

```
using BussinessLayer;
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;
using Entities;

namespace Gym
{
    public partial class Cajas : Form
    {
        #region Instancias
        //Clases internas
        private readonly MetodosGenerales _metodosGenerales;
        private readonly Restricciones _restricciones;

        //Capa negocio
        private readonly BussinessCaja _bussinessCaja;

        //Entidades
        private readonly Entities.Cajas _caja;
        private readonly Entities.Detalles_Cajas _detalles_Cajas;
        private readonly SaldosActualizados _saldos_Actualizados;

        #endregion

        #region Variables
        private int personaLogueada;
        private decimal ingreso;
        private decimal egreso;
        private decimal importeFinalCaja = -1;
        private DataSet DsCajas;
        private int idCajaAbierta;
        private string buscar;
        private bool cajaAbierta;
        private bool primeraCaja;

        #endregion

        public Cajas(int idPersonaLog)
        {
            InitializeComponent();
            _restricciones = new Restricciones();
            _metodosGenerales = new MetodosGenerales();
            _bussinessCaja = new BussinessCaja();
            _caja = new Entities.Cajas();
            _detalles_Cajas = new Entities.Detalles_Cajas();
            _saldos_Actualizados = new SaldosActualizados();
            personaLogueada = idPersonaLog;
            VerificarCajaAbierta();
            ActualizacionDeImportes();
            GetDetallesCajas();
        }
    }
}
```

```
#region Encapsulamientos
```

```
private void VerificarCajaAbierta()
```

```
{
    DateTime fecha = DateTime.Now;
    cajaAbierta = _bussinessCaja.VerificarCajaAbierta(fecha);
    if (cajaAbierta)
    {
        gbAperturaCaja.Enabled = false;
        gbCierreCaja.Enabled = true;
        GetLastCajaID();
    }
    else
    {
        gbAperturaCaja.Enabled = true;
        gbCierreCaja.Enabled = false;
    }
}
```

```
private void GetLastCajaID()
```

```
{
    _bussinessCaja.GetLastCajaID(_caja);
    idCajaAbierta = _caja.Caja_ID;
}
```

```
private void ActualizacionDeImportes()
```

```
{
    //Falta tener en cuenta el importe de ingreso que se registra en caja
    //cuando se abre la caja.
    _saldos_Actualizados.Caja_ID = idCajaAbierta;
    _bussinessCaja.ConsultarSalDOS(_saldos_Actualizados);

    lblImporteInicial.Text = _saldos_Actualizados.Importe_Inicial.ToString();
    lblIngresos.Text = _saldos_Actualizados.Importe_Ingreso.ToString();
    lblEgresos.Text = _saldos_Actualizados.Importe_Egreso.ToString();
    lblTotal.Text = _saldos_Actualizados.Total.ToString();
}
```

```
private void GetDetallesCajas()
```

```
{
    DsCajas = _bussinessCaja.GetDetallesCajas(buscar);

    dtgvCajas.Rows.Clear();
    //Vacío el textbox para resetear en caso de búsqueda específica
    buscar = string.Empty;

    //Y acá traigo los datos
    //Pregunto si el dataset tiene datos, preguntando si tiene filas
    if (DsCajas.Tables[0].Rows.Count > 0)
    {
        //Y por cada fila que haya en el dataset
        foreach (DataRow dr in DsCajas.Tables[0].Rows)
        {
            string fecha = dr[1].ToString();
            fecha = fecha.Substring(0, fecha.Length - 8);

            dtgvCajas.Rows.Add(dr[0].ToString(), fecha, dr[2], dr[3], dr[4], dr[5]);
        }
    }
}
```

```

private void VerificarImportes(KeyPressEventArgs e, string _importe)
{
    string importe = _importe.ToString();
    _restricciones.SoloNumeros(e, importe);
}
private bool VerificarBoxes()
{
    bool estaVacio;

    if (string.IsNullOrEmpty(txtImporteFinal.Text) ||
Convert.ToInt32(txtImporteFinal.Text) < 0)
    {
        estaVacio = true;
    }
    else
    {
        estaVacio = false;
    }

    return estaVacio;
}
private void VerificarimportesFinales()
{
    DialogResult result = MessageBox.Show($"El importe final es de {importeFinalCaja}.
¿Es correcto? De ser correcto, presione \"Aceptar\", " +
        $"de lo contrario presione \"Cancelar\" y reingrese el importe final",
"Verificar Importe de Cierre de Caja");

    if (result == DialogResult.OK)
    {
        CerrarCaja();
    }
}

private void CerrarCaja()
{
    _caja.Caja_ID = idCajaAbierta;
    _caja.Empleado_ID_Cierre = personaLogueada;
    _caja.Fecha_Cierre = DateTime.Now;
    _caja.Importe_Cierre = Convert.ToDecimal(lblTotal.Text);
    _caja.Importe_Cierre_Caja = importeFinalCaja;
    _caja.Caja_Abierta = false;

    _bussinessCaja.CerrarCaja(_caja);
}

#endregion

private void btnAbrirCaja_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtImporteEfectivo.Text))
    {
        MessageBox.Show("Tiene que ingresar el importe de inicio. Si la caja está
vacía, " +
            "ingrese un 0.", "Campos vacíos!");
    }
    else
    {
        _caja.Importe_Apertura = Convert.ToDecimal(txtImporteEfectivo.Text);
        _caja.Empleado_ID_Apertura = personaLogueada;
    }
}

```

```
_caja.Fecha_Apertura = DateTime.Now;
_caja.Caja_Abierta = true;

//Abro la caja del día
_bussinessCaja.AbrirCaja(_caja);
cajaAbierta = true;

//consultamos el último ID
GetLastCajaID();

//Actualizamos los datos para que se muestren los detalles de cada día.
GetDetallesCajas();

gbAperturaCaja.Enabled = false;
gbCierreCaja.Enabled = true;
}
}

private void txtImporteEfectivo_KeyPress(object sender, KeyPressEventArgs e)
{
    string importe = txtImporteEfectivo.Text;
    VerificarImportes(e, importe);
}

private void txtBuscarCajas_KeyPress(object sender, KeyPressEventArgs e)
{
    buscar = txtBuscarCajas.Text;
    if (e.KeyChar == (char)Keys.Enter)
    {
        GetDetallesCajas();
    }
}

private void txtImporteFinal_KeyPress(object sender, KeyPressEventArgs e)
{
    string importe = txtImporteFinal.Text;
    ActualizacionDeImportes();
    VerificarImportes(e, importe);

    if (e.KeyChar == (char)Keys.Enter)
    {
        importeFinalCaja = Convert.ToDecimal(importe);
        lblImporteCajaFinal.Text = txtImporteFinal.Text;
        lblDiferencia.Text =
Convert.ToString(Convert.ToDecimal(lblImporteCajaFinal.Text) -
Convert.ToDecimal(lblTotal.Text));
    }
}

private void btnCerrar_Click(object sender, EventArgs e)
{
    bool estaVacio = VerificarBoxes();
    if (estaVacio)
    {
        MessageBox.Show("Primero debe ingresar el importe de cierre en el cuadro de
texto correspondiente.", "Importe final faltante");
    }
    else
    {
        importeFinalCaja = Convert.ToDecimal(txtImporteFinal.Text);
        decimal total = Convert.ToDecimal(lblTotal.Text);
    }
}
```

```
decimal suma = importeFinalCaja - total;

DialogResult result = MessageBox.Show(
    $"Verifique si los montos son correctos:\n" +
    $"Importe en Caja: ${importeFinalCaja}.\n" +
    $"Importe registrado en el sistema: ${total}.\n" +
    $"Diferencia: {suma}.\n" +
    "Si estos importes son correctos, seleccione 'Aceptar', de lo contrario  

    elija 'Cancelar'." +
    "Recuerde que estos importes no pueden modificarse una vez que elija  

    'Aceptar'.",
    "Leer bien - Cierre de Caja",
    MessageBoxButtons.OKCancel,
    MessageBoxIcon.Exclamation);
if (result == DialogResult.OK)
{
    //Cerramos la caja.
    CerrarCaja();
    VerificarCajaAbierta();
    ActualizacionDeImportes();
    GetDetallesCajas();
    MessageBox.Show("Ha cerrado la caja con éxito!", "Cierre de Caja",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
}
}
```