Asignatura:

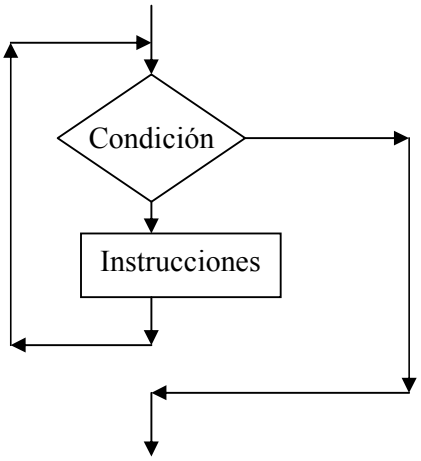
# Informática I

## Trabajo práctico Nº5

# While & do While

# Estructura repetitiva MIENTRAS o while

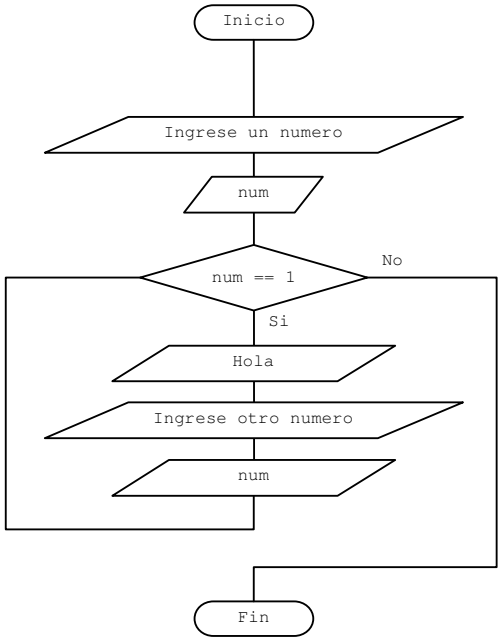
En C la sentencia while (condición o expresión) instrucciones; es seguramente la más utilizada. La sentencia, o grupo de sentencias o instrucciones se ejecutan mientras la evaluación de la expresión sea verdadera.




```
While (Cond)
{
        Instrucciones;
}
```

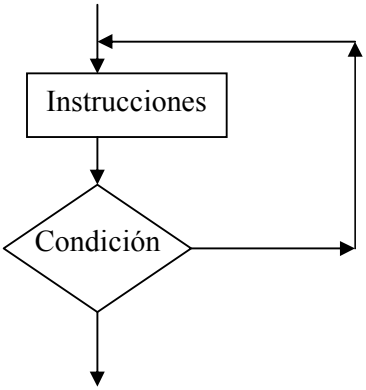
**EJEMPLO X:** Imprimir en pantalla la palabra "Hola", mientras se presione en número 1.

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

printf ("Ingrese un numero: ");

scanf("%d",&num);


while (num == 1)
        {

printf ("\n  Hola\n\n");

printf ("Ingrese otro numero: ");

scanf("%d",&num);

        }

system("PAUSE");
return 0;
}
```



## Estructura repetitiva HACER MIENTRAS o do while

Una variación de la sentencia while es: do instrucciones while ( condición o expresión ); En ella la sentencia se ejecuta al menos una vez, antes de que se evalúe la expresión condicional




```
do
{
        Instrucciones;
}
While (Condición)
```

**Ejemplo de aplicación:**

```
#include <stdio.h>

/* la función main comienza la ejecución del programa */
int main()
{
  int contador = 1;           /* inicializa el contador */

  do {
     printf( "%d  ", contador ); /* despliega el contador */
  } while ( ++contador <= 10 );   /* fin del do...while */

  return 0; /* indica la terminación exitosa del programa */

} /* fin de la función main */
```

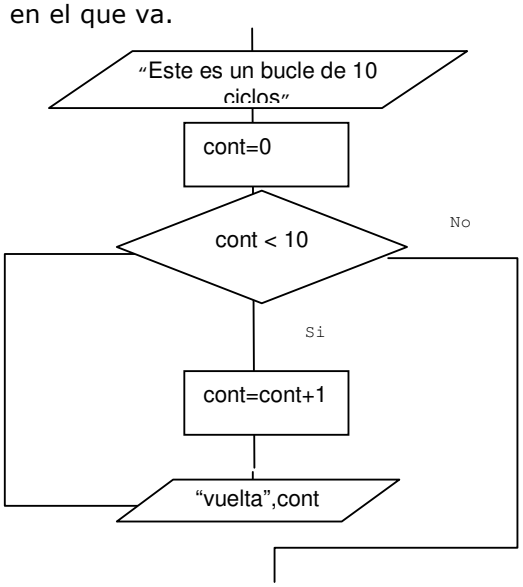
# Estructuras repetitivas

La mayoría de los problemas de programación requieren repetir un conjunto de acciones o sentencias, este ciclo debe repetirse cuando alguna condición de continuación de ciclo sea verdadera .Existen dos clases de repetición y por lo tanto algunas estructuras repetitivas se adecuan mejor a una u otra aplicación, estas son:

- Repetición definida, controlada por contador, cuando se sabe el número exacto de ciclos o de veces a repetir un bloque de acciones.(***Estructura Repetitiva While***)

En este caso se utiliza una variable de control para contar el numero de repeticiones. La variable de control se incrementa (por lo general en 1) cada vez que un grupo de instrucciones se ejecuta .Cuando el valor de la variable de control indica que el numero correcto de repeticiones se ha alcanzado, el ciclo termina.

Este lazo utiliza como contador la variable cont y muestra la palabra vuelta y el número de vuelta en el que va.

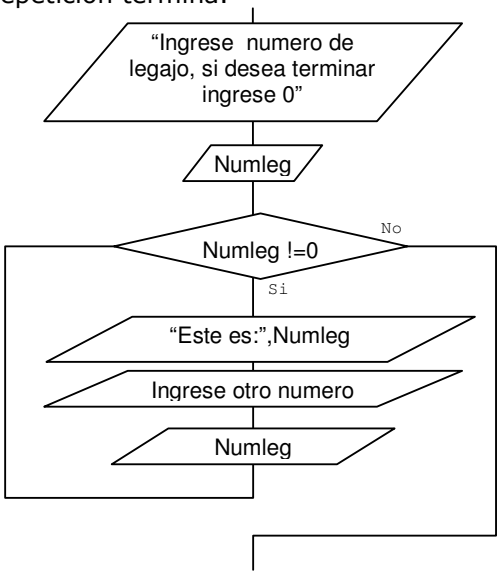



**Repetición controlada por contador**

```
#include <stdio.h>

/* la función main comienza la ejecución del programa */
int main()
{
  int contador = 0; /* inicialización */
  printf ( "Este es un buclede 10 ciclos o vueltas\n", contador );
  while ( contador < 10 ) { /* condición de repetición */
       contador++; /* incremento */
       printf ( "vuelta %d\n", contador ); / * despliega el contador */
       } /* fin del while */
  return 0;

} /* fin de la función main */
```

- Repetición indefinida, controlada por centinela cuando no se sabe el número exacto de ciclos o de veces a repetir un bloque de acciones. (***Estructuras Repetitivas: Mientras y Hacer Mientras).***

En este caso el valor centinela es cero y el programa muestra una cantidad indefinida de números de legajo ya que no existe entre los valores regulares un numero de legajo =0 cuando se quiera terminar basta con ingresar el valor centinela en la variable Numleg y la condición de ejecución dejara de cumplirse y el proceso de repetición termina.




En el caso 2 se utilizan valores centinela cuando, no conocemos por adelantado el numero preciso de repeticiones, así el ciclo incluye instrucciones de lectura de datos cada vez que el ciclo se ejecuta y estos datos son evaluados para ver si toman o no (el que haya sido elegido) el valor centinela, que indica fin de los datos regulares, el centinela siempre debe ser diferente de los valores regulares.

## Break y Continue

Otras sentencias interesantes, aunque menos utilizadas son break y continue. break provoca que se termine la ejecución de una iteración o para salir de la sentencia switch, como ya hemos visto. En cambio, continue provoca que se comience una nueva iteración, evaluándose la expresión de control. Veamos dos ejemplos en trozos de código:

**Ejemplo de aplicacion de break**

```
Cont = 10;
while ( Cont >= 1 )
{
    Cont=Cont -1
    if ( Cont == 4 )
       break;
   printf("%d ", count );
}
```

**Ejemplo de aplicacion de Continue**

```
Cont = 10;
while ( Cont >= 1 )
{
    Cont=Cont -1
    if ( Cont == 4 )
       continue;
   printf("%d ", count );
}
```

## Trabajo Practico numero 5

**Actividad 1:**
Codificar en C y ejecutar cada uno de los ejemplos en el Laboratorio de informática en el caso de los trozos completarlos y correrlos modificar de lugar los incrementos ¿que pasa?

**Actividad 2:**
Modificarlos para que en los dos casos de repetición se utilize do while.

**Actividad 3:**
Diseñar un programa que ingrese las notas y legajos de un curso por teclado utilizando los controles de datos mediante estructuras repetitivas y determine cantidad de aprobados reprobados y promocionados imprimir su legajo con una leyenda A o R o P y el operador deberá decidir cuando termina el ingreso de datos ingresando un legajo negativo
Diagrama de flujo y pseudocódigo

**Actividad 4:**
Codificar en C el ejercicio anterior.
en el Laboratorio de informática

**Actividad 5:**

**¿Que realizan los siguiente programa?**
**comentarlos**

```
#include <stdio.h>

/* la función main inicia la ejecución del programa */
int main()
{
   int aprobados = 0;
   int reprobados = 0;
   int estudiante = 1;  /* */
   int resultado;      /* */

   /* */
   while ( estudiante <= 10 ) {

      /*  */
      printf( "Introduzca el resultado ( 1=aprobado,2=reprobado ): " );
      scanf( "%d", &resultado );

      /*  */
      if ( resultado == 1 ) {
         aprobados = aprobados + 1;
      }
      else { /* */
         reprobados = reprobados + 1;
      }
      estudiante = estudiante + 1;   } /* fin de while */

   /* */
   printf( "Aprobados %d\n", aprobados );
   printf( "Reprobados %d\n", reprobados );

   /* */
   if ( aprobados > 8 ) {
      printf( "Objetivo alcanzado\n" );
   } /* fin de if */

   return 0; }
```

**¿Qué imprime este programa? */**

```
#include <stdio.h>

/* la función main inicia la ejecución del programa */
int main()
{
   int fila = 10; /* inicializa la fila */
   int columna;   /* define la columna */

   while ( fila >= 1 ) { /* repite el ciclo hasta que fila < 1 */
      columna = 1;       /* establece la columna en 1 al comenzar la
iteración */

      while ( columna <= 10 ) {             /* repite 10 veces */
         printf( "%s", fila % 2 ? "<": ">" ); /* salida */
         columna++;                         /* incrementa la columna */
      } /* fin del while interno */

      fila--;         /* decrementa la fila */
      printf( "\n" ); /* comienza la nueva línea de salida */
   } /* fin del while externo */

   return 0; /* indica que el programa terminó con exito */

} /* fin de la función main */
```

**¿Qué imprime este programa? */**

```
#include <stdio.h>

/* la función main inicia la ejecución del programa */
int main()
{
   int contador = 1; /* inicializa contador */

   while ( contador <= 10 ) { /* repite 10 veces */

      /* muestra una línea de texto */
      printf( "%s\n", contador % 2 ? "****" : "++++++++" );
      contador++; /* incrementa contador */
   } /* fin de while */

   return 0; /* indica que el programa terminó con exito */

} /* fin de la función main */
```

```c
#include <stdio.h>

int main()
{
  int x = 1, total = 0, y;


  while ( x <= 10 ) {
    y = x * x;
    printf( "%d\n", y );
        total += y;
        ++x;
  }

  printf( "El total es: %d\n", total );

  return 0;
```

---

```c
#include <stdio.h>

int main()
{
  int x, y, i, potencia;     /* definición de las variables */

  i = 1;
  potencia = 1;      /* inicializa la potencia */
  scanf( "%d", &x ); /* lectura de x del usuario */
  scanf( "%d", &y ); /* lectura de y del usuario */

  while ( i <= y ) { /* repetirá el ciclo while mientras i sea menor o
igual a y */
    potencia *= x;  /* multiplica potencia por x */
    ++i;           /* incrementa i */

  } /* fin del while */

  printf( "%d\n", potencia ); /* despliega la potencia */

  return 0;

} /* fin de la función main */
```

**¿Qué tipo de repetición es?**

```
#include <stdio.h>

/* la función main inicia la ejecución del programa */
int main()
{
   int contador;     /* número de calificaciones introducidas */
   int calificacion; /* valor de la calificación */
   int total;        /* suma de las calificaciones */

   float promedio;   /* número con punto decimal para el promedio */

   /* fase de inicialización */
   total    = 0; /* inicializa el total */
   contador = 0; /* inicializa el contador del ciclo */

   /* fase de proceso */
   /* obtiene la primera calificación del usuario */
   printf( "Introduzca la calificacion, -1 para terminar: " ); /* indicador
para la entrada */
   scanf( "%d", &calificacion ); /* lee la calificación del usuario */

   /* realiza el ciclo mientras no se introduzca el valor centinela */
   while ( calificacion != -1 ) {
      total = total + calificacion; /* suma calificación a total */
      contador = contador + 1; /* incrementa el contador */

      /* obtiene la siguiente calificación del usuario */
      printf( "Introduzca la calificacion, -1 para terminar: " ); /*
indicador para la entrada */
      scanf("%d", &calificacion); /* lee la siguiente calificación */
   } /* fin de while */

   /* fase de terminación */
   /* si el usuario introduce al menos una calificación */
   if ( contador != 0 ) {

      /* calcula el promedio de todas las calificaciones introducidas */
      promedio = ( float ) total / contador; /*evita la truncación*/

      /* despliega el promedio con dos dígitos de precisión */
      printf( " El promedio de la clase es: %.2f\n", promedio );
   } /* fin de if*/
   else { /* si no se introduce calificación alguna, despliega el
mensaje */
      printf( "No se introdujeron calificaciones\n" );
   } /* fin de else */
```

```
   return 0; /* indica que el programa terminó con exito */
} /* fin de la función main */
```