

**PROGRAMA DE ESTUDIOS**

# **DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**HERRAMIENTAS DE  
PROGRAMACIÓN –  
C#**

**Tema**

**EJERCICIOS CON LOS TIPOS DE  
MÉTODOS**

## EJERCICIOS CON LOS TIPOS DE MÉTODOS

### RESUELTOS

**EJERCICIO 01:** Crea una clase Estudiante con un método ObtenerPromedio que retorne el promedio de tres calificaciones (retorna int). Implementa otro método privado ValidarCalificaciones que retorne bool si todas las calificaciones son válidas (mayores o iguales a 0).

```
2 referencias
internal class Estudiante
{
    // Método público que obtiene el promedio de tres calificaciones
    public int ObtenerPromedio(int calificacion1, int calificacion2, int calificacion3)
    {
        // Validar las calificaciones antes de calcular el promedio
        if (ValidarCalificaciones(calificacion1, calificacion2, calificacion3))
        {
            int promedio = (calificacion1 + calificacion2 + calificacion3) / 3;
            return promedio;
        }
        else
        {
            Console.WriteLine("Las calificaciones no son válidas. Deben ser mayores o iguales a 0.");
            return -1; // Retorna un valor inválido si las calificaciones no son válidas
        }
    }

    // Método privado que valida si las tres calificaciones son mayores o iguales a 0
    private bool ValidarCalificaciones(int calificacion1, int calificacion2, int calificacion3)
    {
        return calificacion1 >= 0 && calificacion2 >= 0 && calificacion3 >= 0;
    }
}
```

```
static void Main(string[] args)
{
    Estudiante estudiante = new Estudiante();

    // Ingreso de calificaciones desde el usuario
    Console.WriteLine("Ingrese tres calificaciones:");

    Console.Write("Calificación 1: ");
    int calificacion1 = int.Parse(Console.ReadLine());

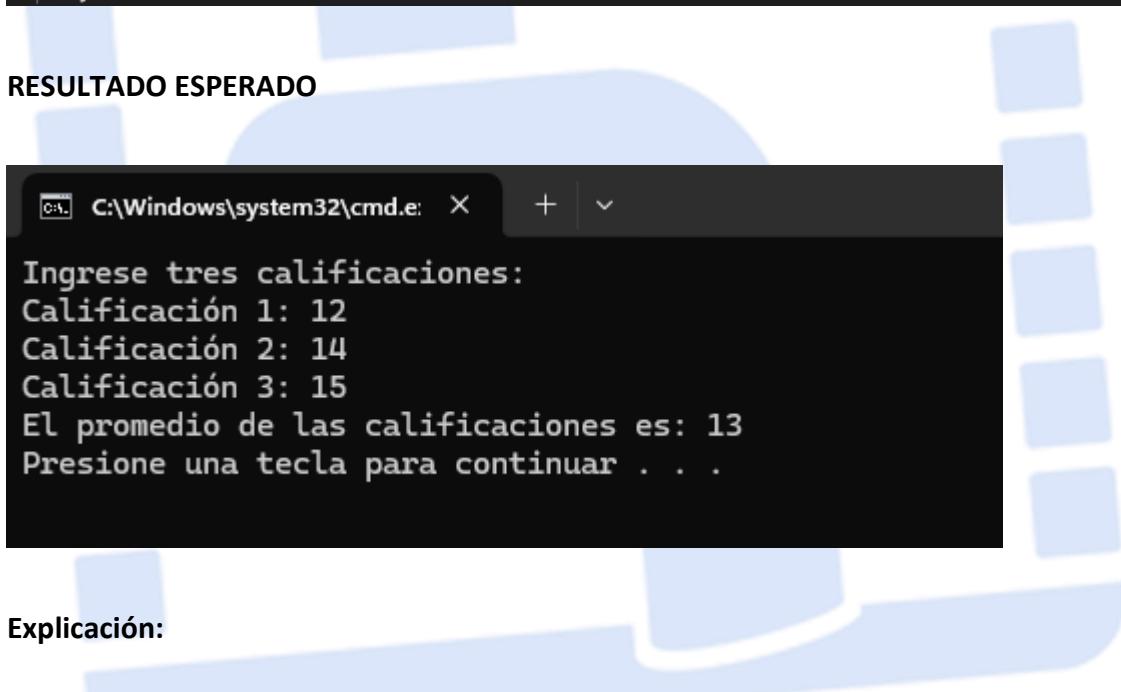
    Console.Write("Calificación 2: ");
    int calificacion2 = int.Parse(Console.ReadLine());

    Console.Write("Calificación 3: ");
    int calificacion3 = int.Parse(Console.ReadLine());

    // Llamar al método ObtenerPromedio
    int promedio = estudiante.ObtenerPromedio(calificacion1, calificacion2, calificacion3);

    // Mostrar el resultado si es válido
    if (promedio != -1)
    {
        Console.WriteLine($"El promedio de las calificaciones es: {promedio}");
    }
}
```

## RESULTADO ESPERADO



```
C:\Windows\system32\cmd.e: X + ▾

Ingrese tres calificaciones:
Calificación 1: 12
Calificación 2: 14
Calificación 3: 15
El promedio de las calificaciones es: 13
Presione una tecla para continuar . . .
```

### Explicación:

#### 1. Método ObtenerPromedio:

- Este método recibe tres calificaciones como parámetros. Antes de calcular el promedio, llama al método privado ValidarCalificaciones para asegurarse de que las calificaciones sean válidas.
- Si las calificaciones son válidas, calcula el promedio sumando las tres calificaciones y dividiendo el resultado entre 3.

- Si las calificaciones no son válidas, retorna un valor -1 e imprime un mensaje de advertencia.

## 2. Método Privado Validar Calificaciones:

- Este método verifica que todas las calificaciones sean mayores o iguales a 0, retornando true si todas son válidas y false si alguna no lo es.

## 3. Ingreso de Datos:

- Se solicitan las tres calificaciones al usuario y se procesan para obtener el promedio si las calificaciones son válidas.

```
internal class CuentaBancaria
{
    // Método público que aumenta el saldo de la cuenta si el depósito es válido
    public void Depositar(decimal monto)
    {
        // Validar si el depósito es positivo antes de procesarlo
        if (ValidarDeposito(monto))
        {
            Console.WriteLine($"Depósito exitoso de {monto:C}.");
        }
        else
        {
            Console.WriteLine("Depósito fallido. El monto debe ser positivo.");
        }
    }

    // Método privado que valida si el monto a depositar es positivo
    private bool ValidarDeposito(decimal monto)
    {
        return monto > 0;
    }
}
```

**EJERCICIO 02:** Implementa una clase CuentaBancaria con un método público Depositar (retorna void) que aumente el saldo de la cuenta, y un método privado ValidarDeposito (retorna bool) que verifique si el monto a depositar es positivo.

```
internal class CuentaBancaria
{
    // Método público que aumenta el saldo de la cuenta si el depósito es válido
    1 referencia
    public void Depositar(decimal monto)
    {
        // Validar si el depósito es positivo antes de procesarlo
        if (ValidarDeposito(monto))
        {
            Console.WriteLine($"Depósito exitoso de {monto:C}.");
        }
        else
        {
            Console.WriteLine("Depósito fallido. El monto debe ser positivo.");
        }
    }

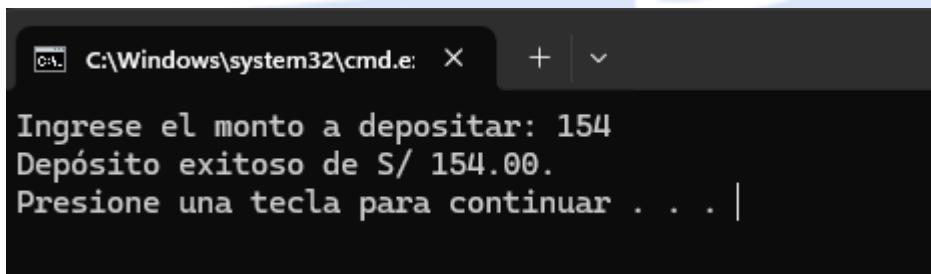
    // Método privado que valida si el monto a depositar es positivo
    1 referencia
    private bool ValidarDeposito(decimal monto)
    {
        return monto > 0;
    }
}
```

```
0 referencias
static void Main(string[] args)
{
    CuentaBancaria cuenta = new CuentaBancaria();

    // Ingreso del monto a depositar desde el usuario
    Console.Write("Ingrese el monto a depositar: ");
    decimal monto;
    while (!decimal.TryParse(Console.ReadLine(), out monto))
    {
        Console.WriteLine("Por favor, ingrese un monto válido.");
        Console.Write("Ingrese el monto a depositar: ");
    }

    // Realizar el depósito
    cuenta.Depositar(monto);
}
```

### RESULTADO ESPERADO



C:\Windows\system32\cmd.e: + ^

```
Ingrese el monto a depositar: 154
Depósito exitoso de S/ 154.00.
Presione una tecla para continuar . . . |
```

**Explicación:**

**1. Método Depositar:**

- Este método recibe un monto como parámetro. Primero llama al método privado ValidarDeposito para verificar si el monto es válido (positivo).
- Si el monto es positivo, muestra un mensaje confirmando el depósito. Si no, imprime un mensaje de error indicando que el depósito ha fallado.

**2. Método Privado ValidarDeposito:**

- Este método verifica si el monto a depositar es mayor que 0 y retorna true si lo es. En

EJERCICIO 03: Crea una clase Empleado con un método público CalcularSalario (retorna int) que calcule el salario mensual en base a horas trabajadas y tarifa por hora. Incluye un método privado ValidarHoras que valide las horas ingresadas. no incluyas atributos de clase

```

internal class Empleado
{
    // Método público que calcula el salario mensual basado en las horas trabajadas y la tarifa por hora
    1 referencia
    public int CalcularSalario(int horasTrabajadas, int tarifaPorHora)
    {
        // Validar si las horas trabajadas son válidas antes de calcular el salario
        if (ValidarHoras(horasTrabajadas))
        {
            int salario = horasTrabajadas * tarifaPorHora;
            return salario;
        }
        else
        {
            Console.WriteLine("Error: El número de horas trabajadas debe ser mayor o igual a 0.");
            return -1; // Retorna un valor inválido si las horas no son válidas
        }
    }

    // Método privado que valida si las horas trabajadas son mayores o iguales a 0
    1 referencia
    private bool ValidarHoras(int horasTrabajadas)
    {
        return horasTrabajadas >= 0;
    }
}

```

```

static void Main(string[] args)
{
    Empleado empleado = new Empleado();

    // Ingreso de datos desde el usuario
    Console.Write("Ingrese las horas trabajadas: ");
    int horasTrabajadas;
    while (!int.TryParse(Console.ReadLine(), out horasTrabajadas))
    {
        Console.WriteLine("Por favor, ingrese un número válido de horas.");
        Console.Write("Ingrese las horas trabajadas: ");
    }

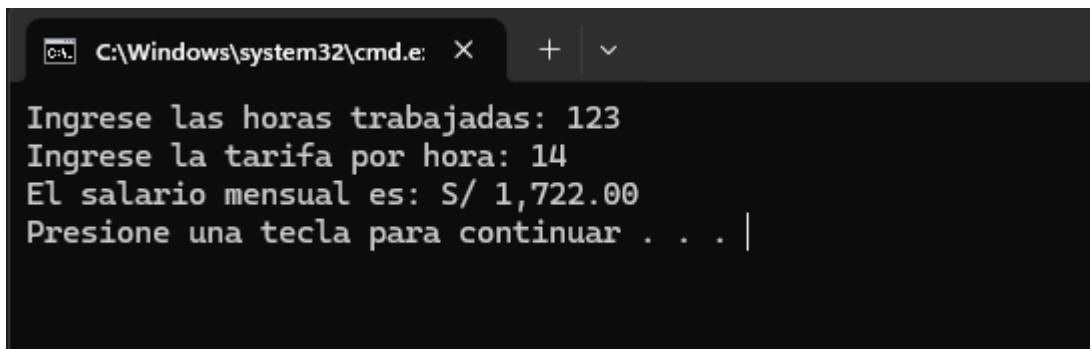
    Console.Write("Ingrese la tarifa por hora: ");
    int tarifaPorHora;
    while (!int.TryParse(Console.ReadLine(), out tarifaPorHora))
    {
        Console.WriteLine("Por favor, ingrese una tarifa válida.");
        Console.Write("Ingrese la tarifa por hora: ");
    }

    // Calcular el salario
    int salario = empleado.CalcularSalario(horasTrabajadas, tarifaPorHora);

    // Mostrar el salario si es válido
    if (salario != -1)
    {
        Console.WriteLine($"El salario mensual es: {salario:C}");
    }
}

```

## RESULTADO ESPERADO



```
C:\Windows\system32\cmd.exe + ^

Ingrese las horas trabajadas: 123
Ingrese la tarifa por hora: 14
El salario mensual es: S/ 1,722.00
Presione una tecla para continuar . . . |
```

### Explicación:

#### 1. Método CalcularSalario:

- Recibe como parámetros las horas trabajadas y la tarifa por hora.
- Antes de calcular el salario, llama al método privado ValidarHoras para asegurarse de que el número de horas trabajadas es válido (mayor o igual a 0).
- Si las horas trabajadas son válidas, multiplica las horas por la tarifa para calcular el salario.
- Si las horas no son válidas, muestra un mensaje de error y retorna -1.

#### 2. Método Privado ValidarHoras:

- Verifica si las horas trabajadas son mayores o iguales a 0. Si lo son, retorna true, en caso contrario, retorna false.

**3. Ingreso de Datos:**

- Se solicitan las horas trabajadas y la tarifa por hora al usuario. Los datos ingresados se validan mediante `int.TryParse` para asegurarse de que sean números válidos.
- Se repite el proceso de entrada hasta que el usuario ingrese datos correctos.

**4. Uso de la Clase:**

- Si el salario calculado es válido, se muestra el salario formateado como moneda

