

Instituto Tecnológico de Las Américas



Nombres: Emil Pérez Barranco

Matrícula: 2018 – 6076

Carrera: Tecnología en Desarrollo de Software

Período: 2019-C-1

Profesor: Ángel Robles

Asignatura: Fundamentos de Programación

Trabajo: Bucles, Excepciones y Recursión

Estructuras de Repetición

Bucle **While**: Este bucle ejecuta instrucciones mientras la condición dada sea verdadera.

```
// while Loops
int firstNum = 0;
while (firstNum < 6)
{
    Console.WriteLine("\n ~~ while Loops ~~");
    firstNum++;
}
```

Bucle **Do-While**: Este bucle ejecuta una función dada antes de la evaluación del While, después de esta evaluación, el While ejecuta sus funciones dadas.

```
// do-while Loops
string doCond = "Ejecución do {}";
int Count = 0;
do {
    Console.WriteLine("\n" + doCond);
    Console.WriteLine("\n ~~~ do-while Loops ~~~");
    Count++;
} while (Count < 6);
```

Bucle For: En este bucle, se inicializa una variable, luego se impone una condición, y después se marcan los pasos a cumplir.

```
// for Loops
string[] firstArray = { "Alpha", "Bravo", "Delta", "Charlie" };
for (int i = 0; i < firstArray.Length; i++)
{
    Console.WriteLine("Tipo: " + firstArray[i]);
}
```

Bucle ForEach: Este bucle ejecuta una instrucción o un bloque de instrucciones para cada elemento en una instancia del tipo que se implementa.

```
//foreach Loops
int[] firstInt = { 22, 98, 45, 73, 53 };
foreach(int i in firstInt)
{
    Console.WriteLine("Número de la Suerte: " + i);
}
```

Serie de Fibonacci

No es otra cosa que una serie especial de número donde el siguiente de la lista se obtiene sumando sus dos predecesores ejemplo: 1,1,2,3,5,8,13... Ejemplo en C#:

```
// Serie Fibonacci
// Este algoritmo desarrolla la serie Fibonacci dependiendo de cuantos se requieran
int A, B, N, i, Aux; // Declaramos las variables
Console.WriteLine("\n Introduzca la cantidad de números que requiere: ");
N = int.Parse(Console.ReadLine()); // Leemos el número introducido
A = 0;
B = 1; // Inicializamos
for (i = 0; i < N; i++) // Se repite para el número de veces deseado
{
    Aux = A;
    A = B; // Intercambio valores
    B = Aux + A; // Calculo el nuevo valor
    Console.WriteLine(Aux); // Imprimo el valor
}
```

El M.C.D.

Es el mayor número que divide exactamente a dos o más números. Aquí tenemos implementado el MCD en C#:

```
// MCD en C#
// Leemos los dos numeros de donde sacaremos el MCD
Console.WriteLine("\n Ingrese el primer numero: ");
int Num1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("\n Ingrese el segundo numero: ");
int Num2 = Convert.ToInt32(Console.ReadLine());

// Inicializando las variables del numero mayor y el numero menor
int X = Math.Max(Num1, Num2);
int Z = Math.Min(Num1, Num2);

// Inicializamos la variable del resultado
int Res;

// Bucle de la operacion
do
{
    Res = Z; // Guardamos el divisor en el resultado
    Z = X % Z; // Guardamos el numero restante en el divisor
    X = Res; // El divisor pasa al dividendo
} while(Z != 0);

// Resultado del M.C.D.
Console.WriteLine("\n El MCD entre " + Num1 + " y " + Num2 + " es: " + Res);
```

El Factorial de un Número

Se define en principio como el producto de todos los números enteros positivos desde 1. *Ej.: $3 \times 2 \times 1 = 6$* . A continuación tenemos un algoritmo que puede hallar el factorial de un número.

```
// Factorial de un Numero
Console.WriteLine("\n ~ Introduzca el numero del que sea extraer el Factorial: ");
int disNum = int.Parse(Console.ReadLine());
int indexNum = 1;
for (int facNum = disNum; facNum > 0; facNum--)
{
    indexNum = facNum * indexNum;
}
Console.WriteLine("\n ~ Resultado: " + indexNum);
```

La Recursión en C#

Un algoritmo es recursivo cuando expresa la solución de un problema en términos de una llamada a sí mismo. En C#:

```
static void Main(string[] args)
{
    // Factorial en Modo Recursivo
    int Fact = 0;
    Fact = Factorial(5);
    Console.WriteLine("El Factorial es: " + Fact);
}

private static int Factorial(int factNum)
{
    if (factNum == 0)
        return 1;
    return factNum * Factorial(factNum - 1);
}
```

Las Excepciones en C#

Las excepciones son un mecanismo que podemos usar en C# para reportar errores en nuestros programas y poder manejarlos. Ex:

```
static void Exceps()
{
    // Division con detección de errores
    Console.WriteLine("\n ~ Digite el Dividendo: ");
    int num1 = int.Parse(Console.ReadLine());
    Console.WriteLine("\n ~ Digite el Divisor: ");
    int num2 = int.Parse(Console.ReadLine());
    try
    {
        int result;
        result = num1 / num2;
        Console.WriteLine("\n {0} dividido por {1} = {2}", num1, num2, result);
    }
    catch (DivideByZeroException e)
    {
        Console.WriteLine("\n No puedes dividir por 0.");
        Console.WriteLine("_____");
        Exceps();
    }
}
```

Como ejemplo del control de excepciones tenemos el **DivideByZeroException**.

Al intentar dividir un número entero o decimal por cero, se produce una excepción **DivideByZeroException**. Para evitar la excepción, debemos asegurarnos de que el denominador en una operación de división con valores enteros o decimales no sea cero.

Manejo de Las Excepciones

La jerarquía en el catch

```
try
{
    int Ex = 0;
    Console.WriteLine(50 / Ex);
}
catch (ArithmeticException)
{
    Console.WriteLine("Error Aritmetico...");
}
catch (Exception e)
{
    Console.WriteLine("Controlador Genérico de Excepciones: {0}", e.ToString());
}
```

Es en el catch en donde es importante la jerarquía de las excepciones. Por ejemplo, en el código anterior, el bloque `catch(ArithmeticException)` atrapará tanto las de tipo:

- `NotFiniteNumberException`
- `OverflowException`
- `DivideByZeroException`

Porque estas heredan de `ArithmeticException`. Los demás errores inesperados serán atrapados por el bloque `catch(Exception e)`.

Usando `throw/finally` para lanzar excepciones

La forma de lanzar una excepción es usando la palabra `throw` seguido de la instancia de excepción que vamos a lanzar, culminando con `finally`:

```
FileStream fs = null;
try
{
    //Abre un archivo
    fs = new FileStream(@"C:\temp\data.txt", FileMode.Open);
    StreamReader sr = new StreamReader(fs);
    string line;

    //Se lee un valor del archivo y es proyectado a la consola.
    line = sr.ReadLine();
    Console.WriteLine(line);
}
catch (FileNotFoundException e)
{
    Console.WriteLine("[Archivo no Existente] {0}", e);
    throw new FileNotFoundException(@"[data.txt not in c:\temp directory]", e);
}
finally
{
    if (fs != null)
        fs.Close();
}
```


Tareas Practicas:

- 1.** Crear un programa que solicite el año de nacimiento de una persona y retorne la edad haciendo uso de funciones o procedimientos.
- 2.** Cree un programa que contenga el siguiente menú:
 - a)** Calcular el Factorial de un Numero. *Debe de Pedir un Numero para calcular el Factorial.
 - b)** Obtener MCD. *Debe de pedir dos números para calcular el MCD.
 - c)** Cálculo de IMC (Índice de Masa Corporal). Investigar que se necesita para calcular el IMC y pedir los datos necesarios.
 - d)** Salir.

****Nota el Programa debe de ser cíclico utilizando recursividad. Utilice métodos para hacer cada una de las funciones especificadas. utilice todo lo que se ha estado viendo en clase hasta el momento. Si es posible repase todo el contenido y lo mandado a investigar hasta el momento.**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CalcsApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Menu();
        }

        static void Menu() {
            // Sets Básicos
            Console.Clear();
            Console.Title = "CalcsApp";
            Console.WriteLine("~~~~~ CalcsApp | 186076 | 06/03/2019 ~~~~~");
            Console.WriteLine("\n ~ Escoja la Función:");
            Console.WriteLine("\n 1. Calculo de Edad");
            Console.WriteLine("\n 2. Factorial");
            Console.WriteLine("\n 3. Máximo Común Divisor");
            Console.WriteLine("\n 4. Índice de Masa Corporal");
            Console.WriteLine("\n 5. Salir");
            Console.WriteLine("\n -----");
            Console.Write("\n ~ Herramienta: ");
            // Variables de Introducción de Datos
            int Ask = int.Parse(Console.ReadLine());

            try
            {
                switch (Ask)
                {
                    case 1:
                        CalcEdad();
                        break;
                    case 2:
                        Factorial();
                        break;
                    case 3:
                        MCD();
                        break;
                    case 4:
                        IMC();
                        break;
                    case 5:
                        Console.WriteLine("\n ~ Saliendo....");
                        Console.ReadKey();
                        break;
                    default:
                        Console.WriteLine("\n ~ Ocurrio un error, sera enviado al
menu...");
                        Menu();
                        break;
                }
            }
        }
    }
}

```

```

        catch (ArithmeticException)
        {
            Menu();
        }
        catch (FormatException)
        {
            Menu();
        }
        catch (Exception ex)
        {
            Menu();
        }
    }

    // Metodo para calcular la edad
    static void CalcEdad()
    {
        Console.WriteLine("\n ~ Inserta tu fecha de nacimiento para Calcular tu Edad: ");
        // Variable de tipo Fecha que contiene el Año de Nacimiento (input)
        DateTime añoNac = DateTime.Parse(Console.ReadLine());
        // Variable de tipo Fecha que contiene el Año de Nacimiento (converted from
input)
        DateTime BirthDay = Convert.ToDateTime(añoNac);
        // Variable que contiene la Fecha de hoy
        DateTime Today = DateTime.Today;
        int Age = Today.Year - BirthDay.Year;
        // Si el año de nacimiento es mayor
        if (BirthDay > Today.AddYears(-Age))
            Age--;
        Console.WriteLine("\n ~ Tu edad es de: " + Age + " años.");
        Console.WriteLine("\n -----");
        // Metodo de repeticion de la herramienta con recursividad
        Console.WriteLine("\n Desea volver a usar la herramienta Calculo de Edad? ");
        string rptFunc = Console.ReadLine();
        if (rptFunc == "Si" || rptFunc == "si" || rptFunc == "s" || rptFunc == "S")
            CalcEdad();
        Menu();
    }

    // Metodo para sacar el Factorial de un numero
    static void Factorial()
    {
        // Factorial de un Numero
        Console.WriteLine("\n ~ Introduzca el numero del que sea extraer el Factorial:
");
        try
        {
            int disNum = int.Parse(Console.ReadLine());
            int indexNum = 1;
            for (int facNum = disNum; facNum > 0; facNum--)
                indexNum = facNum * indexNum;
            int Resp = indexNum;
            Console.WriteLine("\n ~ Resultado: " + Resp);
        }
        catch (FormatException)
        {
            Console.WriteLine("#### Debe introducir el numero de la funcion que desea
emplear! ####");
        }
    }
}

```

```

        Factorial();
    }

    Console.WriteLine("\n -----");
    // Metodo de repeticion de la herramienta con recursividad
    Console.Write("\n Desea volver a usar la herramienta Factorial? ");
    string rptFunc = Console.ReadLine();
    if (rptFunc == "Si" || rptFunc == "si" || rptFunc == "s" || rptFunc == "S")
        Factorial();
    Menu();
}

//Metodo para extraer el Maximo Comun Divisor
static void MCD()
{
    // MCD en C#
    // Leemos los dos numeros de donde sacaremos el MCD
    Console.Write("\n Ingrese el primer numero: ");
    int Num1 = Convert.ToInt32(Console.ReadLine());
    Console.Write("\n Ingrese el segundo numero: ");
    int Num2 = Convert.ToInt32(Console.ReadLine());

    // Inicializando las variables del numero mayor y el numero menor
    int X = Math.Max(Num1, Num2);
    int Z = Math.Min(Num1, Num2);

    // Inicializamos la variable del resultado
    int Res;

    // Bucle de la operacion
    do
    {
        Res = Z; // Guardamos el divisor en el resultado
        Z = X % Z; // Guardamos el numero restante en el divisor
        X = Res; // El divisor pasa al dividendo
    } while (Z != 0);
    // Resultado del M.C.D.
    Console.WriteLine("\n El MCD entre " + Num1 + " y " + Num2 + " es: " + Res);
    Console.WriteLine("\n -----");
    // Metodo de repeticion de la herramienta con recursividad
    Console.Write("\n Desea volver a usar la herramienta MCD? ");
    string rptFunc = Console.ReadLine();
    if (rptFunc == "Si" || rptFunc == "si" || rptFunc == "s" || rptFunc == "S")
        MCD();
    Menu();
}

static void IMC()
{
    // El Indice de Masa Corporal ideal/normal se encuentra entre 18.5 y 25 ~
    //  $\text{Peso (kg)} / \text{Altura} * \text{Altura (m)}$ 
    // Variables sin asignar de Altura, Peso y a lo que equivaldra el IMC
    double Altura, Peso, InMC;
    Console.Write("\n ~ Digite su Peso: ");
    Peso = double.Parse(Console.ReadLine());
    Console.Write("\n ~ Digite su Altura: ");
    Altura = double.Parse(Console.ReadLine());

```

```

// El IMC sera igual al peso dividido por el cuadrado de la altura
InMC = (Peso / (Altura * Altura));
// Display del calculo total del IMC
Console.WriteLine("\n ~ El Indice de Masa Corporal es de: "+ InMC);

//Clasificaciones del IMC dependiendo del resultado obtenido
if (InMC < 18.5) Console.WriteLine("\n ~ Peso por debajo de lo normal");
else if ((InMC == 18.5) || (InMC == 25)) Console.WriteLine("\n ~ Peso
normal");

if ((InMC > 25) || (InMC == 30)) Console.WriteLine("\n ~ Sobre Peso");
else if ((InMC > 30) || (InMC == 35)) Console.WriteLine("\n ~ Grado de
Obesidad I");

if ((InMC > 35) || (InMC == 40)) Console.WriteLine("\n ~ Grado de Obesidad
II");

else if (InMC > 40) Console.WriteLine("\n ~ Grado de Obesidad III");

Console.WriteLine("\n -----");
// Metodo de repeticion de la herramienta con recursividad
Console.Write("\n Desea volver a usar la herramienta MCD? ");
string rptFunc = Console.ReadLine();
if (rptFunc == "Si" || rptFunc == "si" || rptFunc == "s" || rptFunc == "S")
    Program.IMC();
Menu();
}

}
}

```

Anexo: Programa en Ejecución

```

CalcsApp
~~~~~ CalcsApp | 186076 | 06/03/2019 ~~~~~

~ Escoja la Función:
1. Calculo de Edad
2. Factorial
3. Máximo Común Divisor
4. Índice de Masa Corporal
5. Salir

-----

~ Herramienta: 1
~ Inserta tu fecha de nacimiento para Calcular tu Edad: 11/22/2000
~ Tu edad es de: 18 años.

-----

Desea volver a usar la herramienta Calculo de Edad?

```

```
CalcsApp
~~~~~ CalcsApp | 186076 | 06/03/2019 ~~~~~
~ Escoja la Función:
1. Calculo de Edad
2. Factorial
3. Máximo Común Divisor
4. Índice de Masa Corporal
5. Salir
-----
~ Herramienta: 2
~ Introduzca el numero del que sea extraer el Factorial: 7
~ Resultado: 5040
-----
Desea volver a usar la herramienta Factorial?
```

```
CalcsApp
~~~~~ CalcsApp | 186076 | 06/03/2019 ~~~~~
~ Escoja la Función:
1. Calculo de Edad
2. Factorial
3. Máximo Común Divisor
4. Índice de Masa Corporal
5. Salir
-----
~ Herramienta: 3
Ingrese el primer numero: 6
Ingrese el segundo numero: 16
El MCD entre 6 y 16 es: 2
-----
Desea volver a usar la herramienta MCD?
```

```
CalcsApp
~~~~~ CalcsApp | 186076 | 06/03/2019 ~~~~~
~ Escoja la Función:
1. Calculo de Edad
2. Factorial
3. Máximo Común Divisor
4. Índice de Masa Corporal
5. Salir
-----
~ Herramienta: 4
~ Digite su Peso: 90
~ Digite su Altura: 1.77
~ El Índice de Masa Corporal es de: 28.7273771904625
~ Sobre Peso
-----
Desea volver a usar la herramienta MCD?
```