



# Valor de la informática en las organizaciones

.....

Carlos Augusto Arellano Muro

- ▶ Entretenimiento

- ▶ Entretenimiento
- ▶ Publicidad

# Impacto del computador

- ▶ Entretenimiento
- ▶ Publicidad
- ▶ Medicina

# Impacto del computador

- ▶ Entretenimiento
- ▶ Publicidad
- ▶ Medicina
- ▶ Líneas aéreas

- ▶ Entretenimiento: Programas y películas con efectos especiales o edición de vídeo. Músicos, en la composición o grabación de pistas.
- ▶ Publicidad
- ▶ Medicina
- ▶ Líneas aéreas
- ▶ Ciencia

# Impacto del computador

- ▶ Entretenimiento
- ▶ Publicidad: Uso de gráficos y animación para llamar la atención. Periodismo por internet. Edición de imágenes.
- ▶ Medicina
- ▶ Líneas aéreas
- ▶ Ciencia
- ▶ Fábrica automatizada

# Impacto del computador

- ▶ Entretenimiento
- ▶ Publicidad
- ▶ Medicina: Almacén de datos de los pacientes. Monitorización de pacientes
- ▶ Líneas aéreas
- ▶ Ciencia
- ▶ Fábrica automatizada



# Impacto del computador

- ▶ Entretenimiento
- ▶ Publicidad
- ▶ Medicina
- ▶ Líneas aéreas: Diseño y simulación de aeronaves. Simulación de vuelo. Control de tráfico. Sistemas de reservas de vuelos.
- ▶ Ciencia
- ▶ Fábrica automatizada

# Impacto del computador

- ▶ Entretenimiento
- ▶ Publicidad
- ▶ Medicina
- ▶ Líneas aéreas
- ▶ Ciencia: Simulación de fenómenos. Compartir información
- ▶ Fábrica automatizada

# Impacto del computador

- ▶ Entretenimiento
- ▶ Publicidad
- ▶ Medicina
- ▶ Líneas aéreas
- ▶ Ciencia
- ▶ Fábrica automatizada: Realizar inventarios. Seguridad. Máquinas controladas por computadora.

## Definiciones

- ▶ **Bit:** (Binary digit) Es la unidad más pequeña de datos que se puede almacenar en una computadora. Representa el estado de un dispositivo que puede tomar sólo dos estados.

## Definiciones

- ▶ **Bit:** (Binary digit) Es la unidad más pequeña de datos que se puede almacenar en una computadora. Representa el estado de un dispositivo que puede tomar sólo dos estados.
- ▶ **Patrón de bits:** Es una secuencia o cadena de bits para representar datos.

## Definiciones

- ▶ **Bit:** (Binary digit) Es la unidad más pequeña de datos que se puede almacenar en una computadora. Representa el estado de un dispositivo que puede tomar sólo dos estados.
- ▶ **Patrón de bits:** Es una secuencia o cadena de bits para representar datos.

Número de símbolos	Longitud del patrón de bits
2	1
4	2
8	3
16	4
128	7
65536	16

- ▶ Nibble: Cuatro bits

- ▶ Nibble: Cuatro bits
- ▶ Byte: Ocho bits



- ▶ Nibble: Cuatro bits
- ▶ Byte: Ocho bits
- ▶ kilobit: 1024 bits

- ▶ Nibble: Cuatro bits
- ▶ Byte: Ocho bits
- ▶ kilobit: 1024 bits
- ▶ Megabit: 1024 Kb

- ▶ Nibble: Cuatro bits
- ▶ Byte: Ocho bits
- ▶ kilobit: 1024 bits
- ▶ Megabit: 1024 Kb
- ▶ Gigabit: 1024 Mb

- ▶ Nibble: Cuatro bits
- ▶ Byte: Ocho bits
- ▶ kilobit: 1024 bits
- ▶ Megabit: 1024 Kb
- ▶ Gigabit: 1024 Mb
- ▶ Terabit: 1024 Gb

## Tipos de datos

- ▶ **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

## Tipos de datos

- ▶ **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

Se han diseñado diferentes secuencias de patrones de bits (código) para representar símbolos:

## Tipos de datos

- ▶ **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

Se han diseñado diferentes secuencias de patrones de bits (código) para representar símbolos:

- ▶ ASCII: (American Standard Code for Information Interchange)  
Creado por el Instituto Norteamericano de Estándares (ANSI).

## Tipos de datos

- ▶ **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

Se han diseñado diferentes secuencias de patrones de bits (código) para representar símbolos:

- ▶ **ASCII:** (American Standard Code for Information Interchange)  
Creado por el Instituto Norteamericano de Estándares (ANSI).  
Utiliza 7 bits. El primer patrón (0000000) representa el carácter nulo, el último representa el carácter de eliminación.



## Tipos de datos

- ▶ **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

Se han diseñado diferentes secuencias de patrones de bits (código) para representar símbolos:

- ▶ **ASCII:** (American Standard Code for Information Interchange)  
Creado por el Instituto Norteamericano de Estándares (ANSI).  
Utiliza 7 bits. El primer patrón (0000000) representa el carácter nulo, el último representa el carácter de eliminación.  
Hay 31 caracteres de control no imprimibles.

## Tipos de datos

- ▶ **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

Se han diseñado diferentes secuencias de patrones de bits (código) para representar símbolos:

- ▶ **ASCII:** (American Standard Code for Information Interchange)  
Creado por el Instituto Norteamericano de Estándares (ANSI).  
Utiliza 7 bits. El primer patrón (0000000) representa el carácter nulo, el último representa el carácter de eliminación.  
Hay 31 caracteres de control no imprimibles.  
Los caracteres numéricos se representan antes de las letras.

## Tipos de datos

- **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

Se han diseñado diferentes secuencias de patrones de bits (código) para representar símbolos:

- **ASCII:** (American Standard Code for Information Interchange)  
Creado por el Instituto Norteamericano de Estándares (ANSI).  
Utiliza 7 bits. El primer patrón (0000000) representa el carácter nulo, el último representa el carácter de eliminación.  
Hay 31 caracteres de control no imprimibles.  
Los caracteres numéricos se representan antes de las letras.  
Las letras mayúsculas se representan antes que las minúsculas y solo se diferencia un bit entre mayúsculas y minúsculas:  
A: 1000001  
a: 1100001

## Tipos de datos

- **Texto:** Secuencia de símbolos para representar una idea. Cada símbolo se representa con un patrón de bits.

Se han diseñado diferentes secuencias de patrones de bits (código) para representar símbolos:

- **ASCII:** (American Standard Code for Information Interchange)  
Creado por el Instituto Norteamericano de Estándares (ANSI).  
Utiliza 7 bits. El primer patrón (0000000) representa el carácter nulo, el último representa el carácter de eliminación.  
Hay 31 caracteres de control no imprimibles.  
Los caracteres numéricos se representan antes de las letras.  
Las letras mayúsculas se representan antes que las minúsculas y solo se diferencia un bit entre mayúsculas y minúsculas:  
A: 1000001  
a: 1100001  
Hay seis caracteres especiales entre las mayúsculas y minúsculas.

**Ejemplo** Codificar la palabra BYTE en ASCII:

**Ejemplo** Codificar la palabra BYTE en ASCII:

B: 1000010

**Ejemplo** Codificar la palabra BYTE en ASCII:

B: 1000010

Y: 1011001

**Ejemplo** Codificar la palabra BYTE en ASCII:

B: 1000010

Y: 1011001

T: 1010100



**Ejemplo** Codificar la palabra BYTE en ASCII:

B: 1000010

Y: 1011001

T: 1010100

E: 1000101

**Ejemplo** Codificar la palabra BYTE en ASCII:

B: 1000010

Y: 1011001

T: 1010100

E: 1000101

BYTE: 1000010 1011001 1010100 1000101

**Ejemplo** Codificar la palabra BYTE en ASCII:

B: 1000010

Y: 1011001

T: 1010100

E: 1000101

BYTE: 1000010 1011001 1010100 1000101

**Ejercicio:** Codificar la palabra **Palabra** en ASCII

## Tipos de datos

- ▶ **Texto:**

## Tipos de datos

### ► Texto:

- ASCII Extendido: Para que cada patrón sea de un byte, a los patrones de bits de ASCII se les aumenta un 0 a la izquierda, de forma que ahora va de 0000 0000 a 0111 1111.  
O en hexadecimal: 0x00 y 0x7F.

## Tipos de datos

### ► Texto:

- ASCII Extendido: Para que cada patrón sea de un byte, a los patrones de bits de ASCII se les aumenta un 0 a la izquierda, de forma que ahora va de 0000 0000 a 0111 1111.  
O en hexadecimal: 0x00 y 0x7F.

**Ejercicio:** Convertir la palabra anterior a ASCII extendido y expresarla en hexadecimal.

## Tipos de datos

### ► Texto:

- ASCII Extendido: Para que cada patrón sea de un byte, a los patrones de bits de ASCII se les aumenta un 0 a la izquierda, de forma que ahora va de 0000 0000 a 0111 1111.  
O en hexadecimal: 0x00 y 0x7F.

**Ejercicio:** Convertir la palabra anterior a ASCII extendido y expresarla en hexadecimal.

- EBCDIC: (Extended Binary Coded Decimal Interchange Code) Desarrollado por IBM. Usa patrones de 8 bits para representar 256 símbolos, sin embargo solo se usan en mainframes de IBM

## Tipos de datos

### ► Texto:

- ASCII Extendido: Para que cada patrón sea de un byte, a los patrones de bits de ASCII se les aumenta un 0 a la izquierda, de forma que ahora va de 0000 0000 a 0111 1111.  
O en hexadecimal: 0x00 y 0x7F.

**Ejercicio:** Convertir la palabra anterior a ASCII extendido y expresarla en hexadecimal.

- EBCDIC: (Extended Binary Coded Decimal Interchange Code) Desarrollado por IBM. Usa patrones de 8 bits para representar 256 símbolos, sin embargo solo se usan en mainframes de IBM

**Más ejercicio:** Codificar en EBCDIC la frase

**Hola version 2.0**

expresado en hexadecimal.



## Tipos de datos

### ► Texto:

- Unicode: Superconjunto de caracteres basado en ASCII de 2 bytes. Algunas partes del código se usa para símbolos gráficos.
- ISO: La Organización Internacional de Estándares diseñó un código de patrones que utiliza 32 bits. Lo suficiente para representar cualquier símbolo.

## Tipos de datos

- ▶ **Números:** Se representan usando el sistema binario y no código de texto como ASCII.

## Tipos de datos

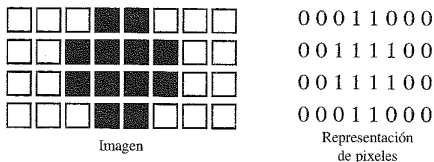
- ▶ **Números:** Se representan usando el sistema binario y no código de texto como ASCII.
- ▶ **Imagen:** Se representa con dos métodos

## Tipos de datos

- ▶ **Números:** Se representan usando el sistema binario y no código de texto como ASCII.
- ▶ **Imagen:** Se representa con dos métodos
  - ▶ Mapa de bits: La imagen se divide en una matriz de *pixeles* (picture elements), donde cada pixel es un punto. El tamaño del punto depende de la resolución. Cada pixel puede tener un patrón de bits.

## Tipos de datos

- ▶ **Números:** Se representan usando el sistema binario y no código de texto como ASCII.
- ▶ **Imagen:** Se representa con dos métodos
  - ▶ Mapa de bits: La imagen se divide en una matriz de *pixeles* (picture elements), donde cada pixel es un punto. El tamaño del punto depende de la resolución. Cada pixel puede tener un patrón de bits.



00011000 00111100 00111100 00011000

Representación lÍneal

## Tipos de datos

### ► Imagen:

#### ► Mapa de bits RGB:

Si la imagen es a color, ésta se compone de tres matrices:

- R: Cada patrón indica el nivel de rojo que hay en el pixel.
- G: Cada patrón indica el nivel de verde que hay en el pixel.
- B: Cada patrón indica el nivel de azul que hay en el pixel.

Ejemplo: pixel con 100 % de intensidad de rojo  $\underbrace{0xFF}_R \underbrace{0x00}_G \underbrace{0x00}_B$

Un pixel blanco sería  $\underbrace{0xFF}_R \underbrace{0xFF}_G \underbrace{0xFF}_B$

## Tipos de datos

- ▶ Imagen:

## Tipos de datos

### ► Imagen:

- Gráficos de vectores: Se compone de instrucciones. Líneas, curvas y sus propiedades (Como el grosor de la línea o el color). Esto hace que se requiera menos memoria para una mayor resolución de la imagen. Cada vez que una imagen se dibuja, las instrucciones se vuelven a evaluar.



## Tipos de datos

### ► Imagen:

- Gráficos de vectores: Se compone de instrucciones. Líneas, curvas y sus propiedades (Como el grosor de la línea o el color). Esto hace que se requiera menos memoria para una mayor resolución de la imagen. Cada vez que una imagen se dibuja, las instrucciones se vuelven a evaluar.

- **Audio:** Se convierte el audio a datos digitales y se usa patrones de bits.

## Tipos de datos

### ► Imagen:

- Gráficos de vectores: Se compone de instrucciones. Líneas, curvas y sus propiedades (Como el grosor de la línea o el color). Esto hace que se requiera menos memoria para una mayor resolución de la imagen. Cada vez que una imagen se dibuja, las instrucciones se vuelven a evaluar.

### ► Audio: Se convierte el audio a datos digitales y se usa patrones de bits.

- La señal se muestrea (Mide el valor de la señal a intervalos iguales)

## Tipos de datos

### ► Imagen:

- Gráficos de vectores: Se compone de instrucciones. Líneas, curvas y sus propiedades (Como el grosor de la línea o el color). Esto hace que se requiera menos memoria para una mayor resolución de la imagen. Cada vez que una imagen se dibuja, las instrucciones se vuelven a evaluar.

### ► Audio: Se convierte el audio a datos digitales y se usa patrones de bits.

- La señal se muestrea (Mide el valor de la señal a intervalos iguales)
- Las muestras se cuantifican (Se asigna un valor a cada muestra)

## Tipos de datos

### ► Imagen:

- Gráficos de vectores: Se compone de instrucciones. Líneas, curvas y sus propiedades (Como el grosor de la línea o el color). Esto hace que se requiera menos memoria para una mayor resolución de la imagen. Cada vez que una imagen se dibuja, las instrucciones se vuelven a evaluar.

### ► **Audio:** Se convierte el audio a datos digitales y se usa patrones de bits.

- La señal se muestrea (Mide el valor de la señal a intervalos iguales)
- Las muestras se cuantifican (Se asigna un valor a cada muestra)
- Los patrones cuantificados se cambian a binario

## Tipos de datos

### ► Imagen:

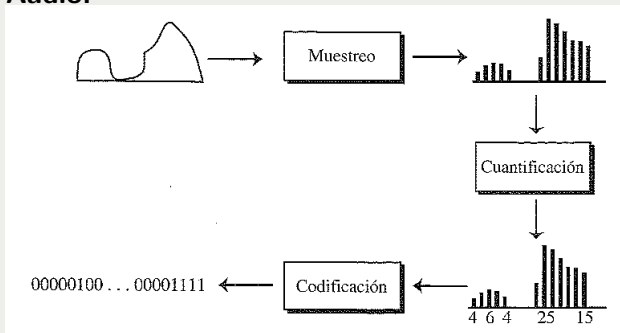
- Gráficos de vectores: Se compone de instrucciones. Líneas, curvas y sus propiedades (Como el grosor de la línea o el color). Esto hace que se requiera menos memoria para una mayor resolución de la imagen. Cada vez que una imagen se dibuja, las instrucciones se vuelven a evaluar.

### ► **Audio:** Se convierte el audio a datos digitales y se usa patrones de bits.

- La señal se muestrea (Mide el valor de la señal a intervalos iguales)
- Las muestras se cuantifican (Se asigna un valor a cada muestra)
- Los patrones cuantificados se cambian a binario
- Los patrones se almacenan.

## Tipo de datos

### ► Audio:



## Tipo de datos

- ▶ **Audio:**

## Tipo de datos

### ► **Audio:**

- MIDI: Análogo al gráfico de vectores, MIDI almacena instrucciones de tono, tiempo y timbre en el audio, así reduciendo el tamaño en memoria pero reconstruyendo la pieza cada vez que se abra.



## Tipo de datos

### ▶ **Audio:**

- ▶ MIDI: Análogo al gráfico de vectores, MIDI almacena instrucciones de tono, tiempo y timbre en el audio, así reduciendo el tamaño en memoria pero reconstruyendo la pieza cada vez que se abra.

### ▶ **Video:** Es una secuencia de imagenes (frames).

## Compresión de datos

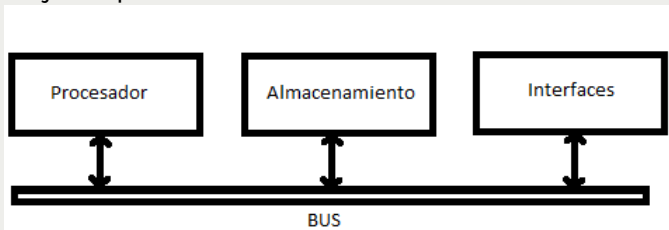
- ▶ Texto: Se crea un *diccionario* solo con las palabras que se usan en el texto y se les asigna un valor numérico. El archivo se compone de una secuencia de estos valores numéricos. Se reconstruye llamando a los caracteres que representa cada valor numérico.

## Compresión de datos

- ▶ Texto: Se crea un *diccionario* solo con las palabras que se usan en el texto y se les asigna un valor numérico. El archivo se compone de una secuencia de estos valores numéricos. Se reconstruye llamando a los caracteres que representa cada valor numérico.
- ▶ Un tipo de compresión de vídeo digital almacena los valores de los píxeles que cambian de un fotograma al siguiente; no hay necesidad de almacenar repetidamente el valor de los píxeles que son iguales en cada fotograma.

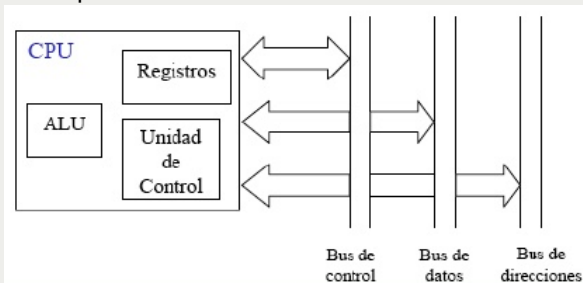
## Flujo de datos en un computador

- Flujo simple:



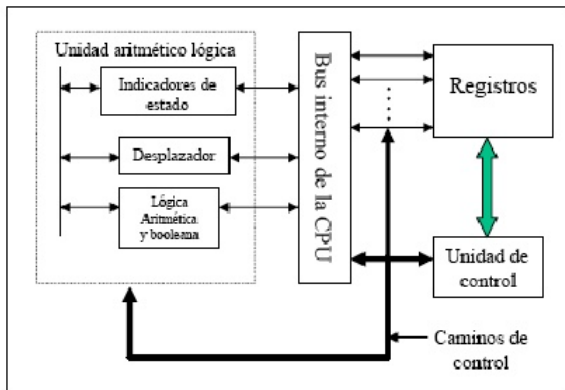
## Flujo de datos en un computador

- ▶ Flujo simple
- ▶ En el procesador: Dividimos el bus



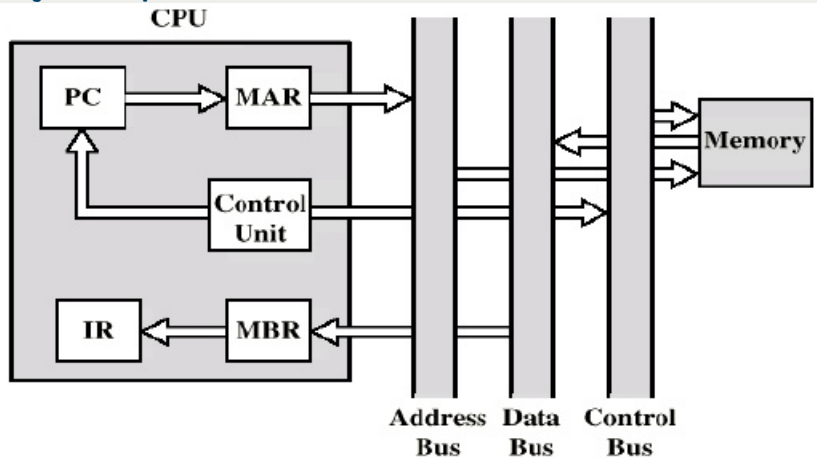
## Flujo de datos en un computador

- ▶ Flujo simple
- ▶ En el procesador: Dividimos el bus
- ▶ Interno:



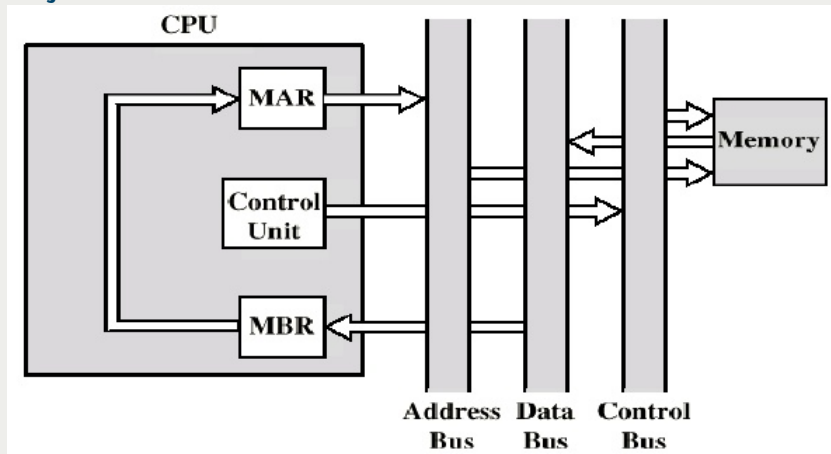
# Datos, información y procesamiento de la información

## Flujo de captación



MBR = Memory buffer register  
MAR = Memory address register  
IR = Instruction register  
PC = Program counter

## Flujo indirecto





## Flujo de interrup

