

REPOSITORIO

Listas genéricas en C# - List<T>



```

namespace Productos
{
    6 references
    public struct Producto
    {
        public string Nombre,Codigo;
        public int Cantidad;
        public double Precio;

        1 reference
        public Producto(string nombre, string codigo, int cantidad, double precio)
        {
            Nombre = nombre;
            Codigo = codigo;
            Cantidad = cantidad;
            Precio = precio;
        }
    }
}

```

```

public class Productos
{
    private static List<Producto> productos = new List<Producto>();

    0 references
    public static void Main()
    {
        int opcion;
        do
        {
            Console.WriteLine("===== Productos =====");
            Console.WriteLine("1. Nuevo Producto");
            Console.WriteLine("2. Todos los Productos");
            Console.WriteLine("3. Modificar Producto");
            Console.WriteLine("4. Eliminar Producto");
            Console.WriteLine("5. Buscar Producto");
            Console.WriteLine("6. Salir");
            Console.WriteLine("\nSeleccione una opción: ");
            opcion = int.Parse(Console.ReadLine());

            switch (opcion)
            {
                case 1: AgregarProducto(); break;
                case 2: LeerProductos(); break;
                case 3: ActualizarProducto(); break;
                case 4: EliminarProducto(); break;
                case 5: BuscarProducto(); break;
                case 6: break;
                default: Console.Clear(); break;
            }
        } while (opcion != 6);
    }
}

```

```

public static void AgregarProducto()
{
    Console.Write("Ingrese el nombre del Producto: ");
    string nombre = Console.ReadLine();
    Console.Write("Ingrese elCodigo al que pertenece el producto: ");
    string codigo = Console.ReadLine();
    Console.Write("Ingrese la cantidad del producto en stock: ");
    int cantidad = int.Parse(Console.ReadLine());
    Console.Write("Ingrese el precio unitario del producto: ");
    double precio = double.Parse(Console.ReadLine());

    Producto producto = new Producto(nombre, codigo, cantidad, precio);
    productos.Add(producto);

    Console.WriteLine("Producto agregado exitosamente.");
    Console.WriteLine("Presione cualquier tecla para continuar...");
    Console.ReadKey();
    Console.Clear();
}

```

```

public static void LeerProductos()
{
    if (productos.Count == 0)
    {
        Console.WriteLine("No hay productos para mostrar.");
    }
    else
    {
        Console.WriteLine("\n\tProductos\n");
        foreach (var producto in productos)
        {
            Console.WriteLine("-----");
            Console.WriteLine($"Nombre: {producto.Nombre}");
            Console.WriteLine($"Codigo: {producto.Codigo}");
            Console.WriteLine($"Cantidad: {producto.Cantidad} unds");
            Console.WriteLine($"Precio unitario: ${producto.Precio}");
            Console.WriteLine("-----");
            Console.WriteLine();
        }
    }

    Console.WriteLine("Presione cualquier tecla para continuar...");
    Console.ReadKey();
    Console.Clear();
}

```

```

public static void EliminarProducto()
{
    Console.WriteLine("Ingrese el Nombre del Producto a eliminar: ");
    string search = Console.ReadLine();
    var producto = productos.FirstOrDefault(p => p.Nombre.ToLower() == search.ToLower());

    if (producto.Nombre != null)
    {
        productos.Remove(producto);
        Console.WriteLine("El Producto fue eliminado con éxito.");
    }
    else
    {
        Console.WriteLine("El Producto no se ha encontrado.");
    }

    Console.WriteLine("Presione cualquier tecla para continuar...");
    Console.ReadKey();
    Console.Clear();
}

1 reference
public static void ActualizarProducto()
{
    Console.WriteLine("Ingrese el nombre del Producto: ");
    string search = Console.ReadLine();
    var index = productos.FindIndex(p => p.Nombre.ToLower() == search.ToLower());

    if (index != -1)
    {
        Producto producto = productos[index];

        Console.WriteLine("Ingrese el nuevoCodigo: ");
        producto.Codigo = Console.ReadLine();
        Console.WriteLine("Ingrese la nueva cantidad: ");
        producto.Cantidad = int.Parse(Console.ReadLine());
        Console.WriteLine("Ingrese el nuevo precio: ");
        producto.Precio = double.Parse(Console.ReadLine());

        productos[index] = producto;

        Console.WriteLine("El producto fue actualizado con éxito.");
    }
    else
    {
        Console.WriteLine("El producto no se ha encontrado.");
    }

    Console.WriteLine("Presione cualquier tecla para continuar.");
    Console.ReadKey();
    Console.Clear();
}

```

```

// BuscarProducto
public static void BuscarProducto()
{
    Console.Write("Ingrese el nombre del Producto: ");
    string search = Console.ReadLine();
    var producto = productos.FirstOrDefault(p => p.Nombre.ToLower() == search.ToLower());

    if (producto.Nombre != null)
    {
        Console.WriteLine($"\\n\\tInformacion del producto buscado.\\n");
        Console.WriteLine("-----");
        Console.WriteLine($"Nombre: {producto.Nombre}\\nCodigo: {producto.Codigo} \\nCantidad: {producto.Cantidad} \\nPrecio: {producto.Precio}");
        Console.WriteLine("-----");
    }
    else
    {
        Console.WriteLine("El Producto no se ha encontrado.");
    }

    Console.WriteLine("Presione cualquier tecla para continuar.");
    Console.ReadKey();
    Console.Clear();
}

```

Productos

1. Nuevo Producto
2. Todos los Productos
3. Modificar Producto
4. Eliminar Producto
5. Buscar Producto
6. Salir

Seleccione una opción: 1

Ingrese el nombre del Producto: Gaseosa

Ingrese el Codigo al que pertenece el producto: 100

Ingrese la cantidad del producto en stock: 200

Ingrese el precio unitario del producto: 10

Producto agregado exitosamente.

Presione cualquier tecla para continuar...

|

Productos

1. Nuevo Producto
2. Todos los Productos
3. Modificar Producto
4. Eliminar Producto
5. Buscar Producto
6. Salir

Seleccione una opción: 2

Productos

Nombre: Gaseosa

Codigo: 100

Cantidad: 200 unds

Precio unitario: \$10

Presione cualquier tecla para continuar...

```
0 references
internal class Listas_Tuplas
{
    0 references
    static void Main(string[] args)
    {
        List<Tuple<string, string>> diccionario = CrearDiccionario();
        Traducir(diccionario);
    }

    1 reference
    public static List<Tuple<string, string>> CrearDiccionario()
    {
        List<Tuple<string, string>> diccionario = new List<Tuple<string, string>>();
        Console.WriteLine("\tIngreso de palabras al diccionario");
        Console.WriteLine("\n_____");
        for (int i = 0; i < 3; i++)
        {
            Console.WriteLine($"Ingrese palabra #{i + 1} en Ingles: ");
            string ing = Console.ReadLine();
            Console.WriteLine($"Ingrese la traduccion de la palabra #{i + 1}: ");
            string esp = Console.ReadLine();
            diccionario.Add(new Tuple<string, string>(ing, esp));
        }
        Console.WriteLine("\n_____");
        return diccionario;
    }
}
```

```

1 reference
public static void Traducir(List<Tuple<string, string>> diccionario)
{
    Console.Clear();
    Console.WriteLine("Ingrese la palabra a traducir: ");
    string pal = Console.ReadLine();
    bool e = false;
    foreach (var duo in diccionario)
    {
        if (duo.Item1.Equals(pal, StringComparison.OrdinalIgnoreCase))
        {
            e = true;
            Console.WriteLine($"La traduccion de {pal} es: {duo.Item2}"); break;
        }
        if (duo.Item2.Equals(pal, StringComparison.OrdinalIgnoreCase))
        {
            e = true;
            Console.WriteLine($"La traduccion de {pal} es: {duo.Item1}"); break;
        }
    }
    if (!e)
    {
        Console.WriteLine($"La palabra {pal} no se ha encontrado en el diccionario");
    }
}
}

```

Ingrese la palabra a traducir: Hand
La traduccion de Hand es: Mano

```

public struct Cuenta
{
    public double Saldo;

    0 references
    public Cuenta(double saldo)
    {
        Saldo = saldo;
    }
}

0 references
internal class Banco
{
    0 references
    public class CuentaBancaria
    {
        public static Cuenta cuenta = new Cuenta();

        0 references
        public static void Main()
        {
            cuenta.Saldo = 5000;
            int opcion;
            do
            {
                Console.WriteLine("===== Cuenta Bancaria =====");
                Console.WriteLine(" ");
                Console.WriteLine("1. Consultar Saldo");
                Console.WriteLine("2. Depositar Dinero");
                Console.WriteLine("3. Retirar Dinero");
                Console.WriteLine("4. Salir");
                Console.WriteLine(" ");
                Console.WriteLine(" ");
                Console.WriteLine("\nSeleccione una opción: ");
                opcion = int.Parse(Console.ReadLine());
            }

```

```

        switch (opcion)
        {
            case 1: ConsultarSaldo(); break;
            case 2: Deposito(); break;
            case 3: Retiro(); break;
            case 4: break;
            default: Console.Clear(); break;
        }
    } while (opcion != 4);
}

1 reference
public static void ConsultarSaldo()
{
    Console.WriteLine();
    Console.Write($"El saldo actual de la cuenta es:{cuenta.Saldo:C2}");
    Console.WriteLine("\n");
    Console.WriteLine("Presione cualquier tecla para continuar...");
    Console.ReadKey();
    Console.Clear();
}

1 reference
public static void Deposito()
{
    Console.Write("Ingrese la cantidad a depositar: $");
    float dep = float.Parse(Console.ReadLine());
    if (dep > 0)
        cuenta.Saldo = cuenta.Saldo + dep;
    else
        Console.WriteLine("El deposito debe ser mayor a 0");
    Console.WriteLine("Presione cualquier tecla para continuar...");
    Console.ReadKey();
    Console.Clear();
}

1 reference
public static void Retiro()
{
    Console.Write("Ingrese la cantidad a Retirar: ");
    float ret = float.Parse(Console.ReadLine());
    if (ret < cuenta.Saldo)
        cuenta.Saldo = cuenta.Saldo - ret;
    else
        Console.WriteLine("La cuenta no dispone de fondos suficientes para realizar el retiro...");
    Console.WriteLine("Presione cualquier tecla para continuar...");
    Console.ReadKey();
    Console.Clear();
}

```


Cuenta Bancaria

1. Consultar Saldo
2. Depositar Dinero
3. Retirar Dinero
4. Salir

Seleccione una opción: 1

El saldo actual de la cuenta es:\$5,000.00

Presione cualquier tecla para continuar...

|