

# CS1112: Programación II

Unidad 1: Introducción al diseño de algoritmos y programación en C++

Sesión de Laboratorio - 2A

## Profesores:

**María Hilda Bermejo** [mbermejo@utec.edu.pe](mailto:mbermejo@utec.edu.pe)

**Estanislao Contreras** [econtreras@utec.edu.pe](mailto:econtreras@utec.edu.pe)

**Jorge Villavicencio** [jvillavicencio@utec.edu.pe](mailto:jvillavicencio@utec.edu.pe)

**Edson Mendiola** [emendiola@utec.edu.pe](mailto:emendiola@utec.edu.pe)

**Ian Paul Brossard** [ibrossard@utec.edu.pe](mailto:ibrossard@utec.edu.pe)

**Jose Chavez** [jchaveza@utec.edu.pe](mailto:jchaveza@utec.edu.pe)

**Julio Yarasca** [jyarascam@utec.edu.pe](mailto:jyarascam@utec.edu.pe)

**Percy Quevedo** [pquevedo@utec.edu.pe](mailto:pquevedo@utec.edu.pe)

**Wilder Nina** [wnina@utec.edu.pe](mailto:wnina@utec.edu.pe)

**José Fiestas** [jfiestas@utec.edu.pe](mailto:jfiestas@utec.edu.pe)

Material elaborado por:

**María Hilda Bermejo, José Fiestas**



# 1.4

Unidad 1:  
Introducción al diseño de algoritmos  
y programación en C++

- Estructuras de Control

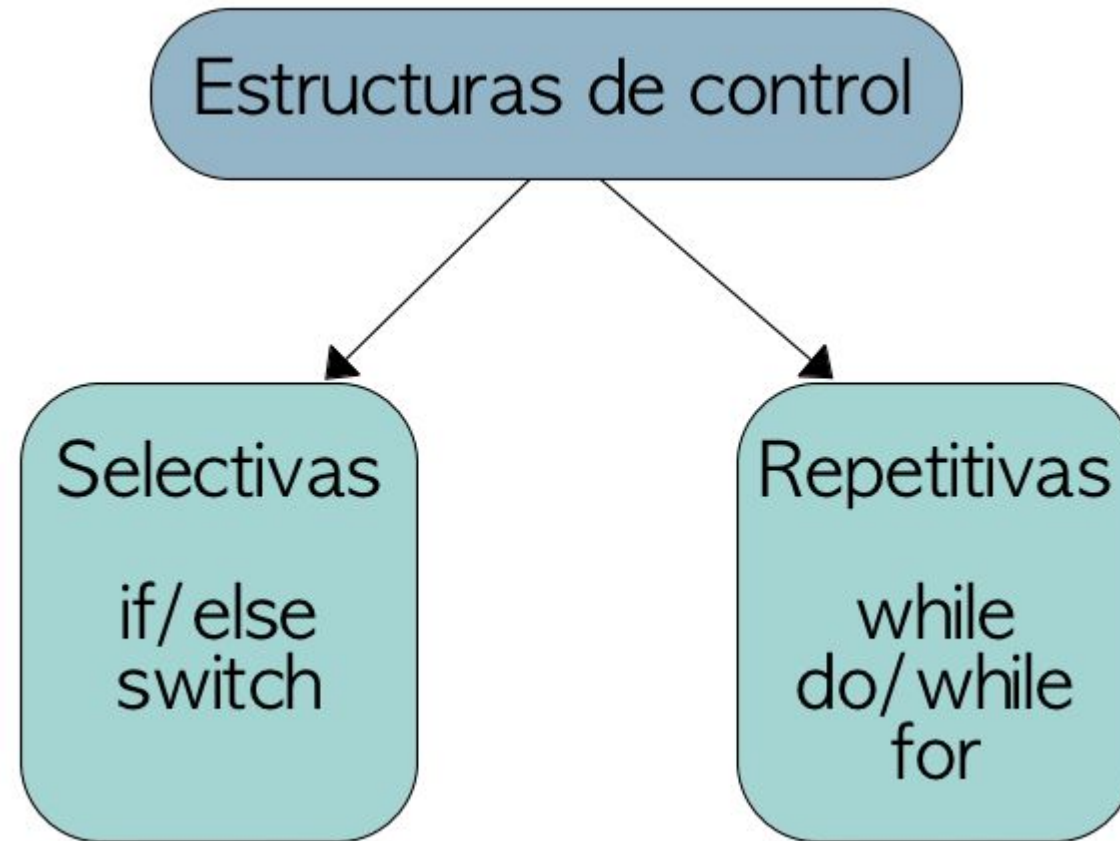
UTEC

## Logro de la sesión:

Al finalizar la sesión, los alumnos podrán:

- Conocer y usar estructuras de control en C++

# Estructuras de Control en C++:



```
#include <stdio.h>
int main(void)
{
    int count;

    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");

    return 0;
}
```

AMEND 10-3

NICE TRY.





# Se imprime 7 veces la palabra UTEC

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int i;
7
8     i=1;
9     while(i<=7)
10    {
11        cout << "UTEC \n";
12        i++;
13    }
14    return(0);
15 }
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int i;
7
8     i=1;
9     do
10    {
11        cout << "UTEC \n";
12        i++;
13    }while(i<=7);
14
15    return(0);
16 }
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6
7     for(int i=1; i<=7; i++)
8         cout << "UTEC \n";
9
10    return(0);
11 }
```

# Ejemplo 1:

Realice un programa, que permita leer varios números enteros hasta que se introduzca el cero.

Luego el programa mostrará lo siguiente:

- La cantidad de números leídos
- La cantidad de números pares
- La cantidad de números impares

El cero, no debe entrar en el conteo.

## Ejemplo 2:

Realice un programa que permita leer un número entero de al menos 3 dígitos, dato que se debe validar, luego el programa indicará:

La cantidad total de dígitos

La cantidad de dígitos pares

La cantidad de dígitos impares

### Ejecución

Numero: 34

Numero: 4

Numero: 345678325

Numero de digitos: 9

Digitos pares: 4

Digitos Impares: 5



## Ejemplo 3:

El valor de PI puede ser aproximadamente la siguiente serie infinita:

$$\pi \approx 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \frac{4}{10 \times 11 \times 12} - \dots$$

Escribir un programa que muestre 60 aproximaciones de PI. La primera aproximación debe de usar solo el primer término de la serie infinita cada aproximación debe incluir un nuevo término.

Observando la lista de aproximaciones podría deducir en qué aproximación se estabiliza el valor de PI.

# Ejemplo 3:

## Ejecución

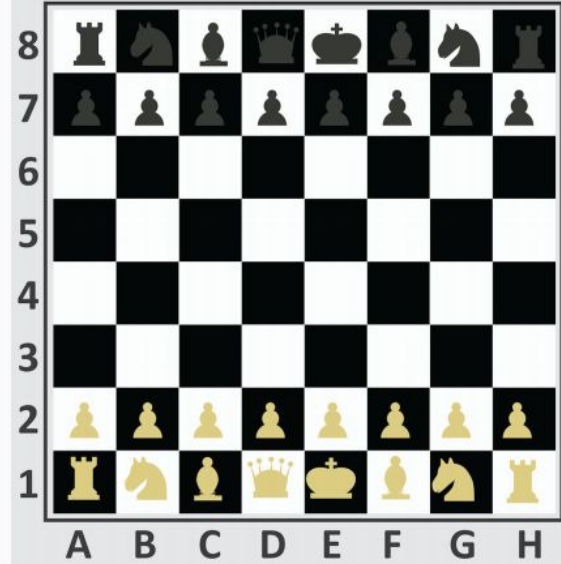
```
El valor de PI para i: 1 es: 3.16667
El valor de PI para i: 2 es: 3.13333
El valor de PI para i: 3 es: 3.14524
El valor de PI para i: 4 es: 3.13968
El valor de PI para i: 5 es: 3.14271
El valor de PI para i: 6 es: 3.14088
El valor de PI para i: 7 es: 3.14207
El valor de PI para i: 8 es: 3.14125
El valor de PI para i: 9 es: 3.14184
El valor de PI para i: 10 es: 3.14141
El valor de PI para i: 11 es: 3.14174
El valor de PI para i: 12 es: 3.14148
El valor de PI para i: 13 es: 3.14168
El valor de PI para i: 14 es: 3.14152
El valor de PI para i: 15 es: 3.14165
El valor de PI para i: 16 es: 3.14154
El valor de PI para i: 17 es: 3.14164
El valor de PI para i: 18 es: 3.14156
El valor de PI para i: 19 es: 3.14162
El valor de PI para i: 20 es: 3.14157
El valor de PI para i: 21 es: 3.14162
El valor de PI para i: 22 es: 3.14157
El valor de PI para i: 23 es: 3.14161
El valor de PI para i: 24 es: 3.14158
El valor de PI para i: 25 es: 3.14161
El valor de PI para i: 26 es: 3.14158
El valor de PI para i: 27 es: 3.1416
El valor de PI para i: 28 es: 3.14158
El valor de PI para i: 29 es: 3.1416
El valor de PI para i: 30 es: 3.14158
El valor de PI para i: 31 es: 3.1416
El valor de PI para i: 32 es: 3.14159
El valor de PI para i: 33 es: 3.1416
El valor de PI para i: 34 es: 3.14159
El valor de PI para i: 35 es: 3.1416
El valor de PI para i: 36 es: 3.14159
El valor de PI para i: 37 es: 3.1416
El valor de PI para i: 38 es: 3.14159
El valor de PI para i: 39 es: 3.1416
El valor de PI para i: 40 es: 3.14159
El valor de PI para i: 41 es: 3.1416
El valor de PI para i: 42 es: 3.14159
El valor de PI para i: 43 es: 3.1416
El valor de PI para i: 44 es: 3.14159
El valor de PI para i: 45 es: 3.1416
El valor de PI para i: 46 es: 3.14159
El valor de PI para i: 47 es: 3.14159
El valor de PI para i: 48 es: 3.14159
El valor de PI para i: 49 es: 3.14159
El valor de PI para i: 50 es: 3.14159
El valor de PI para i: 51 es: 3.14159
El valor de PI para i: 52 es: 3.14159
El valor de PI para i: 53 es: 3.14159
El valor de PI para i: 54 es: 3.14159
El valor de PI para i: 55 es: 3.14159
```

## Ejemplo 4:

Cada cuadrado de un tablero de ajedrez puede ser descrito por una combinación de una letra un número. Por ejemplo: G 5

El siguiente pseudocódigo describe el algoritmo para determinar si un cuadrado del tablero de ajedrez es de color blanco o negro, si se da como dato una letra y un número

Desarrolle un programa en C++ en que lea las coordenadas y el programa indique si corresponde a un cuadrado negro o blanco.



```
if la letra es A, C, E o G
    if el numero es impar
        color=" negro"
    else
        color=" blanco"
else
    if el numero es par
        color=" negro"
    else
        color=" blanco"
```

# Ejercicios



# Ejercicio 1:

Realiza un programa que indique si se está fuera de los límites de velocidad, dependiendo del lugar y el tipo de vía que se use. De acuerdo a la siguiente información:

## **Zona Urbana**

- En zona escolar: 30 km/h
- En calles y jirones: 40 km/h
- En avenidas: 60 km/h
- En vías expresas: 80 km/h

## **Carreteras**

- En caminos rurales: 60 km/h
- Para vehículos de carga: 80 km/h
- Para vehículos de servicio público de transporte de pasajeros: 90 km/h
- Para automoviles, camionetas y motocicletas: 100 km/h

## Ejercicio 2:

Escribir un programa en C++, que permita leer un número entre 1 y 30 e imprima un triángulo con asteriscos. El triángulo tendrá tantas filas como indica el número.

Número: 7

```
*  
**  
***  
****  
*****  
*****  
*****  
*****
```

Número: 9

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****
```

Número: 4

```
*  
**  
***  
****
```



# Ejercicio 3:

Desarrolle un programa que permita dibujar un árbol con números según la cantidad de filas que se indiquen como dato:

- El número de las filas debe ser mayor a 0 y menor a 10.
- El programa deberá verificar el ingreso del dato.
- El triángulo se construye imprimiendo **espacios en blanco** antes de empezar a imprimir los números en cada fila.

Ejemplos.

Filas [1 - 9] : 7

```
  1
 12
123
1234
12345
123456
1234567
```

Filas [1 - 9] : 9

```
  1
 12
123
1234
12345
123456
1234567
12345678
123456789
```

Filas [1 - 9] : 14

Filas [1 - 9] : -2

Filas [1 - 9] : 8

```
  1
 12
123
1234
12345
123456
1234567
12345678
```

# Ejercicio 4:

Diseñar y escribir un programa que solicite un número  $n$  par y que dibuje 2 rectángulos concéntricos donde el rectángulo externo es de lado  $n$  y el rectángulo interno es de lado 2.

Ejemplos:

Ingresa n: 4

```
* * * *  
* + + *  
* + + *  
* * * *
```

Ingresa n: 8

```
* * * * * * * *  
*               *  
*               *  
*       + +     *  
*       + +     *  
*               *  
*               *  
* * * * * * * *
```

Ingresa n: 10

```
* * * * * * * * * *  
*               *  
*               *  
*               *  
*       + +     *  
*       + +     *  
*               *  
*               *  
*               *  
* * * * * * * * * *
```

# Ejercicio 5:

El gerente de Starbucks, realizará un sondeo para determinar cuál de las siguientes bebidas, tiene mayor aceptación entre los turistas que visitan su local.

Las bebidas a considerar son:

- <M> Mango Frappuccino
- <F> Fresa Creme Frappuccino
- <V> Vainilla Creme Frappuccino

Realice un programa, que permita leer como dato el número de personas a las cuales se aplicará la encuesta (N, un número que debe estar en el rango de 5 a 30 inclusive). Luego permita realizar la encuesta y muestre los resultados de las preferencias expresados en porcentajes.

Al realizar la encuesta su programa deberá controlar que el usuario ingrese su preferencia únicamente utilizando las letras : M, m, F, f, V, v.

Vea el ejemplo de la ejecución del programa, en el siguiente slyde:

## Ejemplo:

Cuántas personas harán la encuesta? (entre 5 y 30)

5

¿Qué bebida prefiere?

Mango Frappuccino (M/m)

Fresa Creme Frappuccino (F/f)

Vainilla Creme Frappuccino (V/v)

m

¿Qué bebida prefiere?

Mango Frappuccino (M/m)

Fresa Creme Frappuccino (F/f)

Vainilla Creme Frappuccino (V/v)

m

¿Qué bebida prefiere?

Mango Frappuccino (M/m)

Fresa Creme Frappuccino (F/f)

Vainilla Creme Frappuccino (V/v)

v

¿Qué bebida prefiere?

Mango Frappuccino (M/m)

Fresa Creme Frappuccino (F/f)

Vainilla Creme Frappuccino (V/v)

v

¿Qué bebida prefiere?

Mango Frappuccino (M/m)

Fresa Creme Frappuccino (F/f)

Vainilla Creme Frappuccino (V/v)

f

40% prefieren Mango Frappuccino

20% prefieren Fresa Creme Frappuccino

40% prefieren Vainilla Creme Frappuccino

# Lo aprendido hoy:

- Diferencia entre una estructura selectiva y una estructura repetitiva
- Utilidad de las estructuras de control en un algoritmo

# Bibliografía:

Deitel. P.J. and Deitel. H. M. (2016) C++ How to Program, Prentice Hall.

Stroustrup, Bjarne (2013). The C++ Programming Language, 4th Addison-Wesley.

Eckel, Bruce, 2000. Thinking in C++, Vol 1: Introduction to Standard C++, 2nd Edition, Prentice Hall



¡Nos vemos en la  
siguiente clase!

