

Clase #03 de 27

Notaciones Algorítmicas, Eficiencia & Tipos

Abril 12, Jueves

Agenda para esta clase

- Resolución de Problemas
- Léxico & Notaciones Algorítmicas
- Trabajo #1 -- Resolución de Problemas — Adición
- Intervalo
- Modelo Computacional
- Estilos
- Eficiencia: Espacio & Tiempo
- Introducción a Tipos: En Matemática y en C++
- Trabajo #2

Resolución de Problemas

Problema, Solución y Procesos

- Dominios

- Dominio del Problema
 - “Qué”
 - Obstáculo a un objetivo
 - Análisis del Problemas
- Dominio de la Solución
 - “Como”
 - Plan para lograr el objetivo
 - Síntesis en un Diseño

- Proceso

- Soluciona Problema
 - Qué
- Establece Forma, método
 - Cómo
- Entrada de Datos
- Salida de Resultados
- Secuencia de Acciones o Pasos
 - Secuencia \neq Conjunto
 - $(1,2,3) \neq \{1,2,3\}$
 - $(1,2,3) \neq (1,3,2)$
 - $(1,2,3) \neq (1,2,2,3)$
 - $\{1,2,3\} = \{3,2,2,1\}$.

Léxico & Notaciones Algorítmicas

Léxico & Notaciones Algorítmicas

- Léxico
 - Define las entidades que manipula el algoritmo
 - Usa notación matemática, no usa un lenguaje de programación
 - El léxico es independiente de las notaciones algorítmicas
- Textual: Pseudocódigo
- Visual: Chapin / Lindsay / UTN FRBA
- Visual: Nassi-Shneiderman (NS)
 - Estandarizada
- Visual: Diagrama de Flujo
 - Estandarizada
- Textual: C ++
 - Estandarizada.

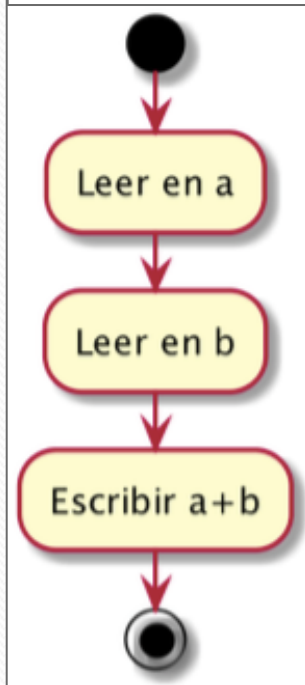
Secuencia:

Comparación de Notaciones Algorítmicas

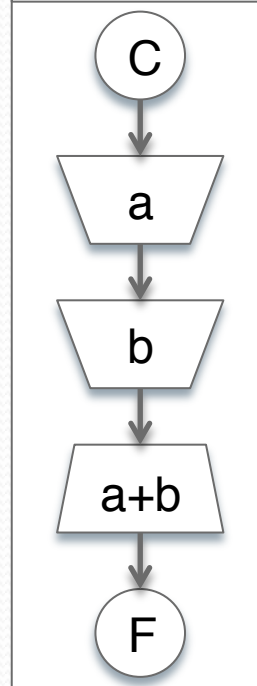
Textual: **Pseudocódigo**

Leer a.
Leer b.
Mostrar a+b.

Visual: Diagrama de Flujo



Visual: Chapin / Lindsay / UTN FRBA



Visual: **Nassi-Shneiderman**

Read a
Read b
Write a + b

Textual: **C++**

```
int main(){  
    int a, b;  
    std::cin >> a;  
    std::cin >> b;  
    std::cout << a + b;  
}
```

Trabajo #1

Resolución de Problemas — Adición

Trabajo #1 – Resolución de Problemas — Adición

- Carpeta: 01-Adición
 - Readme.md
 - Etapa #1: Análisis Problema
 - Transcripción del problema
 - Refinamiento e Hipótesis de trabajo
 - Modelo IPO
 - Etadap #2: Diseño de la Solución
 - Léxico
 - Represetnaciones del Algoritmo
 - Representación Textual y
 - Representación Visual
 - Adición.cpp

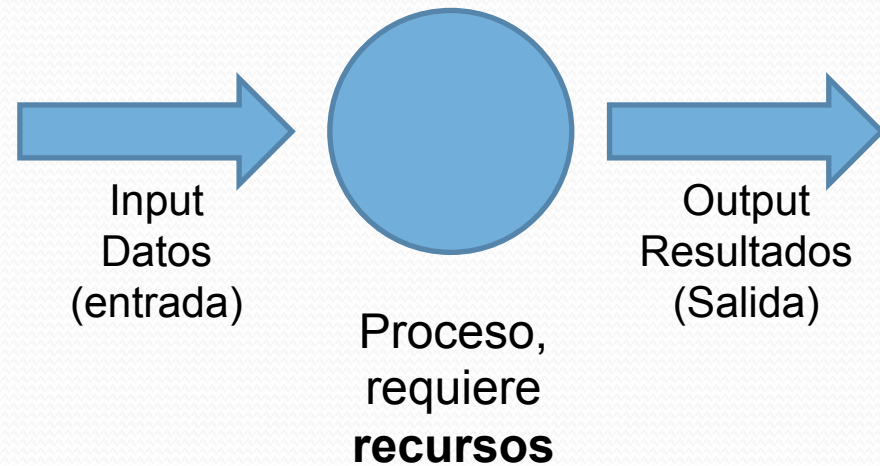
Intervalo

20 minutos

Modelo Computacional

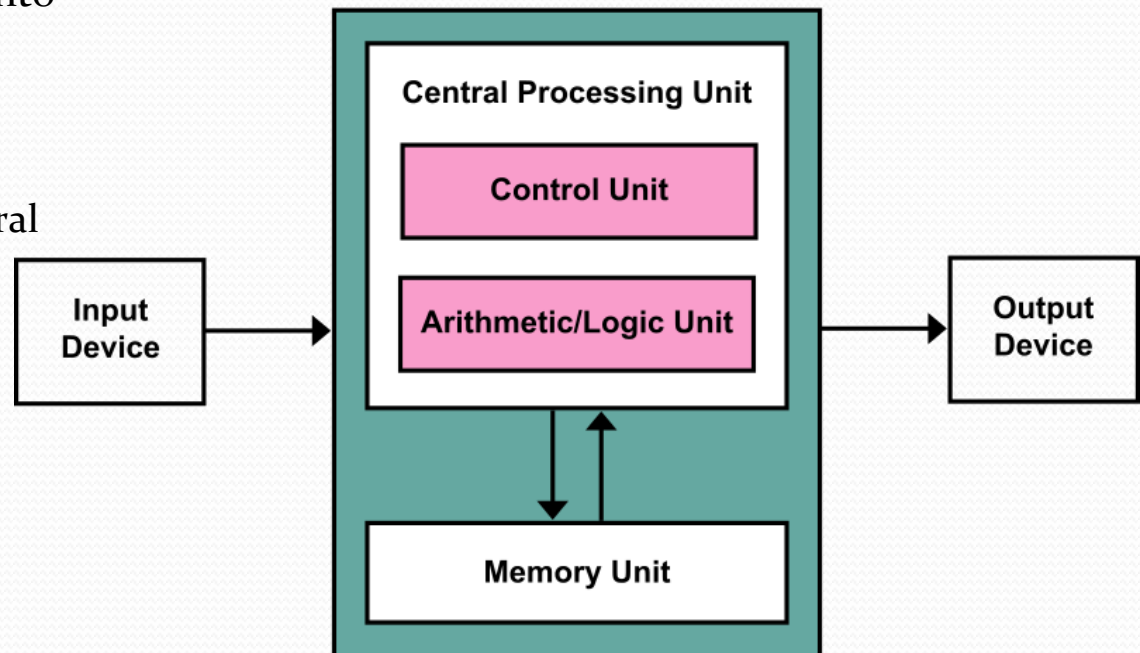
Modelo Computacional

- Modelo de procesamiento
 - Entrada
 - Proceso
 - Salida (y Errores)
 - Modelo IPO
- Arquitectura de Von Neumann (1945)
 - John Von Neumann (1903-1957)
 - Computadora con unidad central de procesamiento que almacena programas y datos en memoria
- Ejemplos
 - Hola.cpp
 - Adición.cpp



Componentes de la Arquitectura von Neumann

- Unidad Central de Procesamiento
 - Registros
 - IR: Registro de instrucción
 - PC: Contador de Programa
 - CA: Aritmético Lógica
 - CC: Unidad de Control Central
- M: Memoria (Principal)
 - Datos
 - Programas
 - Lenguaje Máquina
- I: Entrada
- O: Salida
 - Teclado
 - Pantalla
- R: Memoria Secundaria
- Pros y Cons de Memoria Principal y Secundaria
 - Costo
 - Eficiencia
 - Tamaño
 - Persistencia



Estilos

Comparación Subjetiva

- Legibilidad
- Claridad
- Mantenibilidad
- Los adjetivos "idad" son los atributos de calidad del desarrollo.

```
//1
int a;
cin >> a;
int b;
cin >> b;
cout << a+b;
```

```
//2
int a, b;
cin >> a >> b;
cout << a+b;
```

```
//3
int a;
int b;
int c;
cin >> a;
cin >> b;
c = a+b;
cout << c;
```

```
//4
int a;
cin >> a;
int b;
cin >> b;
int c = a+b;
cout << c;
```

```
//5
int a;
cin >> a;
int b;
cin >> b;
b = a+b;
cout << b;
```

Eficiencia: Espacio & Tiempo

Recursos Utilizados – Comparación Objetiva

		Recurso	
		Espacio	Tiempo
Durante	La traducción del Programa	de memoria secundaria que ocupa el programa fuente; en general, medido en líneas de código (lines of code, LOC).	requerido por el compilador para la traducción del programa en lenguaje de alto nivel a lenguaje máquina.
	La Ejecución del Proceso	de memoria principal utilizado durante la ejecución del proceso de memoria secundaria utilizada como área de trabajo.	requerido para finalizar la ejecución del proceso.

Comparación Objetiva

- Tiempo

- Las declaraciones no insumen tiempo de ejecución, salvo cuando tienen inicialización de una *variable automática*
- Cantidad de operaciones (aproximación)
 - `>>`
 - `<<`
 - `+`
 - `=`

- **Notación O Grande**

- Eje x: cantidad de datos, n
- Eje y: tiempo de procesamiento, t
- Funciones
 - $O(n)$: Lineal
 - $O(1)$: Constante

- Espacio

- Cantidad de variables

```
//1
int a;
int b;
cin >> a;
cin >> b;
cout << a+b;
```

```
//2
int a, b;
cin >> a >> b;
cout << a+b;
```

```
//3
int a;
int b;
int c;
cin >> a;
cin >> b;
c = a+b;
cout << c;
```

```
//4
int a;
int b;
cin >> a;
cin >> b;
int c = a+b;
cout << c;
```

```
//5
int a;
int b;
cin >> a;
cin >> b;
b = a+b;
cout << b;
```

Introducción a Tipos

En Matemática y en C++

¿Qué es un Tipo de Dato?

- Buscar una definición
- Buscar ejemplos
- Representación
- Valores que almacena
- Operaciones que se pueden realizar
- Definición
 - Un *tipo de dato*, o simplemente tipo, es un *conjunto de valores* y un *conjunto de operaciones* sobre esos valores.

Tipos de Datos – Conjunto de Valores

Niveles de abstracción y Ejemplos

Matemática		C++		Microprocesador (general)
Con-junto	Valores	Tipo de Dato	Valores	Representación electrónica
\mathbb{B}	V F	bool	true false	Registro: Bit (0 ó 1)
Σ	A B Ñ ! ... (finito) \approx ASCII	char	'a' 'A' '1' '+' '@' '\' ...	Registro: Byte (8 bytes)
N	Semirecta	unsigned	0 1 2 96 ...	Registro: Word (n bytes)
Z	Recta	int	1 -2 1024 ...	Registro: Word (n bytes)
R	Recta	double	1.5 1.0 1e5 ...	Registro: Floating point 80-bits
Σ^*	A ABA AÑABA ...	string	"a" "aA" "abab" "" ...	Bloque contiguo de memoria

Tipos de Datos – Conjunto de Operaciones

Niveles de abstracción y Ejemplos

Matemática		C++	
Tipo	(Algunas) Operaciones	Tipo	(Algunas) Operaciones
\mathbb{B}	$= \neq \vee \wedge \neg$	bool	== != or and not
Σ	$= \neq$	char	== != <= >= + - * / - %
N, Z	$= \neq \leq \geq + - \cdot / - \text{resto}$	unsigned int	
R	$= \neq \leq \geq + - \cdot / -$	double	== != <= >= + - * / -
Σ^*	$= \neq \cdot $	string	== != <= >= + length

Tipos de Datos Más Abstractos

- Ejemplos Comunes
 - String
 - Date
 - Money o Currency
 - Números Complejos
 - Compuestos
- ¿Qué característica comparten?
- No tienen representación directa en máquina
 - Un int puede ser almacenado en un registros del microprocesador, un string no.
- No son estándar en los lenguajes de programación

Trabajo #2

Ejemplos de Valores y Operaciones de Tipos de Datos

Trabajo #2 – Ejemplos de Valores y Operaciones de Tipos de Datos

- Carpeta: 02-EjemploTipos
 - EjemploTipos.cpp

Términos de la clase #03

Definir cada término con la bibliografía

- Resolución de Problemas
 - Dominio del Problema
 - Análisis
 - Dominio de la Solución
 - Síntesis en un Diseño
 - Definición de Proceso
 - Secuencia vs. Conjunto
- Léxico & Notaciones Algorítmicas
 - Léxico
 - Pseudocódigo
 - Lindsay / Chapin/ UTN FRBA
 - Nassi-Shneiderman (NS)
 - Diagrama de Flujo
 - C++
- Modelo Computacional
 - Arquitectura de John von Neumann
 - Recursos
 - Memoria para datos y programas
 - Registros
 - Memoria principal
 - Memoria secundaria
- Estilos
 - Legibilidad
 - Claridad
 - Mantenibilidad
- Atributos de calidad
- Eficiencia: Espacio & Tiempo
 - Tiempo
 - Espacio
 - Durante la traducción del programa
 - Durante la ejecución del proceso
 - Notación O Grande
 - $O(n)$
 - $O(1)$
- Introducción a Tipos
 - Definición
 - \mathbb{B}
 - $\Sigma = \text{ASCII}$
 - N
 - Z
 - R
 - Σ^*
 - bool
 - char
 - unsigned
 - int
 - double
 - string
 - Representación literal o constante de valores
 - Bit
- byte
- word
- Floating-point
- Bloque de memoria
- Operaciones
- Resto
- Módulo
- Norma o Longitud de una cadena
- Operadores binarios
- Operadores unarios
- Tipo de Datos Más Abstractos

Tareas para la próxima clase

1. Trabajo #1 — Resolución de Problemas — Adición
2. Trabajo #2 — Ejemplos de Valores y Operaciones de Tipos de Datos

¿Consultas?



Fin de la clase