Clase 06 - Codificación

IIC1001 - Algoritmos y Sistemas Computacionales

Cristian Ruz – cruz@uc.cl

Jueves 26-Marzo-2026

Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Contenidos

Contacto

Temas

Complemento a Dos

ASCII

Otras Codificaciones

Cierre

Contenidos

Contacto

Temas

Complemento a Dos

ASCII

Otras Codificaciones

Cierre

Contacto

🐐

vicente.cabra@uc.cl

Vicente Cabra

Ayudante



🐸

alejandro.tapia@uc.cl

Alejandro Tapia

Ayudante



🥇

gabriel.sandoval@uc.cl

Gabriel Sandoval

Ayudante jefe



Contenidos

Contacto

Temas

Complemento a Dos

ASCII

Otras Codificaciones

Cierre

Temas del curso

Sistemas computacionales

- Representación datos, números y compresión
- Funcionamiento hardware, procesadores y memoria.
- Funcionamiento de sistemas operativos: ejemplo scheduling
- Funcionamiento de Internet
- Herramientas computacionales: github + latex

Algoritmos

- Algoritmos y resolución de problemas
- Eficiencia algorítmica
- Estructuras secuenciales y ordenamiento
- Grafos y árboles

Contenidos

Contacto

Temas

Complemento a Dos

ASCII

Otras Codificaciones

Cierre

¿Qué es el complemento a dos?

- Método para representar números negativos en binario
- Usado en la mayoría de los computadores
- Permite realizar operaciones aritméticas fácilmente

Pasos para obtener complemento a dos

Para un número binario:

1. Invertir los bits ($0 \rightarrow 1, 1 \rightarrow 0$)
2. Sumar 1 al resultado

Pasos para obtener complemento a dos

Para un número binario:

1. Invertir los bits ($0 \rightarrow 1$, $1 \rightarrow 0$)
2. Sumar 1 al resultado

Ejemplo: Convertir 5 a -5 en 8 bits

- 5 = 00000101
- Invertir: 11111010
- Sumar 1: 11111011

Ejemplo paso a paso

Convertir 10 a -10 en 8 bits:

$$10 = 00001010$$

$$\text{Invertir} = 11110101$$

$$+1 = 11110110$$

Ejemplo paso a paso

Convertir 10 a -10 en 8 bits:

$$10 = 00001010$$
$$\text{Invertir} = 11110101$$
$$+1 = 11110110$$

Resultado:

$$-10 = 11110110$$

Verificación

Sumamos 10 + (-10):

$$\begin{array}{r} 00001010 \\ +\ 11110110 \\ \hline 00000000 \end{array}$$

Verificación

Sumamos 10 + (-10):

$$\begin{array}{r} 00001010 \\ +\ 11110110 \\ \hline 00000000 \end{array}$$

Resultado: 0 (correcto)

Rango de valores

Para n bits:

$$-2^{n-1} \quad \text{a} \quad 2^{n-1} - 1$$

Ejemplo con 8 bits:

$$-128 \quad \text{a} \quad 127$$

Contenidos

Contacto

Temas

Complemento a Dos

ASCII

Otras Codificaciones

Cierre

¿Qué es ASCII?

- Código para representar caracteres
- Usa 7 bits (0–127)
- Cada número representa un símbolo

Ejemplos ASCII

Carácter	Decimal	Binario
A	65	01000001
B	66	01000010
a	97	01100001
0	48	00110000

Ejemplo de conversión

Convertir ’A’ a binario:

- ’A’ = 65
- 65 en binario = 01000001

Ejemplo de conversión

Convertir ’A’ a binario:

- ’A’ = 65
- 65 en binario = 01000001

Convertir 01100001:

- 01100001 = 97
- 97 = ’a’

Ejercicio

Decodificar:

$$01001000 \quad 01101001$$

Ejercicio

Decodificar:

$$01001000 \quad 01101001$$

Solución:

- 01001000 = 72 = H
- 01101001 = 105 = i

$$\Rightarrow \text{”Hi”}$$

Contenidos

Contacto

Temas

Complemento a Dos

ASCII

Otras Codificaciones

Cierre

Limitaciones de ASCII

- Solo 128 caracteres
- No incluye acentos ni símbolos especiales

UTF-8

- Codificación moderna
- Compatible con ASCII
- Usa 1 a 4 bytes
- Permite emojis y caracteres internacionales

- 'A' = 01000001 (igual que ASCII)
- 'ñ' = 11000011 10110001
- '€' = 11100010 10000010 10101100

Comparación

ASCII	UTF-8
7 bits	1-4 bytes
128 símbolos	Millones de símbolos
Inglés básico	Multilenguaje

¿Cómo UTF-8 determina la longitud?

UTF-8 usa los **bits iniciales** del primer byte para indicar cuántos bytes tiene el carácter:

Patrón	Número de bytes
0xxxxxxx	1 byte (ASCII)
110xxxxx	2 bytes
1110xxxx	3 bytes
11110xxx	4 bytes

¿Cómo UTF-8 determina la longitud?

UTF-8 usa los **bits iniciales** del primer byte para indicar cuántos bytes tiene el carácter:

Patrón	Número de bytes
0xxxxxxx	1 byte (ASCII)
110xxxxx	2 bytes
1110xxxx	3 bytes
11110xxx	4 bytes

Los bytes siguientes siempre comienzan con:

$$10xxxxxx$$

¿Cómo UTF-8 determina la longitud?

Ejemplo:

- 11000011 10110001
- Empieza con 110 $\Rightarrow$ carácter de 2 bytes
- Segundo byte empieza con 10 $\Rightarrow$ válido

$\Rightarrow$ 'ñ'

Estructura de UTF-8 (visual)

Los bits iniciales indican la longitud del carácter:

Patrón	Significado
0xxxxxxx	1 byte
110xxxxx	2 bytes
1110xxxx	3 bytes
11110xxx	4 bytes

Bytes de continuación:

10xxxxxx

¿Cómo UTF-8 determina la longitud?

Ejemplo visual:

$$11000011 \quad 10110001$$

¿Cómo UTF-8 determina la longitud?

Ejemplo visual:

$$11000011 \quad 10110001$$

- 110 $\Rightarrow$ indica 2 bytes
- 10 $\Rightarrow$ byte de continuación

$$\Rightarrow \text{'ñ'}$$

Contenidos

Contacto

Temas

Complemento a Dos

ASCII

Otras Codificaciones

Cierre

Resumen

- El complemento a dos permite representar negativos
- ASCII codifica caracteres básicos
- UTF-8 amplía la representación de símbolos

Actividad sugerida

- Convertir -25 a complemento a dos (8 bits)
- Codificar su nombre en ASCII
- Investigar cómo se representa un emoji en UTF-8