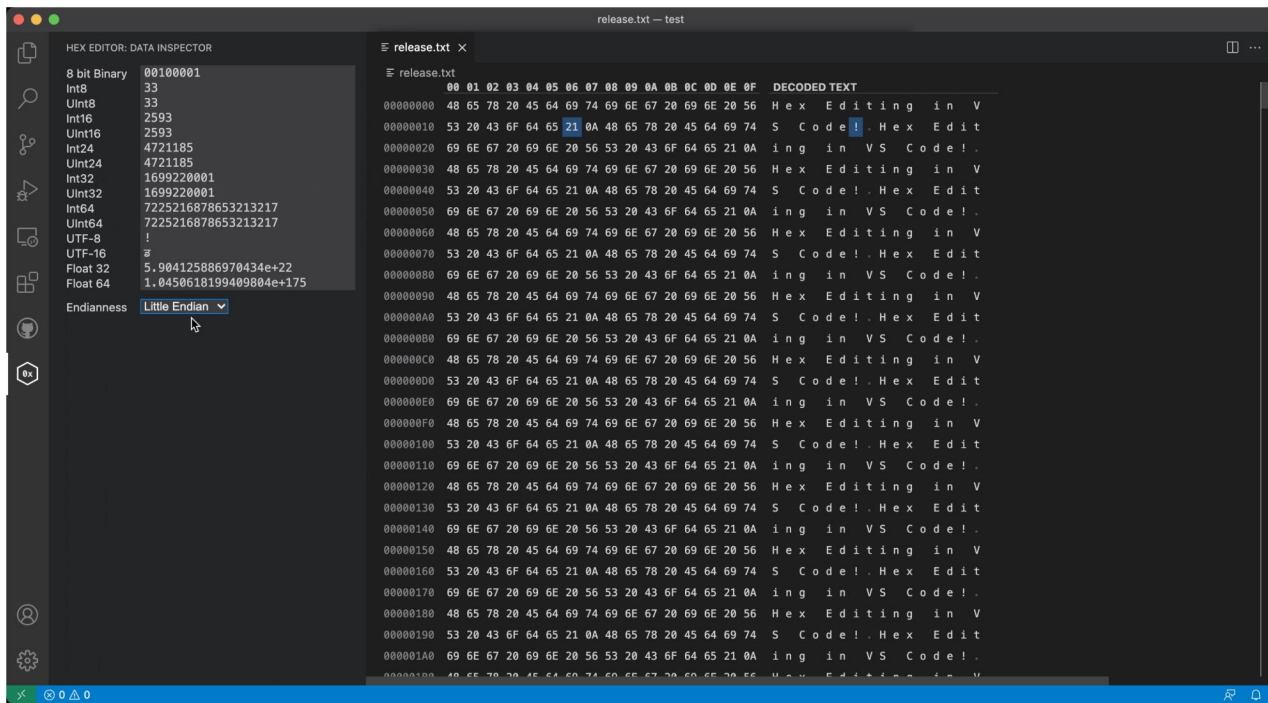
<div> <div> <div>b₇</div> <div>b₆</div> <div>b₅</div> <div>b₄</div> <div>b₃</div> <div>b₂</div> <div>b₁</div> <div>Bits</div> </div> <div> <div>Column</div> <div>Row</div> </div> </div>					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
					0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	DLE	SP	@	P	`	p
0	0	0	1	1	1	SOH	DC1	!	A	Q	a	q
0	0	1	0	2	2	STX	DC2	"	B	R	b	r
0	0	1	1	3	3	ETX	DC3	#	C	S	c	s
0	1	0	0	4	4	EOT	DC4	\$	D	T	d	t
0	1	0	1	5	5	ENQ	NAK	%	E	U	e	u
0	1	1	0	6	6	ACK	SYN	&	F	V	f	v
0	1	1	1	7	7	BEL	ETB	'	G	W	g	w
1	0	0	0	8	8	BS	CAN	(H	X	h	x
1	0	0	1	9	9	HT	EM)	I	Y	i	y
1	0	1	0	10	10	LF	SUB	*	J	Z	j	z
1	0	1	1	11	11	VT	ESC	+	K	[k	{
1	1	0	0	12	12	FF	FS	,	L	\	l	
1	1	0	1	13	13	CR	GS	—	M]	m	}
1	1	1	0	14	14	SO	RS	.	N	^	n	~
1	1	1	1	15	15	SI	US	/	O	_	o	DEL

Representación de datos

- Ahora con una forma de representar números podemos almacenar y **codificar** de distintas formas
- Otro ejemplo es el manejo de imágenes usando 3 Bytes

R	G	B	Hex Code	Color
0	0	0	#000000	Black
255	0	0	#FF0000	Red
0	255	0	#00FF00	Green
255	255	0	#FFFF00	Yellow
128	128	128	#808080	Grey
86	180	233	#56B4E9	Sky blue
230	159	0	#E69F00	Orange
255	255	255	#FFFFFF	White

Representación de datos



¿Dudas?

Ejercicio tipo prueba I1 - 2024-1

- Escriba la representación en decimal del número **0xCAFE**, si este se interpreta como binario en complemento de dos.



Clase 01 - Representaciones numéricas - Números Enteros

Profesor: **IIC2343 - Arquitectura de Computadores**
- Felipe Valenzuela González
Correo:
frvalenzuela@alumni.uc.cl