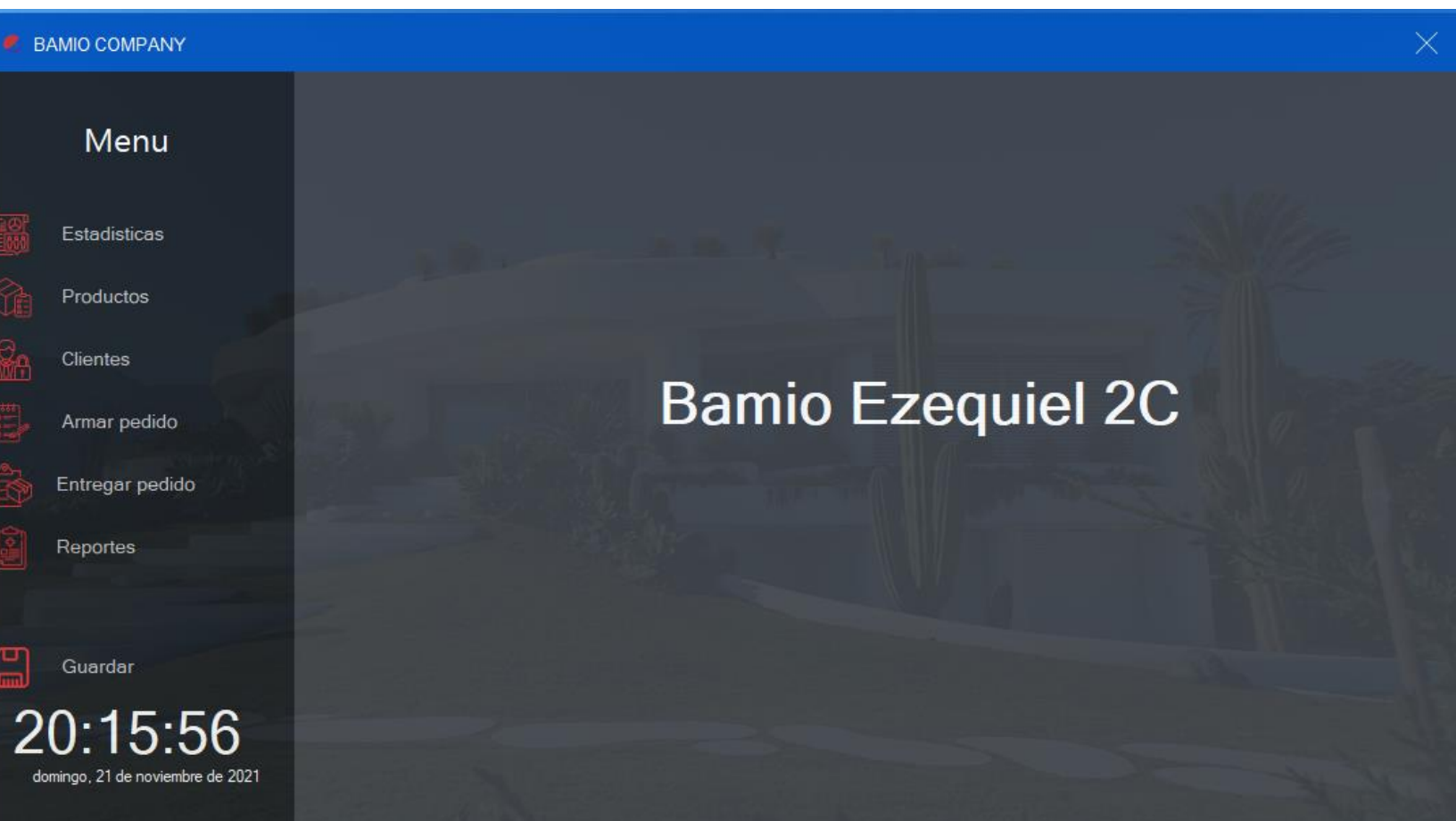


BAMIO EZEQUIEL 2C



**Trabajo practico:
Fabrica de Productos**



CONTENIDOS

1. TP3

- 1.1. Explicación.
- 1.2. Video.
- 1.3. Partes del Proyecto.
- 1.4. Temas utilizados.

2. TP4

- 2.1. Explicación.
- 2.2. Video.
- 2.3. Partes del Proyecto.
- 2.4. Temas utilizados.



1. TP3

1.1. Explicación

EL PROYECTO SIMULA UNA PEQUEÑA FABRICA DE PRODUCTOS EN LA QUE SE PUEDE CREAR PRODUCTOS, CLIENTES Y PEDIDOS. AL HACER UN PEDIDO SE LE GUARDARA UNA FACTURA AL CLIENTE, QUE DICHA FACTURA SE PUEDE OBSERVAR EN EL APARTADO DE "CLIENTES".

1.2. Video

<https://www.youtube.com/watch?v=Lx9DYChFec>

1.3. Partes del proyecto

Producto: En este apartado se podrá ver la lista de productos, crear nuevos, cargar y guardar un archivo de productos.

Clientes: En este apartado se podrá ver la lista de clientes, crear nuevos, cargar y guardar un archivo de clientes.

Estadísticas: En este apartado se podrá ver las estadísticas de la fabrica, el total de compras de clientes, cuantas gaseosas o cervezas compro el cliente y el mejor cliente, cuanto gasto y cuantos productos compro.

Armar pedido: Se podrá armar los pedidos para los clientes. Para armar un pedido es necesario seleccionar los productos y la dirección del cliente.

1.4. Temas utilizados

Archivos: Proyecto "IO".

Usado en: FrmCliente, FmrListaProducto, FrmMenu.

Excepciones: Usado en: Clientes, Fabrica, Productos, Archivos, Formularios.

Test Unitarios: Proyecto "Test"

Tipos Genericos: Usado en IArchivos, IIdentificador.

Serialización: Usado en para cargar Clientes, y productos en Formularios usando Json.

Interfaces: Usado en Archivos, Fabrica



BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

Nuevo cliente

ID

Nombre

Apellido

Direccion

Telefono

6

Crear cliente

Guarda

Cargar

ID del cliente

Eliminar

Listado de Clientes

ID	Nombre	Apellido	Direccion	Telefono
5	Carlos	Surry	2024 Koby Drive	25871963
4	Roberto	Oldis	3833 Hara Islands	98741203
3	Federico	Sanchez	31013 Freida Mall	36528741
2	Conni	Milley	67581 Terry Pass	58962415
1	Johann	Balston	189 Mante Prairie	45879658

23:43:44

jueves, 4 de noviembre de 2021

BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

Ventas

Total ventas

\$ 4950

Gaseosas

\$ 3250

Cervezas

\$ 1700

Fabrica

Total mercaderia

\$ 16595

N° Gaseosas

6

N° Cervezas

7

N° Botellas

7

N° Latas

6

Mejor cliente

Comprado

\$ 1700

Productos

1

ID

2

Nombre

Conni

Apellido

Milley

23:43:35

jueves, 4 de noviembre de 2021

Bamio Ezequiel (C) Copyright (2021) All Rights Reserved

4

BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

Nuevo cliente

ID

Nombre

Apellido

Direccion

Telefono

6

Crear cliente

Guarda

Cargar

Eliminar

Listado de Clientes

ID	Nombre	Apellido	Direccion	Telefono
5	Carlo		2024 Koby Drive	25871963
4	Robe		3833 Hara Islands	98741203
3	Fede		31013 Freida Mall	36528741
2	Conn		67581 Terry Pass	58962415
1	Joha		189 Mante Prairie	45879658

Facturas

Factura #:

Id: 2

Nombre: Conni

Apellido: Milley

Direccion: 67581 Terry Pass

Telefono: 58962415

Productos:

Codigo: 77303

Nombre: Corona

Precio: 1700

Total: \$1700

Aceptar

23:43:48

jueves, 4 de noviembre de 2021

BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

3833 Hara Islands

Confirmar

Codigo	Producto	Nombre	Precio	Tamaño	Tipo
53765	Cerveza	Heineken	1455	Grande	Lata
95901	Gaseosa	Fanta	1800	Grande	Botella
15528	Gaseosa	Fanta	1025	Chico	Botella
84790	Cerveza	Brahama	980	Grande	Botella
87824	Cerveza	Brahama	565	Grande	Lata
42370	Gaseosa	Pepsi	1450	Grande	Botella
72372	Gaseosa	Sprite	650	Grande	Lata
96961	Gaseosa	CocaCola	1800	Grande	Botella
57412	Gaseosa	CocaCola	200	Chico	Lata
52003	Cerveza	Corona	3250	Grande	Botella
28831	Cerveza	Corona	1200	Chico	Lata

23:44:04

jueves, 4 de noviembre de 2021



Menu

Estadísticas

Productos

Cientes

Armar pedido

Guarda

Cargar

Agregar producto

Codigo del producto

Eliminar

Listado de Productos

	Codigo	Producto	Nombre	Precio	Tamaño	Tipo
	53765	Cerveza	Heineken	1455	Grande	Lata
	95901	Gaseosa	Fanta	1800	Grande	Botella
	15528	Gaseosa	Fanta	1025	Chico	Botella
	28831	Cerveza	Corona	1200	Chico	Lata
	52003	Cerveza	Corona	3250	Grande	Botella
	84790	Cerveza	Brahama	980	Grande	Botella
	87824	Cerveza	Brahama	565	Grande	Lata
	42370	Gaseosa	Pepsi	1450	Grande	Botella
	72272	Gaseosa	Coca	650	Grande	Lata

23:43:40

jueves, 4 de noviembre de 2021

1 referencia | cambios | autores, cambios

```

private void btnGuardar_Click(object sender, EventArgs e)
{
    if (!String.IsNullOrEmpty(this.nombreArchivo))
    {
        this.serializacionCliente.Escribir(this.nombreArchivo, this.fabrica.Clientes, false);
        MessageBox.Show("Archivo guardado exitosamente!", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        saveFileDialog = new SaveFileDialog();

        saveFileDialog.Title = "Clientes";
        saveFileDialog.Filter = "json files (*.json)|*.json";

        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            string path = $"{saveFileDialog.FileName}";
            if (Path.GetExtension(path) == ".json")
            {
                this.serializacionCliente.Escribir(path, this.fabrica.Clientes, false);
                MessageBox.Show("Archivo guardado exitosamente!", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
    }
}

```



```

1 referencia | -cambios | -autores, -cambios
private void btnCargar_Click(object sender, EventArgs e)
{
    string archivoNombre = string.Empty;

    openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "json files (*.json)|*.json";

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        archivoNombre = openFileDialog.FileName;
        try
        {
            if (Path.GetExtension(archivoNombre) == ".json")
            {
                this.fabrica.Clientes = this.serializacionCliente.Leer(archivoNombre);
                this.RefrescarLista();
                MessageBox.Show("Se cargo correctamente", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
            this.nombreArchivo = archivoNombre;
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Excepcion: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

#region Operators Producto
8 referencias | -cambios | -autores, -cambios
public static Fabrica operator +(Fabrica fabrica, Producto producto)
{
    if (fabrica != producto)
    {
        fabrica.Productos.Add(producto);
    }
    else
    {
        throw new ProductoInvalidoException("Ese producto ya existe en la lista.");
    }

    return fabrica;
}

3 referencias | -cambios | -autores, -cambios
public static Fabrica operator -(Fabrica fabrica, Producto producto)
{
    if (fabrica == producto)
    {
        fabrica.Productos.Remove(producto);
    }
    else
    {
        throw new ProductoInvalidoException("No se puede eliminar un producto que no esta en la lista.");
    }
}

```



```

1 referencia | -cambios | -autores, -cambios
public interface IIdentificador<T>
{
    2 referencias | -cambios | -autores, -cambios
    public int OrdenarPorId(T objeto1, T objeto2);
    2 referencias | -cambios | -autores, -cambios
    public int GenerarId();
}

```

```

/// Verifica que no se pueda agregar 2 veces el mismo cliente
/// </summary>
[TestMethod]
[ExpectedException(typeof(ClienteInvalidoException))]
0 referencias | -cambios | -autores, -cambios
public void AgregarCliente()
{
    Fabrica fabrica = new Fabrica("Test");
    Cliente cliente = new Cliente(3, "test", "test", "test", 54578781);

    fabrica += cliente;
    fabrica += cliente;
}

/// <summary>
/// Verifica que no se pueda eliminar un cliente que no esta en la lista
/// </summary>
[TestMethod]
[ExpectedException(typeof(ClienteInvalidoException))]
0 referencias | -cambios | -autores, -cambios
public void EliminarCliente()
{
    Fabrica fabrica = new Fabrica("Test");
    Cliente cliente = new Cliente(3, "test", "test", "test", 54578781);
    Cliente cliente2 = new Cliente(2, "asd", "2ws", "sdq", 54578781);

    fabrica += cliente;
    fabrica -= cliente2;
}

/// <summary>
/// Verifica que la id no sea invalida
/// </summary>

```



2. TP4

2.1. Explicación

Es una version mejorada del TP3. Se le añadieron nuevas cosas como por ejemplo “Entregar pedido” que simular las entregas de los pedidos a los clientes. Un reporte, que cuando finaliza la entrega se crea un reporte en la base de datos y se muestra en el apartado “Reportes”. Se modificaron las estadísticas.

2.2. Video

<https://www.youtube.com/watch?v=O0Qwk476LV8>

2.3. Nuevas partes del proyecto

Como se mencionó arriba se mejoraron y agregaron algunas cosas sobre el TP3, como el las estadísticas, se mostraran el porcentaje de tipo de productos más vendidos, los tipos de envase más vendidos, los días con mayores ventas, dinero que se recaudó con las ventas de los productos (Se suman cuando se crea una factura, el dinero no se asigna a la caja hasta que se simule la entrega).

También están las estadísticas actuales de la fábrica, como por ejemplo la cantidad de cada tipo de producto en stock, el porcentaje de los clientes según su sexo y el estado de los clientes (Al día o deudores) y por último el dinero total de la fabrica. Mediante el calendario se podrá filtrar los datos, excepto los datos actuales de la fabrica.

Otras de las cosas que se agregaron fue “Entregar pedido” simula la entrega de los pedidos, pueden pagar o no los clientes (Las facturas pagas se eliminan pero quedan en el historial de facturas), una vez finalizado la entrega se crea un reporte a la base de datos.

Reporte trae y muestra los reportes guardados de la base de datos.

El botón guardar, va a guardar toda la fabrica en formato XML y el historial de facturas en formato Json (Los archivo se encuentra en el bin de VistaForm).

*Haciendo clic en el label “Menú” se hardcodear los datos para probar la aplicación sin datos y con datos y facilitar la prueba.

2.4. Temas utilizados

Base de datos: Utilizado en Reportes. Script en la carpeta del proyecto

Delegados, Hilos, Eventos, Expresiones Lambda: Utilizado en Entregar pedidos.

Métodos de extensión: Utilizado en la clase DataGridView para facilitar al insertar los datos.

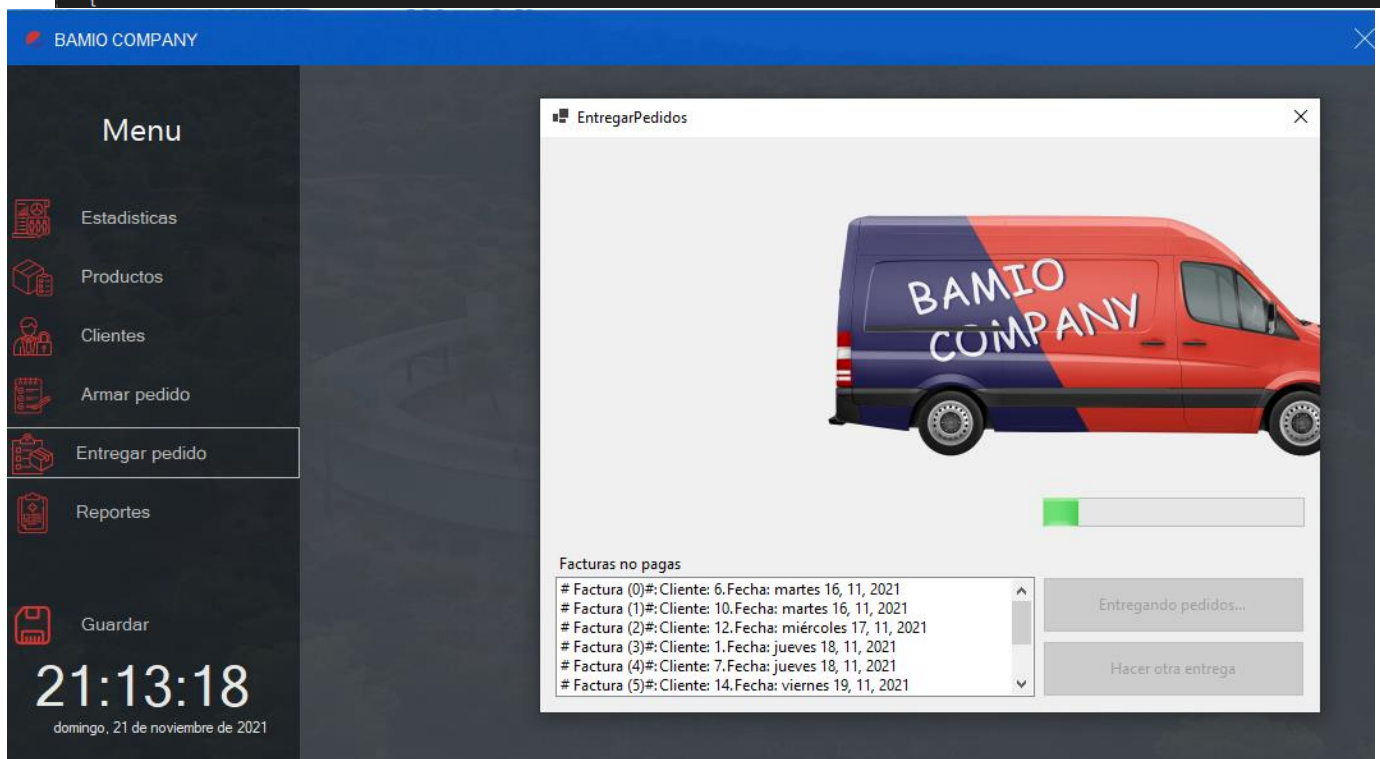


```
3 referencias | 1/1 pasando | - cambios | -autores, - cambios
public void Guardar(Reporte reporte)
{
    using (SqlConnection sqlConnection = new SqlConnection(this.cadenaConexion))
    {
        try
        {
            sqlConnection.Open();
            string command = "INSERT INTO Reportes(totalPedidos, totalPagos, totalNoPagos, recaudado, fecha) VALUES (@totalPedidos, @totalPagos, @totalNoPagos, @recaudado, @fecha)";

            SqlCommand sqlCommand = new SqlCommand(command, sqlConnection);
            sqlCommand.Parameters.AddWithValue("totalPedidos", reporte.TotalPedidos);
            sqlCommand.Parameters.AddWithValue("totalPagos", reporte.TotalPagos);
            sqlCommand.Parameters.AddWithValue("totalNoPagos", reporte.TotalNoPagos);
            sqlCommand.Parameters.AddWithValue("recaudado", reporte.Recaudado);
            sqlCommand.Parameters.AddWithValue("fecha", reporte.Fecha);

            sqlCommand.ExecuteNonQuery();
        }
        catch (Exception)
        {
            throw;
        }
    }
}

4 referencias | 1/1 pasando | - cambios | -autores, - cambios
public List<Reporte> Leer()
{
}
```



BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

Entregar pedido

Reportes

Guardar

21:14:04

domingo, 21 de noviembre de 2021

Tipo de Producto mas vendido

Gaseosas 26%

Cervezas 73%

Tipo de Envase mas vendido

Botella 60%

Lata 40%

Dinero de las ventas

\$26970

noviembre de 2021

lu.	ma.	mi.	ju.	vi.	sá.	do.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Hoy: 21/11/2021

Dia con mayor ventas

Lunes	0%
Martes	16%
Miercoles	8%
Jueves	25%
Viernes	33%
Sabado	16%

Cliente	Cantidad de Productos	Dinero Total
Plevey Carlos	1	3292
Fitzroy Juan	0	0
Rickarsey Pedro	0	0
Govenlock Sara	0	0
Gofford Florencia	0	0

Actual

Producto en stock

Gaseosas 21

Cervezas 8

Cientes

Hombres 46%

Mujeres 53%

Estado

Al día 53%

Deudores 46%

Dinero total de la fabrica

\$6963

referencias | - cambios | -autores, - cambios

public class Camion

```

{
    public event InformacionAvance InformarAvance;
    public event InformacionEntregado InformarEntrega;
    public event PedidoEntregado CobraPedido;

    public delegate void PedidoEntregado(Factura factura);
    public delegate void InformacionAvance();
    public delegate void InformacionEntregado(string mensaje);

    private string patente;
    private int recorrido;
    private int kms;
    private int ubicacion;
    private static int velocidad;
    private List<Factura> facturas;

    0 referencias | - cambios | -autores, - cambios
    public string Patente { get => patente; set => patente = value; }
    7 referencias | - cambios | -autores, - cambios
    public int Ubicacion { get => ubicacion; set => ubicacion = value; }
    0 referencias | - cambios | -autores, - cambios
    public static int Velocidad { get => velocidad; set => velocidad = value; }
    6 referencias | - cambios | -autores, - cambios
    public int Recorrido { get => recorrido; set => recorrido = value; }
    1 referencia | - cambios | -autores, - cambios
    public int Kms { get => kms; set => kms = value; }

    0 referencias | - cambios | -autores, - cambios
    static Camion()
    {
        velocidad = 3;
    }
}

```

Bamio Ezequiel (C) Copyright (2021) All Rights Reserved

11

```

1 referencia | - cambios | -autores, - cambios
private void btnEstado_Click(object sender, EventArgs e)
{
    if (this.fabrica.Facturas.Count > 0)
    {
        this.cancellationToken = cancellationSource.Token;
        if (!this.ejecutando)
        {
            Task task = Task.Run(() =>
            {
                this.ejecutando = true;
                this.camion.InitializeDeliveries(this.cancellationToken);
            });
            btnEstado.Text = "Entregando pedidos...";
            btnEstado.Enabled = false;
        }
    }
    else
    {
        MessageBox.Show("No hay pedidos para entregar.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

1 referencia | - cambios | -autores, - cambios
private void btnReniciar_Click(object sender, EventArgs e)
{
    this.Reniciar();
}

public static class MetodosExtension
{
    public static void InsertarCliente(this DataGridView dataGridView, Fabrica fabrica)
    {
        int i = dataGridView.Rows.Count;

        foreach (Cliente cliente in fabrica.Clientes)
        {
            dataGridView.Rows.Insert(i, cliente.Id, cliente.Nombre, cliente.Apellido, cliente.Sexo, cliente.Direccion, cliente.Telefono, cliente.Estado);
        }
    }

    2 referencias | - cambios | -autores, - cambios
    public static void InsertarProducto(this DataGridView dataGridView, List<Producto> productos)
    {
        int i = dataGridView.Rows.Count;

        foreach (Producto p in productos)
        {
            dataGridView.Rows.Insert(i, p.Codigo, p.TipoProducto, p.Nombre, p.Precio, p.TamanoEnvase, p.TipoEnvase);
        }
    }
}

```

