UNIVERSIDAD DE ANTIOQUIA

1803

Estructuras de control

Informática II

Departamento de Ingeniería Electrónica y Telecomunicaciones

A continuación…

- **Introducción**
- **Estructura, variables y Operadores**
- **Estructuras de Control**
- **Ejemplos**
- **Ejercicios**

```
int main (void){
    //Variables
        int a, b, c;
    //Ingreso de datos
        cout<<"Introduzca el primer numero (entero)";
        cin>>a;
        cout<<"Introduzca el segundo numero (entero)";
        cin>>b;
    //Proceso
        c = a + b
    //Salida
        cout<<"La suma es: "<<c;

}
```

C++

Ensamblador

Alto Nivel

Bajo Nivel

Máquina

```
;Lenguaje ensamblador, sintaxis Intel para
procesadores x86
mov eax,1; //mueve a al registro eax el valor 1
xor ebx, ebx; //pone en 0 el registro ebx
int 80h; //llama interrupción 80h(80h=128sistema
decimal)
```

```
                                        48 6F 6C 61 2C
20 65 73 74 65 20 65 73-20 75 6E 20 70 72 6F 67
72 61 6D 61 20 68 65 63-68 6F 20 65 6E 20 61 73
73 65 6D 62 6C 65 72 20-70 61 72 61 20 6C 61 20
57 69 6B 69 70 65 64 69-61 24
```

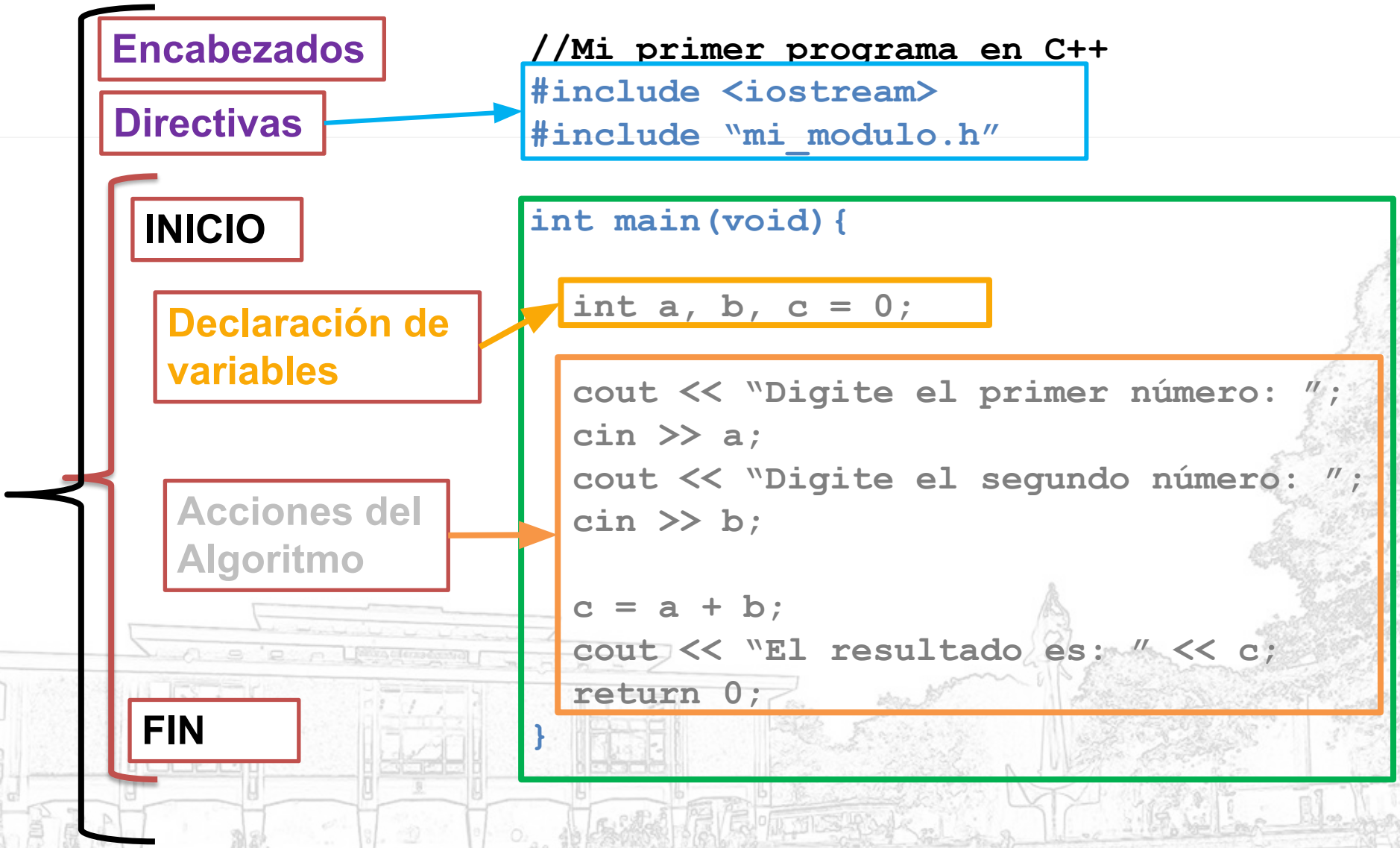
Generalidades del Lenguaje C++

- Ventajas
  - Lenguaje estándar.
  - Se puede realizar código multiparadigma.
  - Más productivo.
  - Fácil de comprender.
  - Código reutilizable.
  - Librería estándar.
  - Portabilidad entre diferentes plataformas.
- Desventajas
  - Gran cantidad de memoria (Flash y RAM).
  - Precio de las herramientas de compilación y depuración algo costoso.
  - Código generado innecesario.

A continuación…

- Introducción
- **Estructura, variables y Operadores**
- Estructuras de Control
- Ejemplos
- Ejercicios

Estructura de un Programa



Lenguaje C++

VARIABLES

Variables y Constantes

- Una variable es un lugar en memoria para almacenar información.
- Cada variable tiene un tamaño específico que le dice a la máquina cuanta memoria necesita reservar.
- Existen variables locales y globales.

Tipo	Número de Bytes
bool	1
char	1
int	4
float	4
double	8
string	N
Otros	…

Variables

- **Locales**
  - Una variable local se declara dentro de una función, son visibles sólo dentro de la función.
  - Ocultan a las variables globales con el mismo nombre.
  - “Nacen” y “mueren” con la función. A no ser que se declaren (static)
- **Globales**
  - Una variable global se declara fuera de todas las funciones es visible en todo el programa.
  - Se le puede cambiar el valor en cualquier parte del programa y lo mantiene.
  - Se recomienda usar pocas o ninguna variable global.

Palabras reservadas

Data type
char
short
signed
unsigned
int
float
long
double

Modifiers
const
static
volatile
restrict

Identifiers
struct
union
void
enum

Selection
if
else
switch
case
default

Storage Specifiers
register
typedef
auto
extern

Iteration
do
while
for

Jump
goto
continue
break
return

Function Specifier
inline

Preprocessing Directives
#include
#define
#undef
#line
#error
#pragma

Conditional Compilation
if
ifdef
ifndef
elif
else
endif

Lenguaje C++

OPERADORES

Aritméticos

Operador	Operación	Clase
+	Suma, adición	Binario
-	Resta, sustracción	Binario
*	Multiplicación, producto	Binario
/	División, cociente	Binario
%	Modulo o residuo de la división	Binario

- Prioridad:
  - *, /, %
  - +, -

Asignación

Operador	Operación	Clase
=	Asignación (valor oper. der a variable oper.izq.)	Binario
+=	Suma y asignación	Binario
-=	Resta y asignación	Binario
*=	Producto y asignación	Binario
/=	División y asignación	Binario
%=	Modulo y asignación	Binario
<<=	Corrimiento a izquierda y asignación	Binario
>>=	Corrimiento a derecha y asignación	Binario
&=	AND binaria y asignación	Binario
^=	XOR binaria y asignación	Binario
|=	OR binaria y asignación	Binario

Incrementos y decrementos

- Aritméticos incrementales - decrementales:
  - Pre…
  - Post..

Operador	Operación	Clase
++	Incremento en 1	Unario
--	Decremento en 1	Unario

Relacionales

Operador	Operación	Clase
==	Igualdad	Binario
!=	Desigualdad	Binario
<	Menor (oper. izq. menor que oper. der.)	Binario
<=	Menor o igual	Binario
>	Mayor	Binario
>=	Mayor o igual	Binario

Lógicos

Operador	Operación	Clase
&&	AND lógica (booleana)	Binario
||	OR lógica (booleana)	Binario
^	XOR lógica (booleana)	Binario
!	NOT lógica (booleana)	Unario

A nivel de bits

Operador	Operación	Clase
&	AND binaria	Binario
|	OR binaria	Binario
^	XOR binaria	Binario
~	NOT (complemento) binaria	Unario
<<	Corrimiento de bits a la izquierda	Binario
>>	Corrimiento de bits a la derecha	Binario

Otros

Operador	Operación	Clase
sizeof()	Tamaño en bytes del operando	Unario
*	Valor en la dirección de un puntero	Unario
+	Signo positivo	Binario
-	Signo negativo	Unario
.	Acceso a un miembro de estructura	Unario
&	Dirección en memoria, referencia	Unario
->	Acceso a un miembro de puntero a estructura	Unario
()	Paréntesis en fórmula matemática. Conversión.	Unario
[]	Corchetes en fórm. mat. Acceso en arreglos.	Unario
?:	Operador ternario de comparación	Ternario
::	Resolución de contexto (ámbito)	Binario
new	Reserva dinámica de memoria en el heap	Unario
delete	Liberación de memoria reservada con op. new	Unario

Operadores

Primary expressions and postfix operators

`( )`	subexpression and function call
`[ ]`	array subscript
`->`	structure pointer
`.`	structure member
`++`	increment(postfix)
`--`	decrement(postfix)

Unary

`!`	logical negation
`~`	one's complement
`++`	increment(prefix)
`--`	decrement(prefix)
`-`	unary negation
`+`	unary plus
`(type)`	type cast
`*`	pointer indirection
`&`	address of
`sizeof`	size of

Assignment

`=`	plain assignment
`+=`	add
`-=`	subtract
`*=`	multiply
`/=`	divide
`%=`	remainder
`<<=`	bit left shift
`>>=`	bit right shift
`&=`	bit AND
`^=`	bit exclusive OR
`|=`	bit inclusive OR

Bitwise

`&`	bitwise AND
`^`	bitwise exclusive OR
`|`	bitwise inclusive OR
`<<`	bitwise left shift
`>>`	bitwise right shift

Math

`*`	multiplication
`/`	division
`%`	modulus(remainder)
`+`	addition
`-`	subtraction

Relational

`<`	less than left to right relational
`<=`	less than or equal
`>`	greater than
`>=`	greater than or equal
`==`	equal test
`!=`	not equal test

Logical

`&&`	logical AND
`||`	logical inclusive OR

Conditional

`?:`	conditional test

Sequence

`,`	comma

¿Cuánto he aprendido?

- ¿Qué diferencias hay entre variables locales y variables globales?
- Encuentre la(s) palabra(s) reservada(s) que no pertenece (n) al grupo:
  - Tipos de datos: char, int, static, long, const
  - Op. Relacionales: ==, ||, >, >=, !=, &&
- Diferencia entre: a++ y ++a

A continuación…

- Introducción
- Estructura, variables y Operadores
- **Estructuras de Control**
- Ejemplos
- Ejercicios

Generalidades del Lenguaje C

- Las estructuras de control permiten dar solución a cualquier problema de programación.

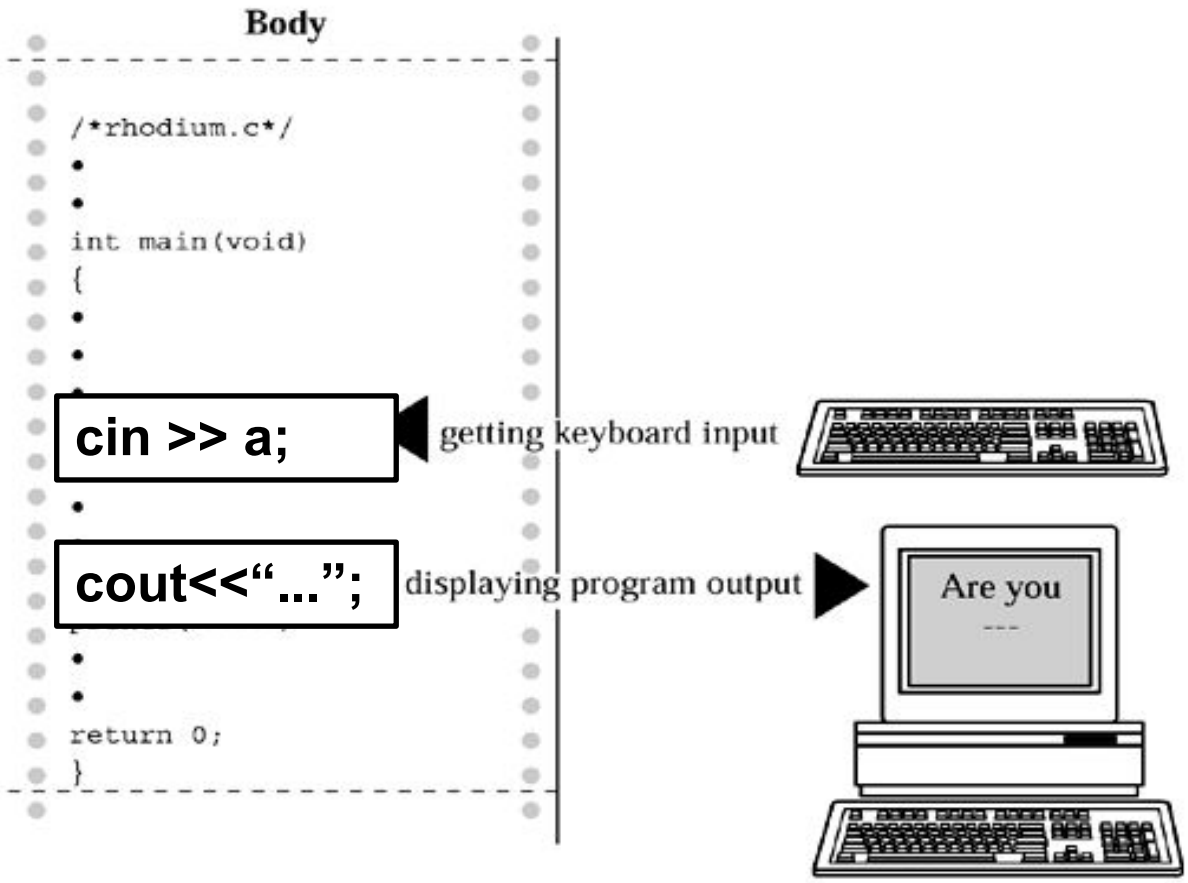
- Existen tres estructuras de control:
  - Secuenciales
  - De Decisión
  - Repetitivas

Entrada y Salida de datos.
Operaciones.

Condicional: **if**
Selección: **switch**

Estructuras cíclicas:
Para: **for**
Mientras: **while**
Haga-Mientras: **do-while**

Secuencial: salida/entrada de datos



Se debe incluir:

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

Condicional: if

Se utiliza cuando interesa realizar acciones que involucren más de dos alternativas

```
if (condicion1) {
      Instrucciones1
}
else if (condicion2){
      Instrucciones2
}
            .
            .
            .
else if (condicionN){
      InstruccionesN
}
else{
     InstruccionesE
}
```

Selección: **switch**

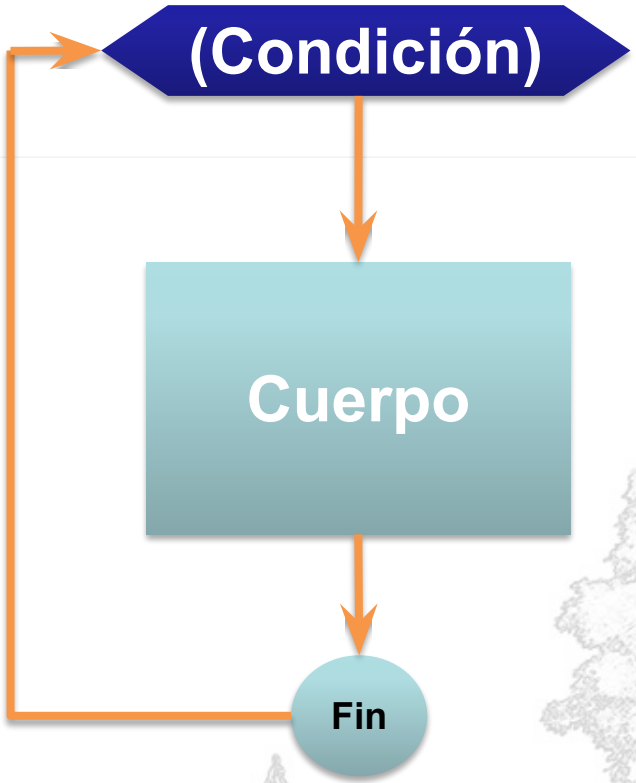
La selección se basa en el valo
de una variable simple o de un
expresión simple denominad
expresión de control o selector.


```
switch (selector) {
      case val1:
            Instrucciones1;
            break;
      case val2:
            Instrucciones2;
            break;
      …
      case valN:
            InstruccionesN;
            break;
      default:

InstruccionesDefault;
}
```

Ciclo: while

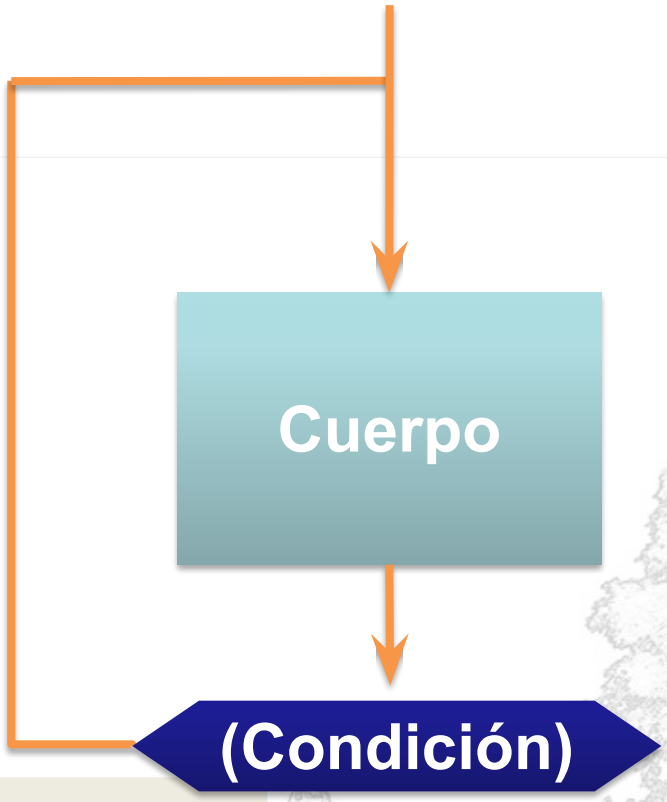
Realiza un conjunto de operaciones mientras una condición se esté cumpliendo, o mientras una expresión sea verdadera




```
while (condición){
   Instrucciones;
}
```

Ciclo: do-while

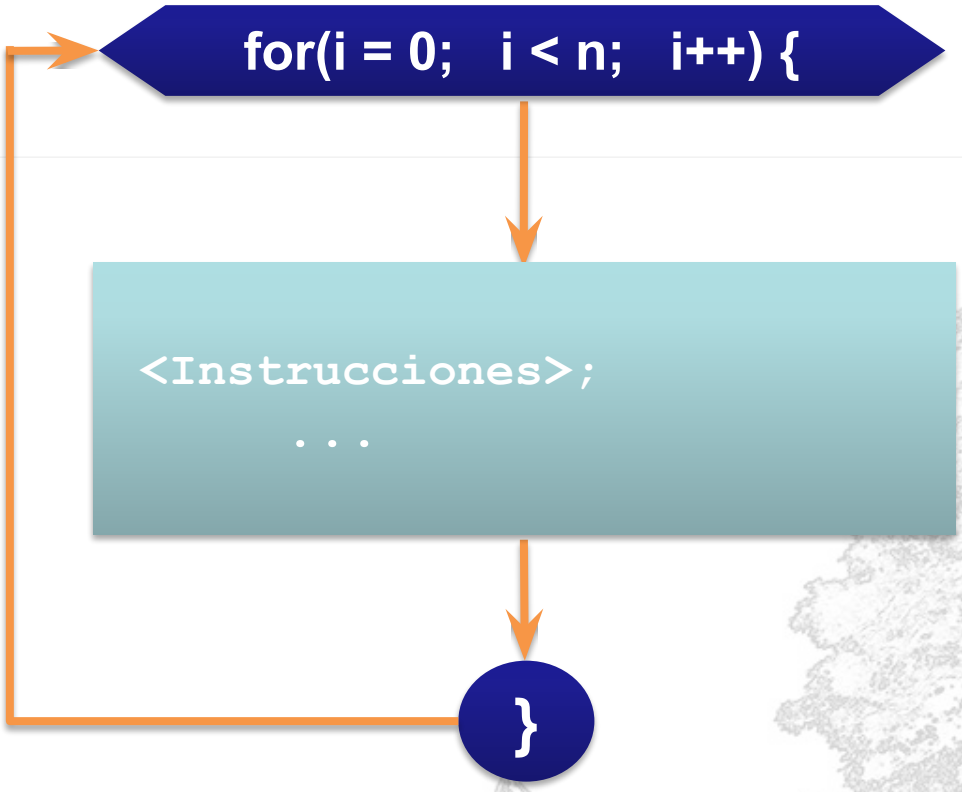
Bucle condicional que se ejecuta al menos una vez (muy semejante al ciclo while).




```
do {
        Instrucciones;
} while (expresión);
```

Ciclo: for

Realiza un conjunto de operaciones un determinado número de veces



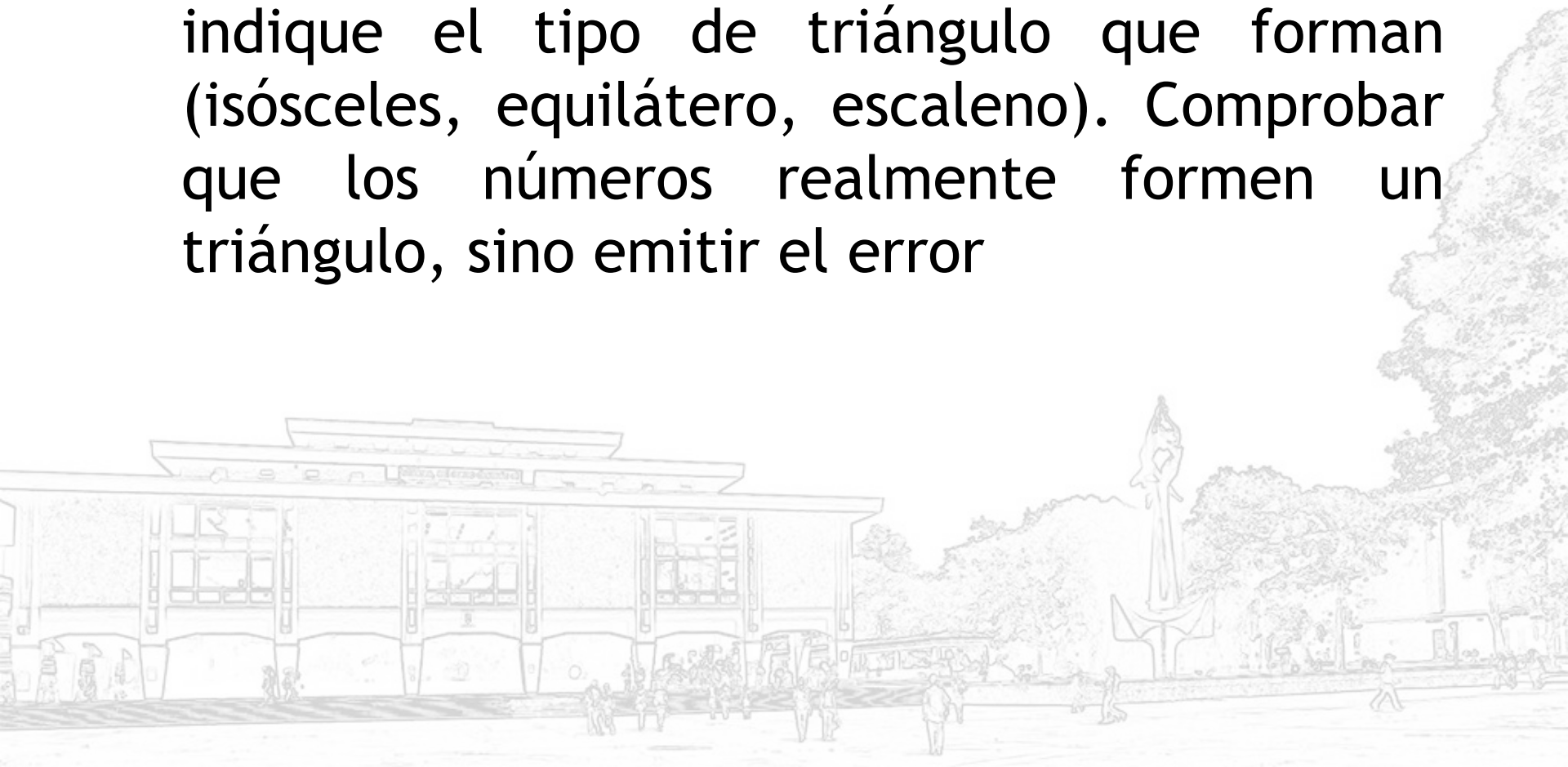

```
for( ValorInicial; condición; INC/DEC){
   instrucciones;
}
```

A continuación…

- Introducción
- Estructura, variables y Operadores
- Estructuras de Control
- **Ejemplos**
- Ejercicios

Ejemplos

- Diseñar un programa que lea tres números e indique el tipo de triángulo que forman (isósceles, equilátero, escaleno). Comprobar que los números realmente formen un triángulo, sino emitir el error



Ejemplos

- Diseñar un programa que lea como entrada tres enteros que representen una fecha como el día, mes, año. Este debe imprimir el número de día, mes y año de la fecha del día siguiente. Típica de entrada: 28 3 1992 La salida típica: Fecha siguiente al día 28:03:1992 es 29:03:1992

Ejemplos

- El curso de informática II está dividido en parte teórica (45%) y parte práctica (55%). En este se realizan tres parciales (uno del 10%, otro del 15% y el final del 20%) y N prácticas de igual porcentaje cada una. El profesor requiere un programa que calcule la nota definitiva de cada estudiante, el promedio definitivo del curso y el porcentaje de estudiantes que ganaron la materia.

MENU

1 Ingresar cantidad de estudiantes

2 Ingresar cantidad de prácticas

3 Ingresar Notas (mostrar definitiva de cada estudiante)

4 Mostrar Promedio del curso

5 Mostrar Porcentaje de estudiantes que ganaron

6 Salir

Seleccione una opción:

Ejemplos

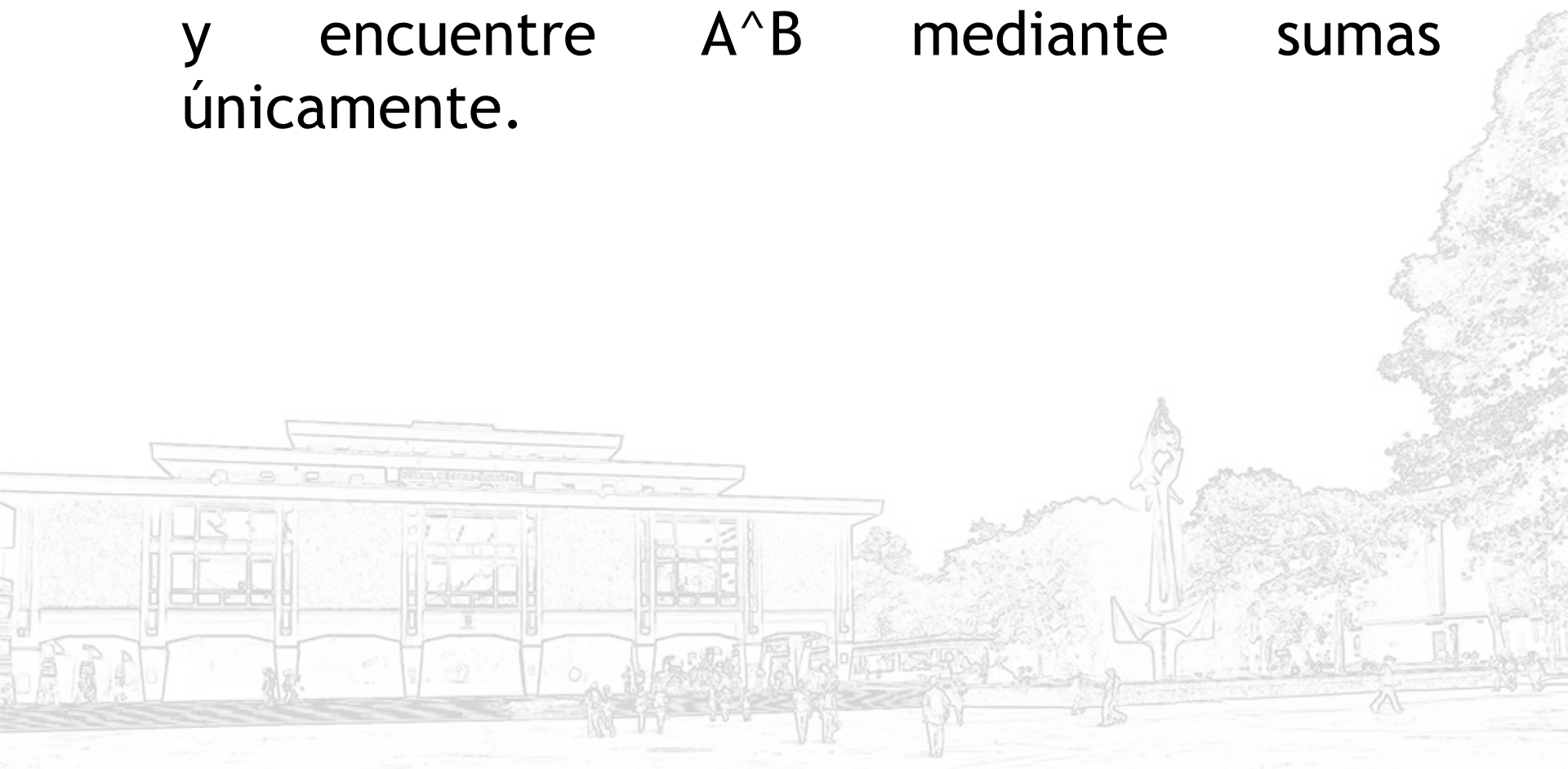
- El profesor indica el número de estudiantes.
- El profesor indica cuántas prácticas se realizaron.
- Todas las notas deben estar entre 0 y 5 (validar datos).
- Al ingresar todas las notas de cada estudiante se debe mostrar su promedio.
- El programa debe tener el menú que se muestra a continuación.
- La única manera de terminar el programa es cuando el profesor seleccione la opción de Salir.

A continuación…

- Introducción
- Estructura, variables y Operadores
- Estructuras de Control
- Ejemplos
- **Ejercicios**

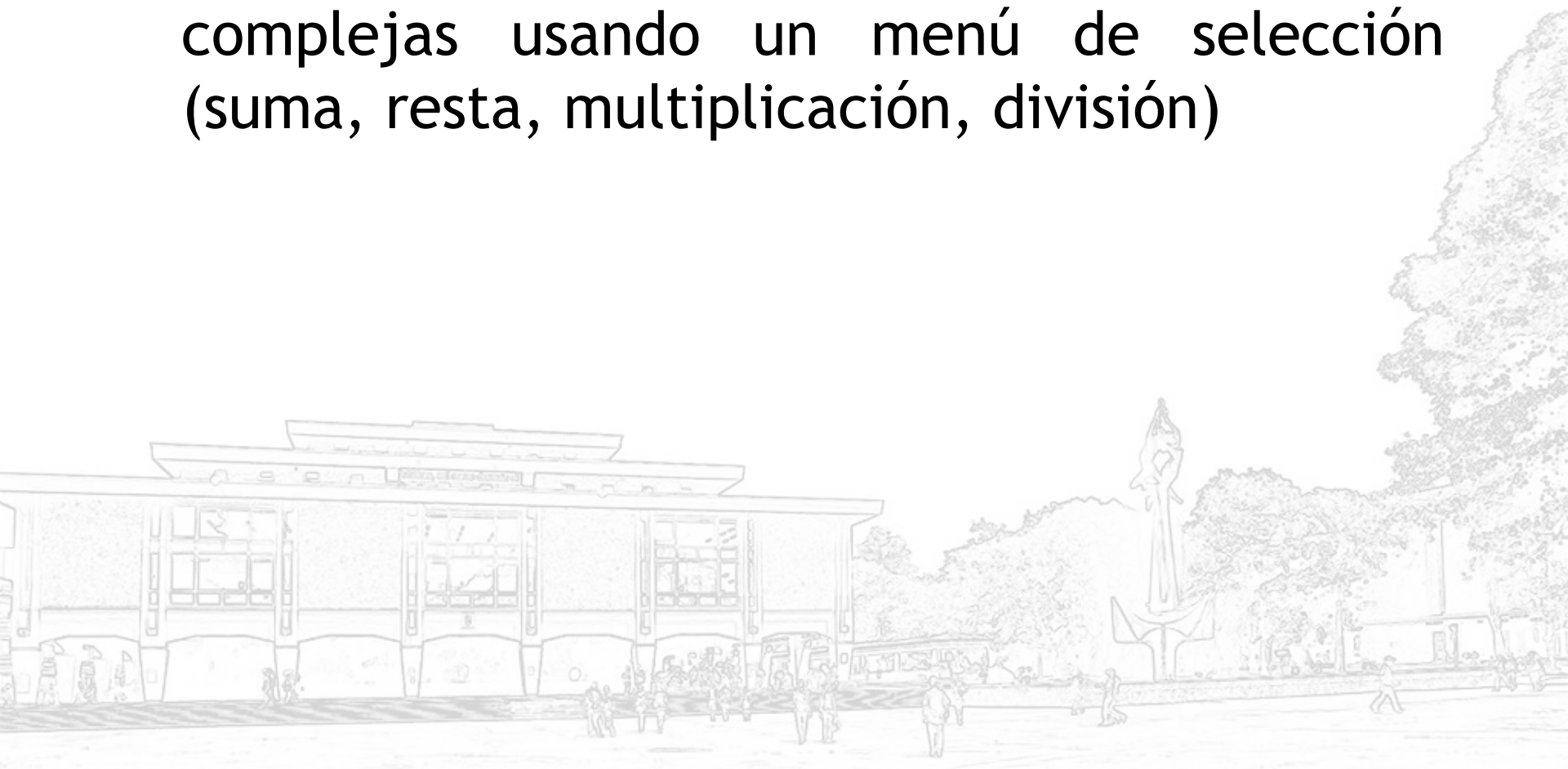
Ejercicios

- Diseñe un algoritmo que lea dos valores A y B, y encuentre A^B mediante sumas únicamente.



Ejercicios

- Realizar calculadora para operar dos variables complejas usando un menú de selección (suma, resta, multiplicación, división)



Para la próxima clase

- Lecturas propuestas
  - Apuntadores, arreglos y funciones
- Motivación:
  - Desarrollo de programas complejos utilizando la modularización mediante funciones y el manejo dinámico de la memoria.
- ¿Preguntas por parte de ustedes?