

### 1. Suma de números pares

Escribe un programa que solicite un número entero positivo  $n$  y sume todos los números pares del 1 hasta  $n$  usando un bucle for.

**Temas: Condicionales, Bucles, Operadores.**

### 2. Calculadora básica

Desarrolla una calculadora que permita realizar suma, resta, multiplicación y división entre dos números según la opción elegida por el usuario.

**Temas: Condicionales, Operadores, Métodos.**

### 3. Tabla de multiplicar

Pide un número al usuario y muestra la tabla de multiplicar del 1 al 10 usando un bucle.

**Temas: Bucles, Condicionales.**

### 4. Promedio de calificaciones

Crea un programa que permita ingresar las calificaciones de 5 estudiantes y calcule el promedio de las mismas.

**Temas: Arrays, Bucles.**

### 5. Búsqueda de palabras en una cadena

Solicita al usuario una oración y una palabra. Busca cuántas veces aparece la palabra en la oración ingresada.

**Temas: Cadenas, Condicionales.**

### 6. Matriz identidad

Crea un programa que genere e imprima una matriz identidad de tamaño  $3 \times 3$ .

**Temas: Matrices, Bucles.**

## 7. Clase Persona y Programador

Crea una clase Persona con propiedades: Nombre, Edad y un método MostrarDatos().

Crea una clase Programador que herede de Persona y añada una propiedad LenguajeFavorito.

Crea un objeto de Programador y muestra los datos.

**Temas: POO, Herencia.**

## 8. Clase Rectángulo

Crea una clase Rectangulo con propiedades: Base y Altura. Añade un método para calcular el área y otro para el perímetro. Crea varios objetos y muestra sus valores.

**Temas: Clases, Métodos.**

## 9. Contador de objetos

Crea una clase Contador que tenga un atributo estático para contar cuántos objetos han sido creados de esa clase.

**Temas: Clases, StaticClass.**

## 10. Interfaz Figura

Crea una interfaz IFigura con un método CalcularArea.

Implementa esta interfaz en las clases Cuadrado y Triangulo. Crea objetos de ambas clases y calcula el área.

**Temas: Interfaces, POO.**

## 11. Uso de Enum

Crea un enum llamado DiasSemana con los días de la semana. Solicita al usuario un número del 1 al 7 e imprime el día correspondiente.

**Temas: Enum, Condicionales.**

## 12. Delegado para operaciones matemáticas

Define un delegado que realice operaciones matemáticas simples como suma, resta, multiplicación y división. Usa métodos para realizar cada operación e invoca el delegado.

**Temas: Delegate, Métodos.**

## 13. Evento de alarma

Crea una clase Alarma con un evento OnAlarmaActivada. Este evento debe activarse cuando se cumpla una condición, como que una temperatura exceda un límite.

**Temas: Eventos, Delegate.**

## 14. Uso de propiedades

Crea una clase Producto con propiedades Nombre, Precio y Cantidad. Calcula el total a pagar multiplicando el precio por la cantidad. Usa propiedades para validar que no se ingresen valores negativos.

**Temas: Propiedades, Clases.**

## 15. Método de extensión para cadenas

Crea un método de extensión para la clase string que cuente el número de palabras en una cadena.

**Temas: Métodos de Extensión, Cadenas.**

## 16. Ejercicio de Anonymous Methods

Filtrar números impares usando un método anónimo

Crea una lista de números enteros del 1 al 10.

Utiliza un método anónimo con `List<T>.FindAll()` para filtrar y mostrar únicamente los números impares.

**Temas: Anonymous Methods, Listas, Delegados.**

## 17. Ejercicio de Lambda

Ordenar nombres con expresiones Lambda

Crea una lista de nombres (ejemplo: "Ana", "Pedro", "Luis", "Carlos").

Utiliza una expresión Lambda con `List<T>.Sort()` para ordenar los nombres en orden alfabético inverso (de Z a A).

Muestra la lista ordenada.

**Temas: Lambda Expressions, Listas.**

## 18. Ejercicio de Expression Trees

Construir una operación matemática simple con Expression Trees

Usa `System.Linq.Expressions` para crear un árbol de expresión que represente la operación matemática:  $x + y$ .

Compila la expresión y ejecútala para valores  $x = 5$  e  $y = 3$ . Muestra el resultado.

**Temas: Expression Trees, Expresiones dinámicas.**