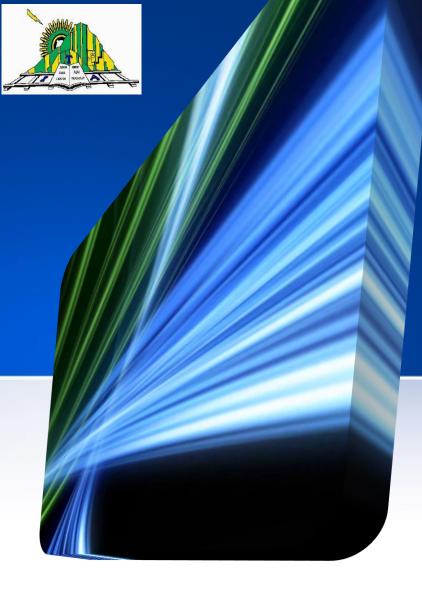
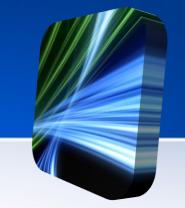


# PROGRAMACIÓN BÁSICA "SISTEMAS AUTOMOTRICES"

M.C. Alicia Cortés Fernández



# BIENVENIDA



# LENGUAJE C++





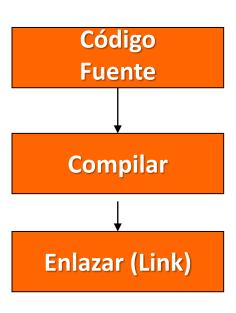
Ver presentación

https://docs.google.com/file/d/0B\_e\_Y9BTFXx1OWJhNzJh OTEtOTcyYi00MjM0LWE1MWQtMjc2OGM5YTQ1YTBk/edit ?pli=1



 Para crear un programa en C, se escribe el código fuente (programa), luego se compila y finalmente se enlaza con las bibliotecas (se hace un link, en nuestra jerga diríamos "se linkea").

Para hacer un programa en C, lo primero que se debe hacer es crear el programa fuente (con extensión ".cpp")





# FORMA GENERAL DEL LENGUAJE C



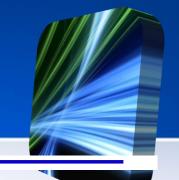
- Todo lenguaje de programación posee palabras claves, estas son aquellas palabras que reserva el lenguaje para identificar ciclos, estructuras y en general cualquier cosa que sea parte de instrucciones.
- Por ejemplo, las palabras while, if, struct son palabras claves en C.
- En C las palabras claves o reservadas deben escribirse en minúsculas, esto ya que C diferencia entre mayúsculas y minúsculas (a diferencia del lenguaje Pascal)

### **Ejemplo del COUT Y CIN**



```
#include <iostream>
 2
      #include <cmath>
 3
 4
      using namespace std;
 5
      int main () {
 6
 7
 8
 9
      float numero;
10
      float resultado;
11
          cout << "Ingresa un numero: \n";
12
13
               cin>>numero;
14
```

# ESTRUCTURA DE UN PROGRAMA EN LENGUAJE C++



```
#include<iostream>
using namespace std; // Esta línea es necesaria cuando
           // utilizamos la librería iostream
int main()
 cout << "Hola Mundo!"; //cout es el equivalente de printf,</pre>
              //si se fijan la sintáxis es muy parecida.
 cin.get(); // cin.get es el equivalente de getch, detiene la ejecución del
programa.
 return 0; /* No es necesaria para que el programa funcione,
       pero es una buena costumbre indicar que nuestro programa
terminó sin errores */
```

# LENGUAJE DE PROGRAMACIÓN C++



### El primer programa clásico....

```
/* "Hola mundo" escrito en C */|
#include <stdio.h>
void main()
   printf( "Hola mundo" );
```

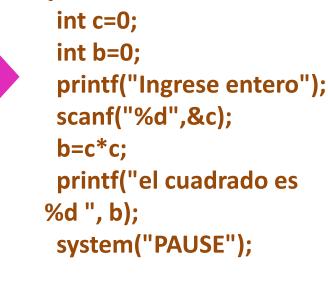
```
/* "Hola mundo" escrito en C++ */
#include <iostream>
void main()
{
   std::cout << "Hola mundo";
}</pre>
```

```
/* "Hola mundo" escrito en C# */
 class HolaMundo
     static void Main()
        System.Console.Write( "Hola mundo" );
```

# Como escribir un programa en

- El **Lenguaje C se escribe** a partir de un programa principal y de funciones, el programa principal también es una función (función principal), y se denomina main(), esta debe estar siempre presente y es la primera en ser llamada.
- Para hacer un programa en C, lo primero que se debe hacer es crear el programa fuente (con extensión ".cpp") de la siguiente manera:

```
#include <stdio.h>
#include <stdlib.h>
int main()
 int c=0;
 int b=0;
 scanf("%d",&c);
 b=c*c;
 printf("el cuadrado es
%d ", b);
 system("PAUSE");
 return 0;
```





# Estructura de un Programa e

Cuando se escribe el programa se recomienda usar el siguiente formato:

- Ilamadas a bibliotecas
- declaración de funciones (prototipos de funciones)



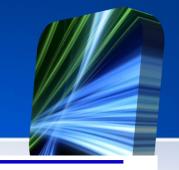
- main()
- {
- declaración de variables locales
- sentencias
- }
- definición de funciones

```
/* esto es un comentario */
#include<stdio.h>
 #include<stdlib.h>
float num;
 float res, cubo(int x);
 main()
 printf("ingrese un
                       número ");
 scanf("%f",&num);
 res=cubo(num);
 printf("%f al cubo es: %f",num,res);
 system("PAUSE");
return 0;
float cubo(float x)
 {return x*x*x;}
```

# Estructura de un Programa en C

- Por lo general las bibliotecas contienen funciones creadas por el programador o entregadas producto. Por ejemplo las funciones de entrada y salida vienen especificadas en la biblioteca de entradas y salidas estándar stdio.h (Standar Input Ouput Header).
- # include <stdio.h>
- Cada vez que se escribe una línea, esta debe llevar ";" al final, con lo cual el compilador entiende que llega al final de una instrucción o proposición.
- Antes de usar una variable dentro de un programa, es necesario indicarle al compilador que va a ser usada, para ello se declara:
- <tipo> de\_variables>; Ejemplo: Int a;
- Como ya es sabido, una asignación es almacenar en una variable el resultado de una operación o proceso. En "C" una asignación se representa como:
- <variable> = <expresión>;
- Ejemplo:
- a=5; /\* a la variable a se le asigna el valor 5 (constante)\*/
- expo=max; /\* a la variable expo se le asigna el contenido de la variable max\*/
- expofin=ult+4; /\* a expofin se le asigna el contenido de la variable ult sumada a la constante 4\*/

# EJEMPLO DE LA ESTRUCTURA DE UN PROGRAMA EN LENGUAJE C++ Y C



```
Para sustituir scanf() se usa cin,
por ejemplo: cin >> edad;
   #include<iostream>
   using namespace std;
   int main()
    int edad;
    cout<<"Cual es tu edad?: ";
    cin >> edad;
    cout<<"ntTu edad es " <<edad;</pre>
    cin.get();cin.get();
    return 0;}
```

```
#include<stdio.h>
#include<stdlib.h>
int main()
 int edad;
 printf("Cual es tu edad?: ");
 scanf("%d",&edad);
 //system("cls"); // limpia
pantalla
 printf("\n\n\t\aTu edad es %d
",edad);
```

# EJEMPLO DE LA ESTRUCTURA DE UN PROGRAMA EN LENGUAJE C CON MÁS COMANDOS



```
#include<stdio.h>
#include<stdlib.h>
int main()
 int edad;
 printf("Cual es tu edad Enrique Castel%cn?: ",160);
 scanf("%d",&edad);
 system("cls");
 printf("\n\n\t\aTu edad es %d a%cos Enrique
Castel%cn", edad, 164, 160);
```



### Cadenas de formatos

%d,% i	entero decimal con signo, (eje: 38)
<b>%</b> 0	entero octal sin signo, (eje: o43)
%u	entero decimal sin signo, (eje: 47)
%x	entero hexadecimal sin signo (en minúsculas), (eje: 0x32)
%X	entero hexadecimal sin signo (en mayúsculas), (eje: 0X32)
%f	Coma flotante en la forma [-]dddd.dddd, (eje: -3.2, 56.896)
%e	Coma flotante en la forma [-]d.dddd e[+/-]ddd, (eje: 2.3e4, 4.879e-4)
%g	Coma flotante según el valor
%E	Como e pero en mayúsculas, (eje: 2.3E4, 4.879E-4)
%G	Como g pero en mayúsculas
%с	un carácter, (eje: a, A)
%s	cadena de caracteres terminada en '\0' o null, (eje: 'juan', 'JUAN')
%5s	primeros cinco caracteres o delimitador
%8.2f	tamaño total de 8 dígitos, con dos decimales
	<u> </u>

### Secuencias de escape

∖a	Alerta
\ <b>b</b>	Espacio atrás
\ <b>f</b>	Salto de página
\ <b>n</b>	Salto de línea
\ <b>r</b>	Retorno de carro
\t	Tabulación horizontal
\ <b>v</b>	Tabulación vertical
//	Barra invertida
\'	Comilla simple
\"	Comillas dobles

# Identificadores

- Estos son simplemente los nombres que pueden asumir vo constantes dentro del programa.
- **Existen algunas restricciones** para los identificadores:
- No deben ser iguales a nombres de palabras reservadas o bibliotecas.
- Deben comenzar con una letra o el caracter '\_'.
- No deben tener en medio el caracter espacio.

### Ejemplo de identificadores:

```
Contador
Conta
var1
_variable3_
Suma_total
```

# Comentarios

# Programación

# Ejemplo:

/\* esto es un comentario \*/

# Entrada / Salida Estándar

La biblioteca (o archivo de cabecera) llamada **stdio.h** contienda las funciones de entrada y salida estándar, entre ellas las funciones **printf()** y **scanf()**. La **f** significa que es entrada o salida con formato. **printf()** 

Esta es la función de salida por pantalla, su formato es el siguiente:

```
printf(cadena_de_control,lista_de_argumentos)
printf("a * b = %d \n", c );
```

Los caracteres **barra invertida** (backslash) y **n** juntos, provocan un retorno de carro (return) en pantalla.

# Entrada / Salida Estándar



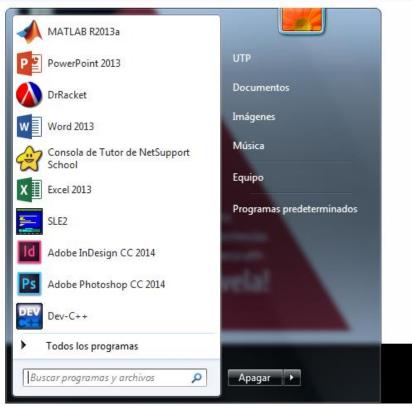
# scanf()

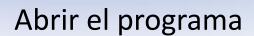
Esta es la función para entrada estándar, se utiliza de un modo parecido a la anterior.

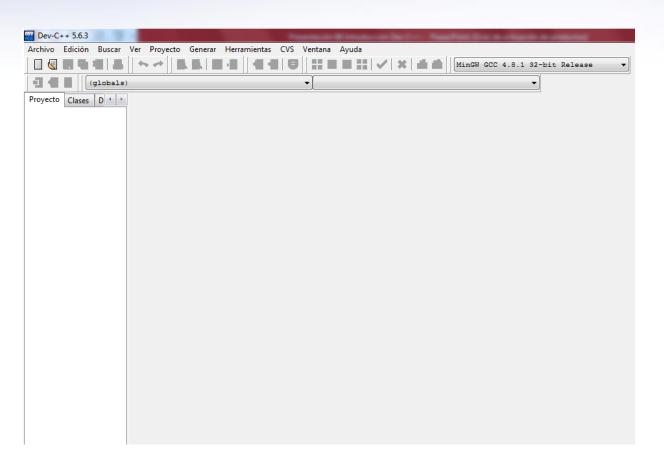
scanf("%d",&numdato);

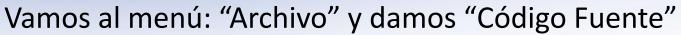
%d indica que se lee con formato entero. En el ejemplo el caracter "&" se usa para indicar la dirección de memoria, es decir, "lee a un valor con formato entero y lo guarda en la dirección de memoria numdato".

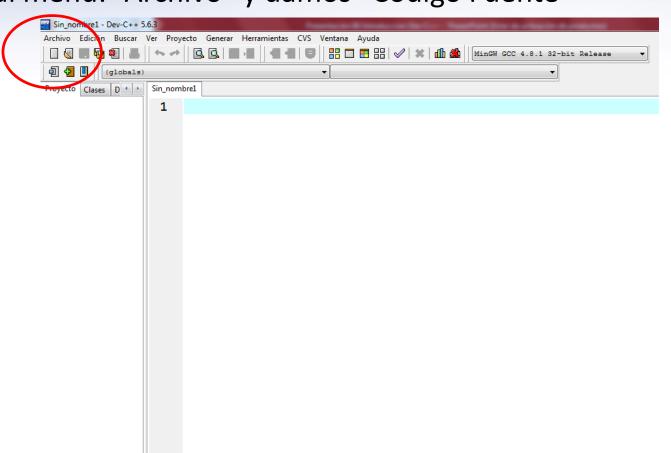
Pasos para iniciar un programa nuevo en Dev C++. Instalar el programa.









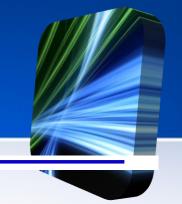


### Ejemplo Lenguaje de programación C y C++



```
/* Programa que nos permite
saber si el número ingresado es
par o no */
#include <stdio.h>
#include <conio.h>
int main () {
int numero;
printf ("Ingrese el valor de
numero: ");
scanf ("%d", &numero);
if(numero%2 == 0) {
printf ("\n***El numero es
par\n"); }
else { printf ("\n***El numero
es impar\n"); }
getch();
return 0 }
```

### Ejemplo con uso de metodología



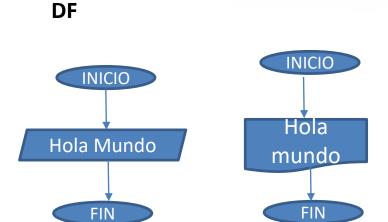
1. PLANTEAMIENTO DEL PROBLEMA.

Imprimir en pantalla "Hola Mundo"

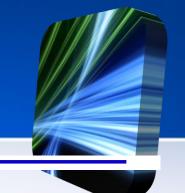
- 2. ANÁLISIS DE LA SOLUCIÓN DEL PROBLEMA
- 3. DISEÑO

### Pseudocódigo

{ imprimir en pantalla "Hola Mundo" }



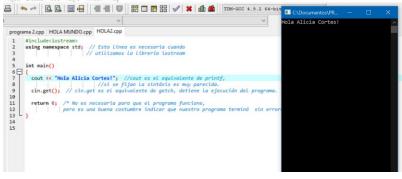
### Ejemplo con uso de metodología

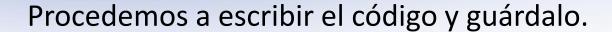


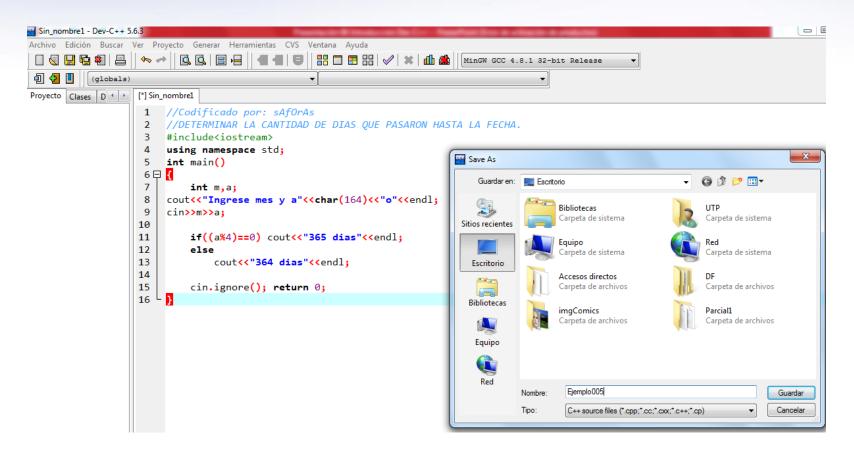
### 4 código

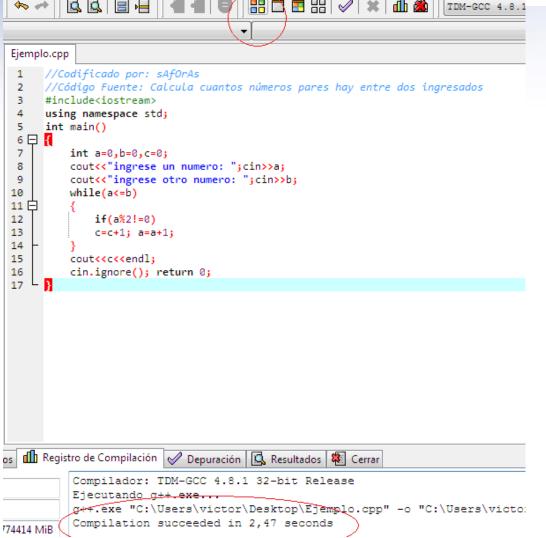
```
#include <stdio.h>
#include <conio.h>
int main () {
 printf ("Hola Mundo ");
}
```

### 5 Ejecución









Luego lo compliantes para que el programa verifique la estructura del programa, donde se nos indica si existen errores.



### Luego lo ejecutamos.





Ver paginas:

http://saforas.wordpress.com/2008/01/04/codigos-sencillos-hechos-en%C2%A0c/



### Referencias utilizadas:

Webgrafia:

http://jorgesaavedra.wordpress.com/2006/12/09/breve-historia-de-c-c-c/



# ii MUCHAS GRACIAS!!

