



BAMIO
EZEQUIEL
2C

Trabajo practico – LabII Fabrica de Productos

Parte 1



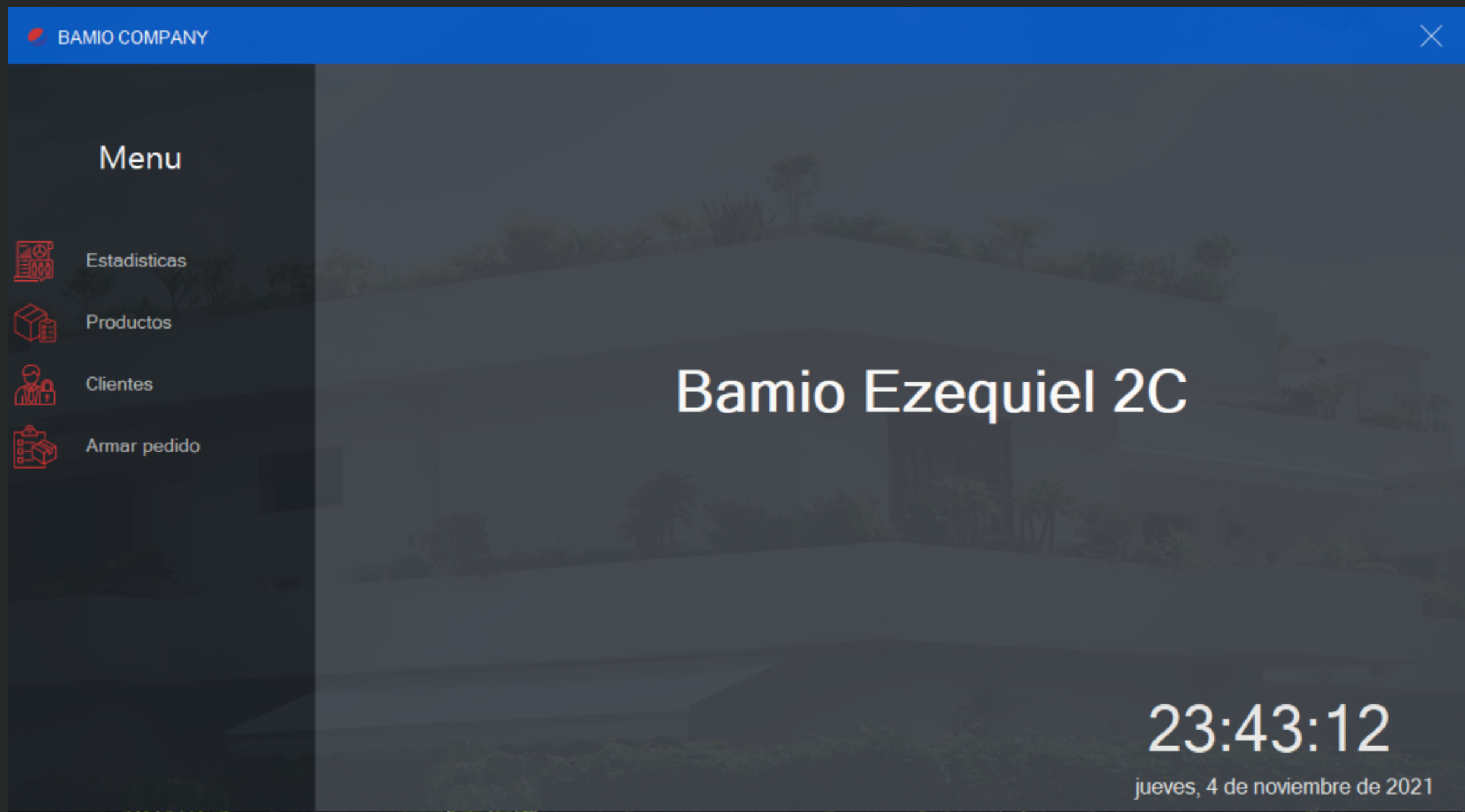
Explicacion

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"

<https://www.youtube.com/watch?v=Lx9DYChrFec>



MENU PRINCIPAL



Productos



EN ESTE APARTADO SE PODRA VER LA LISTA DE PRODUCTOS, CREAR NUEVOS , CARGAR Y GUARDAR UN ARCHIVO DE PRODUCTOS.

BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

Guarda

Cargar

Agregar producto

Nuevo

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



Cientes

EN ESTE APARTADO SE PODRA VER LA LISTA DE CLIENTES, CREAR NUEVOS , CARGAR Y GUARDAR UN ARCHIVO DE CLIENTES.

BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

Guarda

Cargar

Listado de Clientes

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

Nuevo cliente

ID

Nombre

Apellido

Direccion

Telefono

6

Crear cliente

ID del cliente

Eliminar

23:43:44

jueves, 4 de noviembre de 2021

Estadísticas



EN ESTE APARTADO SE PODRA VER LAS ESTADISTICAS 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

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

Armar pedido



EN ESTE APARTADO SE PODRA ARMAR LOS PEDIDOS PARA LOS CLIENTES.
PARA ARMAR UN PEDIDO ES NECESARIO SELECCIONAR LOS PRODUCTOS Y LA DIRECCION DEL CLIENTE.

BAMIO COMPANY

Menu

Estadísticas

Productos

Cientes

Armar pedido

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

Codigo	Producto	Nombre	Precio	Tamaño	Tipo
52003	Cerveza	Corona	3250	Grande	Botella
28831	Cerveza	Corona	1200	Chico	Lata

3833 Hara Islands

Confirmar

23:44:04

jueves, 4 de noviembre de 2021



TEMAS UTILIZADOS

- ARCHIVOS
- EXCEPCIONES
- TEST UNITARIOS
- TIPOS GENERICOS
- SERIALIZACIÓN
- INTERFACES



Archivos y serializacion

PROYECTO "IO".
USADO EN: FRMCLIENTE, FRMLISTAPRODUCTO,
FRMMENU

```
1 referencia | - cambios | - autores | - cambios  
private void btnAbrir_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);  
        }  
    }  
}
```

```
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);  
            }  
        }  
    }  
}
```



Excepciones

USADO EN: CLIENTES, FABRICA, PRODUCTOS, ARCHIVOS, FORMULARIOS.

```
#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.");
    }
}
```

Test Unitarios

USADO EN: TEST



```
/// 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>
```



Tipos genericos - Interfaces

USADO EN: IARCHIVOS, IIDENTIFICADOR

USADO EN: ARCHIVOS, FABRICA

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace IO
{
    public interface IArchivos<T> where T : class
    {
        void Escribir(string rutaCompleta, T dato, bool incluir);

        T Leer(string rutaCompleta);
    }
}
```

```
{
    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();
    }
}
```

```
namespace IO
{
    6 referencias | 0 cambios | 0 autores, 0 cambios
    public class Archivo : IArchivos<string>
    {
        3 referencias | 0 cambios | 0 autores, 0 cambios
        public void Escribir(string rutaCompleta, string texto, bool incluir)
        {
            try
            {
                using (StreamWriter streamWriter = new StreamWriter(rutaCompleta, incluir))
                {
                    streamWriter.WriteLine(texto);
                }
            }
            catch (Exception)
            {
                throw;
            }
        }

        2 referencias | 0 cambios | 0 autores, 0 cambios
        public string Leer(string rutaCompleta)
        {
            string rtn = string.Empty;

            try
            {
                using (StreamReader streamReader = new StreamReader(rutaCompleta))
                {

```