

Facultad: Ingeniería
Escuela: Computación
Asignatura: Programación Orientada a Objetos

GUIA 1:

Rapso sobre uso de C#. Funciones, procedimientos y arreglos.

Competencia

Desarrolla sistemas de información informáticos mediante la integración de principios matemáticos, ciencia computacional y prácticas de ingeniería, considerando estándares de calidad y mejores prácticas validadas por la industria del software

Introducción

Una de las formas de resolver un problema complejo por medio de programación, es recurrir a la segmentación del mismo en pequeños sub problemas más sencillos. Estos segmentos de código que poseen independencia en su compilación con respecto a los otros reciben el nombre de **Módulos** y su implementación se realiza por medio de **Funciones o Procedimientos**.

Los módulos tienen como objetivo resolver un problema concreto y aportar a la solución general del problema, se distingue por tres elementos principales entrada, proceso y salida.

En el momento de la invocación, se deben de dar parámetros que son los valores que procesará el módulo; internamente se define el proceso que realizará con los elementos recibidos (una tarea específica) y finalmente devuelve el resultado o el control al programa que lo llamó.

Para el lenguaje de programación C# que es un lenguaje nativo orientado a objetos, estas funciones y procedimientos reciben el nombre de métodos.

FUNCIONES

Es el conjunto de secuencia de órdenes que hacen una tarea específica para una aplicación más grande.

La declaración de las funciones se distingue por:

- Nombre único en el ámbito. Nombre de la función que se distingue y se identifica con respecto a otras. Este mismo nombre puede ser utilizado únicamente en caso de sobrecarga o de polimorfismo (POO)
- Tipo de dato de retorno. Tipo de dato que será devuelto al terminar la rutina.
- Lista de parámetros. Conjunto de argumentos (desde cero hasta varios) que la función debe recibir para procesarlos en la rutina.

```
Modificador_de_acceso Tipo_Devuelto Nombre_Función (tipo(s)_argumento(s) nombres)
{
    //declaración de datos y cuerpo de la función.
    return (valor)
}
```

PROCEDIMIENTOS

Fragmento de código (subprograma) que realiza una tarea específica y es relativamente independiente del resto del código. Los procedimientos suelen utilizarse para reducir la duplicación de códigos en un programa.

Los procedimientos pueden recibir parámetros, pero no necesitan devolver un valor como es el caso de las funciones.

```
Modificador_de_acceso void Nombre_procedimiento (tipo(s)_argumento(s) nombres)
{
    //declaración de datos y cuerpo de la función
}
```

DECLARACIÓN Y CREACIÓN DE ARREGLOS

Las instancias de los arreglos ocupan espacio en memoria. Al igual que los objetos, los arreglos se crean con la palabra clave **new**. Para crear una instancia de un arreglo, se especifica el tipo y el número de elementos del arreglo, y el número de elementos como parte de una expresión de creación de arreglos, que utiliza la palabra clave **new**. Dicha expresión devuelve una referencia que puede almacenarse en una variable tipo arreglo. La siguiente expresión de declaración y creación de arreglos crea un objeto que contiene 8 elementos int, y almacena la referencia al arreglo en la variable A.

```
int[ ] A = new int[8]
```

Materiales y Equipo

| Nº | Cantidad | Descripción |
|----|----------|--|
| 1 | 1 | Guía de Laboratorio #1 de Programación Orientada a Objetos |
| 2 | 1 | Computadora con programa: ➤ Microsoft Visual C# |
| 3 | 1 | Dispositivo de memoria externo (opcional) |

Procedimiento y desarrollo de práctica

Para iniciar realizaremos unos ejercicios de ejemplo básicos, de forma que recordemos cómo invocar las funciones realizadas:

G2_Ejemplo_01:

Calcular el cuadrado de los números del 1 al 10 utilizando funciones

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

namespace Ejemplo1_Guia2
{
    class Program
    {
        //Prototipo de función
        static Double Potencia(Double x)
        {
            return x * x;
        }

        static void Main(string[] args)
        {
            Console.Title = "Ejercicio Potencia con Funciones";
            Double x;
            //se invoca la función
            for (x = 1; x <= 10; x++)
            {
                Console.WriteLine("\n El cuadrado de " + x + " es: " + Potencia(x));

            }
            Console.ReadKey();
        }
    }
}
```

G2_Ejemplo_02:

Realizar un programa que permita calcular la raíz cuadrada de los primeros 10 números enteros. Utilizar procedimiento.

```

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

namespace Ejemplo2_Guia2
{
    class Program
    {
        static void Raiz()
        {
            Console.WriteLine("Calculando raíces");
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine("\n La raíz cuadrada del número " + i + " es: " + Math.Sqrt(i));
            }
        }

        static void Main(string[] args)
        {
            /*Como es un procedimiento y para ingresar valores utilizamos un for, solamente
             *haremos la invocación del procedimiento en el main*/
            Raiz();
            Console.ReadKey();
        }
    }
}

```

G2_Ejemplo_03:

Realizar un programa que permita convertir temperaturas de grados Fahrenheit a grados Celsius. Utilizando procedimientos.

```

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

namespace Ejemplo_3_Guia2
{
    class Program
    {
        static void Faren()
        {
            Double cel, far;
            //Aparece en pantalla
            Console.Write("Escribe los grados Celsius: ");

            //Se leen los grados Celsius y se convierten a double
            cel = Convert.ToDouble(Console.ReadLine());
            // Se obtienen los grados Fahrenheit
            far = cel * 9.0 / 5.0 + 32;

            //Se muestran los grados Fahrenheit resultantes
            Console.WriteLine("{0} grados Celsius son {1} grados Fahrenheit", cel, far);
        }

        static void Celsi()
        {
            Double cel, far;
            //Aparece en pantalla
            Console.Write("Escribe los grados Fahrenheit: ");

            //Se leen los grados Fahrenheit y se convierten a double
            far = Convert.ToDouble(Console.ReadLine());
            cel = (far - 32) * 5.0 / 9.0;
            Console.WriteLine("{0} grados Fahrenheit son {1} grados Celsius", far, cel);
        }
    }
}

```

```
// Se obtienen los grados Celsius
cel = (far - 32) * 5.0 / 9.0;

//Se muestran los grados Celsius resultantes
Console.WriteLine("{0} grados Fahrenheit son {1} grados Celsius", far, cel);

}

static void Main(string[] args)
{
    string opc;

    Console.WriteLine("\t\t PROGRAMA PARA CONVERTIR TEMPERATURAS");
    Console.WriteLine("\nElige el tipo de conversion");
    Console.WriteLine("\n\nPresiona F si quieres convertir Celsius a Fahrenheit");
    Console.WriteLine("Presiona C si quieres convertir Fahrenheit a Celsius");
    Console.WriteLine("\nElige F ó C: ");

    // Recoger lo que puso el usuario
    opc = Console.ReadLine();

    switch (opc)
    {
        case "F":
        case "f":
            Console.WriteLine("\n Conversion a F");
            //invocación del método
            Faren();
            break;

        case "C":
        case "c":

            Console.WriteLine("\n Conversion a C");
            //invocación del método
            Celsi();
            break;

        default:
            Console.WriteLine("\n Tipo de conversión no válida");
            break;
    }

    //esperando a que el usuario presione una tecla
    Console.ReadKey();
}
}
```

G2_Ejemplo_04:

Se desean guardar los sueldos de 5 secretarias de la agencia Publicidad. Emplee arreglos y métodos.

```
namespace Ejemplo_4_Guia2
{
    class Program
    {
        static void Mostrar(Double[] sueldos)
        {
            Console.WriteLine("\n\t Los sueldos ingresados son:");
            for (int i = 0; i < 5; i++)
            {
                Console.Write("\t" + sueldos[i]);
            }
        }

        static void Main(string[] args)
        {
            Double[] Sueldos;
            Sueldos = new Double[5];
            for (int i = 0; i < 5; i++)
            {
                Console.Write("\n\t El sueldo de la secretaria " + i + " :");
                String Valor;
                Valor = Console.ReadLine();
                Sueldos[i] = Double.Parse(Valor);
            }
            Mostrar(Sueldos);
            Console.ReadKey();
        }
    }
}
```

Una vez hemos realizado los ejemplos sugeridos, vamos a proponer soluciones para los siguientes planteamientos:

- 1) Modifique el tercer ejemplo de forma que pueda hacer conversiones también a grados Kelvin (usados para investigación científica).
- 2) Crear un programa que solicite el año de nacimiento de una persona y retorne la edad haciendo uso de funciones o procedimientos.
- 3) Cree un programa que contenga el siguiente menú:

- a) Dividir (considere que es imposible dividir entre 0)
- b) Obtener cubo.
- c) Cálculo de IMC (Índice de Masa Corporal).
- d) Salir.

Consideraciones:

- El menú debe permanecer disponible hasta que el usuario elija la opción d.
- Utilizar una función o procedimiento para cada opción.
- Para la opción d, utilice la fórmula: IMC = Peso[Kg] / Altura²[Metros].

En todos los ejercicios anteriores se le sugiere considerar:

- i. Realizar un diseño amigable con el usuario e intuitivo
- ii. Incluya validaciones mínimas pero que ayudarán a prevenir errores.
- iii. Incluya comentarios en aquellas secciones que considere es importante destacar.
- iv. Incluya elementos que a usted cómo usuario final le gustaría ver en un programa (ayuda en pantalla, unidades u otros elementos)
- v. En aquellos ítem que no hay una solicitud explícita sea propositivo y sugiera una forma de solución.

Al finalizar el desarrollo de ejercicios compare con sus compañeros de fila

¿Al ingresar valores de entrada iguales se obtienen los mismos resultados?
¿Qué buenas prácticas e ideas observa en el desarrollo de sus ejercicios?

Ejercicios complementarios sugeridos

Los ejercicios sugeridos tienen la finalidad de ser un instrumento para la práctica del estudiante y así reforzar sus competencias.

- 1) Desarrolle un programa en C# haciendo uso de un procedimiento para determinar el resultado de la siguiente serie:

$$\frac{1}{2} - \frac{2}{2^2} + \frac{3}{2^3} - \cdots + \frac{n}{2^n}$$

- 2) Investigue sobre la instrucción foreach y aplíquela en el caso: Realice una sumatoria de pesos de los estudiantes de laboratorio de Programación Orientada a Objetos. Crear un arreglo con las cantidades necesarias de memoria para almacenar los datos correspondientes.

Guarde todo en un archivo comprimido (reporte y soluciones generadas en C#)

