

Introducción a la Programación

Algoritmo: Conjunto ordenado y finito de operaciones que permiten hallar la solución de un problema

Características:

Preciso, Definido y finito.

Conceptos:

Algoritmo, Pseudocódigo / Diagrama, Programa fuente, Lenguaje de programación, Código Máquina y Compilador

Pseudocódigo: Código que escriben las personas para imaginarse las órdenes que le darán a la computadora a través de un lenguaje de programación.

Para diseñar algoritmos hay que:

- * Entender lo que te pide
- * Visualizar cuáles son los datos de entrada y salida
- * Poner en orden los pasos
- * Ser específico en los pasos
- * Pensar en los inconvenientes
- * Tratar de resolverlo en el menor número de pasos
- * Probarlo varias veces analizando los resultados.

El sistema de computo se compone en:

Hardware

- Se puede tocar, tendrá un nombre y forma parte del equipo de computo

Software

Controla el hardware

Se divide en Software de aplicación y Sistema

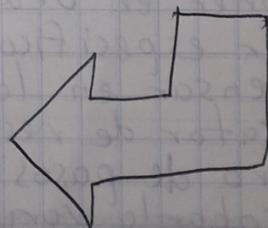
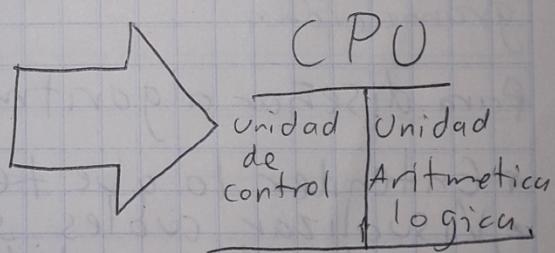
Organización física de la computadora

Dispositivos de entrada

Teclado, ratón, etc.

Dispositivo de salida

Monitor, impresora etc.



• ¿Qué es programación?
R= Es programar

• ¿Qué es programar?
R= elaborar programas

• ¿Qué es un programa?
R= Operaciones, que en un orden determinado, ejecutan ciertas máquinas.

Procesamiento de información en la computadora

Datos de entrada → Proceso → Información de salida

Si es entrada es dato y si es salida es información

Los algoritmos se expresan en pseudocódigo y diagramas de flujo

Pseudocódigo

- No lo puede ejecutar la computadora
- Concentrarse en la lógica
- Fácil de traducir a lenguaje de programación.

Ejemplo: este espacio se llama:
Indentación

- 1 Proceso Saludo
- 2 Definir nombre como carácter
- 3 Escribir '¿Cómo te llamas?' ;
- 4 Leer nombre
- 5 Escribir 'Hola ', nombre;
- 6 FinProceso

La sintaxis de los lenguajes de programación es más sencilla que aprender otro idioma como inglés, francés, etc.

Al final de cada instrucción pon punto y coma (;) en los lenguajes de programación formales por que es obligatorio

En pseudocódigo:

Letras azules: Palabras reservadas con significado especial en pseudocódigo.

Letras rojas: Texto tal cual, aparece cuando pones comillas.

Letras negras: identificadores

Diagramas de flujo

- Representación gráfica del algoritmo
- Utilizan simblos con significados bien definidos.
- Fácil de traducir a lenguaje de programación formal

Simbolo	Nombre	función
	Inicio/ Final	Inicio y final de un proceso
	Línea de Flujo	Orden de ejecución
	Entrada/ Salida	Lectura de datos E/S
	Proceso	Cualquier tipo de operación
	Decisión	Analizar situación

Lenguaje de programación

Conjunto de símbolos y reglas sintácticas que permiten escribir un programa.

- Define sintaxis
- Traduce el programa a un código binario (código objeto)

El pseudocódigo o diagramas de flujo no son comprensibles por la computadora se deben codificar

El algoritmo escrito en un lenguaje de programación se denomina código fuente.

Cada lenguaje tiene:

- * Instrucciones en entrada/salida
- * Instrucciones de cálculo
- * Instrucciones de control

Lenguajes de bajo nivel

Van muy relacionados con la máquina y el sistema operativo

Instucción

Conjunto de datos insertados en una secuencia específica que el procesador interpreta y ejecuta.

Dichas instrucciones están formadas de:

- * Palabras reservadas de cada lenguaje
- * Datos
- * Operaciones
- * Expresiones

Datos

- * Expresión general que describe los objetos con los que opera la computadora.
- * Las computadoras trabajan con varios tipos de datos.
- * En las computadoras, los datos deben ser de un tipo específico.

Tipos de datos

Determinan como se representa en la computadora

Dato	Representación	Operaciones
5	Entero	Operaciones Aritméticas
"5"	Cadena de Caracter	Extraer longitud

- Datos básicos : incluidos dentro del lenguaje
- Compuestos: Construidos a partir de los básicos.
ej: Estructuras, vectores, matrices/ tablas, cadenas.
- En C o C++: punteros

Datos numéricos

- Bool = para true o false
 - Double = para decimal
 - Char
 - Signed char } para caracteres.
 - Unsigned char }
 - short int
 - unsigned short int }
 - Int } para números enteros
 - Unsigned int
 - long int
 - unsigned long int }
 - long long int }
 - unsigned long long int }
 - Float = para decimal
- En javascript para poner un número ponemos .number

Bit \rightarrow 0, 1

Byte \rightarrow 8 bits

Cuando usas un dato numérico para ingresar números asegúrate de cuantos bits es, porque si introduces un número pequeño a un dato numérico de muchos bits, la memoria Ram se va a gastar o llenar mucho.

Datos numéricos:

Entero:

- Se pueden representar con 8, 16, 32 y 64 bits

- Subconjunto finito de números enteros

Reales:

- Cualquier número, entero, con decimal y negativo y positivo

- Para números grandes ponen notaciones científicas.

3650000000...0000... $\rightarrow 3.675201 \times 10^{25}$

Dato cadena (string)

- Sucesión de caracteres que se encuentran delimitados por comillas simples o dobles.

- La longitud es el número de caracteres de la cadena.

- 'Hola mundo'
- "Jack Sparrow"
- '(228) 8-40-35-9t'

A Javascript no le importa si las comillas simples o dobles para poner datos cadena.

Variables y constantes

Constante: Dato que permanece sin cambios durante el desarrollo del algoritmo.

ej: 7, 3.5, "hola", 'm'

Variable: Que puede cambiar

ej: a, x2, res, total, sueldo

Características: tienen un tipo y sólo pueden tomar valores de tipo

El nombre que se le otorga a una variable o constante se llama identificación.

Reglas

- Debe empezar con letra
- No deben existir espacios en blanco intermedios.
- La longitud depende del lenguaje
- NO palabras reservadas (if, while, int)

Recomendaciones

- Los nombres deben ser significativos y tener relación con su contenido

$$a = (b + h)/2$$

$$\text{pato} = [\text{algo1} \text{ "mesa"}]/2$$

- Usar un estandar de nomenclatura (camel case)

area Cuadrado
sum Valores

Para
for

Se utilizan cuando se sabe el numero de veces que se debe repetir

Utiliza una variable numérica para saber cuantas iteraciones deben realizarse

Pseudocódigo:

Para < inicialización > hasta < condición >
con paso < paso > hacer
< acción(es) >
Fin Para

C++

for (< inicialización >; < condición >; < paso >) {
< acción(es) >
}

~~do while~~: es el hacer mientras de Print

Se escribe así:

do {

$$(x += y) = (x = x + y)$$

} while ();

Char: guarda sólo 1 carácter

String: guarda palabras, pero lo muesta hasta que haya un espacio por como trabaja el `Cin >>`, para eso usa `getline`.

Para leer muchas palabras que estén separadas por espacio se usa

`getline(Cin, x)` PON LA DIRECTIVA
#Include <string>

La comilla simple es para caracteres (' ')

Las comillas dobles son para cadenas (" ")

el Char puedes poner: char(64)

↗

código ascii

- * Cuando se usa se dice que se llama o invoca
- * Se puede llamar todas las veces que se deseé
- * Una función puede llamar a otras funciones y puede llamarse a si misma.
- * Puede tomar cero o muchos valores llamados parámetros.
- * Los tipos de los parámetros y resultado siempre debe coincidir.
- * Las variables declaradas dentro del cuerpo se denominan variables locales.
- * De forma predeterminada, los argumentos se pasan a la función por valor (copia). También se pueden pasar por referencia.
- * Cuando una función modifica un argumento que se pasa por referencia, modifica el objeto original.

La computadora puede representar n dimensiones, pero para los seres humanos es muy difícil representar más de 3.

Todos los datos empiezan en 0.

Vector

1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7

↓
índice

elemento.

Operaciones comunes con vectores

Asignación

Lectura y escritura

Recorridos

Búsquedas

Ordenamiento.

Asignación:

int edades [5];

Cuántas edades vas a guardar
No puedes poner números negativos.
porque si lo haces afecta la computadora.

edades [0] = 15;

posición en la que vas a guardar
el dato (empezarán desde 0)

↑ dato que vas a guardar

Métodos: cosas que se pueden aplicar a clases para que hagan cosas específicas.

str.size() // esto dice el tamaño de
 ↑ ↑
 variable método

Los métodos solo se pueden poner en objetos

El objeto es lo más abstracto dentro de la programación

¿Cómo convertir strings a número?

int entero = stoi(x);

int numero = stoi(x)

¿Cómo convertir int, float, double a string?

to_string(x);

while (true) {

} Esto siempre se va a ejecutar

while (false) {

} Esto nunca se va a ejecutar.

Palabra == (palabra Volteada)? return true;
return false;

Otra forma de poner esto:

if (palabra == palabra Volteada) {

 return true;

}

else {

 return false;

}

Matrices

- Vector de vectores
- Todos los elementos son del mismo tipo
- Necesita dos indices para identificar cada elemento del array
- El primer indice se refiere a la fila y el segundo a la columna
- Tipo dato matriz [fila][columna];

• ¿Cómo inicializar una matriz?

int matriz[3][3]
 ↑ ↑
 filas columnas

int matriz[3][3] = { {1,2,3}, // fila 1
 {4,5,6}, // fila 2
 {7,8,9} }; // fila 3

Función trim: elimina los espacios
al inicio y al final de una palabra

trim(" chistian ") da:
"chistian"

cin.ignore()

esto ignora lo que hay pendiente en el cin.

cin>> "tri no mio";
cin.ignore

Si hay esto:

cin>> X;
getline(cin, Y);

puede haber bugs, así que ponlo así

cin>> X;
cin.ignore();
getline(cin, Y);

Objeto

Cualquier cosa que pueda ser representada en el mundo real, como silla, coche, etc.

La clase objeto está omisa de todo

Si tienes un servidor con interfaces gráficas o le conectas una pantalla, consume recursos, por eso los servidores no suelen estar conectados a monitores y no tienen interfaces gráficas

Todos los objetos tienen 2 cosas:
Características y comportamiento.

Estado / los atributos

Un objeto es una unidad de código compuesto de variables y métodos relacionales.

Clase

Automobil

Marco: String

Modelo: string

Color: string

$\nabla \phi_h : \mathbb{R}^d$

3

Elmolde

Objeto

Marca = ford

Modelo: xix

Color: René

Vph: 500 Km/h

La clase [↑] convalecidos

```
class Persona {  
    int altura;  
    string nombre;  
    string edad;
```

La clase

```
class Persona {
```

Private:

```
    int edad;
    string nombre;
    char sexo;
    string dirección;
```

Atributos

Public:

```
void saludar () {
```

```
cout << "Hola a to2" << endl;
```

}

} Método

};

Todos los atributos y métodos en C++ por defecto son privados

Acceder a elementos

```
int main () {
```

```
    Persona P1;
```

```
    P1. saludar
```

Si creas uno o más constructores, que no se te olvide crear el constructor por defecto.

Ej C) {

§

el por defecto es el que no tiene parámetros

Getter: Para cuando un atributo es privado, no se puede acceder desde afuera de la clase.

Es un método, no recibe nada, devuelve la cosa del mismo nombre.

private:

int float;

public: tienen que escribirle get

int getfloat() {
 return float;
}

• Modificar atributos (sets)

Todos los métodos sets son void y si reciben parámetros

```
class gato {  
    private:  
        int chueldo;  
  
    public:  
        void setChueldo(int a) {  
            chueldo = a;  
        }  
}
```

private:

→ float sueldo por hora;

public:

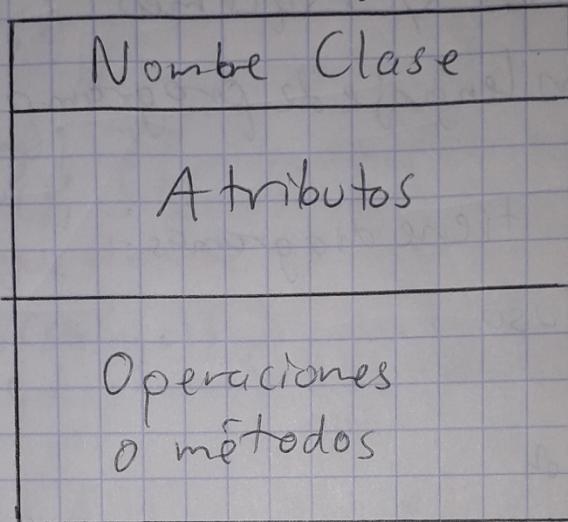
```
void setSueldoPorHora(float sueldoPorHora) {  
    this->sueldoPorHora = sueldoPorHora;  
}
```

↑
3

agregale this para que esto aga me esto,

Elementos diagrama de clase

En uml es representada por un rectángulo dividido en 3 partes



Atributos! variables

+ = públicos

- = privados

= protegidos

Métodos! acciones

Las cosas de la documentación se le llaman artefactos, como los diagramas de clase, casos de uso, etc.

Se usa artefactos porque decir "documentación" da hueva

Simbología

Asociación: relación de una clase y otra. NO ES UNA RELACIÓN FUERTE (o sea es una relación débil)

Relación fuerte: si se rompe algo, lo que esté relacionado con esto se rompe también

Relación débil: si se rompe algo; lo que esté relacionado no afecta, queda intacto.

Cardinalidad: indica el grado y nivel de dependencia de clases

* = Cero, uno o más
0,1 = Cero o uno

1^{*} = Uno o más

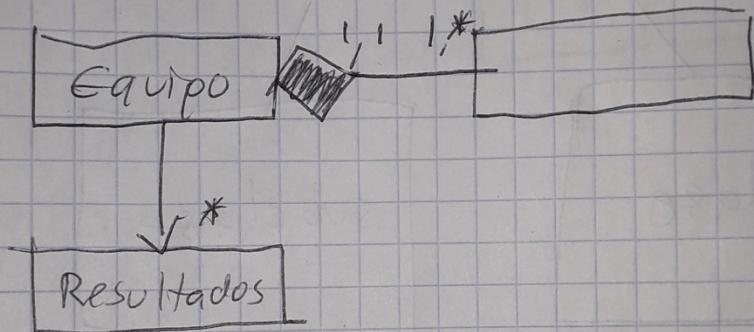
1 = Solo 1

1..5 = está entre 1 y 5

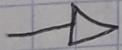
Por referencia  →

Si algo estare relacionado con el, si se rompe,
a lo que este relacionado no afecte.

Asociacion y cardinalidad



Herencia (especialización/generalización)



Indica que una clase hereda los métodos
y atributos de una clase

Cuando pones variables en c++

string x;

en uml:

x: string

Cuando creo código con el uml, quítale los voids a los constructores

En strings siquieres poner muchas cosas:

"x" + x + "y" + y + "z" + z;
}

Se usan los +

Cuando hay esto:

if(x)

Suele ser un boli, por que evalua si es true