

# Clase 2

Viernes, 15 Septiembre 2023

---

[Índice](#)

## Tipos de datos

- Los reales se definen con un 'double'. También es posible usar un float, en caso justificado.
- '**Double**' es un tipo de dato de 64 bits.
- '**Float**' es un tipo de dato de 32 bits. La GPU trabaja con floats

## Lenguajes y sus generaciones:

- **Primera generación**, lenguaje máquina:  
Cada computadora tiene sólo un lenguaje de programación que su procesador puede ejecutar; pues bien, éste es su lenguaje nativo o lenguaje de máquina. Los programas en lenguaje máquina se escriben en el nivel más básico de operación de la computadora. Las instrucciones se codifican como una serie de unos '1' y ceros '0'.
- **Segunda generación**, lenguaje ensamblador:  
Para evitar que los programadores tuvieran que programar directamente en código binario o máquina, se desarrollaron unos programas para traducir instrucciones a código de máquina. Estos programas se llamaron ensambladores, puesto que leían las instrucciones que las personas podían entender en lenguaje ensamblador y las convertía al lenguaje máquina. El lenguaje ensamblador también es de bajo nivel, ya que cada instrucción de este lenguaje corresponde a una instrucción de lenguaje máquina.
- **Tercera generación**, lenguaje de alto nivel:  
Estos lenguajes son parecidos al inglés y facilitan el trabajo de los desarrolladores de software. Existen muchos lenguajes de tercera generación como, por ejemplo, COBOL, BASIC, FORTRAN, C, PASCAL, etc. Con estos lenguajes, los programadores pueden escribir en una sola instrucción lo equivalente a varias instrucciones complicadas de bajo nivel.
- **Cuarta generación**, lenguaje orientado al usuario (4GL):  
Con los lenguajes 4GL, los usuarios finales escriben sus programas de manera sencilla para consultar una base de datos y para crear sistemas de información personales o departamentales. Muchos de estos lenguajes disponen de una interfaz gráfica y sólo obligan al usuario o programador a usar instrucciones sencillas y fáciles de manejar.
- **Quinta generación**, lenguajes naturales:  
Los lenguajes naturales se asemejan más al lenguaje humano que sus antecesores, los lenguajes 4GL. Aunque estos lenguajes se encuentran en sus inicios, la mayoría de las herramientas de uso y trabajo con el ordenador tenderán a este tipo de lenguajes.

## Conocimientos básicos

- **Jerarquía de funciones de clase:**

- Declaración de variables
- Asignaciones
- Condicionales
- Bucles
- Rupturas
- Retornos

- Declaración de variables:

```
,  
int a;  
int b;  
int a, b;  
,
```

- Asignación de variables:

```
,  
a = 5;  
int b = 12;  
,
```

- Condicionales:

```
,  
if(condicion)  
{  
    código ejecutable  
}  
else  
{  
    código ejecutable  
}  
,
```

- Bucles:

```
,  
// While, en caso que NO conozcamos el numero de iteraciones  
while(condicion)  
{  
    código ejecutable  
    actualizar condicion  
}  
  
// For, en caso que SI conozcamos el numero de iteraciones  
for(int i = 0; i < n; i++)  
{  
    código ejecutable en un bucle de n veces  
}  
,
```

- Rupturas:

```
,  
break;  
// Break, hace saltar la línea de compilación hasta el cierre del cuerpo '}'  
  
continue;  
// Continue, hace saltar la línea de compilación hasta el inicio del bucle  
,
```

- Retornos

```
'return [tipo de dato];'  
// termina el metodo o funcion y devuelve el valor indicado.
```

- **Jerarquía de funciones de objeto:**

- Enumerations
- Atributos
- Constructores
- Getters and setters
- Metodos

### Conocimiento básico de funciones

- **FUNCION:** Realiza la suma de dos numeros que se le pasan por parametros

```
,
public class Functions
{
    public static int CalculateSum(int number1, int number2)
    {
        // Opcion 1
        return number1 + number2;

        // Opcion 2
        int result;
        result = number1 + number2;
        return result;
    }
}
```

- Para invocar a una función escribimos el nombre de la clase, seguida de un punto y el nombre de la función.

```
,
public class Program
{
    public static void Main(args)
    {
        // Declarar variables
        int a, b;

        // Asignar valores
        a = 5;
        b = 10;

        // Invocar la funcion de suma de numeros
        Functions.CalculateSum(a,b);
    }
}
```

### Consejos y advertencias

- Existe una guía de estilo para dar formato al código y para nombrar a las distintas funciones. Cada grupo de trabajo tiene su propia guía de estilo. Habitualmente las funciones se nombran con '**Verbo+Sustantivo**' ej: 'CalculateSum' o 'GetIndexAt'.
- Cuando dentro de un *cuerpo* (codigo fuente que se escribe entre {}) solo hay una línea de instrucción, es posible eliminar los indicadores {}.
- Una función debe ocupar como máximo una pantalla del ordenador, y debemos evitar el scroll vertical.
- **JAMÁS** ejecutaremos un 'Console.WriteLine()' dentro del funcionamiento de un método.
- **JAMÁS** se debe escribir una sentencia 'return' dentro de un bucle IF