

ASP.NET

Preguntas Abiertas:

1. ¿Cómo defines la estructura de un Data Set o que contiene?

Los DataSet son bases de datos en memoria. Fueron uno de los primeros métodos usados para interactuar con datos de la DB en C#. Es un poco similar a los DbContext actuales de Entity Framework.

2. ¿Qué contiene un Data Table?

Los registros de una tabla.

3. ¿Cuál es la diferencia entre ASP.NET y ASP.NET MVC?

El primero se refiere al marco de ASP.NET. El segundo es dicho marco aplicando al patrón Modelo-Vista-Controlador.

4. ¿Para ti que es una aplicación por capas?

Una aplicación dividida en funcionalidades independientes. Por ejemplo, un patrón común es aplicaciones web es dividir dicha aplicación en tres proyectos: El núcleo (Core), la infraestructura (datos), y el UI (La interfaz visual). A su vez, cada proyecto puede dividirse en microservicios.

5. ¿Qué es o para que sirve una FOREIGN KEY?

Para indicar que una entidad tiene relación con otra entidad. La entidad tendría un campo con el id de otra tabla.

6. ¿Qué es la PRIMARY KEY?

Representa un campo único e irrepetible de una entidad que sirve como identificador de dicha tabla. Puede ser usado para crear relaciones con otras tablas al usar su valor en un FOREIGN KEY.

Preguntas con opciones:

1.- Al programar con objetos la propiedad que me permite cambiar el nombre de un objeto es:

- a) Label
- b) Name**
- c) Text

NOTA: Creo que le falta contexto a esta pregunta. ¿A que llaman objeto aquí?

2.- La propiedad que me permite presentar el formulario en la pantalla y que debe tener un valor TRUE es:

- a) Key Preview
- b) Visible**
- c) Enable

NOTA: Supondré que hablan de Windows Forms. Depende del tipo de proyecto, cambia. Por ejemplo, en MAUI la propiedad es IsVisible, en WPF es Visibility, etc.

3.- Lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

a) Variables

b) Storyboard

c) Lenguaje de Programacion

4.- Conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa.

a) Código fuente

b) Programador

c) Informática

5.- Mezcla de lenguaje natural con algunas convenciones sintácticas propias de lenguajes de programación

a) Diagramas de flujo

b) Pseudocódigo

c) Variables

6.- Proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales.

a) Compilación

b) Programación

c) Hardware

7.- Conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.

a) Ciclos

b) Estructuras

c) Algoritmo

8.- Son descripciones gráficas de algoritmos; usan símbolos conectados con flechas para indicar la secuencia de instrucciones y están regidos por ISO

a) Diagramas de flujo

b) Pseudocódigo

c) Variables

9.- La Aplicación Windows Forms me permite:

a) Crear un contenedor de objetos

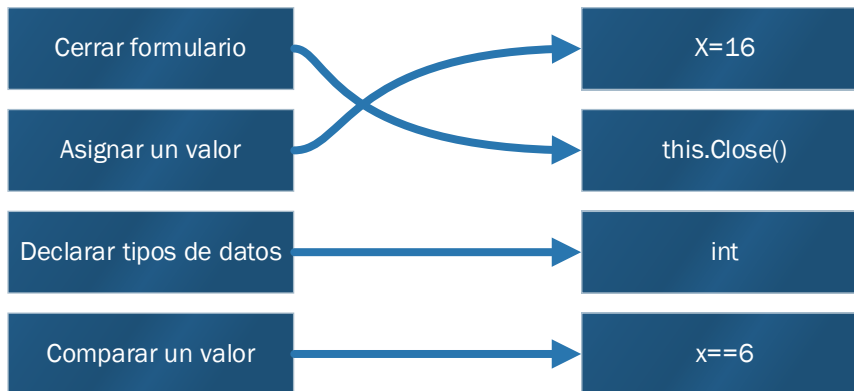
b) Crear un Formulario en blanco para integrar objetos

c) Ninguna de las anteriores

Empareje según corresponda:

Cerrar Formulario
Asignar un valor
Declarar tipos de datos
Comparar un valor

X=16
This.Close()
int
x==6



10.- Resuelve la siguiente operación:

static void Main(string[] args)

```
{  
    int a, b, c, x, r=0, r1=0;  
    decimal resultado = 0;  
    Console.WriteLine("&quot;Ingrese un Valor para A&quot;");  
    a = Convert.ToInt32(Console.ReadLine());  
    Console.WriteLine("&quot;Ingrese un Valor para B&quot;");  
    b = Convert.ToInt32(Console.ReadLine());  
    Console.WriteLine("&quot;Ingrese un Valor para C&quot;");  
    c = Convert.ToInt32(Console.ReadLine());  
    Console.WriteLine("&quot;Ingrese un Valor constante para X&quot;");  
    x = Convert.ToInt32(Console.ReadLine());  
    r = (a * (x * x));  
    r1 = b * x;  
    resultado = r + r1 + c;  
    Console.WriteLine(resultado.ToString());  
    Console.ReadLine();  
}
```

Dando Valores para a=2 b=3 c=4 x=5

La respuesta es

a)56

b)64

c)69

d)-69

11.- Que es lo que realiza el siguiente código

namespace programa

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int ar = 1;
            for (int i = 1; i <= 10; i++)
            {
                ar = i * 2;
                Console.WriteLine(ar.ToString());
            }
            Console.ReadLine();
        }
    }
}
```

Respuesta

- a) Impresión de números Impresión de caracteres
- b) Calcular 10 números impares
- c) Imprimir 10 números pares**
- d) Ninguna de las anteriores

Base de datos

1.- ¿Qué comando extrae datos de la base de datos?

- a) INSERT
- b) SELECT**
- c) UPDATE
- d) ALTER

2.- Una sentencia de SQL que es utilizada para modificar los valores en una tabla.

- a) INSERT
- b) SELECT
- c) UPDATE**
- d) ALTER

3.- Comando que permite modificar la estructura de un objeto.

- c) UPDATE
- d) ALTER**

4.- Comando que elimina un objeto de la base de datos.

- c) DELETE
- d) DROP**

5.- Sentencia de SQL que permite borrar cero o más registros en una tabla.

c) DELETE

d) DROP

6.- Sentencia de SQL que agrega uno o más registros a una tabla.

a) INSERT

b) SELECT

c) UPDATE

d) ALTER

7.- Comando que crea un objeto dentro de la base de datos.

c) CREATE

d) INSERT

8.- ¿Cuál sería la sintaxis correcta para seleccionar la columna llamada "Apellidos" de una tabla llamada "Trabajadores"?

a) EXTRACT Apellidos FROM Trabajadores

b) SELECT Trabajadores, Apellidos

c) SELECT Apellidos FROM Trabajadores

9.- ¿Cuál es la sintaxis correcta para seleccionar todos los campos de una tabla llamada "Trabajadores"?

a) SELECT * FROM Trabajadores

b) SELECT [all] FROM Trabajadores

c) SELECT Trabajadores

10.- ¿Cuál es la sintaxis correcta para seleccionar todos los campos de una tabla llamada Trabajadores, en los que la columna "Nombre" es "Raúl"?

a) SELECT * FROM Trabajadores WHERE Nombre:'Raúl'

b) SELECT * FROM Trabajadores WHERE Nombre='Raúl'

c) SELECT * FROM Trabajadores LIKE 'Raúl'

11.- ¿Qué palabra clave se usa para devolver solamente valores diferentes?

a) NOTSAME

b) COUNT

c) DISTINCT

12.- ¿Qué palabra clave se utiliza para ordenar el resultado?

a) ORDER

b) SORT

c) ORDER BY

13.- ¿Cuál es la sintaxis correcta si se desea insertar una nueva fila en la tabla "Trabajadores" (cuyos campos son "Nombre" y "Apellidos")?

a) INSERT ('Carlos', 'Pérez') INTO Trabajadores

b) INSERT INTO Trabajadores VALUES ('Carlos', 'Pérez')

c) INSERT VALUES ('Carlos', 'Pérez') INTO Trabajadores

14 ¿Cómo cambiarías "Carlos" por "Javier" en la columna "Nombre" de la tabla "Trabajadores"?

a) UPDATE Trabajadores SET Nombre = 'Carlos' INTO Nombre = 'Javier'

b) UPDATE Trabajadores SET Nombre = 'Javier' WHERE Nombre = 'Carlos'

c) SAVE Trabajadores SET Nombre = 'Carlos' INTO Nombre = 'Javier'

15.- ¿Cuál es la sintaxis correcta para borrar los registros cuyo campo "Nombre" sea "Andrés"?

a) DELETE FROM Trabajadores WHERE Nombre = 'Andrés'

b) DELETE ROW Nombre='Andrés' FROM Trabajadores

c) DELETE Nombre='Andrés' FROM Trabajadores

16 - Si quisiéramos contar el número de registros de la tabla "Trabajadores"...

a) SELECT NUMBER FROM Trabajadores

b) SELECT COUNT Trabajadores

c) SELECT COUNT(*) FROM Trabajadores

17.- Podría utilizar: [DELETE FROM usuario], para eliminar la tabla usuario

a) falso

b) verdadero

18.- Para Eliminar todas las filas de una tabla llamada "AUTOR" podría utilizar la Sentencia [TRUNCATE TABLE autor].

a) falso

b) verdadero

19 ¿Qué significan las siglas SQL?

a) Structured Query Language

b) Structured Question Line

c) Strong Question Language

20.- Para mostrar todos los registros de la tabla llamada "nacionalidad" podria aplicar: SELECT FROM * nacionalidad

a) falso

b) verdadero

21.- SELECT MAX(idusuario) FROM usuario , me permitiria ver el primer registro de la tabla usuario

a) falso

b) verdadero

22.- SELECT TOP 1 idusuario from usuario ORDER BY idusuario DESC, me permite mostrar el primer registro de la tabla

a) falso

b) verdadero

23.-- Sentencia SQL para filtrar a todos los usuarios que se llaman "JUAN"; SELECT * FROM USUARIO WHERE nombre like '%JUAN%'

a) falso

b) verdadero

24.- Mostrar todos los campos y todos los registros de usuarios cuyo idusuario es 'LEON'; [SELECT idusuario, password, nombre, telefono FROM usuario WHERE idusuario='LEON' AND password='123']

a) falso

b) verdadero

25.- Mostrar todos los campos y todos los registros de usuarios cuyo password es "123"; [SELECT * FROM usuario WHERE password='123']

a) falso

b) verdadero

26.--Mostrar cuantos registros de la tabla llamada empleado, tienen el nombre ESMERALDA; [SELECT COUNT(nombre) FROM empleado WHERE nombre='ESMERALDA']

a) falso

b) verdadero

27.-Insertar un registro en la tabla prestamo:

INSERT INTO prestamo(idprestamo, fecha, idusuario, idlector, idlibro, idtipo) VALUES ('7','CURRENT_TIMESTAMP','9','8','11','2')

a) falso

b) verdadero

28.-Insertar un registro en la tabla prestamo:

INSERT INTO prestamo(fecha, idusuario, idlector, idlibro, idtipo) VALUES (CURRENT_TIMESTAMP,'9','UBUNTU','11','2')

a) falso

b) verdadero

29.-Para Eliminar todas las filas de una tabla llamada "AUTOR" podria utilizar la Sentencia:

[DELETE FROM autor].

a) falso

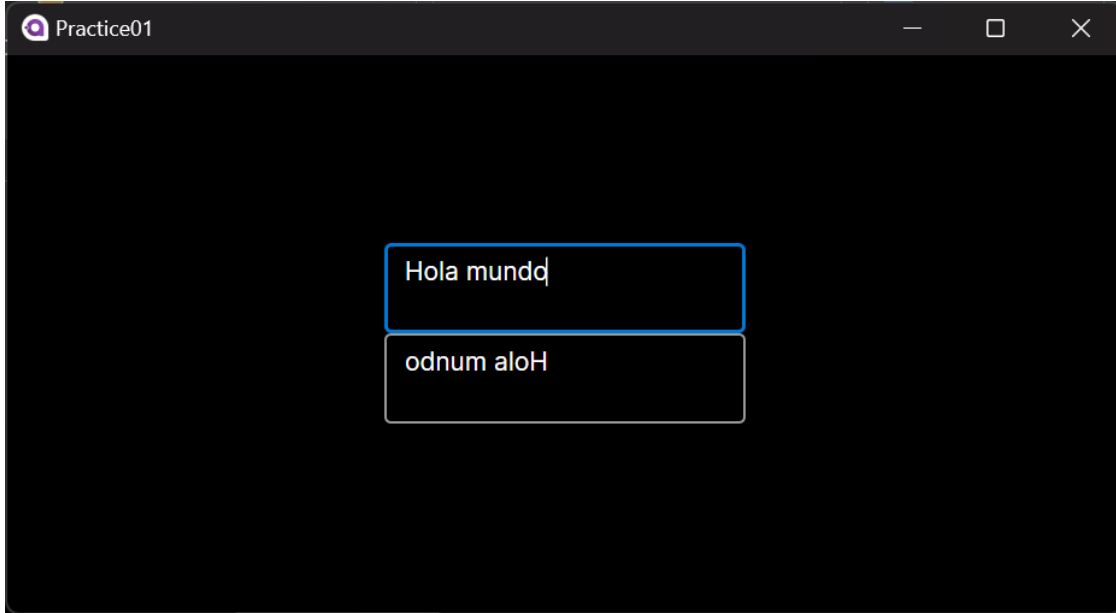
b) verdadero

Practico

Nota: Para los ejercicios se deberá entregar el código fuente y el ejecutable

1.- Realizar un programa donde el objetivo es invertir una cadena de texto, en la cual debe tener una caja de texto para capturar el texto que se desea invertir y una caja de texto para mostrar el resultado.

Resultado



XAML

```
<UserControl xmlns="https://github.com/avaloniaui"
              xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
              xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
              xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
              mc:Ignorable="d" d:DesignWidth="800" d:DesignHeight="450"
              x:Class="Practice01.Views.MainView">

    <StackPanel Orientation="Vertical" HorizontalAlignment="Center"
VerticalAlignment="Center">
        <TextBox x:Name="input" MinWidth="200" Height="50"
TextChanged="input_TextChanged" />
        <TextBox x:Name="output" MinWidth="200" Height="50"
IsReadOnly="True"/>
    </StackPanel>
</UserControl>
```


C#

```
// <summary>
/// Invierte el texto del TextBox
/// </summary>
/// <param name="sender">El objeto que ha lanzado el evento</param>
/// <param name="e">Información sobre el evento</param>
private void input_TextChanged(object sender, TextChangedEventArgs e)
{
    if (sender is TextBox input)
    {
        // Si el texto está vacío, no invertir nada
        if (string.IsNullOrEmpty(input.Text))
        {
            // Vaciar el TextBox de salida
            this.output.Text = string.Empty;
            return;
        }

        // Invertir el texto
        string invertedText = new(input.Text.Reverse().ToArray());

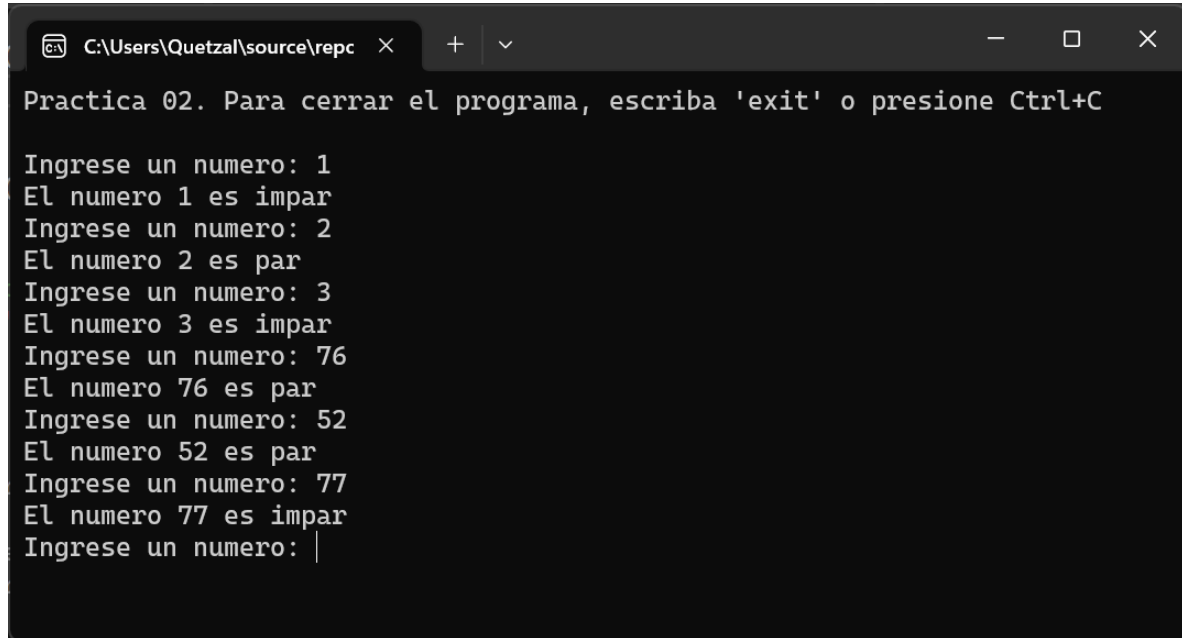
        // Aplicar el texto invertido al TextBox de salida
        this.output.Text = invertedText;
    }
}
```

2.- Se requiere la generación de un código que identifique cuando el número es par o impar.

Mensajes para mostrar:

- Cuando sea par mostrara el siguiente mensaje “El numero (Valor del número) es par”
- Cuando sea impar mostrara el siguiente mensaje “El numero (Valor del número) es impar”

Resultado



```
C:\Users\Quetzal\source\repc x + v
Practica 02. Para cerrar el programa, escriba 'exit' o presione Ctrl+C

Ingrese un numero: 1
El numero 1 es impar
Ingrese un numero: 2
El numero 2 es par
Ingrese un numero: 3
El numero 3 es impar
Ingrese un numero: 76
El numero 76 es par
Ingrese un numero: 52
El numero 52 es par
Ingrese un numero: 77
El numero 77 es impar
Ingrese un numero: |
```

C#

```
// Main program
Console.WriteLine("Practica 02. Para cerrar el programa, escriba 'exit' o
presione Ctrl+C\n");

while (true)
{
    Console.Write("Ingrese un numero: ");
    string? input = Console.ReadLine();

    // Salir del programa
    if (input == "exit")
    {
        break;
    }

    if (int.TryParse(input, out int number))
    {
        var evenMessage = number % 2 == 0 ? "par" : "impar";
        Console.WriteLine("El numero {0} es {1}", number, evenMessage);
    }
    else
    {
        Console.WriteLine("El valor ingresado no es un numero");
    }
}
```

3.- Genere un código que cumpla con lo siguiente:

Se debe contar con un parámetro de entrada, este parámetro tendrá el valor de un color (De los mencionado más adelante), este parámetro debe diferenciarse entre las siguientes opciones "Amarillo", "Morado", "Rosa", "Azul", "Blanco" e imprimirá lo siguiente dependiendo de la opción ingresada por el usuario:

Amarillo = La Tabla del 2.

Morado = Tabla del 7

Rosa = Tabla del 9

Azul = Tabla del 6

Blanco = Tabla del 12

Ejemplo de impresión de las tablas

2x1 = 2

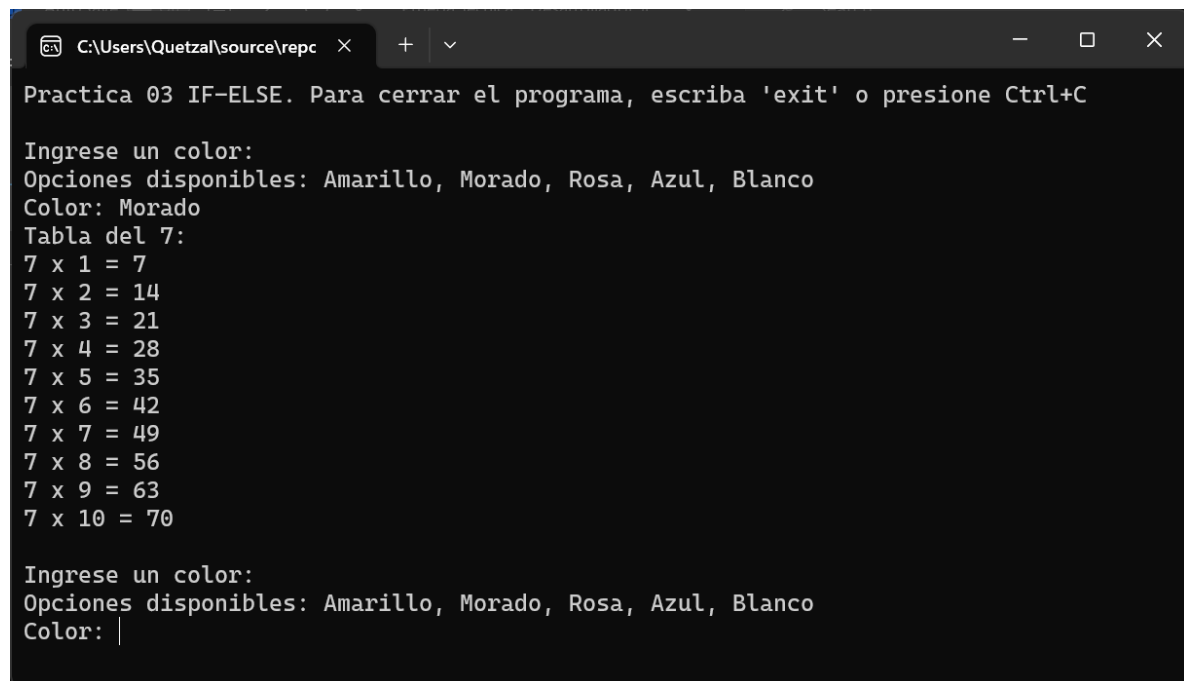
2x2 = 4

2x3 = 6

... etc.

NOTA: Se debe realizar este programa 2 veces, uno utilizando la Función IF y otro utilizando la Funcion SWITCH, ambos programas deben estar dentro de un Metodo que será llamado en el Main.

Resultado IF-ELSE



```
C:\Users\Quetzal\source\repc > Practica 03 IF-ELSE. Para cerrar el programa, escriba 'exit' o presione Ctrl+C

Ingrese un color:
Opciones disponibles: Amarillo, Morado, Rosa, Azul, Blanco
Color: Morado
Tabla del 7:
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Ingrese un color:
Opciones disponibles: Amarillo, Morado, Rosa, Azul, Blanco
Color: |
```

C# IF-ELSE

```
namespace Practice03a;

class Program
{
    static void Main(string[] _)
    {
        Console.WriteLine("Practica 03 IF-ELSE. Para cerrar el programa, escriba 'exit' o presione Ctrl+C");

        while (true)
        {
            Console.WriteLine("\nIngrese un color: ");
            Console.WriteLine("Opciones disponibles: Amarillo, Morado, Rosa, Azul, Blanco");
            Console.Write("Color: ");
            string? input = Console.ReadLine();

            // Salir del programa
            if (input == "exit")
            {
                break;
            }

            // Obtener el número del color
            int? output = GetColorNumber(input);

            if (output == null)
            {
                Console.WriteLine("El color ingresado no existe.");
            }
            else
            {
                // Generar la tabla de multiplicar
                string table = GenerateTable(output.Value);
                Console.WriteLine(table);
            }
        }
    }

    /// <summary>
    /// Obtener el número de color.
    /// </summary>
    /// <param name="color">Nombre del color.</param>
    /// <returns>El número del color o null si el color no existe.</returns>
    private static int? GetColorNumber(string? color)
    {
        if (color == "Amarillo")
        {
            return 2;
        }
        else if (color == "Morado")
        {
            return 7;
        }
        else if (color == "Rosa")
        {
            return 9;
        }
    }
}
```

```

    }
    else if (color == "Azul")
    {
        return 6;
    }
    else if (color == "Blanco")
    {
        return 12;
    }
    else
    {
        return null;
    }
}

/// <summary>
/// Generar la tabla de multiplicar de un número.
/// </summary>
/// <param name="number">El número.</param>
/// <returns>La tabla de multiplicar.</returns>
private static string GenerateTable(int number)
{
    // Buffer para almacenar la tabla
    List<string> table = new(new string[] { $"Tabla del {number}:" });

    // Generar la tabla
    for (int i = 1; i <= 10; i++)
    {
        table.Add($"{number} x {i} = {number * i}");
    }

    return string.Join("\n", table);
}
}

```

Resultado SWITCH

```
C:\Users\Quetzal\source\repc  X + v - □ X
Practica 03 SWITCH. Para cerrar el programa, escriba 'exit' o presione Ctrl+C

Ingrese un color:
Opciones disponibles: Amarillo, Morado, Rosa, Azul, Blanco
Color: Amarillo
Tabla del 2:
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

Ingrese un color:
Opciones disponibles: Amarillo, Morado, Rosa, Azul, Blanco
Color: |
```

C# SWITCH

```
namespace Practice03b;

class Program
{
    static void Main(string[] _)
    {
        Console.WriteLine("Practica 03 SWITCH. Para cerrar el programa,
escriba 'exit' o presione Ctrl+C");

        while (true)
        {
            Console.WriteLine("\nIngrese un color: ");
            Console.WriteLine("Opciones disponibles: Amarillo, Morado,
Rosa, Azul, Blanco");
            Console.Write("Color: ");
            string? input = Console.ReadLine();

            // Salir del programa
            if (input == "exit")
            {
                break;
            }

            // Obtener el número del color
            int? output = GetColorNumber(input);

            if (output == null)
            {

```

```

        Console.WriteLine("El color ingresado no existe.");
    }
    else
    {
        // Generar la tabla de multiplicar
        string table = GenerateTable(output.Value);
        Console.WriteLine(table);
    }
}

/// <summary>
/// Obtener el número de color.
/// </summary>
/// <param name="color">Nombre del color.</param>
/// <returns>El número del color o null si el color no existe.</returns>
private static int? GetColorNumber(string? color) => color switch
{
    "Amarillo" => 2,
    "Morado" => 7,
    "Rosa" => 9,
    "Azul" => 6,
    "Blanco" => 12,
    _ => null
};

/// <summary>
/// Generar la tabla de multiplicar de un número.
/// </summary>
/// <param name="number">El número.</param>
/// <returns>La tabla de multiplicar.</returns>
private static string GenerateTable(int number)
{
    // Buffer para almacenar la tabla
    List<string> table = new(new string[] { $"Tabla del {number}:" });

    // Generar la tabla
    for (int i = 1; i <= 10; i++)
    {
        table.Add($"{number} x {i} = {number * i}");
    }

    return string.Join("\n", table);
}
}

```

Los ejercicios de base de datos deben integrarse en un bloc de notas y adjuntarse con los ejercicios de código.

4.- Genere la sintaxis para crear una tabla llamada "Oficinas" con los siguientes campos

ID – Int

Nombre – Varchar

Direccion – Varchar

Ciudad - Varchar

Estado - Varchar

Pais – Varchar

División - Int

Donde el ID es un campo único y ningún campo acepta nulos.

```
CREATE TABLE Oficinas (  
    ID INT PRIMARY KEY,  
    Nombre VARCHAR(128) NOT NULL,  
    Direccion VARCHAR(128) NOT NULL,  
    Ciudad VARCHAR(128) NOT NULL,  
    Estado VARCHAR(128) NOT NULL,  
    Pais VARCHAR(128) NOT NULL,  
    Division INT NOT NULL,  
);
```


5.- Genere un Stored Procedure que se encargue realizar un insert en la tabla Oficinas (Declarada en el ejemplo anterior).

Reglas: El SP debe tener un parámetro que si el valor es 1 se realiza el insert, si es 0 se realiza el un update al campo direccion.

```
CREATE PROCEDURE dbo.insertOrUpdate
    @mode int
AS
BEGIN
    -- Si el modo es 1, insertar un nuevo registro.
    IF @mode = 1
    BEGIN
        INSERT INTO Oficinas (ID, Nombre, Direccion, Ciudad,
Estado, Pais, Division)
        VALUES (1, 'The Office', 'Direccion 1', 'Monterrey',
'Nuevo Leon', 'Mexico', 1)
    END
    -- Si el modo es 0, actualizar el campo direccion del
registro con ID = 1.
    ELSE IF @mode = 0
    BEGIN
        UPDATE Oficinas
        SET Direccion = 'Direccion 2'
        WHERE ID = 1
    END
END
```

Verificación

```
-- Borrar todos los registros.
TRUNCATE TABLE Oficinas;

-- Ver registros actuales.
SELECT * FROM Oficinas;

-- Insertar un nuevo registro.
EXEC dbo.insertOrUpdate 1;

-- Ver registros actuales.
SELECT * FROM Oficinas;
```


6.- Genera un trigger que se ejecute al insertar un registro en la tabla Oficina. Este trigger consultara la secuencia IDOFICINA_SEC y le integrara el valor de la secuencia al campo ID.

```
-- Crear una nueva secuencia llamada "IDOFICINA_SEC".
CREATE SEQUENCE IDOFICINA_SEC
  START WITH 100
  INCREMENT BY 1;

-- Crear un nuevo trigger llamado "OficinaInsertTrigger".
GO
CREATE OR ALTER TRIGGER OficinaInsertTrigger
  ON Oficinas
  INSTEAD OF INSERT
AS
BEGIN
  -- Insertar el valor de la secuencia IDOFICINA_SEC en el
  campo ID.
  INSERT INTO Oficinas
    (ID, Nombre, Direccion, Ciudad, Estado, Pais,
  Division)
    SELECT NEXT VALUE FOR IDOFICINA_SEC, Nombre, Direccion,
  Ciudad, Estado, Pais, Division
    FROM inserted;
END
GO
```

Verificación

```
-- Borrar todos los registros.
TRUNCATE TABLE Oficinas;

-- Ver registros actuales.
SELECT * FROM Oficinas;

-- Insertar un nuevo registro.
INSERT INTO Oficinas
  (Nombre, Direccion, Ciudad, Estado, Pais, Division)
VALUES
  ('The Office', 'Direccion 1', 'Monterrey', 'Nuevo Leon',
'Mexico', 1),
```

```

        ('The Office 2', 'Direccion 1', 'Guadalupe', 'Nuevo Leon',
'Mexico', 1),
        ('The Office 3', 'Direccion 1', 'San Pedro Garza Garcia',
'Nuevo Leon', 'Mexico', 1);

-- Ver registros actuales.
SELECT * FROM Oficinas;

```

Resultados

Results		Messages					
	ID	Nombre	Direccion	Ciudad	Estado	Pais	Division
1	100	The Office	Direccion 1	Monterrey	Nuevo Leon	Mexico	1
2	101	The Office 2	Direccion 1	Guadalupe	Nuevo Leon	Mexico	1
3	102	The Office 3	Direccion 1	San Pedro Garza Garcia	Nuevo Leon	Mexico	1