

Prof. Blanca Tallaj

ENGI 2310, sec. 07, 04

Texto: Tony Gaddis, 6th edition
Starting Out with C++ From
Control Structures through Objects.

Deitel & Deitel C++: How To Program, 4th edic.

www.cplusplus.com

↳ C++ Tutorial

Oficina: Dpto. Ing. Eléctrica

Hrs. de Oficina: Lu y Mi 10:00 - 11:15 am
Ma y Ju 2:00 - 2:30 pm
~~Ma y Ju 10:00 - 11:15 am~~
bltallaj@pupr.edu

Evaluación:

3 exámenes	20% c/u = 60%
3 proyectos (grupales)	40%
	<hr/> 100%

60 ó + de prom. en exámenes

Introducción

¿Qué es un computador?

Es una máquina o equipo que permite manipular información a velocidades extraordinariamente mayores que las de un ser humano.

2 tipos de problemas

Problemas con
solución algorítmica

Su solución se puede definir como una secuencia de pasos que, si se sigue, se consigue siempre la solución.

Ej: balancear una chequera

↓
Si manipulan datos, se pueden programar

Problemas con
solución heurística.

Su solución no se puede definir como una sec. de pasos o, si se define, no hay garantía de que funcione bien cada vez. Su solución se basa en conocim. y experiencia previos y un proceso de intento y error.

Ej: la educación de los hijos.-

E → P → S
 Computador



dispositivos de entrada
 Permiten entrar info. en el comp. desde el mundo exterior

- teclado
- mouse
- cámara
- lector de código de barra
- microfono
- scanner

dispositivos de salida.
 Permiten sacar info. desde adentro del computador hacia el mundo exterior.

- pantalla
- impresora
- parlantes

dispositivos de almacenamiento

A veces funcionan como disp. de entrada (lectura) y	<ul style="list-style-type: none"> disco duro CD DVD zip disks 	<ul style="list-style-type: none"> pen drive tapes etc. 	a veces como disp. de salida (escritura)

Computador

- hardware Parte "dura", lo que se puede tocar.
- software Son los programas, las instrucciones que hacen que un computador haga algo

- Sist. operativo
- word
- excel
- powerpoint
- autocad
- quicken / MS Money
- Sists. de inventario
- Sists. de nómina
- lenguajes de programación de alto nivel: C, C++, C#, pascal, fortran, Visual Basic, basic, Java, etc.

- Almacenam. secundario o permanente: ^{memoria secundaria}
Se almacena info. que permanece más allá del momento en que se apaga el computador. (CD, DVD, pen drive, disco duro).

- Almacenam. primario o volátil: ^{memoria primaria} (RAM)
Se almacena info. temporalmente en lo que se hace algo con ella. Esta info. se pierde cuando se apaga el computador.

Evolución de los lenguajes de programac!

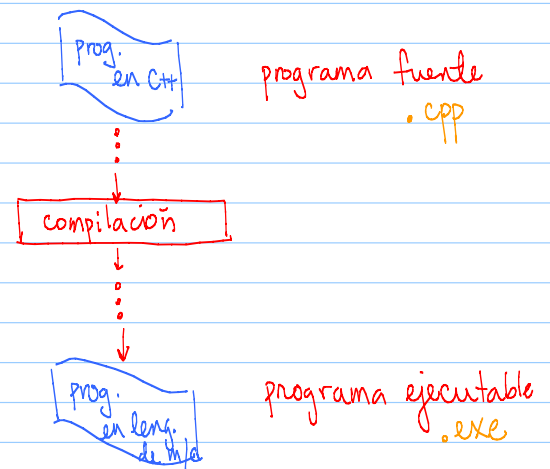
1º Leng. de máquina - leng. natural del computador (binario) y es el único que puede entender. Depende del hardware.

2º "Assembly Language": En forma de comandos. Necesita traducirse el prog. a leng. de m/c para poder ent. ejecutarse. Esto lo hace un traductor (ensamblador).

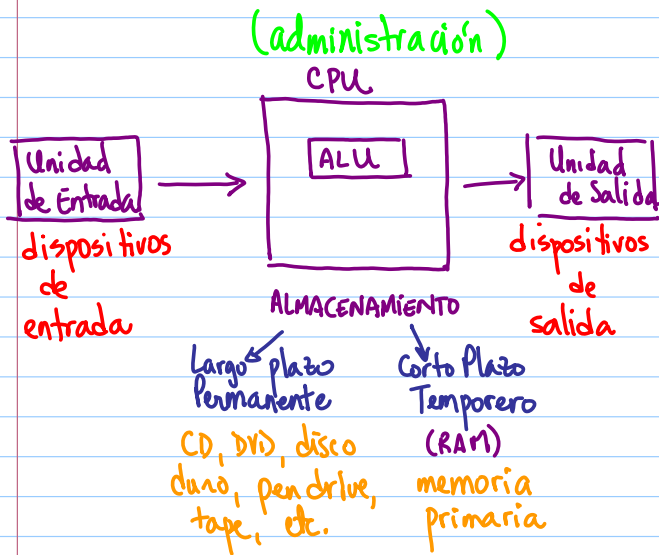
3º Leng. de prog. de alto nivel - como C++. Se escriben las insts. en C++ y un compilador las traduce a leng. de m/c para que se puedan ejecutar.

Lenguajes - Resumen

- Vamos a programar en C++.
- El computador NO entiende C++ directamente.
- El computador solo entiende 1 lenguaje: LENGUAJE DE MÁQUINA. (leng. binario, de 1's y 0's).
- C++ NO es un leng. natural de comput. (pg. no es binario) y las instrucciones de C++ hay que traducirlas 1º a leng. de máquina para que se puedan ejecutar. Ese trabajo lo hace el COMPILADOR de C++.



Organización de un computador.



ALU: Unidad Aritmética y Lógica

- Cálculos
- Comparaciones
- Toma de decisiones
- etc.

CPU: Unidad Central de Procesamiento

6 pasos en la solución de problemas:

- 1) Identificar el problema.
- 2) Entender el problema.
 - a) Comentar sobre el problema.
 - b) Limitaciones/restricciones
- 3) Listar las alternativas de solución. Ventajas y desventajas de cada una.

Analisis

Alternativa	Ventajas	Desventajas
1.	=	=
2.	=	≡
3.	=	=

- 4) Seleccionar la mejor alternativa y justificar.

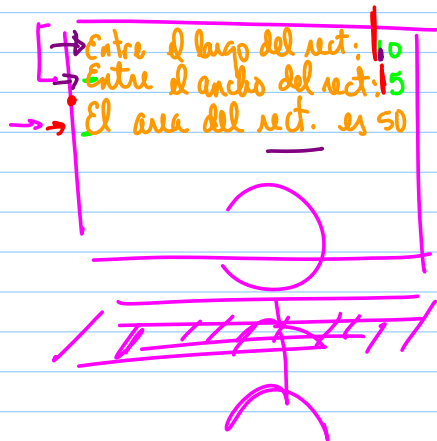
Programa 5) Listar, paso a paso, las instrucciones que conducen a la solución del problema con la alt. seleccionada.

Prueba 6) Probar la solución ¿funciona?

FASE

- ① Análisis → Análisis de Entradas, Procesam. y Salidas.
- ② Programa → C++
(Microsoft Visual Studio)
↳ MS Visual C++
- ③ Prueba → En la computadora.
↳ Manual.

Ej: Calcular el área de un rectángulo.



FASE 1: Análisis

Análisis

(2) Entradas	(3) Procesam.	(1) Salidas
• largo del rectángulo (l) • ancho del rectángulo (an)	1) Solicitar el largo (l) y el ancho (an) del rect.	• El área del rectángulo (A)

2) Calcular el área (A) del rect:

$$A = l \times an$$

3) Imprimir el área del rect. (A)

Objetivos del análisis:

- Extraer la información importante para fines de la solución del problema y desechar la que no es importante.

- Sirve de comunicación con el cliente, para que confirme si eso que se plantea en el análisis es lo que él quiere.

** Usar un leng. sencillo que todo el mundo pueda entender aunque no sepa programar **

NO USO LENG. TÉCNICO

- En las entradas y las salidas NO se usan verbos, sólo listado de datos.

- El procesam. es un listado de pasos (si usa verbos) en forma de "bullets".

FASE 2: Programa (en C++)

```
#include <iostream>
using namespace std;
```

```
① int main()
{
  ② double l, an, A;

  // 1) Solicitar el largo (l) y el ancho (an) del rect.
  ③ cout << "Entre el largo del rect: ";
  ④ cin >> l;

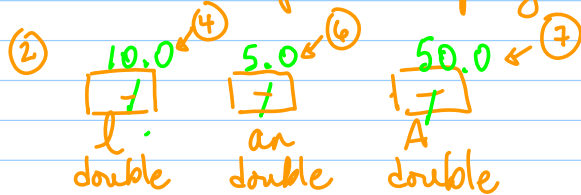
  ⑤ cout << "Entre el ancho del rect: ";
  ⑥ cin >> an;

  // 2) Calcular el área (A) del rect:  $A = l \times an$ 
  ⑦  $A = l * an;$ 

  // 3) Imprimir el área del rectángulo (A)
  ⑧ cout << "El área del rect. es" << A;
  ⑨ return 0;
}
```

FASE 3: Prueba Manual.

① Se inicia la ejecuc. del prog.



③ (Vea pant.)

④ $l \leftarrow 10$

⑤ (Vea pant.)

⑥ $an \leftarrow 5$

⑦ $A = l * an;$
②

⑧ (Vea pant.)

⑨ Fin del prog. → Se liberan las vars.

③ Entre el largo del rect: 10
⑤ Entre el ancho del rect: 5
⑧ El area del rect. es 50.0

Note Title

3/15/2011

El propósito de la prueba manual es:

- 1) demostrar que el prog. hace lo que se solicitó (es decir, cumple con las especificaciones en nuestro caso, calcula el área de un rectángulo), y
- 2) lo hace bien. Es decir, para un largo 5 y un ancho 3, que fueron los datos de entrada usados en la prueba, se produjo una resp. de 15, que es la que se supone.

Tarea:

- 1) Programa que calcula el área de un triángulo. $A = \frac{1}{2}bh$
- 2) Programa que calcula el área de un círculo. $A = \pi r^2$

FASE 1 → Analisis

FASE 2 → Programas

FASE 3 → Prueba Manual (Test).

Programa anterior que se entrob en C++.

```
#include <stdafx.h>
using namespace System;

#include <iostream>
using namespace std;

int main()
{
    double l, an, A;

    cout<<"Entre el largo del rect: ";
    cin>>l;

    cout<<"Entre el ancho del rect: ";
    cin>>an;

    A = l * an;

    cout<<"El area del rectangulo es "<<A;

    return 0;
}
```

Variable

Es una celda de memoria primaria cuyo contenido puede cambiar a lo largo de la ejecución de un programa.

Para cada dato que un prog. tiene que manipular y que podría cambiar de una ejecución a otra, e incluso dentro de una misma ejecución, se necesita una var. que lo represente.

Toda var. debe definirse antes de usarse por 1^{ra} vez.

Una definición de variable tiene 2 partes:

double l;

↑ ↓
tipo de dato nombre único

double an; { double an, A;
double A;

memoria

	X		-	A
an	-	X	X	
	X		/	l

19.0
5.5
0.0
9.0
9.55
-10.44

Nombre único - regla:

Cualquier combinación de letras, dígitos y el símbolo especial de "underscore" (`_`), siempre y cuando comience con una letra.

✓ = buen nombre (no da error, cumple con la regla) ✗ = mal nombre

Ejercicio:

l	✓	l Rect	✗	l%	✗
largo	✓	lRect1	✓	l!	✗
largoR	✓	123Rect	✗	l!	✓
largo1	✓	"Rect"	✗	Juan	✓
l_2	✓	R*R	✗	nombre	✓
l-2	✗	RxR	✓		
l#	✗	'R'	✗		

Tipo de Dato

Es el tipo de información que se puede almacenar en una variable.

Los tipos de datos básicos son:

Númericos → int
→ double

Alfanuméricos → char
→ string

Lógicos → bool

i) Numéricos: Almacenan información numérica.

a) int Almacenan números enteros

Ej: `int n;`

- Número de estudiantes matriculados (4 bytes)
- Cant. de artículos comprados
- Código de un producto
- Número de estudiantes en la excursión
- hora
- minuto } horario
- segundo
- día (1-31) } fecha
- mes (1-12)
- año

$$0.035 \rightarrow 3.5 \times 10^{-2}$$

b) **double** Almacenan números reales (parte entera + parte decimal)

3.5e-2
 2.0
 -179.99
 2.0
 4.48
 3.754194
 [F]
 p
 double

Ej: **double** p;

(8 bytes)

- precio
- velocidad
- aceleración
- distancia
- largo
- ancho
- area
- volumen
- la coordenada x
- altura
- tiempo transcurrido en segundos (ej: 3.25)
- promedio final
- nota (numérica) obtenida en un examen.
- densidad
- pH

2) Alfanuméricos: Almacenan TEXTO

a) **char** Almacena una letra, un dígito o un símbolo especial por vez, entre ' ' para diferenciar la letra de un nombre de variable.

✗
 ✗
 ✗
 ✗ 'A'
 ✗ 'B'
 ✗
 ✗
 ✗
 ✗
 ✗

Ej: **char** letra;

[F]
 letra
 char

(1 byte)

A ≠ 'A'

↑

↑

la var. A

la letra A.

Un valor char, que podría almacenar en la var. letra, pq. está definida char

Ejemplo: `int A = 10; char letra = 'A';` → [10] [A]

`cout << letra;` → A

`cout << "letra";` → letra

`cout << A;` → 10

`cout << 'A';` → A

int char

1 ≠ '1'

↑

↑

es una letra,

valor int
 Se puede almacenar en una var. int

valor char
 Se puede almacenar en una var. char

• calificación (A, B, C, D o F)

• sexo (F o M)

b) string

Un string es un conj. de caracteres. Por lo tanto, una var. string puede almacenar cualquier combinación de letras, dígitos y símbolos especiales, entre " " para diferenciarlos de los nombres de variables.

teórico

"598-98-2899"

"(09) 342-4911"

"Hola, R2D2!!"

"Bugn dia"

"Calle 1 #25"

"x+y+z=39"

"Jugar del Pueblo"

Ej: string S;

"Blanca Tallaj"
S
string

- nombre
- dirección
- teléfono
- seguro social
- código de una cuenta de banco
- nombre de un color ("azul", "amarillo"...)

3) lógicos

a) bool

Almacenan solamente uno de dos posibles valores: true (1) o false (0)

false (0)
true (1)

1
lic

bool
(1 byte)

Ej: bool lic;

- Si cliente tiene o no tiene una lic. válida (true / false)

Ejercicio (resp. sec. 07)

Para cada dato, llene las columnas de la tabla con: nombre de var., tipo de dato y la def. que Ud. le daría a la var. correspondiente.

dato	nombre de var.	tipo de dato	def. de var.
1.) precio de un artículo	precio	double	double precio;
2.) número de presos	presos	int	int presos;
3.) código de un plan médico (C para Cosvi, M para MCS, S para SSS)	CodigoPlan	char	char CodigoPlan;
4.) Nombre del plan médico	PlanMedico	string	string PlanMedico;

dato	nombre de var.	tipo de dato	def. de var.
5.) Si cliente tiene lic. válida (true o false)	lv	bool	bool lv;
6.) Si cliente tiene lic. válida ("SI" o "NO")	lVal	string	string lVal;
7.) Si cliente tiene lic. válida ('S' o 'N')	licV	char	char licV;

Ejercicio (resp. sec. 06)

Para cada dato, llene las columnas de la tabla con: nombre de var., tipo de dato y la def. que Ud. le daría a la var. correspondiente.

dato	nombre de var.	tipo de dato	def. de var.
1.) precio de un artículo	precio	double	double precio;
2.) número de presos	presos	int	int presos;
3.) código de un plan médico (C para Cosvi, M para MCS, S para SSS)	CodPlanMed	char	char CodPlanMed;
4.) Nombre del plan médico	PlanMed	string	string PlanMed;

dato	nombre de var.	tipo de dato	def. de var.
5.) Si cliente tiene lic. válida (true o false)	licValid	bool	bool licValid;
6.) Si cliente tiene lic. válida ("SI" o "NO")	LValida	string	string LValida;
7.) Si cliente tiene lic. válida ('S' o 'N')	licV	char	char licV;
8.) Sexo de una persona (F = Femenino, M = Masculino)	S	char	char S;

