



Implementación de las clases en C#



TEMA 2

MOTIVACIÓN

Persona

cadena nombre

cadena apellidos

entero añoNacimiento

entero Edad(entero unAño)

cadena ToString()



OBJETIVO

Caracterizar la construcción de clases en C#.





TEMÁTICA

- I. Estructura de una clase en C#.
- II. Definición y uso de constantes.
- III. Sobrecarga de métodos.
- IV. Manejo de instancias. Constructores.
- V. Semántica de Referencia vs Semántica de Valor.
- VI. Casos de estudio.

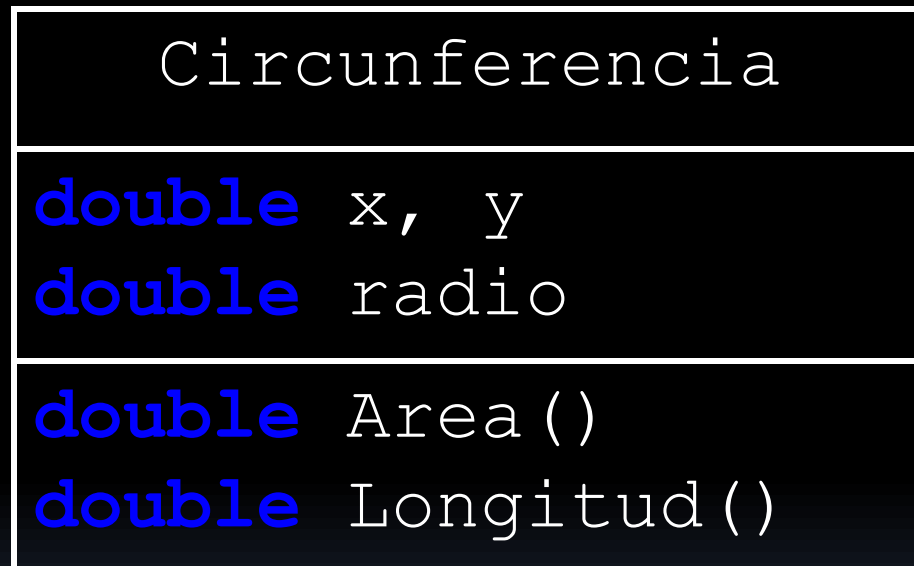
CLASE PERSONA EN C#

```
class Persona
{
    string nombre;
    string apellidos;
    int añoNacimiento;

    int Edad(int unAño)
    {
        return unAño - añoNacimiento;
    }

    string ToString()
    {
        return nombre + " " + apellidos;
    }
}
```

CLASE CIRCUNFERENCIA EN UML



CLASE CIRCUNFERENCIA EN C# 1

```
class Circunferencia
{
    double x, y;
    double radio;

    double Area()
    {
        return 3.14159 * radio * radio;
    }
    double Longitud()
    {
        return 2 * 3.14159 * radio;
    }
}
```

CLASE CIRCUNFERENCIA EN C# 2

```
class Circunferencia
{
    double x, y;
    double radio;
    const double PI = 3.14159;

    double Area()
    {
        return PI * radio * radio;
    }
    double Longitud()
    {
        return 2 * PI * radio;
    }
}
```


CLASE CIRCUNFERENCIA EN C# 3

```
class Circunferencia
{
    double x, y;
    double radio;

    double Area()
    {
        return Math.PI * radio * radio;
    }

    double Longitud()
    {
        return 2 * Math.PI * radio;
    }
}
```

MÉTODOS. SOBRECARGA (1)

```
class Persona
{
    string nombre;
    string apellidos;
    int añoNacimiento;

    int Edad(int unAño) {...}

    int Edad()
    {
        return System.DateTime.Now.Year -
            añoNacimiento;
    }
    ...
}
```

MÉTODOS. SOBRECARGA (2)

```
{  
  
    ...  
    juan.Edad(2003);  
  
    ...  
  
    juan.Edad();  
}
```

MANEJO DE LAS INSTANCIAS

```
{  
  
    Persona maría;  
  
    maría = new Persona();  
  
    // Persona maría = new Persona();  
  
    maría.nombre = "María";  
  
    maría.apellidos = "González";  
  
    maría.añoNacimiento = 1980;  
  
}
```

CONSTRUCTORES (1)

```
class Persona
```

```
{
```

```
    string nombre;
```

```
    string apellidos;
```

```
    int añoNacimiento;
```

```
    ...
```

```
Persona(string Nomb, string Ap, int AN)
```

```
{
```

```
    nombre = Nomb;
```

```
    apellidos = Ap;
```

```
    añoNacimiento = AN;
```

```
}
```

```
}
```

CONSTRUCTORES (2)

```
Persona maría =  
    new Persona("María", "González", 1980);
```

CONSTRUCTORES (3)

```
class Persona
{
    string nombre;
    string apellidos;
    int añoNacimiento;
    ...
    Persona(string nombre,
            string apellidos,
            int añoNacimiento )
    {
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.añoNacimiento =
            añoNacimiento;
    }
}
```

REFERENCIA VS VALOR (1)

```
{  
    Persona juan =  
        new Persona("Juan", "Ferrer", 1980);  
  
    Persona pedro = juan;  
  
    pedro.nombre = "Pedro";  
  
    Console.WriteLine(pedro.nombre);  
  
    Console.WriteLine(juan.nombre);  
}
```


REFERENCIA VS VALOR (2)

```
{
```

```
    uint edadJuan = 20;
```

```
    uint edadPedro = 50;
```

```
    edadJuan = edadPedro;
```

```
    edadJuan++;
```

```
}
```

REFERENCIA VS VALOR (3)

```
{  
    string s1 = "Hola";  
  
    string s2 = s1;
```

```
    Console.WriteLine(s2);
```

```
}
```

CLASE TEMPERATURA

Diseñe e implemente una clase Temperatura con responsabilidades para almacenar un valor de temperatura en grados Celsius (C) y obtener su correspondiente en grados Fahrenheit (F) ($F = C * 9 / 5 + 32$).

CLASE TEMPERATURA UML

Temperatura

double celsius

Temperatura(**double** unaTempCelsius)

double Fahrenheit()

CLASE TEMPERATURA C#

```
class Temperatura
```

```
{  
    double celsius;
```

```
    Temperatura(double unaTempCelsius)  
    {  
        celsius = unaTempCelsius;  
    }
```

```
    double Fahrenheit()  
    {  
        return celsius * 9 / 5 + 32;  
    }
```

```
}
```



Conclusiones