

EJERCICIO 1:

Crear un programa que simule la gestión de un inventario en una tienda. Utilizar un menú para agregar, eliminar, modificar y consultar productos en el inventario. Cada producto tendrá un código, nombre, cantidad y precio. Menú de opciones: 1. Agregar producto 2. Eliminar producto 3. Modificar producto 4. Consultar producto 5. Mostrar todos los productos 6. Salir

CÓDIGO EJERCICIO 1:

```
using System;
using System.Collections.Generic;

namespace GestionInventario
{
    // Clase Producto que define las propiedades de un producto
    class Producto
    {
        public intCodigo { get; set; }

        public stringNombre { get; set; }

        public intCantidad { get; set; }

        public doublePrecio { get; set; }

        // Constructor para inicializar un producto
        public Producto(int codigo, string nombre, int cantidad, double precio)
        {
            Codigo = codigo;
            Nombre = nombre;
            Cantidad = cantidad;
            Precio = precio;
        }

        // Método para mostrar los detalles del producto
        public void MostrarProducto()
        {
            Console.WriteLine($"Código: {Codigo}, Nombre: {Nombre}, Cantidad: {Cantidad}, Precio: {Precio}");
        }
    }

    class Program
    {
        // Lista genérica para almacenar productos
        static List<Producto> inventario = new List<Producto>();

        // Función para agregar un producto
        static void AgregarProducto()
    }
}
```

```

1 referencia
static void AgregarProducto()
{
    Console.Write("Ingrese el código del producto: ");
    int codigo = int.Parse(Console.ReadLine());

    Console.Write("Ingrese el nombre del producto: ");
    string nombre = Console.ReadLine();

    Console.Write("Ingrese la cantidad del producto: ");
    int cantidad = int.Parse(Console.ReadLine());

    Console.Write("Ingrese el precio del producto: ");
    double precio = double.Parse(Console.ReadLine());

    inventario.Add(new Producto(codigo, nombre, cantidad, precio));
    Console.WriteLine($"Producto {nombre} agregado con éxito.");
}

```

```

// Función para eliminar un producto
1 referencia
static void EliminarProducto()
{
    Console.Write("Ingrese el código del producto a eliminar: ");
    int codigo = int.Parse(Console.ReadLine());

    Producto producto = inventario.Find(p => p.Codigo == codigo);

    if (producto != null)
    {
        inventario.Remove(producto);
        Console.WriteLine($"Producto con código {codigo} eliminado con éxito.");
    }
    else
    {
        Console.WriteLine("Producto no encontrado.");
    }
}

```

```

    Console.WriteLine($"Producto con código {codigo} modificado con éxito.");
}
else
{
    Console.WriteLine("Producto no encontrado.");
}
}

// Función para consultar un producto
1 referencia
static void ConsultarProducto()
{
    Console.Write("Ingrese el código del producto a consultar: ");
    int codigo = int.Parse(Console.ReadLine());

    Producto producto = inventario.Find(p => p.Codigo == codigo);

    if (producto != null)
    {
        producto.MostrarProducto();
    }
}

```

1 referencia

```
static void ModificarProducto()
{
    Console.WriteLine("Ingrese el código del producto a modificar: ");
    int codigo = int.Parse(Console.ReadLine());

    Producto producto = inventario.Find(p => p.Codigo == codigo);

    if (producto != null)
    {
        Console.WriteLine("Ingrese el nuevo nombre del producto: ");
        string nuevoNombre = Console.ReadLine();

        Console.WriteLine("Ingrese la nueva cantidad del producto: ");
        int nuevaCantidad = int.Parse(Console.ReadLine());

        Console.WriteLine("Ingrese el nuevo precio del producto: ");
        double nuevoPrecio = double.Parse(Console.ReadLine());

        producto.Nombre = nuevoNombre;
        producto.Cantidad = nuevaCantidad;
        producto.Precio = nuevoPrecio;
    }
}
```

```
    Console.WriteLine($"Producto con código {codigo} modificado con éxito.");
}
else
{
    Console.WriteLine("Producto no encontrado.");
}
}
```

// Función para consultar un producto

1 referencia

```
static void ConsultarProducto()
{
    Console.WriteLine("Ingrese el código del producto a consultar: ");
    int codigo = int.Parse(Console.ReadLine());

    Producto producto = inventario.Find(p => p.Codigo == codigo);

    if (producto != null)
    {
        producto.MostrarProducto();
    }
}
```

// Función para mostrar todos los productos

```
static void MostrarTodosLosProductos()
{
    if (inventario.Count > 0)
    {
        Console.WriteLine("Productos en inventario:");
        foreach (var producto in inventario)
        {
            producto.MostrarProducto();
        }
    }
    else
    {
        Console.WriteLine("No hay productos en el inventario.");
    }
}
```

```

0 referencias
static void Main(string[] args)
{
    while (true)
    {
        Console.WriteLine("\nMenú de opciones:");
        Console.WriteLine("1. Agregar producto");
        Console.WriteLine("2. Eliminar producto");
        Console.WriteLine("3. Modificar producto");
        Console.WriteLine("4. Consultar producto");
        Console.WriteLine("5. Mostrar todos los productos");
        Console.WriteLine("6. Salir");
        Console.Write("Elige una opción: ");
        int opcion = int.Parse(Console.ReadLine());

        switch (opcion)
        {

```

```

            case 1:
                AgregarProducto();
                break;
            case 2:
                EliminarProducto();
                break;
            case 3:
                ModificarProducto();
                break;
            case 4:
                ConsultarProducto();
                break;
            case 5:
                MostrarTodosLosProductos();
                break;
            case 6:
                return;
            default:
                Console.WriteLine("Opción no válida, intenta de nuevo.");

```

```

            case 5:
                MostrarTodosLosProductos();
                break;
            case 6:
                return;
            default:
                Console.WriteLine("Opción no válida, intenta de nuevo.");
                break;
        }
    }
}

```

FUNCIONAMIENTO CÓDIGO EJERCICIO 1:

```
Menú de opciones:
1. Agregar producto
2. Eliminar producto
3. Modificar producto
4. Consultar producto
5. Mostrar todos los productos
6. Salir
Elige una opción: 1
Ingrese el código del producto: 100
Ingrese el nombre del producto: MANZANA ROJA
Ingrese la cantidad del producto: 1000
Ingrese el precio del producto: 10
Producto MANZANA ROJA agregado con éxito.

Menú de opciones:
1. Agregar producto
2. Eliminar producto
3. Modificar producto
4. Consultar producto
5. Mostrar todos los productos
6. Salir
Elige una opción: 4
Ingrese el código del producto a consultar: 100
Código: 100, Nombre: MANZANA ROJA, Cantidad: 1000, Precio: $10.00

Menú de opciones:
1. Agregar producto
2. Eliminar producto
3. Modificar producto
4. Consultar producto
5. Mostrar todos los productos
6. Salir
Elige una opción: 3
```

```
Menú de opciones:
1. Agregar producto
2. Eliminar producto
3. Modificar producto
4. Consultar producto
5. Mostrar todos los productos
6. Salir
Elige una opción: 3
Ingrese el código del producto a modificar: 100
Ingrese el nuevo nombre del producto: MANZANA VERDE
Ingrese la nueva cantidad del producto: 100
Ingrese el nuevo precio del producto: 10
Producto con código 100 modificado con éxito.

Menú de opciones:
1. Agregar producto
2. Eliminar producto
3. Modificar producto
4. Consultar producto
5. Mostrar todos los productos
6. Salir
Elige una opción: 2
Ingrese el código del producto a eliminar: 100
Producto con código 100 eliminado con éxito.

Menú de opciones:
1. Agregar producto
2. Eliminar producto
3. Modificar producto
4. Consultar producto
5. Mostrar todos los productos
6. Salir
```

EJERCICIO 2:

Desarrollar un programa que permita al usuario gestionar una cuenta bancaria. El programa deberá utilizar un menú que permita realizar diferentes operaciones sobre el saldo de la cuenta. Menú de opciones: 1. Consultar saldo 2. Depositar dinero 3. Retirar dinero 4. Salir. El programa debe permitir al usuario ingresar la cantidad para depositar o retirar, actualizar el saldo y mostrar los resultados. Si se elige la opción de retiro, debe verificar que el saldo sea suficiente.

CÓDIGO EJERCICIO 2:

```
using System;
using System.Collections.Generic;

namespace GestionInventario
{
    // Clase Producto que define las propiedades de un producto
    class Producto
    {
        public intCodigo { get; set; }
        public string Nombre { get; set; }
        public int Cantidad { get; set; }
        public double Precio { get; set; }

        // Constructor para inicializar un producto
        public Producto(int codigo, string nombre, int cantidad, double precio)
        {
            Codigo = codigo;
            Nombre = nombre;
            Cantidad = cantidad;
            Precio = precio;
        }
    }
}
```

```
    // Método para mostrar los detalles del producto
    public void MostrarProducto()
    {
        Console.WriteLine($"Código: {Codigo}, Nombre: {Nombre}, Cantidad: {Cantidad}, Precio: ${Precio:0.00}");
    }

    class Program
    {
        // Lista genérica para almacenar productos
        static List<Producto> inventario = new List<Producto>();

        // Función para agregar un producto
        static void AgregarProducto()
        {
            Console.Write("Ingrese el código del producto: ");
            int codigo = int.Parse(Console.ReadLine());

            Console.Write("Ingrese el nombre del producto: ");
            string nombre = Console.ReadLine();

            Console.Write("Ingrese la cantidad del producto: ");
        }
    }
}
```

```
// Función para agregar un producto
static void AgregarProducto()
{
    Console.WriteLine("Ingrese el código del producto: ");
    int codigo = int.Parse(Console.ReadLine());

    Console.WriteLine("Ingrese el nombre del producto: ");
    string nombre = Console.ReadLine();

    Console.WriteLine("Ingrese la cantidad del producto: ");
    int cantidad = int.Parse(Console.ReadLine());

    Console.WriteLine("Ingrese el precio del producto: ");
    double precio = double.Parse(Console.ReadLine());

    inventario.Add(new Producto(codigo, nombre, cantidad, precio));
    Console.WriteLine($"Producto {nombre} agregado con éxito.");
}
}
```

```
// Función para modificar un producto
static void ModificarProducto()
{
    Console.WriteLine("Ingrese el código del producto a modificar: ");
    int codigo = int.Parse(Console.ReadLine());

    Producto producto = inventario.Find(p => p.Codigo == codigo);

    if (producto != null)
    {
        Console.WriteLine("Ingrese el nuevo nombre del producto: ");
        string nuevoNombre = Console.ReadLine();

        Console.WriteLine("Ingrese la nueva cantidad del producto: ");
        int nuevaCantidad = int.Parse(Console.ReadLine());

        Console.WriteLine("Ingrese el nuevo precio del producto: ");
        double nuevoPrecio = double.Parse(Console.ReadLine());

        producto.Nombre = nuevoNombre;
        producto.Cantidad = nuevaCantidad;
        producto.Precio = nuevoPrecio;
    }
}
```

```
        Console.WriteLine($"Producto con código {codigo} modificado con éxito.");
    }
    else
    {
        Console.WriteLine("Producto no encontrado.");
    }
}
}
```

```
// Función para consultar un producto
static void ConsultarProducto()
{
    Console.WriteLine("Ingrese el código del producto a consultar: ");
    int codigo = int.Parse(Console.ReadLine());

    Producto producto = inventario.Find(p => p.Codigo == codigo);

    if (producto != null)
    {
        producto.MostrarProducto();
    }
    else
    {
        Console.WriteLine("Producto no encontrado.");
    }
}
}
```

```
// Función para mostrar todos los productos
static void MostrarTodosLosProductos()
{
    if (inventario.Count > 0)
    {
        Console.WriteLine("Productos en inventario:");
        foreach (var producto in inventario)
        {
            producto.MostrarProducto();
        }
    }
    else
    {
        Console.WriteLine("No hay productos en el inventario.");
    }
}
```

```
static void Main(string[] args)
{
    while (true)
    {
        Console.WriteLine("\nMenú de opciones:");
        Console.WriteLine("1. Agregar producto");
        Console.WriteLine("2. Eliminar producto");
        Console.WriteLine("3. Modificar producto");
        Console.WriteLine("4. Consultar producto");
        Console.WriteLine("5. Mostrar todos los productos");
        Console.WriteLine("6. Salir");
        Console.Write("Elige una opción: ");
        int opcion = int.Parse(Console.ReadLine());
```

```
switch (opcion)
{
    case 1:
        AgregarProducto();
        break;
    case 2:
        EliminarProducto();
        break;
    case 3:
        ModificarProducto();
        break;
    case 4:
        ConsultarProducto();
        break;
    case 5:
        MostrarTodosLosProductos();
        break;
    case 6:
        return;
    default:
        Console.WriteLine("Opción no válida, intenta de nuevo.");
}
```



```

        break;
    case 4:
        ConsultarProducto();
        break;
    case 5:
        MostrarTodosLosProductos();
        break;
    case 6:
        return;
    default:
        Console.WriteLine("Opción no válida, intenta de nuevo.");
        break;
    }
}
}
}
}

```

FUNCIONAMIENTO DE CODIGO DE EJERCICIO 2:

```

Menú de opciones:
1. Consultar saldo
2. Depositar dinero
3. Retirar dinero
4. Salir
Elige una opción: 2

Ingrese la cantidad a depositar: 1000000
Se han depositado $1000000.00. Nuevo saldo: $1000000.00

Menú de opciones:
1. Consultar saldo
2. Depositar dinero
3. Retirar dinero
4. Salir
Elige una opción: 1

El saldo actual es: $1000000.00

Menú de opciones:
1. Consultar saldo
2. Depositar dinero
3. Retirar dinero
4. Salir
Elige una opción: 3

Ingrese la cantidad a retirar: 10000
Se han retirado $10000.00. Nuevo saldo: $990000.00

Menú de opciones:
1. Consultar saldo
2. Depositar dinero
3. Retirar dinero
4. Salir
Elige una opción: 

```

EJERCICIO 3:

Desarrollar un programa que se comporte como un diccionario Inglés-Español; esto es, solicitará una palabra en inglés y escribirá la correspondiente palabra en español. Para hacer más sencillo el ejercicio, el número de parejas de palabras estará limitado a 5. Por ejemplo, suponer que introducimos las siguientes parejas de palabras: book libro green verde mouse ratón Una vez finalizada la introducción de las listas de palabras, pasamos al modo traducción, de forma que si introducimos green, la respuesta ha de ser verde. Si la palabra no se encuentra, se emitirá un mensaje que lo indique. El programa constará de dos métodos, aparte de Main(): 1. crearDiccionario(). Este método creará el diccionario. 2. traducir(). Este método realizará la labor de traducción

CÓDIGO EJERCICIO 3:

```
using System;
using System.Collections.Generic;

namespace DiccionarioInglesEspanol
{
    class Program
    {
        // Método para crear el diccionario con las 5 parejas de palabras

        static Dictionary<string, string> crearDiccionario()
        {
            // Crear un diccionario para almacenar las traducciones
            Dictionary<string, string> diccionario = new Dictionary<string, string>();

            Console.WriteLine("Introduce 5 parejas de palabras (Inglés - Español):");

            // Limitar a 5 entradas
            for (int i = 0; i < 5; i++)
            {
                Console.Write($"Introduce la palabra en inglés #{i + 1}: ");
                string palabraIngles = Console.ReadLine().ToLower();

                for (int j = 0; j < 5; j++)
                {
                    Console.Write($"Introduce la palabra en español para '{palabraIngles}': ");
                    string palabraEspanol = Console.ReadLine().ToLower();

                    diccionario.Add(palabraIngles, palabraEspanol);
                    Console.WriteLine($"Pareja '{palabraIngles} - {palabraEspanol}' agregada.");
                }
            }

            return diccionario;
        }

        // Método para realizar la traducción
        static void traducir(Dictionary<string, string> diccionario)
        {
            Console.WriteLine("\nModo traducción. Escribe una palabra en inglés para obtener su traducción (escribe una palabra o 'fin' para salir):");

            while (true)
            {
                string palabraIngles = Console.ReadLine().ToLower();

                if (palabraIngles == "fin")
                {
                    break;
                }

                if (diccionario.ContainsKey(palabraIngles))
                {
                    string palabraEspanol = diccionario[palabraIngles];
                    Console.WriteLine($"Traducción de '{palabraIngles}': {palabraEspanol}");
                }
                else
                {
                    Console.WriteLine($"No se encontró la traducción para '{palabraIngles}'");
                }
            }
        }
    }
}
```

```

    {
        Console.WriteLine("Introduce una palabra en inglés: ");
        string palabraIngles = Console.ReadLine().ToLower();

        if (palabraIngles == "salir")
        {
            break;
        }

        if (diccionario.ContainsKey(palabraIngles))
        {
            string traduccion = diccionario[palabraIngles];
            Console.WriteLine($"La traducción de '{palabraIngles}' es: {traduccion}");
        }
        else
        {
            Console.WriteLine("La palabra no se encuentra en el diccionario.");
        }
    }
}

```

FUNCIONAMIENTO DE CÓDIGO DE EJERCICIO 3:

```

Introduce 5 parejas de palabras (Inglés - Español):
Introduce la palabra en inglés #1: CAR
Introduce la palabra en español para 'car': CARRO
Pareja 'car - carro' agregada.
Introduce la palabra en inglés #2: RUN
Introduce la palabra en español para 'run': CORRER
Pareja 'run - correr' agregada.
Introduce la palabra en inglés #3: PLAY
Introduce la palabra en español para 'play': JUGAR
Pareja 'play - jugar' agregada.
Introduce la palabra en inglés #4: LOVE
Introduce la palabra en español para 'love': AMOR
Pareja 'love - amor' agregada.
Introduce la palabra en inglés #5: HI
Introduce la palabra en español para 'hi': HOLA
Pareja 'hi - hola' agregada.

Modo traducción. Escribe una palabra en inglés para obtener su traducción (escribe 'salir' para terminar):
Introduce una palabra en inglés:

```

CAPTURA DE COMMIT

Commits

main
All users
All time

Commits on Oct 19, 2024

archivos

EnocSalinas committed 1 minute ago

792f5e5

LINK DEL REPOSITORIO

<https://github.com/EnocSalinas/TRABAJOFINAL>