**1. Ejercicio1: Hola Mundo Básico**

* **Enunciado**: Escribe un programa en C# que muestre el mensaje “Hola, Mundo” en la consola. Usa el método Console.WriteLine para imprimir el texto.

using System;

class Program

{

static void Main()

{

Console.WriteLine("Hola, Mundo");

}

}

**Ejercicio 2: Información Personal**

* **Enunciado**: Escribe un programa que imprima en consola tu nombre, edad y tu ciudad de residencia.

using System;

class Program

{

static void Main()

{

string nombre = "Gabriel";

int edad = 25;

string ciudad = "Madrid";

Console.WriteLine($"Nombre: {nombre}");

Console.WriteLine($"Edad: {edad}");

Console.WriteLine($"Ciudad: {ciudad}");

}

}

**Tipos de Datos, Variables y Constantes**

**Ejercicio 3: Calcular el área de un círculo**

* **Enunciado**: Escribe un programa en C# que calcule el área de un círculo con un radio dado. Define el valor de PI como una constante y utiliza una variable para el radio.

using System;

class Program

{

static void Main()

{

const double PI = 3.1416;

double radio = 5.0;

double area = PI \* radio \* radio;

Console.WriteLine($"El área del círculo es: {area}");

}

}

**Ejercicio 4: Conversión de Temperatura**

* **Enunciado**: Escribe un programa que convierta de grados Celsius a Fahrenheit. Solicita al usuario la temperatura en Celsius y muestra el resultado en Fahrenheit.

using System;

class Program

{

static void Main()

{

Console.Write("Introduce la temperatura en Celsius: ");

double celsius = Convert.ToDouble(Console.ReadLine());

double fahrenheit = (celsius \* 9 / 5) + 32;

Console.WriteLine($"La temperatura en Fahrenheit es: {fahrenheit}");

}

}

**Control de Flujo: if-else**

**Ejercicio 5: Número Par o Impar**

* **Enunciado**: Escribe un programa que solicite un número al usuario y determine si es par o impar.

using System;

class Program

{

static void Main()

{

Console.Write("Introduce un número: ");

int numero = Convert.ToInt32(Console.ReadLine());

if (numero % 2 == 0)

{

Console.WriteLine("El número es par.");

}

else

{

Console.WriteLine("El número es impar.");

}

}

}

#### Ejercicio 6: Edad para Conducir

* **Enunciado**: Escribe un programa que verifique si una persona puede obtener una licencia de conducir. La edad mínima es 18 años.

using System;

class Program

{

static void Main()

{

Console.Write("Introduce tu edad: ");

int edad = Convert.ToInt32(Console.ReadLine());

if (edad >= 18)

{

Console.WriteLine("Puedes obtener una licencia de conducir.");

}

else

{

Console.WriteLine("No puedes obtener una licencia de conducir.");

}

}

}

**Control de Flujo: switch-case**

**Ejercicio 7: Día de la Semana**

Enunciado: Escribe un programa que reciba un número (del 1 al 7) y muestre el día de la semana correspondiente.

using System;

class Program

{

static void Main()

{

Console.Write("Introduce un número (1-7) para el día de la semana: ");

int dia = Convert.ToInt32(Console.ReadLine());

switch (dia)

{

case 1: Console.WriteLine("Lunes"); break;

case 2: Console.WriteLine("Martes"); break;

case 3: Console.WriteLine("Miércoles"); break;

case 4: Console.WriteLine("Jueves"); break;

case 5: Console.WriteLine("Viernes"); break;

case 6: Console.WriteLine("Sábado"); break;

case 7: Console.WriteLine("Domingo"); break;

default: Console.WriteLine("Número inválido"); break;

}

}

}

**Ejercicio 8: Mes del Año**

* **Enunciado**: Escribe un programa que reciba un número (del 1 al 12) e imprima el nombre del mes correspondiente.

using System;

class Program

{

static void Main()

{

Console.Write("Introduce un número (1-12) para el mes: ");

int mes = Convert.ToInt32(Console.ReadLine());

switch (mes)

{

case 1: Console.WriteLine("Enero"); break;

case 2: Console.WriteLine("Febrero"); break;

case 3: Console.WriteLine("Marzo"); break;

case 4: Console.WriteLine("Abril"); break;

case 5: Console.WriteLine("Mayo"); break;

case 6: Console.WriteLine("Junio"); break;

case 7: Console.WriteLine("Julio"); break;

case 8: Console.WriteLine("Agosto"); break;

case 9: Console.WriteLine("Septiembre"); break;

case 10: Console.WriteLine("Octubre"); break;

case 11: Console.WriteLine("Noviembre"); break;

case 12: Console.WriteLine("Diciembre"); break;

default: Console.WriteLine("Número inválido"); break;

}

}

}

**Control de Flujo: Ciclo for**

**Ejercicio 9: Sumar los primeros 10 números**

* **Enunciado**: Escribe un programa que sume los números del 1 al 10 usando un ciclo for y muestre el resultado.

using System;

class Program

{

static void Main()

{

int suma = 0;

for (int i = 1; i <= 10; i++)

{

suma += i;

}

Console.WriteLine($"La suma de los primeros 10 números es: {suma}");

}

}

**Ejercicio 10: Tabla de Multiplicar**

* **Enunciado**: Solicita al usuario un número y usa un ciclo for para imprimir su tabla de multiplicar del 1 al 10.

using System;

class Program

{

static void Main()

{

Console.Write("Introduce un número para la tabla de multiplicar: ");

int numero = Convert.ToInt32(Console.ReadLine());

for (int i = 1; i <= 10; i++)

{

Console.WriteLine($"{numero} x {i} = {numero \* i}");

}

}

}

**Control de Flujo: Ciclo while y do-while**

**Ejercicio 11: Contar hasta 10**

* **Enunciado**: Escribe un programa que use un ciclo while para contar del 1 al 10 e imprima los números en la consola.

using System;

class Program

{

static void Main()

{

int contador = 1;

while (contador <= 10)

{

Console.WriteLine(contador);

contador++;

}

}

}

**Ejercicio 12: Sumar hasta que se ingrese un 0**

* **Enunciado**: Escribe un programa que sume números ingresados por el usuario hasta que introduzca el número 0.

using System;

class Program

{

static void Main()

{

int suma = 0;

int numero;

do

{

Console.Write("Introduce un número (0 para terminar): ");

numero = Convert.ToInt32(Console.ReadLine());

suma += numero;

} while (numero != 0);

Console.WriteLine($"La suma total es: {suma}");

}

}

**Ejercicio 1: Definir una estructura de coordenadas**

* **Enunciado**: Crea una estructura llamada Coordenadas que contenga dos campos enteros X y Y. Escribe un método dentro de la estructura que calcule la distancia entre las coordenadas actuales y otro punto (otro objeto Coordenadas). Luego, crea un objeto de esta estructura y calcula la distancia entre dos puntos.

using System;

struct Coordenadas

{

public int X { get; set; }

public int Y { get; set; }

public Coordenadas(int x, int y)

{

X = x;

Y = y;

}

public double CalcularDistancia(Coordenadas otroPunto)

{

int deltaX = otroPunto.X - X;

int deltaY = otroPunto.Y - Y;

return Math.Sqrt(deltaX \* deltaX + deltaY \* deltaY);

}

}

class Program

{

static void Main()

{

Coordenadas punto1 = new Coordenadas(3, 4);

Coordenadas punto2 = new Coordenadas(7, 1);

double distancia = punto1.CalcularDistancia(punto2);

Console.WriteLine($"La distancia entre los puntos es: {distancia}");

}

}

**Ejercicio 2: Definir una estructura para representar rectángulos**

* **Enunciado**: Crea una estructura llamada Rectangulo que tenga dos propiedades: Ancho y Alto. Incluye un método para calcular el área del rectángulo y otro para calcular su perímetro. Luego, crea un objeto de esta estructura y calcula el área y el perímetro para un rectángulo con valores específicos de ancho y alto.

using System;

struct Rectangulo

{

public double Ancho { get; set; }

public double Alto { get; set; }

public Rectangulo(double ancho, double alto)

{

Ancho = ancho;

Alto = alto;

}

public double CalcularArea()

{

return Ancho \* Alto;

}

public double CalcularPerimetro()

{

return 2 \* (Ancho + Alto);

}

}

class Program

{

static void Main()

{

Rectangulo rect = new Rectangulo(5, 3);

Console.WriteLine($"Área del rectángulo: {rect.CalcularArea()}");

Console.WriteLine($"Perímetro del rectángulo: {rect.CalcularPerimetro()}");

}

}

**Ejercicio 3: Definir una estructura para manejar fechas simples**

* **Enunciado**: Crea una estructura llamada FechaSimple que contenga tres campos: Dia, Mes, y Año. Incluye un método que verifique si la fecha es válida (por ejemplo, día entre 1 y 31, mes entre 1 y 12, etc.). Luego, crea un objeto de esta estructura e imprime si la fecha es válida o no.

using System;

struct FechaSimple

{

public int Dia { get; set; }

public int Mes { get; set; }

public int Año { get; set; }

public FechaSimple(int dia, int mes, int año)

{

Dia = dia;

Mes = mes;

Año = año;

}

public bool EsFechaValida()

{

if (Año < 1) return false;

if (Mes < 1 || Mes > 12) return false;

if (Dia < 1 || Dia > 31) return false;

// Meses con 30 días

if ((Mes == 4 || Mes == 6 || Mes == 9 || Mes == 11) && Dia > 30)

return false;

// Febrero y años bisiestos

if (Mes == 2)

{

bool esBisiesto = (Año % 4 == 0 && Año % 100 != 0) || (Año % 400 == 0);

if (Dia > 29 || (Dia == 29 && !esBisiesto))

return false;

}

return true;

}

}

class Program

{

static void Main()

{

FechaSimple fecha = new FechaSimple(29, 2, 2023);

if (fecha.EsFechaValida())

{

Console.WriteLine("La fecha es válida.");

}

else

{

Console.WriteLine("La fecha no es válida.");

}

}

}

**Ejercicio 4: Estructura para representar un color en RGB**

* **Enunciado**: Crea una estructura llamada ColorRGB que contenga tres campos R, G, B para los componentes de color rojo, verde, y azul (con valores entre 0 y 255). Escribe un método para verificar si el color es válido (es decir, todos los valores deben estar en el rango de 0 a 255). Luego, crea un objeto de esta estructura e imprime si el color es válido o no.

using System;

struct ColorRGB

{

public int R { get; set; }

public int G { get; set; }

public int B { get; set; }

public ColorRGB(int r, int g, int b)

{

R = r;

G = g;

B = b;

}

public bool EsColorValido()

{

return (R >= 0 && R <= 255) && (G >= 0 && G <= 255) && (B >= 0 && B <= 255);

}

}

class Program

{

static void Main()

{

ColorRGB color = new ColorRGB(120, 255, 300);

if (color.EsColorValido())

{

Console.WriteLine("El color es válido.");

}

else

{

Console.WriteLine("El color no es válido.");

}

}

}