

## Programa del BancoSimple2T1 antes de las mejoras

### Base de Datos

SQLQuery1.sql - D:\U8CM\Usuario (60))\*

```
Create database BancoSimple2T1
GO
Use BancoSimple2T1
GO
Create table Cliente (
  ClienteId int primary key identity (1,1),
  Nombre Nvarchar (50) not null,
  Identificacion Nvarchar (14) not null
);
Create table Cuenta (
  CuentaId Int primary key identity (1,1),
  NumeroCuenta Nvarchar (20) not null,
  Saldo Decimal (18,2) not null default 0,
  Activa Bit not null default 1,
  ClienteId int not null,
  Foreign key (ClienteId) references Cliente (ClienteId)
);
create table Transacciones (
  TransaccionId Int primary key identity (1,1),
  Monto decimal (18,2) not null,
  Fecha datetime not null default getdate (),
  Descripcion Nvarchar (200),
  CuentaOrigenId int,
  CuentaDestinoId int,
  Foreign key (CuentaOrigenId) references Cuenta (CuentaId),
  Foreign key (CuentaDestinoId) references Cuenta (CuentaId)
);
```

## Programa

### ✓ *Carpeta Data*

### ➤ *Clase BancoSimpleContext*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore;
using BancoSimple2T1.Models;

namespace BancoSimple2T1.Data
{
    6 referencias
    public class BancoSimpleContext : DbContext
    {
        3 referencias
        public DbSet <Cliente> Cliente { get; set; }
        7 referencias
        public DbSet <Cuenta> Cuenta { get; set; }
        2 referencias
        public DbSet <Transaccion> Transacciones { get; set; }

        0 referencias
        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer(@"Server = DESKTOP-QHPU8CM\SQLEXPRESS; database =BancoSimple2T1; trusted_C
```

```
0 referencias
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseSqlServer(@"Server = DESKTOP-QHPU8CM\SQLEXPRESS; database =BancoSimple2T1; trusted_Connection = true; trustservercertificate = true;");
}

//Definicion de filtro global
0 referencias
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Cuenta>().HasQueryFilter ( c => c.Activa);
}
```

✓ **Carpeta Models**

➤ **Clase Cliente**

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

namespace BancoSimple2T1.Models
{
    4 referencias
    public class Cliente
    {
        0 referencias
        public int ClienteId { get; set; }
        5 referencias
        public string Nombre { get; set; }
        1 referencia
        public string Identificacion { get; set; }
        0 referencias
        public List<Cuenta> Cuentas { get; set; } = new List<Cuenta>();
    }
}
```

➤ **Clase Cuenta**

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

namespace BancoSimple2T1.Models
{
    6 referencias
    public class Cuenta
    {
        5 referencias
        public int CuentaId { get; set; }
        4 referencias
        public string NumeroCuenta { get; set; }
        6 referencias
        public decimal Saldo { get; set; }
        7 referencias
        public bool Activa { get; set; } = true;
        2 referencias
        public int ClienteId { get; set; }
        6 referencias
        public Cliente cliente { get; set; }
    }
}
```

➤ **Clase Transaccion**

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

namespace BancoSimple2T1.Models
{
    2 referencias
    public class Transaccion
    {
        0 referencias
        public int TransaccionId { get; set; }
        1 referencia
        public decimal Monto { get; set; }
        1 referencia
        public DateTime Fecha { get; set; } = DateTime.Now;
        1 referencia
        public string Descripcion { get; set; }
        1 referencia
        public int ? CuentaOrigenId { get; set; }
        1 referencia
        public int ? CuentaDestinoId { get; set; }
    }
}
```

✓ **Formulario Principal**

➤ **Diseño**

The image shows a Windows application window titled "Form1". The window contains a main form with a light gray background. At the top left, there is a button labeled "Buscar Cliente" next to a text input field. Below this, the form is divided into two columns. The left column is labeled "Cliente" and contains a large gray rectangular area. The right column is labeled "Cuenta" and also contains a large gray rectangular area. At the bottom of the form, there are five buttons: "AgregarCliente" on the left, and "AgregarCuenta", "Transferencia", "Ver Transferencia", and "Desactivar Cuenta" on the right. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

➤ **Código de Form1.cs**

```
using BancoSimple2T1.Data;
using BancoSimple2T1.Models;
using Microsoft.EntityFrameworkCore;
namespace BancoSimple2T1
{
    3 referencias
    public partial class Form1 : Form
    {
        private BancoSimpleContext _db = new BancoSimpleContext();
        1 referencia
        public Form1()
        {
            InitializeComponent();
            CargarInfo();
        }

        5 referencias
        private void CargarInfo()
        {
            dgvClientes.DataSource = _db.Cliente.ToList();
            var cuenta = _db.Cuenta.
                Include(c => c.cliente).Where(c => c.Activa).
                Select(c => new
                {
                    c.CuentaId,
                    c.NumeroCuenta,
                    c.Saldo,
                    NombreCliente = c.cliente.Nombre,
                });
        }
    }
}
```

```

        NombreCliente = c.cliente.Nombre,
        c.Activa,
        c.ClienteId
    })).ToList();
    dgvCuentas.DataSource = cuenta;
}

1 referencia
private void btnAgregarCliente_Click(object sender, EventArgs e)
{
    var form = new AgregarClienteForm();
    if (form.ShowDialog() == DialogResult.OK)
    {
        _db.Cliente.Add(form.NuevoCliente);
        _db.SaveChanges();
        CargarInfo();
    }
}

1 referencia
private void btnAgregarCuenta_Click(object sender, EventArgs e)
{
    if (dgvClientes.SelectedRows.Count == 0)
    {
        MessageBox.Show("Seleccione un cliente primero");
        return;
    }
    var clienteId = (int)dgvClientes.SelectedRows[0].Cells["ClienteId"].Value;

```

```

    }
    var clienteId = (int)dgvClientes.SelectedRows[0].Cells["ClienteId"].Value;
    var form = new AgregarCuentaForm(clienteId);
    if (form.ShowDialog() == DialogResult.OK)
    {
        _db.Cuenta.Add(form.NuevaCuenta);
        _db.SaveChanges();
        CargarInfo();
    }
}

//Transacciones
1 referencia
private void RealizarTransaccion(int cuentaOrigenId, int cuentaDestinoId, decimal monto)
{
    //implementacion
    //Nivel de aislamiento
    using var transferencia = _db.Database.BeginTransaction(System.Data.IsolationLevel.Serializable);
    try
    {
        //Filtro y ordenacion
        var cuentaOrigen = _db.Cuenta.FirstOrDefault(c => c.CuentaId == cuentaOrigenId);
        var cuentaDestino = _db.Cuenta.FirstOrDefault(c => c.CuentaId == cuentaDestinoId);

        if (cuentaOrigen == null || cuentaDestino == null)
            throw new Exception("Una o ambas cuentas no existen");

        if (!cuentaOrigen.Activa || !cuentaDestino.Activa)
            throw new Exception("Una o ambas cuentas están inactivas");
        if (cuentaOrigen.Saldo < monto)

```



```

if (cuentaOrigen.Saldo < monto)
    throw new Exception("Saldo Insuficiente en la cuenta Origen");

cuentaOrigen.Saldo -= monto;
cuentaDestino.Saldo += monto;

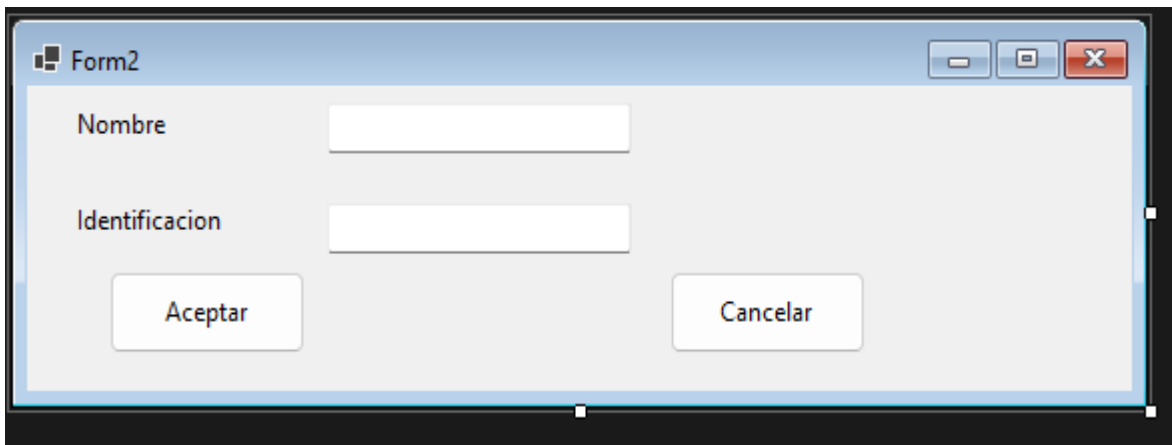
//Registrar la transaccion
_db.Transacciones.Add(new Transaccion
{
    Monto = monto,
    Fecha = DateTime.Now,
    Descripcion = "Transferencia entre cuentas",
    CuentaOrigenId = cuentaOrigenId,
    CuentaDestinoId = cuentaDestinoId
});
_db.SaveChanges();
//Transaccion completada
transferencia.Commit();
MessageBox.Show("Transferencia realizada");
CargarInfo();
}
catch (Exception ex)
{
    transferencia.Rollback();

    var inner = ex.InnerException?.Message ?? "No hay InnerException";
    MessageBox.Show($"Error al guardar:\n{ex.Message}\n\nDetalle:\n{inner}");
}

```

## ✓ **Formulario AgregarClienteForm**

### ➤ **Diseño**



The image shows a Windows Forms application window titled "Form2". Inside the window, there are two text input fields. The first is labeled "Nombre" and the second is labeled "Identificacion". Below these fields, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel). The window has a standard Windows title bar with minimize, maximize, and close buttons.

## ✓ **Código AgregarClienteForm.cs**

```

using BancoSimple2T1.Models;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace BancoSimple2T1
{
    3 referencias
    public partial class AgregarClienteForm : Form
    {
        2 referencias
        public Cliente NuevoCliente { get; private set; }
        1 referencia
        public AgregarClienteForm()
        {
            InitializeComponent();
        }

        1 referencia
        private void btnAceptar_Click(object sender, EventArgs e)
        {

```

```

1 referencia
private void btnAceptar_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtNombre.Text) || string.IsNullOrEmpty(txtIdentificacion.Text))
    {
        MessageBox.Show("Todos los campos son necesarios");
        return;
    }
    NuevoCliente = new Cliente
    {
        Nombre = txtNombre.Text,
        Identificacion = txtIdentificacion.Text
    };
    DialogResult = DialogResult.OK;
    Close();
}

1 referencia
private void btnCancelar_Click(object sender, EventArgs e)
{
    DialogResult = DialogResult.Cancel;
    Close();
}

```

## ✓ **Formulario AgregarCuentaForm**

### ➤ **Diseño**

The screenshot shows a Windows Form titled "Form2". The form has a light gray background and a blue title bar. It contains two labels: "Numero de cuenta" and "Saldo Inicial". The "Numero de cuenta" label is positioned to the left of a text box. The "Saldo Inicial" label is positioned to the left of a numeric spinner box showing the value "0". At the bottom center of the form is a button labeled "Aceptar".

✓ Código AgregarCuentaForm.cs

```
using BancoSimple2T1.Models;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace BancoSimple2T1
{
    3 referencias
    public partial class AgregarCuentaForm : Form
    {
        2 referencias
        public Cuenta NuevaCuenta { get; private set; }
        private int _clienteId;
        1 referencia
        public AgregarCuentaForm(int clienteId)
        {
            InitializeComponent();
            _clienteId = clienteId;
        }
    }
}
```

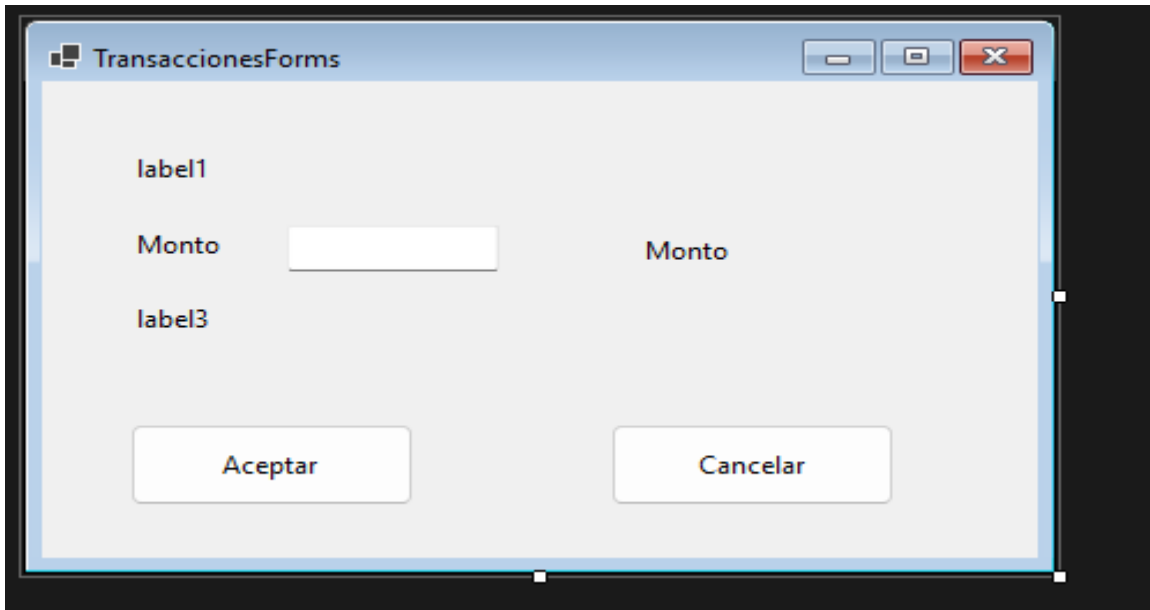
```
1 referencia
private void btnAceptar_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtNumeroCuenta.Text))
    {
        MessageBox.Show("El numero de cuenta es requerido");
        return;
    }

    NuevaCuenta = new Cuenta
    {
        NumeroCuenta = txtNumeroCuenta.Text,
        Saldo = numSaldoInicial.Value,
        ClienteId = _clienteId,
        Activa = true
    };

    DialogResult = DialogResult.OK;
    Close();
}
}
```

✓ **Formulario TransaccionesForms**

➤ **Diseño**



➤ **Código TransaccionesForms.cs**

```
using BancoSimple2T1.Data;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace BancoSimple2T1
{
    3 referencias
    public partial class TransaccionesForms : Form
    {
        2 referencias
        public decimal Monto { get; private set; }
        private int _cuentaOrigenId;
        private int _cuentaDestinoId;
        private BancoSimpleContext db;
        1 referencia
        public TransaccionesForms(int cuentaOrigenId, int cuentaDestinoId)
        {
            InitializeComponent();
        }
    }
}
```

```
InitializeComponent();
_cuentaOrigenId = cuentaOrigenId;
_cuentaDestinoId = cuentaDestinoId;
```

```
db = new BancoSimpleContext();
CargarInfoCuenta();
```

```
}
```

1 referencia

```
private void CargarInfoCuenta()
```

```
{
```

```
    var cuentaOrigen = db.Cuenta.
        Include(c => c.cliente).
        First(c => c.CuentaId == _cuentaOrigenId);
```

```
    var cuentaDestino = db.Cuenta.
        Include(c => c.cliente).
        First(c => c.CuentaId == _cuentaDestinoId);
```

```
    lblOrigen.Text = $"Nombre: {cuentaOrigen.cliente.Nombre} cuenta {cuentaOrigen.NumeroCuenta}";
    lblDestino.Text = $"Nombre: {cuentaDestino.cliente.Nombre} cuenta {cuentaDestino.NumeroCuenta}";
    lblDisponible.Text = $"Saldo Disponible : {cuentaOrigen.Saldo:c}";
```

```
}
```

1 referencia

```
private void lbl1_Click(object sender, EventArgs e)
```

```
{
```

```
}
```

1 referencia

```
private void btnAceptar_Click(object sender, EventArgs e)
```

```
{
```

```
    if (decimal.TryParse(txtSaldo.Text, out decimal monto) && monto > 0)
    {
        Monto = monto;
        DialogResult = DialogResult.OK;
        Close();
    }
```

```
    else
```

```
    {
```

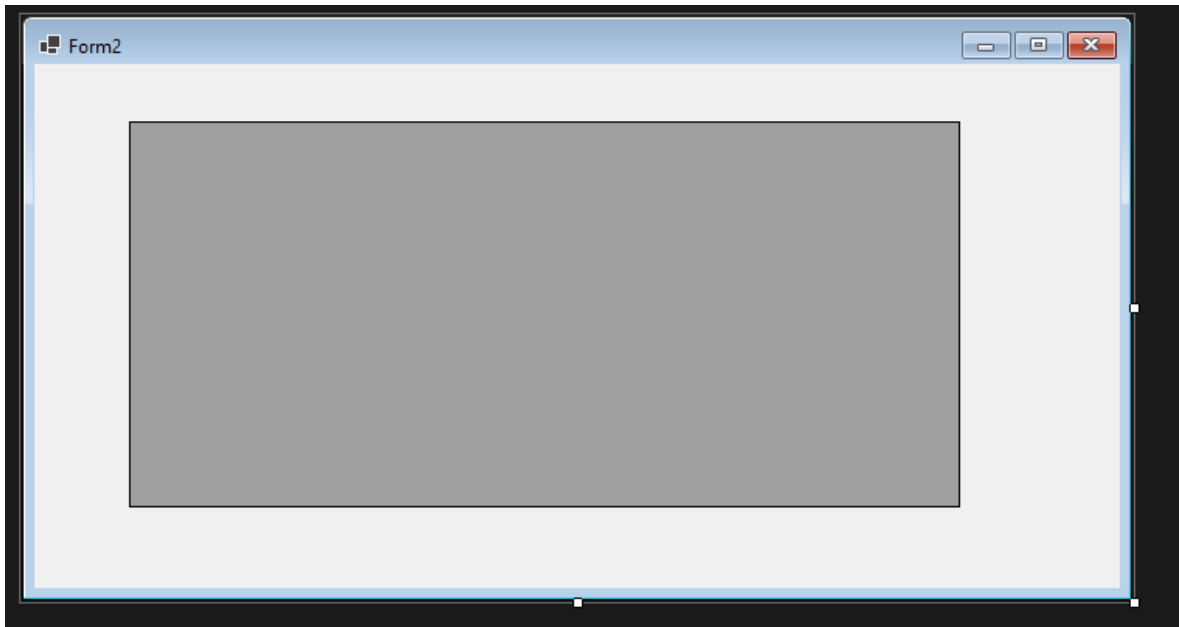
```
        MessageBox.Show("Ingrese un monto mayor a 0");
```

```
    }
```

```
}
```

✓ **Formulario VerTransferenciaForms**

➤ **Diseño**



✓ **Código VerTransferenciaForms.cs**

```
using BancoSimple2T1.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace BancoSimple2T1
{
    3 referencias
    public partial class VerTransferenciaForms : Form
    {
        private BancoSimpleContext con = new BancoSimpleContext();
        1 referencia
        public VerTransferenciaForms()
        {
            InitializeComponent();
            CargarTransferencias();
        }
    }
}
```

```

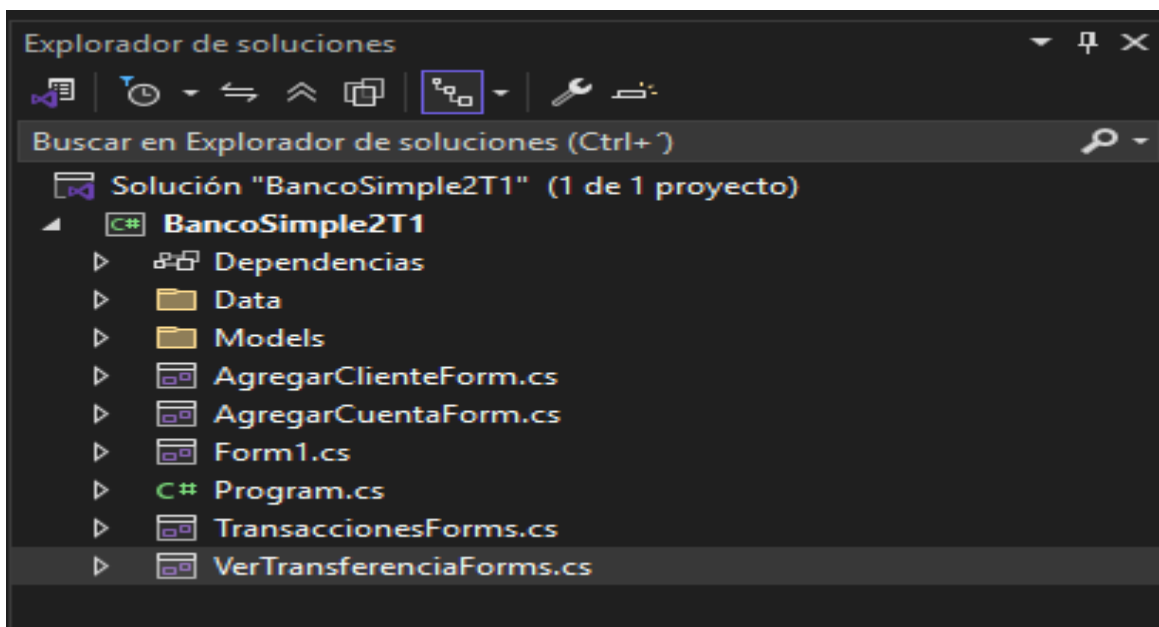
1 referencia
private void CargarTransferencias()
{
    dgvTransferencias.DataSource = con.Transacciones.ToList();
}

1 referencia
private void dgvTransferencias_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}

1 referencia
private void VerTransferenciaForms_Load(object sender, EventArgs e)
{
}

```

### ✓ *Explorador de soluciones*



Se puede observar el antes del programa tanto su base de datos como también las carpetas, los diseños y los códigos de cada uno de los formularios, además, esto nos ayudara a identificar las mejoras que se hicieron en el programa, las cuales se hicieron conforme a cada uno de los puntos a mejorar de los requisitos por parte de la asignación pedida. Por lo tanto, luego de este documento se subirán los cambios tanto del programa como también de los documentos para su mejor observación, por medio de GitBash y GitHub.



