

Indice

ANEXO B. Código fuente

B.1. CÓDIGO ITERACION.CS	3
B.2. CÓDIGO LISTAMAQUINAS.CS	4
B.3. CÓDIGO LISTAOPERACION.CS	6
B.4. CÓDIGO LISTAPIEZAS.CS	8
B.5. CÓDIGO MAQUINARIA.CS	11
B.6. CÓDIGO MAQUINARIAFORM.CS	13
B.7. CÓDIGO MAQUINARIAFORM.DESIGNER.CS	15
B.8. CÓDIGO OPERACION.CS	18
B.9. CÓDIGO PIEZA.CS	21
B.10. CÓDIGO PIEZAFORM.CS	26
B.11. CÓDIGO PIEZAFORMDESIGNER.CS	29
B.12. CÓDIGO PRINCIPAL.CS	35
B.13. CÓDIGO PRINCIPAL.DESIGNER.CS	48
B.14. CÓDIGO PROGRAM.CS	53
B.15. CÓDIGO SOLUCION.CS	54



B.1. Código Iteracion.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class Proceso
    {
        Operacion _Op;
        int _Iteracion;

        public Operacion Op
        {
            get
            {
                return _Op;
            }
            set
            {
                _Op = value;
            }
        }

        public int Iteracion
        {
            get
            {
                return _Iteracion;
            }
            set
            {
                _Iteracion = value;
            }
        }

        public Proceso(Operacion o, int it)
        {
            _Op = new Operacion(o);
            _Iteracion = it;
        }
    }
}
```



B.2. Código ListaMaquinas.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class ListaMaquinas
    {
        List<Maquinaria> _LMaquinaria;

        public List<Maquinaria> LMaquinaria {
            get
            {
                return _LMaquinaria;
            }
            set
            {
                _LMaquinaria=value;
            }
        }

        public ListaMaquinas()
        {
            _LMaquinaria = new List<Maquinaria>();
        }

        public ListaMaquinas(List<Maquinaria> Lm)
        {
            _LMaquinaria = new List<Maquinaria>(Lm);
        }

        public Maquinaria Find(Maquinaria m)
        {
            return LMaquinaria.Find(delegate(Maquinaria Maq){ return m.Nombre==Maq.Nombre;});
        }

        public Maquinaria Find(string m)
        {
            Maquinaria resultado=null;
            int indice = 0;
            while (indice < LMaquinaria.Count)
            {
                if (LMaquinaria[indice].Nombre == m)
                {
                    resultado = LMaquinaria[indice];
                }
                indice++;
            }
            return resultado;
        }

        public void Inicializar()
        {
            int indice = 0;
            while (indice < _LMaquinaria.Count)
            {
                _LMaquinaria[indice].Tiempo = 0;
                indice++;
            }
        }
    }
}
```



```
}  
}  
}
```



B.3. Código ListaOperacion.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class ListaOperacion
    {
        List<Operacion> _LOperacion;

        public List<Operacion> LOperacion
        {
            get
            {
                return _LOperacion;
            }
            set
            {
                _LOperacion = value;
            }
        }

        public ListaOperacion()
        {
            _LOperacion = new List<Operacion>();
        }

        public ListaOperacion(List<Operacion> Lo)
        {
            _LOperacion = new List<Operacion>(Lo);
        }

        public ListaOperacion(ListaOperacion LOp)
        {
            _LOperacion = LOp._LOperacion;
        }

        public ListaOperacion OperacionesActivas()
        {
            int indice = 0, operacion = -1;
            ListaOperacion LResultado=new ListaOperacion();
            while (indice < _LOperacion.Count)
            {
                if (!_LOperacion[indice].Finalizado)
                {
                    if (operacion == _LOperacion[indice].Orden || operacion == -1)
                    {
                        operacion = _LOperacion[indice].Orden;
                        LResultado._LOperacion.Add(LOperacion[indice]);
                    }
                }
                indice++;
            }
            return LResultado;
        }

        public void Concatenar(ListaOperacion LOp)
        {
            int indice = 0;
```



```
while (indice < LOp._LOperacion.Count)
{
    _LOperacion.Add(LOp.LOperacion[indice]);
    indice++;
}

public Operacion Find(Operacion op)
{
    return _LOperacion.Find(delegate(Operacion Oper)
    {
        return (op.Orden == Oper.Orden && op.Maquina == Oper.Maquina);
    });
}
}
```



B.4. Código ListaPiezas.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class ListaPiezas
    {
        List<Pieza> _LPiezas;

        public List<Pieza> LPiezas
        {
            get
            {
                return _LPiezas;
            }
            set
            {
                _LPiezas = value;
            }
        }

        public ListaPiezas()
        {
            _LPiezas = new List<Pieza>();
        }

        public ListaPiezas(ListaPiezas Lp)
        {
            _LPiezas = new List<Pieza>(Lp._LPiezas);
        }

        public ListaPiezas(List<Pieza> Lp)
        {
            _LPiezas = new List<Pieza>(Lp);
        }

        public ListaOperacion OperacionesCandidatas()
        {
            int indice = 0;
            ListaOperacion LResultado=new ListaOperacion(), LOpAux;
            while (indice < _LPiezas.Count)
            {
                LOpAux = _LPiezas[indice].LOperacion.OperacionesActivas();
                if (LOpAux != null)
                {
                    LResultado.Concatenar(LOpAux);
                }
                LOpAux = null;
                indice++;
            }
            return LResultado;
        }

        public Pieza Find(Pieza p)
        {
            return _LPiezas.Find(delegate(Pieza pieza){ return pieza.Nombre==p.Nombre;});
        }
    }
}
```




```
public Pieza Find(string p)
{
    return _LPiezas.Find(delegate(Pieza pieza) { return pieza.Nombre == p; });
}

public bool Finalizado()
{
    bool fin=true;
    int indice = 0;
    while (indice < _LPiezas.Count)
    {
        if (!_LPiezas[indice].Finalizado())
        {
            fin = false;
        }
        indice++;
    }
    return fin;
}

public Pieza SeleccionarPieza()
{
    Pieza p_aux;
    Pieza p=null;
    int indice = 0;
    while (indice < LPiezas.Count)
    {
        p_aux = LPiezas[indice];
        if (!p_aux.Finalizado())
        {
            if (p == null)
            {
                p = p_aux;
            }
            else
            {
                p = Criterios(p, p_aux);
            }
        }
        indice = indice + 1;
    }
    return p;
}

private Pieza Criterios(Pieza p, Pieza p_aux)
{
    double a = p.Restraso / p.Cantidad * 100;
    double b = p_aux.Restraso / p_aux.Cantidad * 100;
    if (a < b)
    {
        return p;
    }
    else
    {
        if (a > b)
        {
            return p_aux;
        }
        else
        {
            return p;
        }
    }
}
}
```



}
}



B.5. Código Maquinaria.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class Maquinaria
    {
        string _Nombre;
        double _Tiempo;
        double _Preparacion;

        #region Propiedades

        public string Nombre
        {
            get
            {
                return _Nombre;
            }
            set
            {
                _Nombre = value;
            }
        }

        public double Tiempo
        {
            get
            {
                return _Tiempo;
            }
            set
            {
                _Tiempo = value;
            }
        }

        public double Preparacion
        {
            get
            {
                return _Preparacion;
            }
            set
            {
                _Preparacion = value;
            }
        }
    }

    #endregion

    public Maquinaria(string nombre, double tiempo, double preparacion)
    {
        _Nombre = nombre;
        _Tiempo = tiempo;
        _Preparacion = preparacion;
    }
}
```



}
}



B.6. Código MaquinariaForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Planificador
{
    public partial class MaquinariaForm : Form
    {
        public MaquinariaForm()
        {
            InitializeComponent();
        }

        private void MaquinariaForm_Load(object sender, EventArgs e)
        {
            if (Principal.LMaquinas != null)
            {
                TablaMaquinaria.DataSource = Principal.LMaquinas.LMaquinaria;
            }
            TiempoTxt.Text = "0";
        }

        private void InsertarBtn_Click(object sender, EventArgs e)
        {
            Maquinaria m;
            if (MaquinaTxt.Text != "" && TiempoTxt.Text != "" && PreparacionTxt.Text != "")
            {
                m = new Maquinaria((string)MaquinaTxt.Text, double.Parse(TiempoTxt.Text),
                double.Parse(PreparacionTxt.Text));
                if (Principal.LMaquinas == null)
                {
                    Principal.LMaquinas = new ListaMaquinas();
                }
                if (Principal.LMaquinas.Find(m) == null)
                {
                    Principal.LMaquinas.LMaquinaria.Add(m);
                    TablaMaquinaria.DataSource = null;
                    TablaMaquinaria.DataSource = Principal.LMaquinas.LMaquinaria;
                    MaquinaTxt.Text = "";
                    TiempoTxt.Text = "0";
                }
            }
            else
            {
                MessageBox.Show("Rellene los campos por favor", "ERROR!");
            }
        }

        private void TablaMaquinaria_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
        }

        private void BorrarBtn_Click(object sender, EventArgs e)
        {
            Principal.LMaquinas.LMaquinaria.Clear();
        }
    }
}

```



```
    TablaMaquinaria.DataSource = null;  
    TablaMaquinaria.DataSource=Principal.LMaquinas.LMaquinaria;  
  }  
}  
}
```



B.7. Código MaquinariaForm.Designer.cs

```

namespace Planificador
{
    partial class MaquinariaForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.MaquinaTxt = new System.Windows.Forms.TextBox();
            this.TiempoTxt = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.InsertarBtn = new System.Windows.Forms.Button();
            this.BorrarBtn = new System.Windows.Forms.Button();
            this.TablaMaquinaria = new System.Windows.Forms.DataGridView();
            this.label3 = new System.Windows.Forms.Label();
            this.PreparacionTxt = new System.Windows.Forms.TextBox();
            ((System.ComponentModel.ISupportInitialize)(this.TablaMaquinaria)).BeginInit();
            this.SuspendLayout();
            //
            // MaquinaTxt
            //
            this.MaquinaTxt.Location = new System.Drawing.Point(181, 196);
            this.MaquinaTxt.Name = "MaquinaTxt";
            this.MaquinaTxt.Size = new System.Drawing.Size(139, 20);
            this.MaquinaTxt.TabIndex = 2;
            //
            // TiempoTxt
            //
            this.TiempoTxt.Location = new System.Drawing.Point(181, 231);
            this.TiempoTxt.Name = "TiempoTxt";
            this.TiempoTxt.Size = new System.Drawing.Size(139, 20);
            this.TiempoTxt.TabIndex = 3;
            //
            // label1
            //
            this.label1.AutoSize = true;

```



```
this.label1.Location = new System.Drawing.Point(59, 199);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(51, 13);
this.label1.TabIndex = 4;
this.label1.Text = "Máquina:";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(59, 238);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(73, 13);
this.label2.TabIndex = 5;
this.label2.Text = "Tiempo Inicio:";
//
// InsertarBtn
//
this.InsertarBtn.Location = new System.Drawing.Point(256, 314);
this.InsertarBtn.Name = "InsertarBtn";
this.InsertarBtn.Size = new System.Drawing.Size(109, 34);
this.InsertarBtn.TabIndex = 6;
this.InsertarBtn.Text = "Añadir";
this.InsertarBtn.UseVisualStyleBackColor = true;
this.InsertarBtn.Click += new System.EventHandler(this.InsertarBtn_Click);
//
// BorrarBtn
//
this.BorrarBtn.Location = new System.Drawing.Point(12, 314);
this.BorrarBtn.Name = "BorrarBtn";
this.BorrarBtn.Size = new System.Drawing.Size(109, 34);
this.BorrarBtn.TabIndex = 7;
this.BorrarBtn.Text = "Borrar Lista";
this.BorrarBtn.UseVisualStyleBackColor = true;
this.BorrarBtn.Click += new System.EventHandler(this.BorrarBtn_Click);
//
// TablaMaquinaria
//
this.TablaMaquinaria.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.TablaMaquinaria.Location = new System.Drawing.Point(12, 12);
this.TablaMaquinaria.Name = "TablaMaquinaria";
this.TablaMaquinaria.Size = new System.Drawing.Size(353, 163);
this.TablaMaquinaria.TabIndex = 8;
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(59, 271);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(105, 13);
this.label3.TabIndex = 10;
this.label3.Text = "Tiempo Preparación:";
//
// PreparacionTxt
//
this.PreparacionTxt.Location = new System.Drawing.Point(181, 268);
this.PreparacionTxt.Name = "PreparacionTxt";
this.PreparacionTxt.Size = new System.Drawing.Size(139, 20);
this.PreparacionTxt.TabIndex = 9;
//
// MaquinariaForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
```




```
this.ClientSize = new System.Drawing.Size(377, 360);
this.Controls.Add(this.label3);
this.Controls.Add(this.PreparacionTxt);
this.Controls.Add(this.TablaMaquinaria);
this.Controls.Add(this.BorrarBtn);
this.Controls.Add(this.InsertarBtn);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Controls.Add(this.TiempoTxt);
this.Controls.Add(this.MaquinaTxt);
this.Name = "MaquinariaForm";
this.Text = "Maquinaria";
this.Load += new System.EventHandler(this.MaquinariaForm_Load);
((System.ComponentModel.ISupportInitialize)(this.TablaMaquinaria)).EndInit();
this.ResumeLayout(false);
this.PerformLayout();

}

#endregion

private System.Windows.Forms.TextBox MaquinaTxt;
private System.Windows.Forms.TextBox TiempoTxt;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Button InsertarBtn;
private System.Windows.Forms.Button BorrarBtn;
private System.Windows.Forms.DataGridView TablaMaquinaria;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.TextBox PreparacionTxt;
}
```



B.8. Código Operacion.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class Operacion
    {
        int _Orden;
        string _Maquina;
        double _Tiempo;
        bool _Finalizado;
        double _Tiempo_Inicio;
        double _Tiempo_Fin;
        string _Pieza;

        #region Propiedades

        public int Orden
        {
            get
            {
                return _Orden;
            }
            set
            {
                _Orden = value;
            }
        }

        public string Maquina
        {
            get
            {
                return _Maquina;
            }
            set
            {
                _Maquina = value;
            }
        }

        public double Tiempo
        {
            get
            {
                return _Tiempo;
            }
            set
            {
                _Tiempo = value;
            }
        }

        public bool Finalizado
        {
            get
            {
                return _Finalizado;
            }
        }
    }
}
```



```
    }
    set
    {
        _Finalizado = value;
    }
}

public double Tiempo_Inicio
{
    get
    {
        return _Tiempo_Inicio;
    }
    set
    {
        _Tiempo_Inicio = value;
    }
}

public double Tiempo_Fin
{
    get
    {
        return _Tiempo_Fin;
    }
    set
    {
        _Tiempo_Fin = value;
    }
}

public string Pieza
{
    get
    {
        return _Pieza;
    }
    set
    {
        _Pieza = value;
    }
}

#endregion

public Operacion(int orden, string maquina, double tiempo, string pieza)
{
    _Orden = orden;
    _Maquina = maquina;
    _Tiempo = tiempo;
    _Pieza = pieza;
    _Finalizado = false;
    _Tiempo_Inicio = 0;
    _Tiempo_Fin = 0;
}

public Operacion(Operacion o)
{
    _Orden = o._Orden;
    _Maquina = o._Maquina;
    _Tiempo = o._Tiempo;
    _Pieza = o._Pieza;
    _Finalizado = o._Finalizado;
    _Tiempo_Inicio = o._Tiempo_Inicio;
}
```



```
    } _Tiempo_Fin = o._Tiempo_Fin;  
  }  
}
```



B.9. Código Pieza.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class Pieza
    {
        string _Nombre;
        int _Cantidad;
        int _DRetraso;
        double _Demanda_Mensual;
        double _Duracion_Total;
        DateTime _Fecha;
        ListaOperacion _LOperacion;
        double _Retraso;
        Operacion _Operacion_Anterior;

        #region Propiedades

        public string Nombre
        {
            get
            {
                return _Nombre;
            }
            set
            {
                _Nombre=value;
            }
        }

        public int Cantidad
        {
            get
            {
                return _Cantidad;
            }
            set
            {
                _Cantidad = value;
            }
        }

        public int DRetraso
        {
            get
            {
                return _DRetraso;
            }
            set
            {
                _DRetraso = value;
            }
        }

        public double Demanda_Mensual
        {
            get
```



```
{
    return _Demanda_Mensual;
}
set
{
    _Demanda_Mensual = value;
}
}

public DateTime Fecha
{
    get
    {
        return _Fecha;
    }
    set
    {
        _Fecha = value;
    }
}

public double Duracion_Total
{
    get
    {
        return _Duracion_Total;
    }
    set
    {
        _Duracion_Total = value;
    }
}

}

public ListaOperacion LOperacion
{
    get
    {
        return _LOperacion;
    }
    set
    {
        _LOperacion = value;
    }
}

}

public double Retraso
{
    get
    {
        return _Retraso;
    }
    set
    {
        _Retraso = value;
    }
}

}

public Operacion Operacion_anterior
{
    get
    {
        return _Operacion_Anterior;
    }
}
```



```

    }
    set
    {
        _Operacion_Anterior = value;
    }
}

#endregion

public Pieza(string nombre, int cantidad, int d_retrazo, double demanda, DateTime fecha, double
retrazo, ListaOperacion Loperacion)
{
    _Nombre = nombre;
    _Cantidad = cantidad;
    _DRetraso = d_retrazo;
    _Demanda_Mensual = demanda;
    _Fecha = fecha;
    _Retraso = retraso;
    _LOperacion = Loperacion;
    _Operacion_Anterior = null;
    _Duracion_Total = DuracionPendiente();
}

public void Inicializar()
{
    int indice = 0;
    _Operacion_Anterior = null;
    while (indice < LOperacion.LOperacion.Count)
    {
        LOperacion.LOperacion[indice].Finalizado = false;
        LOperacion.LOperacion[indice].Tiempo_Fin = 0;
        LOperacion.LOperacion[indice].Tiempo_Inicio = 0;
        indice++;
    }
    _Duracion_Total = DuracionPendiente();
}

private double DuracionPendiente()
{
    int indice = 0, orden = 1;
    double duracion = 0, duracion_pendiente = 0;
    while (indice < _LOperacion.LOperacion.Count)
    {
        if (orden == _LOperacion.LOperacion[indice].Orden)
        {
            if (_LOperacion.LOperacion[indice].Tiempo > duracion)
            {
                duracion = _LOperacion.LOperacion[indice].Tiempo;
            }
        }
        else
        {
            duracion_pendiente = duracion_pendiente + duracion * _Cantidad;
            orden = _LOperacion.LOperacion[indice].Orden;
            duracion = _LOperacion.LOperacion[indice].Tiempo;
        }
        indice++;
    }
    duracion_pendiente = duracion_pendiente + duracion * _Cantidad;
    return duracion_pendiente;
}

public bool Finalizado()
{

```



```

int indice = 0;
Operacion op_aux;
bool fin = false;
bool parar = false;
if (_Operacion_Anterior != null)
{
    op_aux = _Operacion_Anterior;
    while (indice < LOperacion.LOperacion.Count && !parar)
    {
        if (LOperacion.LOperacion[indice].Orden <= op_aux.Orden)
        {
            LOperacion.LOperacion[indice].Finalizado = true;
        }
        else
        {
            parar = true;
        }
        indice++;
    }
    if (indice == LOperacion.LOperacion.Count && !parar)
    {
        fin = true;
    }
}
return fin;
}

public Operacion EjecutarOperacion()
{
    int indice = 0;
    Operacion op=null;
    Operacion op_aux;
    Maquinaria m, m1;

    if (_Operacion_Anterior == null)
    {
        op = LOperacion.LOperacion[indice];
        while (indice < LOperacion.LOperacion.Count)
        {
            op_aux = LOperacion.LOperacion[indice];
            if (op.Orden == LOperacion.LOperacion[indice].Orden)
            {
                m = Principal.LMaquinas.Find(op_aux.Maquina);
                m1 = Principal.LMaquinas.Find(op.Maquina);
                if ((m.Tiempo + op_aux.Tiempo) < (m1.Tiempo + op.Tiempo))
                {
                    op = op_aux;
                }
            }
            indice++;
        }
    }
    else
    {
        op= _Operacion_Anterior;
        while (LOperacion.LOperacion[indice].Orden <= op.Orden && indice <
LOperacion.LOperacion.Count)
        {
            indice++;
        }
        op=LOperacion.LOperacion[indice];
        while (indice < LOperacion.LOperacion.Count)
        {
            op_aux = LOperacion.LOperacion[indice];

```



B.10. Código PiezaForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Planificador
{
    public partial class PiezaForm : Form
    {
        private static ListaOperacion LOperaciones;

        public PiezaForm()
        {
            InitializeComponent();
        }

        private void PiezaForm_Load(object sender, EventArgs e)
        {
            if (Principal.LPiezas.LPiezas.Count > 0)
            {
                NombreCmb.DataSource = Principal.LPiezas.LPiezas;
                NombreCmb.DisplayMember = "Nombre";
                NombreCmb.ValueMember = "Nombre";
            }
            MaquinaCmb.DataSource = Principal.LMaquinas.LMaquinaria;
            MaquinaCmb.DisplayMember = "Nombre";
            MaquinaCmb.ValueMember = "Nombre";
        }

        private void NombreCmb_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (Principal.LPiezas != null)
            {
                Pieza p = (Pieza)NombreCmb.SelectedItem;
                if (Principal.LPiezas.Find(p) != null)
                {
                    CantidadTxt.Text = p.Cantidad.ToString();
                    DRetrasoTxt.Text = p.DRetraso.ToString();
                    DemandaTxt.Text = p.Demanda_Mensual.ToString();
                    RetrasoTxt.Text = p.Re retraso.ToString();
                    Fecha.Text = p.Fecha.ToString();
                    LOperaciones = new ListaOperacion(p.LOperacion.LOperacion);
                    TablaOperacion.DataSource = null;
                    TablaOperacion.DataSource = LOperaciones.LOperacion;
                }
            }
        }

        private void InsertarOrdenBtn_Click(object sender, EventArgs e)
        {
            if (OrdenTxt.Text != "" && MaquinaCmb.Text != "" && TiempoTxt.Text != "" && NombreCmb.Text != "")
            {
                Operacion op;
                op = new Operacion(int.Parse(OrdenTxt.Text), MaquinaCmb.Text, double.Parse(TiempoTxt.Text),
                NombreCmb.Text);
                if (LOperaciones != null)

```



```

    {
        if (LOperaciones.Find(op) == null)
        {
            LOperaciones.LOperacion.Add(op);
            TablaOperacion.DataSource = null;
            TablaOperacion.DataSource = LOperaciones.LOperacion;
            TablaOperacion.Columns["Maquina"].ToString();
        }
    }
    else
    {
        LOperaciones = new ListaOperacion();
        LOperaciones.LOperacion.Add(op);
        TablaOperacion.DataSource = null;
        TablaOperacion.DataSource = LOperaciones.LOperacion;
        TablaOperacion.Columns["Maquina"].ToString();
    }
}
else
{
    MessageBox.Show("Rellene los campos orden máquina y tiempo", "ERROR!");
}
}

private void InsertarBtn_Click(object sender, EventArgs e)
{
    Pieza p;
    Pieza b;
    if (LOperaciones.LOperacion != null && NombreCmb.Text != "" && CantidadTxt.Text != "" &&
    DRetrasoTxt.Text != "")
    {
        p = new Pieza(NombreCmb.Text, int.Parse(CantidadTxt.Text), int.Parse(DRetrasoTxt.Text),
        double.Parse(DemandaTxt.Text), DateTime.Parse(Fecha.Text), double.Parse(RetrasoTxt.Text),
        LOperaciones);
        if (Principal.LPiezas.LPiezas != null)
        {
            b = Principal.LPiezas.Find(p);
            if (b != null)
            {
                b.DRetraso = int.Parse(DRetrasoTxt.Text);
                b.Cantidad = int.Parse(CantidadTxt.Text);
                b.Demanda_Mensual = double.Parse(DemandaTxt.Text);
                b.Fecha = DateTime.Parse(Fecha.Text);
                b.Restraso = double.Parse(RetrasoTxt.Text);
                b.Nombre = NombreCmb.Text;
                b.LOperacion = LOperaciones;
            }
        }
        else
        {
            Principal.LPiezas.LPiezas.Add(p);
            NombreCmb.DataSource = null;
            NombreCmb.DataSource = Principal.LPiezas.LPiezas;
            NombreCmb.ValueMember = "Nombre";
            NombreCmb.DisplayMember = "Nombre";
            TablaOperacion.DataSource = null;
            LOperaciones.LOperacion.Clear();
        }
    }
}
else
{
    Principal.LPiezas = new ListaPiezas();
    Principal.LPiezas.LPiezas.Add(p);
    NombreCmb.DataSource = null;
    NombreCmb.DataSource = Principal.LPiezas.LPiezas;
}
}

```



```
        NombreCmb.ValueMember = "Nombre";
        NombreCmb.DisplayMember = "Nombre";
        TablaOperacion.DataSource = null;
        LOperaciones.LOperacion.Clear();
    }
}
else
{
    MessageBox.Show("Debe rellenar todos los campos", "ERROR!");
}
}

private void BorrarLstBtn_Click(object sender, EventArgs e)
{
    TablaOperacion.DataSource = null;
    LOperaciones.LOperacion.Clear();
}
}
```



B.11. Código PiezaFormDesigner.cs

```

namespace Planificador
{
    partial class PiezaForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.TablaOperacion = new System.Windows.Forms.DataGridView();
            this.NombreCmb = new System.Windows.Forms.ComboBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.CantidadTxt = new System.Windows.Forms.TextBox();
            this.Fecha = new System.Windows.Forms.DateTimePicker();
            this.label3 = new System.Windows.Forms.Label();
            this.DRetrasoTxt = new System.Windows.Forms.TextBox();
            this.label4 = new System.Windows.Forms.Label();
            this.OperacionesGrp = new System.Windows.Forms.GroupBox();
            this.BorrarLstBtn = new System.Windows.Forms.Button();
            this.InsertarOrdenBtn = new System.Windows.Forms.Button();
            this.label7 = new System.Windows.Forms.Label();
            this.TiempoTxt = new System.Windows.Forms.TextBox();
            this.label6 = new System.Windows.Forms.Label();
            this.MaquinaCmb = new System.Windows.Forms.ComboBox();
            this.label5 = new System.Windows.Forms.Label();
            this.OrdenTxt = new System.Windows.Forms.TextBox();
            this.InsertarBtn = new System.Windows.Forms.Button();
            this.DemandaTxt = new System.Windows.Forms.TextBox();
            this.label8 = new System.Windows.Forms.Label();
            this.RetrasoTxt = new System.Windows.Forms.TextBox();
            this.label9 = new System.Windows.Forms.Label();
            ((System.ComponentModel.ISupportInitialize)(this.TablaOperacion)).BeginInit();
            this.OperacionesGrp.SuspendLayout();
            this.SuspendLayout();
            //
            // TablaOperacion
            //
        }
    }
}

```



```
this.TablaOperacion.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.TablaOperacion.Location = new System.Drawing.Point(23, 19);
this.TablaOperacion.Name = "TablaOperacion";
this.TablaOperacion.Size = new System.Drawing.Size(540, 132);
this.TablaOperacion.TabIndex = 0;
//
// NombreCmb
//
this.NombreCmb.FormattingEnabled = true;
this.NombreCmb.Location = new System.Drawing.Point(107, 19);
this.NombreCmb.Name = "NombreCmb";
this.NombreCmb.Size = new System.Drawing.Size(120, 21);
this.NombreCmb.TabIndex = 1;
this.NombreCmb.SelectedIndexChanged += new
System.EventHandler(this.NombreCmb_SelectedIndexChanged);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(21, 22);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(47, 13);
this.label1.TabIndex = 2;
this.label1.Text = "Nombre:";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(21, 49);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(52, 13);
this.label2.TabIndex = 3;
this.label2.Text = "Cantidad:";
//
// CantidadTxt
//
this.CantidadTxt.Location = new System.Drawing.Point(107, 46);
this.CantidadTxt.Name = "CantidadTxt";
this.CantidadTxt.Size = new System.Drawing.Size(120, 20);
this.CantidadTxt.TabIndex = 3;
//
// Fecha
//
this.Fecha.Location = new System.Drawing.Point(107, 72);
this.Fecha.Name = "Fecha";
this.Fecha.Size = new System.Drawing.Size(211, 20);
this.Fecha.TabIndex = 5;
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(21, 76);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(80, 13);
this.label3.TabIndex = 6;
this.label3.Text = "Fecha Entrega:";
//
// DRetrasoTxt
//
this.DRetrasoTxt.Location = new System.Drawing.Point(436, 19);
this.DRetrasoTxt.Name = "DRetrasoTxt";
this.DRetrasoTxt.Size = new System.Drawing.Size(120, 20);
this.DRetrasoTxt.TabIndex = 2;
```



```
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(331, 22);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(83, 13);
this.label4.TabIndex = 7;
this.label4.Text = "Días de retraso:";
//
// OperacionesGrp
//
this.OperacionesGrp.Controls.Add(this.BorrarLstBtn);
this.OperacionesGrp.Controls.Add(this.InsertarOrdenBtn);
this.OperacionesGrp.Controls.Add(this.label7);
this.OperacionesGrp.Controls.Add(this.TiempoTxt);
this.OperacionesGrp.Controls.Add(this.label6);
this.OperacionesGrp.Controls.Add(this.MaquinaCmb);
this.OperacionesGrp.Controls.Add(this.label5);
this.OperacionesGrp.Controls.Add(this.OrdenTxt);
this.OperacionesGrp.Controls.Add(this.TablaOperacion);
this.OperacionesGrp.Location = new System.Drawing.Point(24, 138);
this.OperacionesGrp.Name = "OperacionesGrp";
this.OperacionesGrp.Size = new System.Drawing.Size(586, 252);
this.OperacionesGrp.TabIndex = 9;
this.OperacionesGrp.TabStop = false;
this.OperacionesGrp.Text = "Operaciones:";
//
// BorrarLstBtn
//
this.BorrarLstBtn.Location = new System.Drawing.Point(388, 219);
this.BorrarLstBtn.Name = "BorrarLstBtn";
this.BorrarLstBtn.Size = new System.Drawing.Size(90, 27);
this.BorrarLstBtn.TabIndex = 12;
this.BorrarLstBtn.Text = "Borrar Lista";
this.BorrarLstBtn.UseVisualStyleBackColor = true;
this.BorrarLstBtn.Click += new System.EventHandler(this.BorrarLstBtn_Click);
//
// InsertarOrdenBtn
//
this.InsertarOrdenBtn.Location = new System.Drawing.Point(490, 219);
this.InsertarOrdenBtn.Name = "InsertarOrdenBtn";
this.InsertarOrdenBtn.Size = new System.Drawing.Size(90, 27);
this.InsertarOrdenBtn.TabIndex = 10;
this.InsertarOrdenBtn.Text = "Añadir";
this.InsertarOrdenBtn.UseVisualStyleBackColor = true;
this.InsertarOrdenBtn.Click += new System.EventHandler(this.InsertarOrdenBtn_Click);
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(410, 174);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(45, 13);
this.label7.TabIndex = 14;
this.label7.Text = "Tiempo:";
//
// TiempoTxt
//
this.TiempoTxt.Location = new System.Drawing.Point(461, 171);
this.TiempoTxt.Name = "TiempoTxt";
this.TiempoTxt.Size = new System.Drawing.Size(71, 20);
this.TiempoTxt.TabIndex = 9;
//
```



```
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(206, 174);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(51, 13);
this.label6.TabIndex = 12;
this.label6.Text = "Máquina:";
//
// MaquinaCmb
//
this.MaquinaCmb.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.MaquinaCmb.FormattingEnabled = true;
this.MaquinaCmb.Location = new System.Drawing.Point(263, 171);
this.MaquinaCmb.Name = "MaquinaCmb";
this.MaquinaCmb.Size = new System.Drawing.Size(120, 21);
this.MaquinaCmb.TabIndex = 8;
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(20, 174);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(71, 13);
this.label5.TabIndex = 10;
this.label5.Text = "Nº Operación";
//
// OrdenTxt
//
this.OrdenTxt.Location = new System.Drawing.Point(97, 171);
this.OrdenTxt.Name = "OrdenTxt";
this.OrdenTxt.Size = new System.Drawing.Size(71, 20);
this.OrdenTxt.TabIndex = 7;
//
// InsertarBtn
//
this.InsertarBtn.Location = new System.Drawing.Point(539, 409);
this.InsertarBtn.Name = "InsertarBtn";
this.InsertarBtn.Size = new System.Drawing.Size(71, 28);
this.InsertarBtn.TabIndex = 11;
this.InsertarBtn.Text = "Guardar";
this.InsertarBtn.UseVisualStyleBackColor = true;
this.InsertarBtn.Click += new System.EventHandler(this.InsertarBtn_Click);
//
// DemandaTxt
//
this.DemandaTxt.Location = new System.Drawing.Point(436, 46);
this.DemandaTxt.Name = "DemandaTxt";
this.DemandaTxt.Size = new System.Drawing.Size(120, 20);
this.DemandaTxt.TabIndex = 4;
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(331, 49);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(99, 13);
this.label8.TabIndex = 11;
this.label8.Text = "Demanda Mensual:";
//
// RetrasoTxt
//
this.RetrasoTxt.Location = new System.Drawing.Point(436, 72);
this.RetrasoTxt.Name = "RetrasoTxt";
```




```

    this.RetrasoTxt.Size = new System.Drawing.Size(120, 20);
    this.RetrasoTxt.TabIndex = 6;
    //
    // label9
    //
    this.label9.AutoSize = true;
    this.label9.Location = new System.Drawing.Point(331, 76);
    this.label9.Name = "label9";
    this.label9.Size = new System.Drawing.Size(58, 13);
    this.label9.TabIndex = 12;
    this.label9.Text = "% Retraso:";
    //
    // PiezaForm
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(622, 449);
    this.Controls.Add(this.RetrasoTxt);
    this.Controls.Add(this.label9);
    this.Controls.Add(this.DemandaTxt);
    this.Controls.Add(this.label8);
    this.Controls.Add(this.InsertarBtn);
    this.Controls.Add(this.OperacionesGrp);
    this.Controls.Add(this.DRetrasoTxt);
    this.Controls.Add(this.label4);
    this.Controls.Add(this.label3);
    this.Controls.Add(this.Fecha);
    this.Controls.Add(this.CantidadTxt);
    this.Controls.Add(this.label2);
    this.Controls.Add(this.label1);
    this.Controls.Add(this.NombreCmb);
    this.Name = "PiezaForm";
    this.Text = "Formulario de piezas";
    this.Load += new System.EventHandler(this.PiezaForm_Load);
    ((System.ComponentModel.ISupportInitialize)(this.TablaOperacion)).EndInit();
    this.OperacionesGrp.ResumeLayout(false);
    this.OperacionesGrp.PerformLayout();
    this.ResumeLayout(false);
    this.PerformLayout();

}

#endregion

private System.Windows.Forms.DataGridView TablaOperacion;
private System.Windows.Forms.ComboBox NombreCmb;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.TextBox CantidadTxt;
private System.Windows.Forms.DateTimePicker Fecha;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.TextBox DRetrasoTxt;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.GroupBox OperacionesGrp;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.TextBox TiempoTxt;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.ComboBox MaquinaCmb;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.TextBox OrdenTxt;
private System.Windows.Forms.Button InsertarOrdenBtn;
private System.Windows.Forms.Button InsertarBtn;
private System.Windows.Forms.Button BorrarLstBtn;
private System.Windows.Forms.TextBox DemandaTxt;

```



```
private System.Windows.Forms.Label label8;  
private System.Windows.Forms.TextBox RetrasoTxt;  
private System.Windows.Forms.Label label9;  
}
```



B.12. Código Principal.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Planificador
{
    public partial class Principal : Form
    {
        static ListaMaquinas _LMaquinas;
        static ListaPiezas _LPiezas;

        public static ListaMaquinas LMaquinas { get { return _LMaquinas; } set { _LMaquinas = value; } }
        public static ListaPiezas LPiezas { get { return _LPiezas; } set { _LPiezas = value; } }

        public Principal()
        {
            InitializeComponent();
        }

        private void Principal_Load(object sender, EventArgs e)
        {
            LPiezas = new ListaPiezas();
        }

        private void MaquinariaBtn_Click(object sender, EventArgs e)
        {
            MaquinariaForm MF = new MaquinariaForm();
            MF.Show();
        }

        private void PiezasBtn_Click(object sender, EventArgs e)
        {
            if (LMaquinas != null)
            {
                PiezaForm PF=new PiezaForm();
                PF.Show();
            }
            else
            {
                MessageBox.Show("Debe existir almenos una maquina", "ERROR!");
            }
        }

        private void GenerarBtn_Click(object sender, EventArgs e)
        {
            string fich_p = @"C:\tmp\retrasos.txt";
            ListaOperacion LOpAux;
            Solucion SolucionFinal;
            List<Solucion> LS= new List<Solucion>();
            Operacion op;
            Proceso p;
            double aux_fecha=0;
            int orden;
            int iteracion = 1;
            int indice, npiezas, pos_op, npiezasretrasadas;
        }
    }
}
```



```

double resultado, aux;
SolucionFinal = new Solucion();
if (LPiezas.LPiezas.Count > 0)
{
    System.IO.StreamWriter sw = new System.IO.StreamWriter(fich_p);
    if (!Aleatoridad.Checked)
    {
        if (RepeticionesTxt.Text != "1")
        {
            RepeticionesTxt.Text = "1";
            MessageBox.Show("Solamente se hará 1 iteración");
        }
    }
    while (iteracion <= int.Parse(RepeticionesTxt.Text))
    {
        indice = 0;
        npiezas = 0;
        npiezasretrasadas = 0;
        pos_op = 0;
        resultado = 0;
        aux = 0;
        while (!LPiezas.Finalizado()) //Selección de operaciones
        {
            LOpAux = _LPiezas.OperacionesCandidatas();
            LOpAux = SeleccionSegunReadys(LOpAux);
            LOpAux = MenorTiempoProceso(LOpAux);
            op = ElegirOperacion(LOpAux);
            FinalizarOperacion(op);
            op.Tiempo_Inicio = op.Tiempo_Inicio / 60;
            op.Tiempo_Fin = op.Tiempo_Fin / 60;
            p = new Proceso(op, iteracion);
            SolucionFinal.Add_Proceso(p);
            //MessageBox.Show(op.Pieza.ToString() + " " + op.Orden.ToString() + " " +
op.Maquina.ToString() + " " + op.Tiempo_Inicio.ToString() + " " + op.Tiempo_Fin.ToString());
            op.Tiempo_Inicio = op.Tiempo_Inicio * 60;
            op.Tiempo_Fin = op.Tiempo_Fin * 60;
            LOpAux.LOperacion.Clear();
        }
        // Calculo de retraso medio de la pieza enésima
        while (indice < LPiezas.LPiezas.Count)
        {
            if (LPiezas.LPiezas[indice].Cantidad != 0)
            {
                npiezas = npiezas + 1;
                LOpAux = LPiezas.LPiezas[indice].LOperacion;
                pos_op = LOpAux.LOperacion.Count - 1;
                orden = LOpAux.LOperacion[pos_op].Orden;

                while (orden == LOpAux.LOperacion[pos_op].Orden &&
LOpAux.LOperacion[pos_op].Tiempo_Fin == 0)
                {
                    pos_op--;
                }
                switch (LPiezas.LPiezas[indice].Fecha.DayOfWeek.ToString())
                {
                    case "Monday": aux = 16 * 60;
                        aux_fecha = 7 * 16 * 60;
                        break;
                    case ("Tuesday"): aux = 32 * 60;
                        aux_fecha = 7 * 16 * 60;
                        break;
                    case ("Wednesday"): aux = 48 * 60;
                        aux_fecha = 7 * 16 * 60;
                        break;
                }
            }
        }
    }
}

```



```

        case ("Thursday"): aux = 64 * 60;
            aux_fecha = 7 * 16 * 60;
            break;
        case ("Friday"): aux = 80 * 60;
            aux_fecha = 7 * 16 * 60;
            break;
        case ("Saturday"): aux = 96 * 60;
            aux_fecha = 7 * 16 * 60;
            break;
        case ("Sunday"): aux = 0;
            aux_fecha = 0 * 16 * 60;
            break;
    }
    aux = Math.Max((LOpAux.LOperacion[pos_op].Tiempo_Fin-aux_fecha), 0);
    //aux = Math.Max((LOpAux.LOperacion[pos_op].Tiempo_Fin - aux), 0);
    if (aux > 0)
    {
        npiezasretrasadas++;
    }
    resultado = resultado + aux;
    aux = aux / 60;
    sw.WriteLine(iteracion.ToString() + " " + LOpAux.LOperacion[pos_op].Pieza.ToString() + " "
+ LOpAux.LOperacion[pos_op].Orden.ToString() + " " + aux.ToString());
    //MessageBox.Show(LOpAux.LOperacion[pos_op].Pieza.ToString() + " " +
LOpAux.LOperacion[pos_op].Orden.ToString() + " " + aux.ToString());
    aux = 0;
    pos_op = 0;
}
indice++;
}
resultado = resultado / npiezas / 60;
sw.WriteLine("Nº de piezas en retraso=" + npiezasretrasadas.ToString());
SolucionFinal.AñadirRetrasoMedio(resultado);
//Fin retraso medio pieza enésima
indice = 0;
//Reseteamos tiempos de maquinas y piezas
LMaquinas.Inicializar();
while (indice < LPiezas.LPiezas.Count)
{
    LPiezas.LPiezas[indice].Inicializar();
    indice++;
}
iteracion = iteracion + 1;
// Fin Reset
}

sw.Close();
SolucionFinal.Save();
MessageBox.Show("Secuenciador Finalizado", "FIN");
// Mostrar mejor repeticion
indice = 0;
pos_op = -1;
while (indice < SolucionFinal.RetrasoMedio.Count)
{
    if (pos_op < 0)
    {
        pos_op = indice;
    }
    else
    {
        if (SolucionFinal.RetrasoMedio[indice] < SolucionFinal.RetrasoMedio[pos_op])
        {
            pos_op = indice;
        }
    }
}

```



```

    }
    indice++;
}
pos_op++;
MessageBox.Show("Retraso Medio = " + SolucionFinal.RetrasoMedio[pos_op - 1].ToString(),
"Mejor Solución: " + pos_op.ToString());
// Fin Mostrar mejor repeticion
//Reseteamos tiempos de maquinas y piezas
LMaquinas.Inicializar();
while (indice < LPiezas.LPiezas.Count)
{
    LPiezas.LPiezas[indice].Inicializar();
    indice++;
}
iteracion = iteracion + 1;
// Fin Reset
}
else
{
    MessageBox.Show("No existe ninguna pieza", "ERROR!");
}
}

```

```

private ListaOperacion MenorTiempoProceso(ListaOperacion LOp)
{
    ListaOperacion LResultado = new ListaOperacion(LOp);
    int indice = 0;
    int cont = 0, aux=0;
    double min=0;
    while (indice < LOp.LOperacion.Count)
    {
        while (cont < LOp.LOperacion.Count)
        {
            if (LOp.LOperacion[cont].Pieza == LOp.LOperacion[indice].Pieza)
            {
                if (LOp.LOperacion[cont].Tiempo <= LOp.LOperacion[indice].Tiempo)
                {
                    min = LOp.LOperacion[cont].Tiempo;
                    aux=cont;
                }
            }
            cont++;
        }
        if (LOp.LOperacion[indice].Pieza == LOp.LOperacion[aux].Pieza &&
LOp.LOperacion[indice].Tiempo > min)
        {
            LOp.LOperacion.RemoveAt(indice);
        }
        cont = 0;
        indice++;
    }
    return LResultado;
}

```

```

private ListaOperacion SeleccionSegunReadys(ListaOperacion LOp)
{
    ListaOperacion LResultado=new ListaOperacion();
    Pieza pieza_aux;
    int indice = 0;
    double ready = -1, ready_aux = 0;
    Maquinaria maquina_aux;
    Operacion op_aux;
    while (indice < LOp.LOperacion.Count)
    {

```



```

op_aux = LOp.LOperacion[indice];
pieza_aux = LPiezas.Find(op_aux.Pieza);
maquina_aux = _LMaquinas.Find(op_aux.Maquina);
if (pieza_aux.Operacion_anterior != null)
{
    if (op_aux.Maquina == pieza_aux.Operacion_anterior.Maquina)
    {
        ready_aux = pieza_aux.Operacion_anterior.Tiempo_Fin;
    }
    else
    {
        if (op_aux.Tiempo >= pieza_aux.Operacion_anterior.Tiempo)
        {
            ready_aux = pieza_aux.Operacion_anterior.Tiempo_Inicio + 2 + 15 *
pieza_aux.Operacion_anterior.Tiempo;
        }
        else
        {
            ready_aux = pieza_aux.Operacion_anterior.Tiempo_Fin - (op_aux.Tiempo *
pieza_aux.Cantidad + maquina_aux.Preparacion) + 2 + 15 * pieza_aux.Operacion_anterior.Tiempo;
        }
    }
}
else
{
    ready_aux = 0;
}
ready_aux = Math.Max(ready_aux, maquina_aux.Tiempo);
if (ready_aux <= ready || ready == -1)
{
    if (ready_aux < ready)
    {
        LResultado.LOperacion.Clear();
    }
    LResultado.LOperacion.Add(LOp.LOperacion[indice]);
    ready = ready_aux;
}
indice++;
}
return LResultado;
}

private Operacion ElegirOperacion(ListaOperacion LOp)
{
    int indice = 1, posicion = 0;
    Pieza pieza, pieza_aux;

    pieza = LPiezas.Find(LOp.LOperacion[0].Pieza);
    while (indice < LOp.LOperacion.Count)
    {
        pieza_aux = LPiezas.Find(LOp.LOperacion[indice].Pieza);
        if (Criterios(pieza, pieza_aux) != pieza)
        {
            posicion = indice;
            pieza = pieza_aux;
        }
        indice++;
    }
    return (LOp.LOperacion[posicion]);
}

private Pieza Criterios(Pieza p, Pieza p_aux)
{
    double a = (double)p.Restraso;

```



```
double b = (double)p_aux.Retraso;
Pieza resultado=null;
// Criterio de seleccion de operacion sin aleatoriedad.
if (!Aleatoriedad.Checked)
{
    if (a < b)
    {
        resultado = p_aux;
    }
    else
    {
        if (a > b)
        {
            resultado = p;
        }
        else
        {
            if (p.DRetraso > p_aux.DRetraso)
            {
                resultado = p;
            }
            else
            {
                if (p.DRetraso < p_aux.DRetraso)
                {
                    resultado = p_aux;
                }
                else
                {
                    if (p.Fecha < p_aux.Fecha)
                    {
                        resultado = p;
                    }
                    else
                    {
                        if (p.Fecha > p_aux.Fecha)
                        {
                            resultado = p_aux;
                        }
                    }
                    else
                    {
                        if (p.Duracion_Total > p_aux.Duracion_Total)
                        {
                            resultado = p;
                        }
                        else
                        {
                            if (p.Duracion_Total < p_aux.Duracion_Total)
                            {
                                resultado = p_aux;
                            }
                            else
                            {
                                if (p.Demanda_Mensual > p_aux.Demanda_Mensual)
                                {
                                    resultado = p;
                                }
                                else
                                {
                                    resultado = p_aux;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```




```

    }
  }
}
n_azar= n_azar - int.Parse(RetrasoDiasTxt.Text);
if (n_azar <= int.Parse(FechaTxt.Text) && n_azar > 0)
{
  if (p.Fecha < p_aux.Fecha)
  {
    resultado = p;
  }
  else
  {
    if (p.Fecha > p_aux.Fecha)
    {
      resultado = p_aux;
    }
    else
    {
      if (p.Duracion_Total > p_aux.Duracion_Total)
      {
        resultado = p;
      }
      else
      {
        if (p.Duracion_Total < p_aux.Duracion_Total)
        {
          resultado = p_aux;
        }
        else
        {
          if (p.Demanda_Mensual > p_aux.Demanda_Mensual)
          {
            resultado = p;
          }
          else
          {
            resultado = p_aux;
          }
        }
      }
    }
  }
}
}
n_azar = n_azar - int.Parse(FechaTxt.Text);
if (n_azar <= int.Parse(DuracionTxt.Text) && n_azar > 0)
{
  if (p.Duracion_Total > p_aux.Duracion_Total)
  {
    resultado = p;
  }
  else
  {
    if (p.Duracion_Total < p_aux.Duracion_Total)
    {
      resultado = p_aux;
    }
    else
    {
      if (p.Demanda_Mensual > p_aux.Demanda_Mensual)
      {
        resultado = p;
      }
      else
      {

```



```

        {
            resultado = p_aux;
        }
    }
}
n_azar = n_azar - int.Parse(DuracionTxt.Text);
if (n_azar <= int.Parse(DemandaTxt.Text) && n_azar > 0)
{
    if (p.Demanda_Mensual > p_aux.Demanda_Mensual)
    {
        resultado = p;
    }
    else
    {
        resultado = p_aux;
    }
}
}
return resultado;
}

private void FinalizarOperacion(Operacion op_aux)
{
    Maquinaria m;
    Pieza p;
    Operacion op;
    m = LMaquinas.Find(op_aux.Maquina);
    p = LPiezas.Find(op_aux.Pieza);
    op = p.LOperacion.Find(op_aux);
    if (p.Operacion_anterior != null)
    {
        if (op.Maquina == p.Operacion_anterior.Maquina)
        {
            op.Tiempo_Inicio = m.Tiempo;
        }
        else
        {
            if (op.Tiempo >= p.Operacion_anterior.Tiempo)
            {
                op.Tiempo_Inicio = p.Operacion_anterior.Tiempo_Inicio + 2 + 15 *
p.Operacion_anterior.Tiempo;
                if (op.Tiempo_Inicio < m.Tiempo)
                {
                    op.Tiempo_Inicio = m.Tiempo;
                }
            }
            else
            {
                op.Tiempo_Inicio = p.Operacion_anterior.Tiempo_Fin - (op.Tiempo * p.Cantidad +
m.Preparacion) + 2 + 15 * p.Operacion_anterior.Tiempo;
                if (op.Tiempo_Inicio < m.Tiempo)
                {
                    op.Tiempo_Inicio = m.Tiempo;
                }
            }
        }
    }
}
else
{
    if (p.Cantidad != 0)
    {
        op.Tiempo_Inicio = m.Tiempo;
    }
}
}
}

```



```

        else
        {
            op.Tiempo_Inicio = 0;
        }
    }
    if (p.Cantidad != 0)
    {
        op.Tiempo_Fin = op.Tiempo_Inicio + (op.Tiempo * p.Cantidad) + m.Preparacion;
        m.Tiempo = op.Tiempo_Fin;
    }
    else
    {
        op.Tiempo_Fin = 0;
    }
    ActivarFinalizacionOrden(op);
    p.Operacion_anterior = op;
}

private void ActivarFinalizacionOrden(Operacion Op)
{
    int indice = 0, orden = 0;
    double duracion=0;
    Pieza p;
    p = LPiezas.Find(Op.Pieza);
    orden = Op.Orden;
    while (indice < p.LOperacion.LOperacion.Count)
    {
        if (p.LOperacion.LOperacion[indice].Orden == orden)
        {
            if (p.LOperacion.LOperacion[indice].Tiempo * p.Cantidad > duracion)
            {
                duracion = p.LOperacion.LOperacion[indice].Tiempo * p.Cantidad;
            }
            p.LOperacion.LOperacion[indice].Finalizado = true;
        }
        indice++;
    }
    p.Duracion_Total = p.Duracion_Total - duracion;
}

private void SaveBtn_Click(object sender, EventArgs e)
{
    string fich_m = @"c:\tmp\maquinas.txt";
    string fich_p = @"C:\tmp\piezas.txt";
    int indice=0, indice_aux=0;
    System.IO.StreamWriter sw=new System.IO.StreamWriter(fich_m);
    if (LMaquinas != null)
    {
        while (indice < LMaquinas.LMaquinaria.Count)
        {
            sw.WriteLine(LMaquinas.LMaquinaria[indice].Nombre + " " +
LMaquinas.LMaquinaria[indice].Tiempo + " " + LMaquinas.LMaquinaria[indice].Preparacion);
            indice++;
        }
    }
    sw.Close();
    sw = new System.IO.StreamWriter(fich_p);
    indice = 0;
    if (LPiezas != null)
    {
        while (indice < LPiezas.LPiezas.Count)
        {
            sw.Write(LPiezas.LPiezas[indice].Nombre + " ");
            sw.Write(LPiezas.LPiezas[indice].Cantidad + " ");

```



```

sw.Write(LPiezas.LPiezas[indice].Demanda_Mensual + " ");
sw.Write(LPiezas.LPiezas[indice].Retraso + " ");
sw.Write(LPiezas.LPiezas[indice].DRetraso + " ");
sw.WriteLine(LPiezas.LPiezas[indice].Fecha);
indice_aux = 0;
while (indice_aux < LPiezas.LPiezas[indice].LOperacion.LOperacion.Count)
{
    sw.Write(LPiezas.LPiezas[indice].LOperacion.LOperacion[indice_aux].Pieza + " ");
    sw.Write(LPiezas.LPiezas[indice].LOperacion.LOperacion[indice_aux].Maquina + " ");
    sw.Write(LPiezas.LPiezas[indice].LOperacion.LOperacion[indice_aux].Orden + " ");
    sw.WriteLine(LPiezas.LPiezas[indice].LOperacion.LOperacion[indice_aux].Tiempo);
    indice_aux++;
}
indice++;
}
}
sw.Close();
MessageBox.Show("Maquinas y piezas guardadas", "OK");
}

private void LoadBtn_Click(object sender, EventArgs e)
{
    string fich_m = @"c:\tmp\maquinas.txt";
    string fich_p = @"C:\tmp\piezas.txt";
    string aux="";
    string orden, maquina, pieza, preparacion;
    string nombre = "", cantidad = "", demanda = "", retraso = "", volumen = "", tiempo = "";
    Pieza p;
    Maquinaria m;
    Operacion op;
    ListaOperacion LOp;
    LPiezas = new ListaPiezas();
    LMaquinas = new ListaMaquinas();

    System.IO.StreamReader sr = new System.IO.StreamReader(fich_m);
    while (!sr.EndOfStream)
    {
        aux = sr.ReadLine();
        maquina=aux.Substring(0,aux.IndexOf(' '));
        aux = aux.Substring(aux.IndexOf(' ')+1);
        tiempo = aux.Substring(0, aux.IndexOf(' '));
        aux = aux.Substring(aux.IndexOf(' ') + 1);
        preparacion = aux;
        m = new Maquinaria(maquina, double.Parse(tiempo), double.Parse(preparacion));
        LMaquinas.LMaquinaria.Add(m);
    }
    sr.Close();

    sr = new System.IO.StreamReader(fich_p);
    nombre = "";
    LOp=null;
    p = null;
    while (!sr.EndOfStream)
    {
        aux = sr.ReadLine();
        if (aux.Substring(0, aux.IndexOf(' ')) == nombre)
        {
            pieza = aux.Substring(0, aux.IndexOf(' '));
            aux = aux.Substring(aux.IndexOf(' ') + 1);
            maquina = aux.Substring(0, aux.IndexOf(' '));
            aux = aux.Substring(aux.IndexOf(' ') + 1);
            orden = aux.Substring(0, aux.IndexOf(' '));
            aux = aux.Substring(aux.IndexOf(' ') + 1);
            op = new Operacion(int.Parse(orden), maquina, double.Parse(aux), pieza);

```



```
        LOp.LOperacion.Add(op);
    }
    else
    {
        if (LOp == null)
        {
            LOp = new ListaOperacion();
        }
        else
        {
            p = new Pieza(nombre, int.Parse(cantidad), int.Parse(volumen), double.Parse(demanda),
DateTime.Parse(tiempo), double.Parse(retraso), LOp);
            LPiezas.LPiezas.Add(p);
            LOp = new ListaOperacion();
        }
        nombre = aux.Substring(0, aux.IndexOf(' '));
        aux = aux.Substring(aux.IndexOf(' ') + 1);
        cantidad = aux.Substring(0, aux.IndexOf(' '));
        aux = aux.Substring(aux.IndexOf(' ') + 1);
        demanda = aux.Substring(0, aux.IndexOf(' '));
        aux = aux.Substring(aux.IndexOf(' ') + 1);
        retraso = aux.Substring(0, aux.IndexOf(' '));
        aux = aux.Substring(aux.IndexOf(' ') + 1);
        volumen = aux.Substring(0, aux.IndexOf(' '));
        aux = aux.Substring(aux.IndexOf(' ') + 1);
        tiempo = aux;
    }
}

p = new Pieza(nombre, int.Parse(cantidad), int.Parse(volumen), double.Parse(demanda),
DateTime.Parse(tiempo), double.Parse(retraso), LOp);
LPiezas.LPiezas.Add(p);

sr.Close();
MessageBox.Show("Maquinas y piezas cargadas", "OK");
}
}
}
```



B.13. Código Principal.Designer.cs

```

namespace Planificador
{
    partial class Principal
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.MaquinariaBtn = new System.Windows.Forms.Button();
            this.PiezasBtn = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.RetrasoPorCientoText = new System.Windows.Forms.TextBox();
            this.RetrasoDiasTxt = new System.Windows.Forms.TextBox();
            this.FechaTxt = new System.Windows.Forms.TextBox();
            this.DuracionTxt = new System.Windows.Forms.TextBox();
            this.DemandaTxt = new System.Windows.Forms.TextBox();
            this.label7 = new System.Windows.Forms.Label();
            this.GenerarBtn = new System.Windows.Forms.Button();
            this.label8 = new System.Windows.Forms.Label();
            this.RepeticionesTxt = new System.Windows.Forms.TextBox();
            this.SaveBtn = new System.Windows.Forms.Button();
            this.LoadBtn = new System.Windows.Forms.Button();
            this.Aleatoridad = new System.Windows.Forms.CheckBox();
            this.SuspendLayout();
            //
            // MaquinariaBtn
            //
            this.MaquinariaBtn.Location = new System.Drawing.Point(12, 12);
            this.MaquinariaBtn.Name = "MaquinariaBtn";
            this.MaquinariaBtn.Size = new System.Drawing.Size(116, 35);
            this.MaquinariaBtn.TabIndex = 0;
            this.MaquinariaBtn.Text = "Maquinaria";
        }
    }
}

```




```
this.MaquinariaBtn.UseVisualStyleBackColor = true;
this.MaquinariaBtn.Click += new System.EventHandler(this.MaquinariaBtn_Click);
//
// PiezasBtn
//
this.PiezasBtn.Location = new System.Drawing.Point(134, 12);
this.PiezasBtn.Name = "PiezasBtn";
this.PiezasBtn.Size = new System.Drawing.Size(116, 35);
this.PiezasBtn.TabIndex = 1;
this.PiezasBtn.Text = "Piezas";
this.PiezasBtn.UseVisualStyleBackColor = true;
this.PiezasBtn.Click += new System.EventHandler(this.PiezasBtn_Click);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(24, 91);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(105, 13);
this.label1.TabIndex = 2;
this.label1.Text = "Criterios de prioridad:";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(24, 128);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(71, 13);
this.label2.TabIndex = 3;
this.label2.Text = "(%) de retraso";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(24, 160);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(80, 13);
this.label3.TabIndex = 4;
this.label3.Text = "Días de retraso";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(24, 189);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(91, 13);
this.label4.TabIndex = 5;
this.label4.Text = "Fecha de entrega";
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(24, 219);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(109, 13);
this.label5.TabIndex = 6;
this.label5.Text = "> Duración pendiente";
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(24, 250);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(104, 13);
```



```
this.label6.TabIndex = 7;
this.label6.Text = "> Demanda mensual";
//
// RetrasoPorCientoText
//
this.RetrasoPorCientoText.Location = new System.Drawing.Point(169, 125);
this.RetrasoPorCientoText.Name = "RetrasoPorCientoText";
this.RetrasoPorCientoText.Size = new System.Drawing.Size(50, 20);
this.RetrasoPorCientoText.TabIndex = 8;
this.RetrasoPorCientoText.Text = "60";
//
// RetrasoDiasTxt
//
this.RetrasoDiasTxt.Location = new System.Drawing.Point(169, 157);
this.RetrasoDiasTxt.Name = "RetrasoDiasTxt";
this.RetrasoDiasTxt.Size = new System.Drawing.Size(50, 20);
this.RetrasoDiasTxt.TabIndex = 9;
this.RetrasoDiasTxt.Text = "20";
//
// FechaTxt
//
this.FechaTxt.Location = new System.Drawing.Point(169, 186);
this.FechaTxt.Name = "FechaTxt";
this.FechaTxt.Size = new System.Drawing.Size(50, 20);
this.FechaTxt.TabIndex = 10;
this.FechaTxt.Text = "10";
//
// DuracionTxt
//
this.DuracionTxt.Location = new System.Drawing.Point(169, 216);
this.DuracionTxt.Name = "DuracionTxt";
this.DuracionTxt.Size = new System.Drawing.Size(50, 20);
this.DuracionTxt.TabIndex = 11;
this.DuracionTxt.Text = "5";
//
// DemandaTxt
//
this.DemandaTxt.Location = new System.Drawing.Point(169, 247);
this.DemandaTxt.Name = "DemandaTxt";
this.DemandaTxt.Size = new System.Drawing.Size(50, 20);
this.DemandaTxt.TabIndex = 12;
this.DemandaTxt.Text = "5";
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(166, 91);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(21, 13);
this.label7.TabIndex = 13;
this.label7.Text = "(%)";
//
// GenerarBtn
//
this.GenerarBtn.Location = new System.Drawing.Point(487, 12);
this.GenerarBtn.Name = "GenerarBtn";
this.GenerarBtn.Size = new System.Drawing.Size(117, 35);
this.GenerarBtn.TabIndex = 14;
this.GenerarBtn.Text = "Generar";
this.GenerarBtn.UseVisualStyleBackColor = true;
this.GenerarBtn.Click += new System.EventHandler(this.GenerarBtn_Click);
//
// label8
//
```



```
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(304, 91);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(72, 13);
this.label8.TabIndex = 15;
this.label8.Text = "Repeticiones:";
//
// RepeticionesTxt
//
this.RepeticionesTxt.Location = new System.Drawing.Point(382, 88);
this.RepeticionesTxt.Name = "RepeticionesTxt";
this.RepeticionesTxt.Size = new System.Drawing.Size(50, 20);
this.RepeticionesTxt.TabIndex = 16;
this.RepeticionesTxt.Text = "2";
//
// SaveBtn
//
this.SaveBtn.Location = new System.Drawing.Point(486, 239);
this.SaveBtn.Name = "SaveBtn";
this.SaveBtn.Size = new System.Drawing.Size(116, 35);
this.SaveBtn.TabIndex = 17;
this.SaveBtn.Text = "Guardar";
this.SaveBtn.UseVisualStyleBackColor = true;
this.SaveBtn.Click += new System.EventHandler(this.SaveBtn_Click);
//
// LoadBtn
//
this.LoadBtn.Location = new System.Drawing.Point(364, 239);
this.LoadBtn.Name = "LoadBtn";
this.LoadBtn.Size = new System.Drawing.Size(116, 35);
this.LoadBtn.TabIndex = 18;
this.LoadBtn.Text = "Cargar";
this.LoadBtn.UseVisualStyleBackColor = true;
this.LoadBtn.Click += new System.EventHandler(this.LoadBtn_Click);
//
// Aleatoridad
//
this.Aleatoridad.AutoSize = true;
this.Aleatoridad.Location = new System.Drawing.Point(307, 127);
this.Aleatoridad.Name = "Aleatoridad";
this.Aleatoridad.Size = new System.Drawing.Size(85, 17);
this.Aleatoridad.TabIndex = 19;
this.Aleatoridad.Text = "Aleatoriedad";
this.Aleatoridad.UseVisualStyleBackColor = true;
//
// Principal
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(616, 302);
this.Controls.Add(this.Aleatoridad);
this.Controls.Add(this.LoadBtn);
this.Controls.Add(this.SaveBtn);
this.Controls.Add(this.RepeticionesTxt);
this.Controls.Add(this.label8);
this.Controls.Add(this.GenerarBtn);
this.Controls.Add(this.label7);
this.Controls.Add(this.DemandaTxt);
this.Controls.Add(this.DuracionTxt);
this.Controls.Add(this.FechaTxt);
this.Controls.Add(this.RetrasoDiasTxt);
this.Controls.Add(this.RetrasoPorCientoText);
this.Controls.Add(this.label6);
this.Controls.Add(this.label5);
```



```
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Controls.Add(this.PiezasBtn);
this.Controls.Add(this.MaquinariaBtn);
this.Name = "Principal";
this.Text = "Principal";
this.Load += new System.EventHandler(this.Principal_Load);
this.ResumeLayout(false);
this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button MaquinariaBtn;
private System.Windows.Forms.Button PiezasBtn;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.TextBox RetrasoPorCientoText;
private System.Windows.Forms.TextBox RetrasoDiasTxt;
private System.Windows.Forms.TextBox FechaTxt;
private System.Windows.Forms.TextBox DuracionTxt;
private System.Windows.Forms.TextBox DemandaTxt;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.Button GenerarBtn;
private System.Windows.Forms.Label label8;
private System.Windows.Forms.TextBox RepeticionesTxt;
private System.Windows.Forms.Button SaveBtn;
private System.Windows.Forms.Button LoadBtn;
private System.Windows.Forms.CheckBox Aleatoridad;
}
}
```



B.14. Código Program.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Planificador
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Principal());
        }
    }
}
```



B.15. Código Solucion.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Planificador
{
    public class Solucion
    {
        List<Proceso> _LProceso;
        List<double> _RetrasoMedio;

        public List<Proceso> ListaProceso
        {
            get
            {
                return _LProceso;
            }
            set
            {
                _LProceso = value;
            }
        }

        public List<double> RetrasoMedio
        {
            get
            {
                return _RetrasoMedio;
            }
            set
            {
                _RetrasoMedio = value;
            }
        }

        public Solucion()
        {
            _LProceso = new List<Proceso>();
            _RetrasoMedio = new List<double>();
        }

        public Solucion(List<Proceso> LP)
        {
            _LProceso = new List<Proceso>(LP);
        }

        public void AñadirRetrasoMedio (double RM)
        {
            _RetrasoMedio.Add(RM);
        }

        public void Add_Proceso(Proceso p)
        {
            _LProceso.Add(p);
        }

        public void Save()
        {
            string fich = @"c:\tmp\solucion.txt";
```



```

int indice = 0, dia, hora;
double minuto;
int it = 1;
System.IO.StreamWriter sw = new System.IO.StreamWriter(fich);
while (indice < _LProceso.Count)
{
    if (it != _LProceso[indice].Iteracion)
    {
        sw.WriteLine(_RetrasoMedio[it-1].ToString());
        it = _LProceso[indice].Iteracion;
    }
    sw.Write(_LProceso[indice].Iteracion + " ");
    sw.Write(_LProceso[indice].Op.Orden + " ");
    sw.Write(_LProceso[indice].Op.Pieza + " ");
    sw.Write(_LProceso[indice].Op.Maquina + " ");
    sw.Write(_LProceso[indice].Op.Tiempo_Inicio + " ");
    sw.Write(_LProceso[indice].Op.Tiempo_Fin + " ");
    dia=(int)_LProceso[indice].Op.Tiempo_Inicio / 16;
    switch (dia)
    {
        case 0: sw.Write("Lunes "); break;
        case 1: sw.Write("Martes "); break;
        case 2: sw.Write("Miércoles "); break;
        case 3: sw.Write("Jueves "); break;
        case 4: sw.Write("Viernes "); break;
        case 5: sw.Write("Sábado "); break;
        case 6: sw.Write("Lunes2 "); break;
        case 7: sw.Write("Martes2 "); break;
        case 8: sw.Write("Miércoles2 "); break;
        case 9: sw.Write("Jueves2 "); break;
        case 10: sw.Write("Viernes2 "); break;
        case 11: sw.Write("Sábado2 "); break;
    }
    hora = (int)_LProceso[indice].Op.Tiempo_Inicio % 16 + 6;
    sw.Write(hora + "h" + " ");
    minuto=(double)(_LProceso[indice].Op.Tiempo_Inicio % 16 - (hora-6));
    minuto=minuto * 60;
    sw.Write(minuto + "" + " ");
    dia = (int)_LProceso[indice].Op.Tiempo_Fin / 16;
    switch (dia)
    {
        case 0: sw.Write("Lunes "); break;
        case 1: sw.Write("Martes "); break;
        case 2: sw.Write("Miércoles "); break;
        case 3: sw.Write("Jueves "); break;
        case 4: sw.Write("Viernes "); break;
        case 5: sw.Write("Sábado "); break;
        case 6: sw.Write("Lunes2 "); break;
        case 7: sw.Write("Martes2 "); break;
        case 8: sw.Write("Miércoles2 "); break;
        case 9: sw.Write("Jueves2 "); break;
        case 10: sw.Write("Viernes2 "); break;
        case 11: sw.Write("Sábado2 "); break;
    }
    hora = (int)_LProceso[indice].Op.Tiempo_Fin % 16 + 6;
    sw.Write(hora + "h" + " ");
    minuto = (double)(_LProceso[indice].Op.Tiempo_Fin % 16 - (hora - 6));
    minuto = minuto * 60;
    sw.WriteLine(minuto + "");
    indice++;
}
sw.WriteLine(_RetrasoMedio[it - 1].ToString());
sw.Close();
}

```



}
}

