UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CONSTRUCCIÓN DE SOFTWARE

GRUPO #4:

- ENDARA TIGRERO BRAYAN ARIEL
- OBREGON GUAMAN JOEL FERNANDO
- OSORIO MORALES NAOMI YERLIN
- SOLANO MONCADA DANNY ISMAEL
- VALENCIA MOSQUERA DENNYS SAMUEL

DOCENTE:

ING. ZUMBA GAMBOA JOHANNA

CURSO:

SOF-S-NO-6-7

2024-2025

CONTENIDO

Proyecto sistema votación

1. Clases de conectar

a. Clase conexión

```
using System;
using System.Collections.Generic;
using MySql.Data.MySqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.conectar
{
    public class Conexion
    {
        private static string servidor = "localhost";
        private static string bd = "BDVotacion";
        private static string usuario = "root";
        private static string password = "0929453835";
        private static string port = "3306";

        public static MySqlConnection GetConexion()
        {
            string cadenaConexio = "Server=" + servidor + ";Port=" + port +
";Database=" + bd + ";Uid=" + usuario + ";password=" + password + ";";
            try
            {
                MySqlConnection conexioBD = new
MySqlConnection(cadenaConexio);
                return conexioBD;

            }
            catch (MySqlException ex)
            {
                MessageBox.Show(ex.Message, "Error Conexion");
                return null;
            }
        }
    }
}
```

b. Clase carrera

```
using SistemaVotacion.modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MySql.Data.MySqlClient;
using System.Windows.Forms;

namespace SistemaVotacion.conectar
{
    public class Carrera_con
    {

        public static List<Carrera> Listar()
```

```
        {
            List<Carrera> carreras = new List<Carrera>();

            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * from carrera";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);

                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Carrera carrera = new Carrera
                        {
                            IdCarrera =
int.Parse(reader["idCarrera"].ToString()),
                            NomCarrera = reader["nomCarrera"].ToString(),

                        };

                        carreras.Add(carrera);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return carreras;
        }
    }
}
```

c. Clase estudiante

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using SistemaVotacion.modelo;

namespace SistemaVotacion.conectar
{
    public class Estudiante_con
    {
        public static int Insertar(Estudiante e)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
```

```
                string consulta = "INSERT INTO estudiante(cedula,nombre,
apellidos, correo, celular, idSemestre, contrasenia)" +
                                  " VALUES (@cedula, @nombre, @apellidos,
@correo, @celular, @idSemestre, @contrasenia)";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@nombre", e.Nombre);
                SQLComando.Parameters.AddWithValue("@apellidos",
e.Apellidos);
                SQLComando.Parameters.AddWithValue("@correo", e.Correo);
                SQLComando.Parameters.AddWithValue("@celular", e.Celular);
                SQLComando.Parameters.AddWithValue("@idSemestre",
e.IdSemestre);
                SQLComando.Parameters.AddWithValue("@contrasenia",
e.Contrasenia);
                SQLComando.Parameters.AddWithValue("@cedula", e.Cedula);

                SQLComando.ExecuteNonQuery();

                int ultimoIdInsertado = (int)SQLComando.LastInsertedId;
                return ultimoIdInsertado;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return -1;
            }
            finally
            {
                conn.Close();
            }
        }

        public static bool Actualizar(Estudiante e)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "UPDATE estudiante SET nombre = @nombre,
apellidos = @apellidos, correo = @correo, celular = @celular, " +
                                  "idSemestre = @idSemestre, contrasenia =
@contrasenia WHERE idEstudiante = @idEstudiante";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@nombre", e.Nombre);
                SQLComando.Parameters.AddWithValue("@apellidos",
e.Apellidos);
                SQLComando.Parameters.AddWithValue("@correo", e.Correo);
                SQLComando.Parameters.AddWithValue("@celular", e.Celular);
                SQLComando.Parameters.AddWithValue("@idSemestre",
e.IdSemestre);
                SQLComando.Parameters.AddWithValue("@contrasenia",
e.Contrasenia);
                SQLComando.Parameters.AddWithValue("@idEstudiante",
e.IdEstudiante);

                int filasAfectadas = SQLComando.ExecuteNonQuery();
                return filasAfectadas > 0;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return false;
```

```
            }
            finally
            {
                conn.Close();
            }
        }


        public static List<Estudiante> Estudiantes_votaron()
        {
            MySqlConnection conn = Conexion.GetConexion();
            List<Estudiante> estudiantes = new List<Estudiante>();

            try
            {
                conn.Open();
                string consulta = "SELECT e.*,s.nomSemestre FROM estudiante
e INNER JOIN votacion_estudiante ve ON ve.idEstudiante=e.idEstudiante INNER
JOIN semestre s ON s.idSemestre= e.idSemestre";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);

                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Estudiante estudiante = new Estudiante
                        {
                            Cedula = reader["cedula"].ToString(),
                            IdEstudiante =
int.Parse(reader["idEstudiante"].ToString()),
                            Nombre = reader["nombre"].ToString(),
                            Apellidos = reader["apellidos"].ToString(),
                            Correo = reader["correo"].ToString(),
                            Celular = reader["celular"].ToString(),
                            NomSemestre = reader["nomSemestre"].ToString(),
                            Contrasenia = reader["contrasenia"].ToString()

                        };

                        estudiantes.Add(estudiante);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return estudiantes;
        }

        public static List<Estudiante> ConsultaCedula(string cedula)
        {
            MySqlConnection conn = Conexion.GetConexion();
            List<Estudiante> estudiantes = new List<Estudiante>();
            try
            {
                conn.Open();
```

```
                string consulta = "SELECT e.*, s.nomSemestre FROM estudiante
e INNER JOIN semestre s ON e.idSemestre=s.idSemestre WHERE e.cedula =
@buscar";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@buscar", "%" + cedula +
"%");

                MySqlDataReader reader = SQLComando.ExecuteReader();

                while (reader.Read())
                {
                    Estudiante estudiante = new Estudiante
                    {
                        IdEstudiante = reader.GetInt32("idEstudiante"),
                        Nombre = reader.GetString("nombre"),
                        Apellidos = reader.GetString("apellidos"),
                        Cedula = reader.GetString("cedula"),
                        Correo = reader.GetString("correo"),
                        Celular = reader.GetString("celular"),
                        IdSemestre = reader.GetInt32("idSemestre"),
                        Contrasenia = reader.GetString("contrasenia"),
                        NomSemestre = reader.GetString("nomSemestre")
                    };
                    estudiantes.Add(estudiante);

                }
                return estudiantes;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return null;
            }
            finally
            {
                conn.Close();
            }
        }

        public static List<Estudiante> ConsultarApellido(string apellido)
        {
            MySqlConnection conn = Conexion.GetConexion();
            List<Estudiante> estudiantes = new List<Estudiante>();
            try
            {
                conn.Open();
                string consulta = "SELECT e.*, s.nomSemestre FROM estudiante
e INNER JOIN semestre s ON e.idSemestre=s.idSemestre WHERE e.apellidos =
@buscar";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@buscar", "%" + apellido
+ "%");

                MySqlDataReader reader = SQLComando.ExecuteReader();

                while (reader.Read())
                {
                    Estudiante estudiante = new Estudiante
                    {
                        IdEstudiante = reader.GetInt32("idEstudiante"),
                        Nombre = reader.GetString("nombre"),
                        Apellidos = reader.GetString("apellidos"),
                        Cedula = reader.GetString("cedula"),
```

```
                    Correo = reader.GetString("correo"),
                    Celular = reader.GetString("celular"),
                    IdSemestre = reader.GetInt32("idSemestre"),
                    Contrasenia = reader.GetString("contrasenia"),
                    NomSemestre = reader.GetString("nomSemestre")
                };
                estudiantes.Add(estudiante);

            }
            return estudiantes;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
            return null;
        }
        finally
        {
            conn.Close();
        }
    }

    public static List<Estudiante> ConsultarSemesntre(string semestre)
    {
        MySqlConnection conn = Conexion.GetConexion();
        List<Estudiante> estudiantes = new List<Estudiante>();
        try
        {
            conn.Open();
            string consulta = "SELECT e.*, s.nomSemestre FROM estudiante
e INNER JOIN semestre s ON e.idSemestre=s.idSemestre WHERE s.nomSemestre =
@buscar";
            MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
            SQLComando.Parameters.AddWithValue("@buscar", "%" + semestre
+ "%");

            MySqlDataReader reader = SQLComando.ExecuteReader();

            while (reader.Read())
            {
                Estudiante estudiante = new Estudiante
                {
                    IdEstudiante = reader.GetInt32("idEstudiante"),
                    Nombre = reader.GetString("nombre"),
                    Apellidos = reader.GetString("apellidos"),
                    Cedula = reader.GetString("cedula"),
                    Correo = reader.GetString("correo"),
                    Celular = reader.GetString("celular"),
                    IdSemestre = reader.GetInt32("idSemestre"),
                    Contrasenia = reader.GetString("contrasenia"),
                    NomSemestre = reader.GetString("nomSemestre")
                };
                estudiantes.Add(estudiante);

            }
            return estudiantes;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
            return null;
        }
        finally
```

```
            {
                conn.Close();
            }
        }

        public static Estudiante ObtenerEstudiantePorId(int idEstudiante)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * FROM estudiante WHERE
idEstudiante = @idEstudiante";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idEstudiante",
idEstudiante);

                MySqlDataReader reader = SQLComando.ExecuteReader();

                if (reader.Read())
                {
                    Estudiante estudiante = new Estudiante
                    {
                        IdEstudiante = reader.GetInt32("idEstudiante"),
                        Nombre = reader.GetString("nombre"),
                        Apellidos = reader.GetString("apellidos"),
                        Cedula = reader.GetString("cedula"),
                        Correo = reader.GetString("correo"),
                        Celular = reader.GetString("celular"),
                        IdSemestre = reader.GetInt32("idSemestre"),
                        Contrasenia = reader.GetString("contrasenia")
                    };
                    return estudiante;
                }
                else
                {
                    return null;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return null;
            }
            finally
            {
                conn.Close();
            }
        }

        public static List<Estudiante> Listar()
        {
            List<Estudiante> estudiantes = new List<Estudiante>();

            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT e.*,s.nomSemestre FROM estudiante
e INNER JOIN semestre s ON e.idSemestre=s.idSemestre";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
```

```
                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Estudiante estudiante = new Estudiante
                        {
                            Cedula = reader["cedula"].ToString(),
                            IdEstudiante =
int.Parse(reader["idEstudiante"].ToString()),
                            Nombre = reader["nombre"].ToString(),
                            Apellidos = reader["apellidos"].ToString(),
                            Correo = reader["correo"].ToString(),
                            Celular = reader["celular"].ToString(),
                            NomSemestre = reader["nomSemestre"].ToString(),
                            Contrasenia = reader["contrasenia"].ToString(),
                        };

                        estudiantes.Add(estudiante);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return estudiantes;
        }

        public static bool Eliminar(int idEstudiante)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "DELETE FROM estudiante WHERE idEstudiante
= @idEstudiante";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idEstudiante",
idEstudiante);

                int filasEliminadas = SQLComando.ExecuteNonQuery();
                return filasEliminadas > 0;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return false;
            }
            finally
            {
                conn.Close();
            }
        }
```

```
        public static bool ActualizarContrasenia(string cedula, string
nuevaContrasenia)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "UPDATE estudiante SET contrasenia =
@nuevaContrasenia WHERE cedula = @cedula";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@nuevaContrasenia",
nuevaContrasenia);
                SQLComando.Parameters.AddWithValue("@cedula", cedula);

                int filasAfectadas = SQLComando.ExecuteNonQuery();
                return filasAfectadas > 0;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return false;
            }
            finally
            {
                conn.Close();
            }
        }

    }
}
```

d. Clase lista

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using SistemaVotacion.modelo;

namespace SistemaVotacion.conectar
{
    public class Lista_con
    {
        public static int Insertar(Lista l)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "INSERT INTO lista(nomLista,
nomPresidente, nomSecretario, nomPrimerVocal, idCarrera, fotoP,
descripcion,cantVotos)" +
                                  " VALUES (@nomLista, @nomPresidente,
@nomSecretario, @nomPrimerVocal, @idCarrera, @fotoP,
@descripcion,@cantVotos)";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@nomLista", l.NomLista);
```

```
                SQLComando.Parameters.AddWithValue("@nomPresidente",
l.NomPresidente);
                SQLComando.Parameters.AddWithValue("@nomSecretario",
l.NomSecretario);
                SQLComando.Parameters.AddWithValue("@nomPrimerVocal",
l.NomPrimerVocal);
                SQLComando.Parameters.AddWithValue("@idCarrera", l.Carrera);
                SQLComando.Parameters.AddWithValue("@fotoP", l.FotoP);
                SQLComando.Parameters.AddWithValue("@descripcion",
l.Descripcion);
                SQLComando.Parameters.AddWithValue("@cantVotos",
l.CantVotos);
                SQLComando.ExecuteNonQuery();

                int ultimoIdInsertado = (int)SQLComando.LastInsertedId;
                return ultimoIdInsertado;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return -1;
            }
            finally
            {
                conn.Close();
            }
        }

        public static List<Lista> Listar()
        {
            List<Lista> listas = new List<Lista>();

            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT l.*,c.nomCarrera FROM lista l
INNER JOIN carrera c ON c.idCarrera=l.idCarrera";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);

                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Lista lista = new Lista
                        {
                            NomLista = reader["nomLista"].ToString(),
                            IdLista =
int.Parse(reader["idLista"].ToString()),
                            NomPresidente =
reader["nomPresidente"].ToString(),
                            NomSecretario =
reader["nomSecretario"].ToString(),
                            NomPrimerVocal =
reader["nomPrimerVocal"].ToString(),
                            NomCarrera = reader["nomCarrera"].ToString(),
                            FotoP = reader["fotoP"].ToString(),
                            Descripcion = reader["descripcion"].ToString(),
                            CantVotos=
int.Parse(reader["cantVotos"].ToString()),
                        };
```

```
                        listas.Add(lista);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return listas;
        }

        public static Lista ObtenerListaPorId(int idLista)
        {
            Lista lista = null;

            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT l.*, c.nomCarrera FROM lista l
INNER JOIN carrera c ON c.idCarrera = l.idCarrera WHERE l.idLista =
@idLista";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idLista", idLista);

                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        lista = new Lista
                        {
                            NomLista = reader["nomLista"].ToString(),
                            IdLista =
int.Parse(reader["idLista"].ToString()),
                            NomPresidente =
reader["nomPresidente"].ToString(),
                            NomSecretario =
reader["nomSecretario"].ToString(),
                            NomPrimerVocal =
reader["nomPrimerVocal"].ToString(),
                            NomCarrera = reader["nomCarrera"].ToString(),
                            FotoP = reader["fotoP"].ToString(),
                            Descripcion = reader["descripcion"].ToString(),
                            Carrera=
int.Parse(reader["idCarrera"].ToString()),
                        };
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }
```

```
            return lista;
        }

        public static bool Actualizar(Lista lista)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "UPDATE lista SET nomLista = @nomLista,
nomPresidente = @nomPresidente, nomSecretario = @nomSecretario, " +
                                  "nomPrimerVocal = @nomPrimerVocal,
idCarrera = @idCarrera, fotoP = @fotoP, descripcion = @descripcion " +
                                  "WHERE idLista = @idLista";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@nomLista",
lista.NomLista);
                SQLComando.Parameters.AddWithValue("@nomPresidente",
lista.NomPresidente);
                SQLComando.Parameters.AddWithValue("@nomSecretario",
lista.NomSecretario);
                SQLComando.Parameters.AddWithValue("@nomPrimerVocal",
lista.NomPrimerVocal);
                SQLComando.Parameters.AddWithValue("@idCarrera",
lista.Carrera);
                SQLComando.Parameters.AddWithValue("@fotoP", lista.FotoP);
                SQLComando.Parameters.AddWithValue("@descripcion",
lista.Descripcion);
                SQLComando.Parameters.AddWithValue("@idLista",
lista.IdLista);

                int filasActualizadas = SQLComando.ExecuteNonQuery();

                return filasActualizadas > 0;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return false;
            }
            finally
            {
                conn.Close();
            }
        }

        public static bool Eliminar(int idLista)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "DELETE FROM lista WHERE idLista =
@idLista";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idLista", idLista);

                int filasEliminadas = SQLComando.ExecuteNonQuery();
                return filasEliminadas > 0;
            }
```

```
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return false;
            }
            finally
            {
                conn.Close();
            }
        }


        public static bool AumentarPuntuacion(int idLista, int idEstudiante)
        {
            MySqlConnection conn = Conexion.GetConexion();
            try
            {
                conn.Open();
                string consulta = "UPDATE lista SET cantVotos= cantVotos+1
WHERE idLista=@idLista";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idLista", idLista);
                SQLComando.ExecuteNonQuery();

                consulta = "INSERT votacion_estudiante(idLista,idEstudiante)
values(@idLista, @idEstudiante)";
                SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idLista", idLista);
                SQLComando.Parameters.AddWithValue("@idEstudiante",
idEstudiante);
                SQLComando.ExecuteNonQuery();
                return true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return false;
        }

        public static bool ExisteVotacionPorEstudiante(int idEstudiante)
        {
            MySqlConnection conn = Conexion.GetConexion();
            try
            {
                conn.Open();
                string consulta = "SELECT COUNT(*) FROM votacion_estudiante
WHERE idEstudiante = @idEstudiante";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idEstudiante",
idEstudiante);

                int count = Convert.ToInt32(SQLComando.ExecuteScalar());
                return count > 0;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
```

```
                return false;
            }
            finally
            {
                conn.Close();
            }
        }



    }

}
```

e. Clase propuesta

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using SistemaVotacion.modelo;

namespace SistemaVotacion.conectar
{
    public class Propuesta_con
    {
        public static int Insertar(Propuesta p)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "INSERT INTO propuestas(dscripcion,
idLista)" +
                                  " VALUES (@dscripcion, @idLista)";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@dscripcion",
p.Descripcion);
                SQLComando.Parameters.AddWithValue("@idLista", p.IdLista);


                SQLComando.ExecuteNonQuery();

                int ultimoIdInsertado = (int)SQLComando.LastInsertedId;
                return ultimoIdInsertado;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                return -1;
            }
            finally
            {
                conn.Close();
            }
        }
```

```
        public static List<Propuesta> ObtenerPropuestasPorIdLista(int
idLista)
        {
            List<Propuesta> propuestas = new List<Propuesta>();

            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * FROM propuestas WHERE idLista =
@idLista";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idLista", idLista);

                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Propuesta propuesta = new Propuesta
                        {
                            IdPropuesta =
int.Parse(reader["idPropuesta"].ToString()),
                            Descripcion = reader["dscripcion"].ToString(),
                            IdLista =
int.Parse(reader["idLista"].ToString())
                        };

                        propuestas.Add(propuesta);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return propuestas;
        }


        public static void EliminarPropuestasPorIdLista(int idLista)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "DELETE FROM propuestas WHERE idLista =
@idLista";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idLista", idLista);

                SQLComando.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
```

```
            finally
            {
                conn.Close();
            }
        }


    }
}
```

f. Clase semestre

```
using MySql.Data.MySqlClient;
using SistemaVotacion.modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.conectar
{
    public class Semestre_con
    {
        public static List<Semestre> Listar()
        {
            List<Semestre> semestres = new List<Semestre>();

            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * from semestre";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);

                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Semestre semestre = new Semestre
                        {
                            IdSemestre =
int.Parse(reader["idSemestre"].ToString()),
                            NomSemestre = reader["nomSemestre"].ToString(),

                        };

                        semestres.Add(semestre);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return semestres;
```

```
        }
    }
}
```

g. Clase usuario

```
using MySql.Data.MySqlClient;
using SistemaVotacion.modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.conectar
{
    public class Usuario_con
    {

        public static Administrador
ValidarIngresoAdministrador(Administrador u)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * FROM usuario WHERE
correo=@correo AND contrasenia=@pass";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@correo", u.Correo);
                SQLComando.Parameters.AddWithValue("@pass", u.Contrasenia);
                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        return u;
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return null;
        }

        public static Estudiante ValidarIngresoEstudiante(Estudiante e)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * FROM estudiante WHERE
correo=@correo AND contrasenia=@pass";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
```

```
                SQLComando.Parameters.AddWithValue("@correo", e.Correo);
                SQLComando.Parameters.AddWithValue("@pass", e.Contrasenia);
                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        e.IdEstudiante = reader.GetInt32("idEstudiante");
                        e.Cedula = Convert.ToString(reader["cedula"]);
                        return e;

                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return null;
        }
    }
}
```

h. Clase votación

```
using MySql.Data.MySqlClient;
using SistemaVotacion.modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.conectar
{
    public class Votacion_con
    {
        public static bool Insertar(Votacion v)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "INSERT INTO votacion(idCandidata,
matricula, tipo)" +
                                  " VALUES (@idCandidata, @matricula,
@tipo)";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@idCandidata",
v.IdCandidata);
                SQLComando.Parameters.AddWithValue("@matricula",
v.Matricula);
                SQLComando.Parameters.AddWithValue("@tipo", v.Tipo);
                SQLComando.ExecuteNonQuery();
                return true;
            }
            catch (Exception ex)
```

```
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }
            return false;
        }

        public static bool BuscarVotacionMissFotogenica(string matricula)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * FROM votacion WHERE
matricula=@matricula AND tipo=1";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@matricula", matricula);
                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        return true;
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                conn.Close();
            }

            return false;
        }

        public static bool BuscarVotacionReinaa(string matricula)
        {
            MySqlConnection conn = Conexion.GetConexion();

            try
            {
                conn.Open();
                string consulta = "SELECT * FROM votacion WHERE
matricula=@matricula AND tipo=2";
                MySqlCommand SQLComando = new MySqlCommand(consulta, conn);
                SQLComando.Parameters.AddWithValue("@matricula", matricula);
                using (MySqlDataReader reader = SQLComando.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        return true;
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
```

```
            }
            finally
            {
                conn.Close();
            }

            return false;
        }
    }
}
```

2. Clases controladoras

a. Buscar controlador

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.Utilidad;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmBuscar_Controlador
    {
        private FrmBuscar vista;
        private FrmPrincipal principal;
        private Estudiante estudiante;

        public FrmBuscar_Controlador(FrmBuscar vista, Estudiante estudiante,
FrmPrincipal principal)
        {
            this.vista = vista;
            this.estudiante = estudiante;
            this.principal = principal;

            this.vista.btnBuscarApellido.Click += BtnBuscarApellido_Click;
            this.vista.btnBuscarCedula.Click += BtnBuscarCedula_Click;
            this.vista.btnBuscarSemestre.Click += BtnBuscarSemestre_Click;


        }

        private void BtnBuscarApellido_Click(object sender, EventArgs e)
        {
            FrmBuscarApellido frm = new FrmBuscarApellido();
            FrmBuscarApellido_Controlador controlador = new
FrmBuscarApellido_Controlador(frm, new Estudiante());
            AbrirHijo(frm);
        }

        private void BtnBuscarCedula_Click(object sender, EventArgs e)
        {
            FrmBuscarCedula frm = new FrmBuscarCedula();
            FrmBuscarCedula_Controlador controlador = new
FrmBuscarCedula_Controlador(frm, new Estudiante());
            AbrirHijo(frm);
```

```
        }

        private void BtnBuscarSemestre_Click(object sender, EventArgs e)
        {
            FrmBuscarSemestre frm = new FrmBuscarSemestre();
            FrmBuscarSemestre_Controlador controlador = new
FrmBuscarSemestre_Controlador(frm, new Estudiante());
            AbrirHijo(frm);
        }




        private void AbrirHijo(object fornHijo)
        {
            if (this.vista.panel_datos.Controls.Count > 0)
            {
                this.vista.panel_datos.Controls.RemoveAt(0);
            }
            Form fh = fornHijo as Form;
            fh.TopLevel = false;
            fh.Dock = DockStyle.Fill;
            this.vista.panel_datos.Controls.Add(fh);
            this.vista.panel_datos.Tag = fh;

            fh.Show();
        }



    }
}
```

b. Buscar apellido

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.Utilidad;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmBuscarApellido_Controlador
    {
        private FrmBuscarApellido vista;
        private Estudiante estudiante;

        public FrmBuscarApellido_Controlador(FrmBuscarApellido vista,
Estudiante estudiante)
        {
            this.vista = vista;
            this.estudiante = estudiante;
            this.vista.txtBuscar.KeyPress += new
KeyPressEventHandler(txtBuscar_KeyPress);
            CargarEstudiantes(Estudiante_con.Listar());
            EstiloTabla.EstilizarDataGridView(this.vista.dgvEstudiantes);
            this.vista.Show();
```

```
        }

        private void txtBuscar_KeyPress(object sender, KeyPressEventArgs e)
        {
            string buscar = this.vista.txtBuscar.Text;
            CargarEstudiantes(Estudiante_con.ConsultarApellido(buscar));

        }

        private void CargarEstudiantes(List<Estudiante> estudiantes)
        {
            vista.dgvEstudiantes.Rows.Clear();
            for (int i = 0; i < estudiantes.Count; i++)
            {
                vista.dgvEstudiantes.Rows.Add(new object[]
{estudiantes[i].IdEstudiante,estudiantes[i].Nombre,
estudiantes[i].Apellidos,
                estudiantes[i].Cedula, estudiantes[i].Correo,
estudiantes[i].Celular, estudiantes[i].NomSemestre});
            }
        }
    }
}
```

c. Buscar cédula

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.Utilidad;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    internal class FrmBuscarCedula_Controlador
    {
        private FrmBuscarCedula vista;
        private Estudiante estudiante;

        public FrmBuscarCedula_Controlador(FrmBuscarCedula vista, Estudiante
estudiante)
        {
            this.vista = vista;
            this.estudiante = estudiante;
            this.vista.txtBuscar.KeyPress += new
KeyPressEventHandler(txtBuscar_KeyPress);
            CargarEstudiantes(Estudiante_con.Listar());
            EstiloTabla.EstilizarDataGridView(this.vista.dgvEstudiantes);
            this.vista.Show();
        }

        private void txtBuscar_KeyPress(object sender, KeyPressEventArgs e)
        {
            string buscar = this.vista.txtBuscar.Text;
            CargarEstudiantes(Estudiante_con.ConsultaCedula(buscar));

        }
```

```
        private void CargarEstudiantes(List<Estudiante> estudiantes)
        {
            vista.dgvEstudiantes.Rows.Clear();
            for (int i = 0; i < estudiantes.Count; i++)
            {
                vista.dgvEstudiantes.Rows.Add(new object[]
{estudiantes[i].IdEstudiante,estudiantes[i].Nombre,
estudiantes[i].Apellidos,
                estudiantes[i].Cedula, estudiantes[i].Correo,
estudiantes[i].Celular, estudiantes[i].NomSemestre});
            }
        }
    }
}
```

d. Buscar semester

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.Utilidad;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmBuscarSemestre_Controlador
    {
        private FrmBuscarSemestre vista;
        private Estudiante estudiante;

        public FrmBuscarSemestre_Controlador(FrmBuscarSemestre vista,
Estudiante estudiante)
        {
            this.vista = vista;
            this.estudiante = estudiante;
            this.vista.txtBuscar.KeyPress += new
KeyPressEventHandler(txtBuscar_KeyPress);
            CargarEstudiantes(Estudiante_con.Listar());
            EstiloTabla.EstilizarDataGridView(this.vista.dgvEstudiantes);
            this.vista.Show();
        }

        private void txtBuscar_KeyPress(object sender, KeyPressEventArgs e)
        {
            string buscar = this.vista.txtBuscar.Text;
            CargarEstudiantes(Estudiante_con.ConsultarSemesntre(buscar));

        }

        private void CargarEstudiantes(List<Estudiante> estudiantes)
        {
            vista.dgvEstudiantes.Rows.Clear();
            for (int i = 0; i < estudiantes.Count; i++)
            {
                vista.dgvEstudiantes.Rows.Add(new object[]
{estudiantes[i].IdEstudiante,estudiantes[i].Nombre,
estudiantes[i].Apellidos,
```

```
                estudiantes[i].Cedula, estudiantes[i].Correo,
estudiantes[i].Celular, estudiantes[i].NomSemestre});
            }
        }
    }
}
```

e. Cambiar contraseña

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmCambiarContrasenia_Controlador
    {
        private FrmCambiarContrasenia vista;
        private Estudiante estudiante;


        public FrmCambiarContrasenia_Controlador(FrmCambiarContrasenia
vista, Estudiante estudiante)
        {
            this.vista = vista;
            this.estudiante = estudiante;
            this.vista.btnCambiarPass.Click += BtnCambiar_Click;
            this.vista.ShowDialog();
        }


        private void BtnCambiar_Click(object sender, EventArgs e)
        {
            string msj = validarCamposVacios();
            if (msj.Equals(""))
            {
                string matricula = vista.txtCedula.Text;
                string pass1 = vista.txtPass.Text;
                string pass2 = vista.txtPass2.Text;
                if (!pass1.Equals(pass2)) {
                    MessageBox.Show("Las contraseñas no coinciden");
                }
                else
                {
                    bool estado =
Estudiante_con.ActualizarContrasenia(matricula, pass1);
                    if (estado)
                    {
                        MessageBox.Show("Actualizó su contraseña
correctamente", "Actualizar Contraseña");
                        vista.Dispose();
                    }
                    else
                    {
                        MessageBox.Show("Error al intentar actualizar
contraseña", "Actualizar Contraseña");
                    }
```

```
                }
            }
            else
            {
                MessageBox.Show(msj, "Actualizar Contraseña");
            }

        }



        private string validarCamposVacios()
        {
            string msj = "";
            if (vista.txtCedula.Text.Equals("")) msj+= "Falta ingresar la
matrícula\n";
            if (vista.txtPass.Text.Equals("")) msj += "Falta ingresar
contraseña\n";
            if (vista.txtPass2.Text.Equals("")) msj += "Falta ingresar
reptir nueva contraseña\n";

            return msj;
        }
    }
}
```

f. Consulta controlador

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.Utilidad;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmConsulta_Controlador
    {
        private FrmConsulta vista;
        private FrmPrincipal principal;
        private Lista lista;
        private Estudiante estudiante;

        public FrmConsulta_Controlador(FrmConsulta vista, Lista lista,
Estudiante estudiante,FrmPrincipal principal)
        {
            this.vista = vista;
            this.lista = lista;
            this.estudiante = estudiante;
            this.principal = principal;
            CargarEstudiantes();
            CargarLista();
            EstiloTabla.EstilizarDataGridView(vista.dgvCandidatos);
            EstiloTabla.EstilizarDataGridView(vista.dgvEstudiantes);

            this.vista.btnEditCandidato.Click += BtnEditarCandidato_Click;
            this.vista.btnEditEstudiante.Click += BtnEditarEstudiante_Click;
```

```
            this.vista.btnEliminarCandidato.Click +=
BtnEliminarCandidato_Click;
            this.vista.btnEliminarEstudiante.Click +=
BtnEliminarEstudiante_Click;
        }

        private void CargarEstudiantes()
        {
            vista.dgvEstudiantes.Rows.Clear();
            List<Estudiante> estudiantes = Estudiante_con.Listar();
            for (int i = 0; i < estudiantes.Count; i++)
            {
                vista.dgvEstudiantes.Rows.Add(new object[]
{estudiantes[i].IdEstudiante,estudiantes[i].Nombre,
estudiantes[i].Apellidos,
                estudiantes[i].Cedula, estudiantes[i].Correo,
estudiantes[i].Celular, estudiantes[i].NomSemestre});
            }
        }

        private void CargarLista()
        {
            vista.dgvCandidatos.Rows.Clear();
            List<Lista> listas = Lista_con.Listar();
            for (int i = 0; i < listas.Count; i++)
            {
                vista.dgvCandidatos.Rows.Add(new object[]
{listas[i].IdLista,
                listas[i].NomLista,listas[i].NomPresidente,
listas[i].NomSecretario,
                listas[i].NomPrimerVocal, listas[i].NomCarrera});
            }
        }

        private void BtnEditarCandidato_Click(object sender, EventArgs e)
        {
            if (this.vista.dgvCandidatos.SelectedRows.Count > 0)
            {
                DataGridViewRow filaSeleccionada =
this.vista.dgvCandidatos.SelectedRows[0];
                int id =
int.Parse(filaSeleccionada.Cells[0].Value.ToString());
                FrmRegistroLista frm = new FrmRegistroLista();
                FrmRegistroLista_Controlador controlador = new
FrmRegistroLista_Controlador(frm, new Lista(), 2,id);
                AbrirHijo(frm);
            }
            else
            {
                MessageBox.Show("Seleccione una fila de la tabla
candidatos");
            }
        }

        private void BtnEditarEstudiante_Click(object sender, EventArgs e)
        {
            if (this.vista.dgvEstudiantes.SelectedRows.Count > 0)
            {
                DataGridViewRow filaSeleccionada =
this.vista.dgvEstudiantes.SelectedRows[0];
                int id =
int.Parse(filaSeleccionada.Cells[0].Value.ToString());
                FrmRegistroEstudiante frm = new FrmRegistroEstudiante();
```

```
                FrmRegistroEstudiante_Controlador controlador = new
FrmRegistroEstudiante_Controlador(frm, new Estudiante(), 2, id);
                CargarEstudiantes();
            }
            else
            {
                MessageBox.Show("Seleccione una fila de la tabla
estudiantes");
            }

        }

        private void BtnEliminarCandidato_Click(object sender, EventArgs e)
        {
            if (this.vista.dgvCandidatos.SelectedRows.Count > 0)
            {
                DataGridViewRow filaSeleccionada =
this.vista.dgvCandidatos.SelectedRows[0];
                int id =
int.Parse(filaSeleccionada.Cells[0].Value.ToString());
                if (Lista_con.Eliminar(id))
                {
                    MessageBox.Show("Lista eliminado", "Elimnar Lista");
                    CargarLista();
                }
                else
                {
                    MessageBox.Show("Error al eliminar lista", "Elimnar
Lista");
                }
            }
            else
            {
                MessageBox.Show("Seleccione una fila de la tabla
candidatos");
            }
        }

        private void BtnEliminarEstudiante_Click(object sender, EventArgs e)
        {
            if (this.vista.dgvEstudiantes.SelectedRows.Count > 0)
            {
                DataGridViewRow filaSeleccionada =
this.vista.dgvEstudiantes.SelectedRows[0];
                int id =
int.Parse(filaSeleccionada.Cells[0].Value.ToString());
                if (Estudiante_con.Eliminar(id))
                {
                    MessageBox.Show("Estudiante eliminado", "Elimnar
Estudiante");
                    CargarEstudiantes();
                }
                else
                {
                    MessageBox.Show("Error al eliminar estudiante", "Elimnar
Estudiante");
                }
            }
            else
            {
                MessageBox.Show("Seleccione una fila de la tabla
estudiantes");
            }
```

```
        }



        private void AbrirHijo(object fornHijo)
        {
            if (this.principal.panel3.Controls.Count > 0)
            {
                this.principal.panel3.Controls.RemoveAt(0);
            }
            Form fh = fornHijo as Form;
            fh.TopLevel = false;
            fh.Dock = DockStyle.Fill;
            this.principal.panel3.Controls.Add(fh);
            this.principal.panel3.Tag = fh;

            fh.Show();
        }

    }
}
```

g. Login controlador

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmLogin_Controlador
    {
        private FrmLogin vista;

        public FrmLogin_Controlador(FrmLogin vista)
        {
            this.vista = vista;
            this.vista.btnIngresar.Click += BtnIngresar_Click;
            this.vista.cbTipo.SelectedIndex = 0;

            this.vista.linkContrasenia.Click += BtnOlvidaContrasenia_Click;
            this.vista.linkRegistro.Click += BtnRegistro_Click;

        }

        public void Iniciar()
        {
            this.vista.ShowDialog();
        }

        private void BtnIngresar_Click(object sender, EventArgs e)
        {
            object objeto=null;
            if (vista.cbTipo.SelectedIndex == 0 ||
vista.cbTipo.SelectedIndex == 1)
            {
```

```
                if(vista.cbTipo.SelectedIndex == 0)  objeto =
Usuario_con.ValidarIngresoAdministrador(new
Administrador(vista.txtCorreo.Text, vista.txtPass.Text));
                else objeto = Usuario_con.ValidarIngresoEstudiante(new
Estudiante(vista.txtCorreo.Text, vista.txtPass.Text));

                if (objeto != null)
                {
                    vista.txtCorreo.Text = "";
                    vista.txtPass.Text = "";
                    vista.cbTipo.SelectedIndex = 0;
                    FrmPrincipal frm = new FrmPrincipal();
                    FrmPrincipal_Controlador controlador = new
FrmPrincipal_Controlador(frm, vista, objeto);
                    this.vista.Hide();
                    controlador.iniciar();

                }
                else
                {
                    MessageBox.Show("Correo y/o Contraseña incorrecto");
                }
            }
            else
            {
                vista.txtCorreo.Text = "";
                vista.txtPass.Text = "";
                vista.cbTipo.SelectedIndex = 0;
                FrmPrincipal frm = new FrmPrincipal();
                FrmPrincipal_Controlador controlador = new
FrmPrincipal_Controlador(frm, vista, objeto);
                this.vista.Hide();
                controlador.iniciar();
            }

        }

        private void BtnOlvidaContrasenia_Click(object sender, EventArgs e)
        {
            FrmCambiarContrasenia frm = new FrmCambiarContrasenia();
            FrmCambiarContrasenia_Controlador controlador = new
FrmCambiarContrasenia_Controlador(frm, new Estudiante());
        }

        private void BtnRegistro_Click(object sender, EventArgs e)
        {
            FrmRegistroEstudiante frm = new FrmRegistroEstudiante();
            FrmRegistroEstudiante_Controlador controlador = new
FrmRegistroEstudiante_Controlador(frm,new Estudiante(),1,0);
        }
    }
}
```

h. Principal controlador

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.Utilidad;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmPrincipal_Controlador
    {
        private FrmPrincipal vista;
        private FrmLogin vistaLogin;
        private Object usuario;
        private static int votar = 0;

        public FrmPrincipal_Controlador(FrmPrincipal vista,FrmLogin
vistaLogin, Object usuario)
        {
            Archivo archivo = new Archivo("iniciar");
            this.vista = vista;
            this.vistaLogin = vistaLogin;
            this.usuario = usuario;

            this.vista.btnInscripcion.Click += BtnInscripcion_Click;
            this.vista.btnConsulta.Click += BtnConsulta_Click;
            this.vista.btnResultados.Click += BtnResultados_Click;
            this.vista.btnListas.Click+= BtnListas_Click;
            this.vista.btnBuscar.Click += BtnBuscar_Click;
            this.vista.btnVisualizar.Click += BtnVisualizar_Click;

            this.vista.btnGuardar.Click += BtnGuardar_Click;
            this.vista.FormClosing += Vista_FormClosing;
            this.vista.btnCerrar.Click += BtnCerrar_Click;

            try
            {
                string inicio = archivo.Obtener();
                if (inicio.Equals(""))
                {
                    vista.radBDetener.Checked = true;
                    votar = 0;
                }
                else if (int.Parse(inicio) == 0)
                {
                    vista.radBDetener.Checked = true;
                    votar = 0;
                }
                else
                {
                    vista.radBIniciar.Checked = true;
                    votar = 1;
                }
            }
            catch (Exception ) {  }


            if (usuario is Administrador)//administrador
            {
                this.vista.btnListas.Visible = false;
                this.vista.btnVisualizar.Visible = false;
            }
            else if(usuario is Estudiante)//estudiante
            {
                this.vista.btnInscripcion.Visible = false;
                this.vista.btnConsulta.Visible = false;
                this.vista.panel_iniciar.Visible = false;
```

```
                this.vista.btnBuscar.Visible = false;

            }
            else//invitado
            {
                this.vista.btnBuscar.Visible = false;
                this.vista.btnInscripcion.Visible=false;
                this.vista.btnListas.Visible=false;
                this.vista.btnConsulta.Visible = false;
                this.vista.btnResultados.Visible=false;
                this.vista.panel_iniciar.Visible = false;
            }
        }


        private void Vista_FormClosing(object sender, FormClosingEventArgs
e)
        {
            vista.Dispose();
            vistaLogin.Show();
        }

        public void iniciar()
        {
            this.vista.ShowDialog();
        }

        private void BtnCerrar_Click(object sender, EventArgs e)
        {
            vista.Dispose();
            vistaLogin.Show();
        }

        private void BtnGuardar_Click(object sender, EventArgs e)
        {
            Archivo archivo = new Archivo("iniciar");
            string dato = "1";
            if (vista.radBDetener.Checked)
            {
                dato = "0";
            }

            if (archivo.Guardar(dato)) MessageBox.Show("Se Guardó
exitosamente");
            else MessageBox.Show("Error al guardar");

        }

        private void BtnVisualizar_Click(object sender, EventArgs e)
        {
            FrmVisualizarLista frm = new FrmVisualizarLista();
            FrmVisualizarLista_Controlador controlador = new
FrmVisualizarLista_Controlador(frm, new Lista());
            AbrirHijo(frm);
        }

        private void BtnBuscar_Click(object sender, EventArgs e)
        {
            FrmBuscar frm = new FrmBuscar();
            FrmBuscar_Controlador controlador = new
FrmBuscar_Controlador(frm, new Estudiante(), vista);
            AbrirHijo(frm);
        }
```

```
        private void BtnInscripcion_Click(object sender, EventArgs e)
        {
            FrmRegistroLista frm = new FrmRegistroLista();
            FrmRegistroLista_Controlador controlador = new
FrmRegistroLista_Controlador(frm, new Lista(),1,0);
            AbrirHijo(frm);
        }

        private void AbrirHijo(object fornHijo)
        {
            if (this.vista.panel3.Controls.Count > 0)
            {
                this.vista.panel3.Controls.RemoveAt(0);
            }
            Form fh = fornHijo as Form;
            fh.TopLevel = false;
            fh.Dock = DockStyle.Fill;
            this.vista.panel3.Controls.Add(fh);
            this.vista.panel3.Tag = fh;

            fh.Show();
        }



        private void BtnConsulta_Click(object sender, EventArgs e)
        {
            FrmConsulta frm = new FrmConsulta();
            FrmConsulta_Controlador controlador = new
FrmConsulta_Controlador(frm,new Lista(), new Estudiante(),vista);
            AbrirHijo(frm);
        }

        private void BtnResultados_Click(object sender, EventArgs e)
        {
            FrmResultado frm = new FrmResultado();
            FrmResultado_Controlador controlador = new
FrmResultado_Controlador(frm, new Lista());
            AbrirHijo(frm);
        }

        private void BtnListas_Click(object sender, EventArgs e)
        {
            Estudiante est = (Estudiante)usuario;
            if (Lista_con.ExisteVotacionPorEstudiante(est.IdEstudiante))
            {
                MessageBox.Show("Usted ya realizó su votación");
                return;
            }

            if (FrmPrincipal_Controlador.votar == 0)
            {
                MessageBox.Show("Las votaciones estan deshabilitadas");
                return;
            }
            FrmVotarLista frm = new FrmVotarLista();
            FrmVotarLista_Controlador controlador = new
FrmVotarLista_Controlador(frm, new Lista(), est.IdEstudiante);
            AbrirHijo(frm);
        }
    }
}
```

i. Registro estudiante

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmRegistroEstudiante_Controlador
    {
        private FrmRegistroEstudiante vista;
        private Estudiante estudiante;
        private int op;
        private int id;


        public FrmRegistroEstudiante_Controlador(FrmRegistroEstudiante
vista, Estudiante estudiante, int op, int id)
        {
            this.vista = vista;
            this.estudiante = estudiante;
            this.vista.btnAgregar.Click += BtnAgregar_Click;
            this.vista.txtApellidos.KeyPress += new
KeyPressEventHandler(txtLetras_KeyPress);
            this.vista.txtNombre.KeyPress += new
KeyPressEventHandler(txtLetras_KeyPress);
            this.vista.txtCedula.KeyPress += new
KeyPressEventHandler(txtNumeros_KeyPress);
            this.vista.txtCelular.KeyPress += new
KeyPressEventHandler(txtNumeros_KeyPress);
            this.op = op;
            this.id = id;
            CargarSemestres();
            CargarDatos();
            this.vista.ShowDialog();
        }

        private void txtLetras_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar)
&& e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }
        }

        private void txtNumeros_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!char.IsDigit(e.KeyChar) && e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }
        }

        private void CargarDatos()
        {
```

```
            if (op == 2)
            {
                Estudiante e = Estudiante_con.ObtenerEstudiantePorId(id);
                vista.txtApellidos.Text = e.Apellidos;
                vista.txtCedula.Text = e.Cedula;
                vista.txtCelular.Text = e.Celular;
                vista.txtContrasenia.Text = e.Contrasenia;
                vista.txtCorreo.Text = e.Correo;
                vista.txtNombre.Text = e.Nombre;


                for (int i = 0; i < vista.cbSemestre.Items.Count; i++)
                {
                    Semestre s = (Semestre)vista.cbSemestre.Items[i];
                    if (s.IdSemestre == e.IdSemestre)
                    {
                        vista.cbSemestre.SelectedIndex = i;
                        break;
                    }
                }
            }
        }

        private void CargarSemestres()
        {
            vista.cbSemestre.Items.Clear();
            List<Semestre> semestres = Semestre_con.Listar();
            for (int i = 0; i < semestres.Count; i++)
            {
                vista.cbSemestre.Items.Add(semestres[i]);
            }
        }

        private void BtnAgregar_Click(object sender, EventArgs e)
        {
            string msj = validarCamposVacios();
            if (op == 1)
            {
                if (msj.Equals(""))
                {
                    estudiante = new Estudiante();
                    Semestre s = null;

                    estudiante.Apellidos = vista.txtApellidos.Text;
                    estudiante.Cedula = vista.txtCedula.Text;
                    estudiante.Celular = vista.txtCelular.Text;
                    estudiante.Contrasenia = vista.txtContrasenia.Text;
                    estudiante.Correo = vista.txtCorreo.Text;
                    estudiante.Nombre = vista.txtNombre.Text;
                    if (vista.cbSemestre.SelectedIndex >= 0)
                    {
                        s = (Semestre)vista.cbSemestre.SelectedItem;
                        estudiante.IdSemestre = s.IdSemestre;
                    }


                    int idEstudiante = Estudiante_con.Insertar(estudiante);
                    if (idEstudiante != -1)
                    {

                        MessageBox.Show("Se registró exitosamente",
"Registrar Estudiante");
                        vista.Dispose();
```

```
                    }
                    else
                    {
                        MessageBox.Show("Error al intentar registrarse",
"Registrar Estudiante");
                    }

                }
                else
                {
                    MessageBox.Show(msj, "Registrar Estudiante");
                }
            }
            else//editar
            {
                if (msj.Equals(""))
                {
                    estudiante = new Estudiante();
                    Semestre s = null;
                    estudiante.IdEstudiante = id;
                    estudiante.Apellidos = vista.txtApellidos.Text;
                    estudiante.Cedula = vista.txtCedula.Text;
                    estudiante.Celular = vista.txtCelular.Text;
                    estudiante.Contrasenia = vista.txtContrasenia.Text;
                    estudiante.Correo = vista.txtCorreo.Text;
                    estudiante.Nombre = vista.txtNombre.Text;
                    if (vista.cbSemestre.SelectedIndex >= 0)
                    {
                        s = (Semestre)vista.cbSemestre.SelectedItem;
                        estudiante.IdSemestre = s.IdSemestre;
                    }


                    bool estado = Estudiante_con.Actualizar(estudiante);
                    if (estado)
                    {

                        MessageBox.Show("Se actualizó el estudiante",
"Actualizar Estudiante");
                        vista.Dispose();



                    }
                    else
                    {
                        MessageBox.Show("Error al intentar actualizar",
"Actualizar Estudiante");
                    }

                }
                else
                {
                    MessageBox.Show(msj, "Actualizar Estudiante");
                }

            }

        }



        private string validarCamposVacios()
```

```
        {
            string msj = "";
            if (vista.txtApellidos.Text.Equals("")) msj += "Falta ingresar
apellidos\n";
            if (vista.txtCedula.Text.Equals("")) msj += "Falta ingresar
cédula\n";
            if (vista.txtContrasenia.Text.Equals("")) msj += "Falta ingresar
Contraseña\n";
            if (vista.txtCedula.Text.Length != 10) msj += "La cédula debe
tener 10 dígitos\n";
            if (vista.txtCelular.Text.Length != 10) msj += "El teléfono debe
tener 10 dígitos\n";
            if (! new Regex(@"^(09|120)\d*$").IsMatch(vista.txtCedula.Text))
msj += "La cédula debe empezar con 09 o 120\n";
            if (!new
Regex(@"^(096|097|098|099)\d*$").IsMatch(vista.txtCelular.Text)) msj += "El
teléfono debe empezar con 096, 097, 098 o 099\n";
            if (!new Regex(@"^[a-zA-Z0-9._%+-
]+@ug\.edu\.ec$").IsMatch(vista.txtCorreo.Text)) msj += "El correo es
inválido\n";
            if (vista.txtCorreo.Text.Equals("")) msj += "Falta ingresar
correo\n";
            if (vista.txtNombre.Text.Equals("")) msj += "Falta ingresar
nombre\n";
            if (vista.cbSemestre.SelectedIndex < 0) msj += "Falta
seleccionar semestre\n";
            return msj;
        }
    }
}
```

j. Registro lista

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmRegistroLista_Controlador
    {
        private FrmRegistroLista vista;
        private Lista lista;
        private string nombreUnico;
        private int opcion;
        private int id;

        /*public FrmRegistroLista_Controlador(int opcion,int id)
        {
            nombreUnico = "";
            CargarCarreras();
            this.opcion = opcion;
            this.id = id;
            CargarDatos();
        }*/
```

```
        public FrmRegistroLista_Controlador(FrmRegistroLista vista, Lista
lista, int opcion,int id)
        {
            this.vista = vista;
            this.lista = lista;
            this.nombreUnico = "";
            CargarCarreras();
            this.vista.btnAgregar.Click += BtnAgregar_Click;
            this.vista.btnAgregarFoto.Click += BtnAgregarFoto_Click;
            this.vista.btnPro.Click += BtnAgregarPropuesta_Click;
            this.vista.btnQuitar.Click += BtnQuitar_Click;
            this.vista.btnQuitarFoto.Click += BtnQuitarFoto_Click;
            this.vista.txtNomLista.KeyPress += new
KeyPressEventHandler(txtNomLista_KeyPress);
            this.vista.txtNomPresidente.KeyPress += new
KeyPressEventHandler(txtNomLista_KeyPress);
            this.vista.txtNomSecretario.KeyPress += new
KeyPressEventHandler(txtNomLista_KeyPress);
            this.vista.txtNomVocal.KeyPress += new
KeyPressEventHandler(txtNomLista_KeyPress);
            this.opcion = opcion;
            this.id = id;

            CargarDatos();
        }

        private void CargarDatos()
        {
            if (opcion == 2)
            {
                this.lista = Lista_con.ObtenerListaPorId(id);
                List<Propuesta> propuestas =
Propuesta_con.ObtenerPropuestasPorIdLista(id);

                this.vista.txtNomLista.Text = lista.NomLista;
                vista.txtNomPresidente.Text = lista.NomPresidente;
                vista.txtNomSecretario.Text = lista.NomSecretario;
                vista.txtNomVocal.Text = lista.NomPrimerVocal;
                for(int i=0; i<vista.cbCarrera.Items.Count; i++)
                {
                    Carrera c = (Carrera)vista.cbCarrera.Items[i];
                    if (c.IdCarrera == lista.Carrera)
                    {
                        vista.cbCarrera.SelectedIndex = i;
                        break;
                    }

                }

                vista.listPropuestas.Items.Clear();
                for(int i=0; i<propuestas.Count; i++)
                {

this.vista.listPropuestas.Items.Add(propuestas[i].Descripcion);
                }

                vista.txtDescripcion.Text = lista.Descripcion;
                if (!string.IsNullOrEmpty(lista.FotoP))
                {
                    string rutaImagen =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "foto_lista",
lista.FotoP);
```

```
                    if (File.Exists(rutaImagen))
                    {
                        vista.pbImagen.Image = Image.FromFile(rutaImagen);
                    }
                }

            }
        }

        private void txtNomLista_KeyPress(object sender, KeyPressEventArgs
e)
        {
            if (!char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar)
&& e.KeyChar != (char)Keys.Back)
            {
                e.Handled = true;
            }
        }


        private void BtnQuitarFoto_Click(object sender, EventArgs e)
        {
            vista.pbImagen.Image = null;
            nombreUnico = "";
        }

        private void BtnQuitar_Click(object sender, EventArgs e)
        {
            if (vista.listPropuestas.SelectedIndex != -1)
            {
                int indiceSeleccionado = vista.listPropuestas.SelectedIndex;
                vista.listPropuestas.Items.RemoveAt(indiceSeleccionado);
            }
            else
            {
                MessageBox.Show("Por favor, seleccione un fila de la lista
de propuestas para eliminar.", "Eliminar Propuesta");
            }
        }

        private void BtnAgregarPropuesta_Click(object sender, EventArgs e)
        {
            string propuesta = vista.txtPropuesta.Text;
            if (propuesta.Equals(""))
            {
                MessageBox.Show("Ingrese una propuesta");
            }
            else
            {
                vista.listPropuestas.Items.Add(propuesta);
                vista.txtPropuesta.Text = "";
            }
        }

        private void BtnAgregar_Click(object sender, EventArgs e)
        {
            if (opcion == 1)
            {
                string msj = validarCamposVacios();
                if (msj.Equals(""))
                {
                    Carrera c = null;
                    Propuesta p = new Propuesta();
```

```
                    lista = new Lista();

                    lista.NomLista = vista.txtNomLista.Text;
                    lista.NomPresidente = vista.txtNomPresidente.Text;
                    lista.NomSecretario = vista.txtNomSecretario.Text;
                    lista.NomPrimerVocal = vista.txtNomVocal.Text;
                    lista.Descripcion = vista.txtDescripcion.Text;
                    if (vista.cbCarrera.SelectedIndex >= 0)
                    {
                        c = (Carrera)vista.cbCarrera.SelectedItem;
                        lista.Carrera = c.IdCarrera;
                    }
                    lista.FotoP = nombreUnico;
                    lista.CantVotos = 0;


                    int idLista = Lista_con.Insertar(lista);
                    if (idLista != -1)
                    {
                        foreach (object item in vista.listPropuestas.Items)
                        {
                            p = new Propuesta();
                            p.IdLista = idLista;
                            p.Descripcion = item.ToString();
                            Propuesta_con.Insertar(p);
                        }

                        MessageBox.Show("Se registró nueva lista",
"Registrar Lista");
                        Limpiar();
                    }
                    else
                    {
                        MessageBox.Show("Error al intentar registrar nueva
lista", "Registrar Lista");
                    }

                }
                else
                {
                    MessageBox.Show(msj, "Registrar Lista");
                }
            }
            else//modificar
            {
                string msj = validarCamposVacios();
                if (msj.Equals(""))
                {
                    Carrera c = null;
                    Propuesta p = new Propuesta();
                    if (nombreUnico.Equals("")) nombreUnico = lista.FotoP;
                    lista = new Lista();
                    lista.IdLista = id;
                    lista.NomLista = vista.txtNomLista.Text;
                    lista.NomPresidente = vista.txtNomPresidente.Text;
                    lista.NomSecretario = vista.txtNomSecretario.Text;
                    lista.NomPrimerVocal = vista.txtNomVocal.Text;
                    lista.Descripcion = vista.txtDescripcion.Text;
                    if (vista.cbCarrera.SelectedIndex >= 0)
                    {
                        c = (Carrera)vista.cbCarrera.SelectedItem;
                        lista.Carrera = c.IdCarrera;
                    }
```

```
                    lista.FotoP = nombreUnico;


                    bool estado = Lista_con.Actualizar(lista);
                    if (estado)
                    {
                        Propuesta_con.EliminarPropuestasPorIdLista(id);
                        foreach (object item in vista.listPropuestas.Items)
                        {
                            p = new Propuesta();
                            p.IdLista = id;
                            p.Descripcion = item.ToString();
                            Propuesta_con.Insertar(p);
                        }

                        MessageBox.Show("Se actualizó la lista", "Actualizar
Lista");
                    }
                    else
                    {
                        MessageBox.Show("Error al intentar actualizar la
lista", "Actualizar Lista");
                    }
                }
                else
                {
                    MessageBox.Show(msj, "Actualizar Lista");
                }
            }
        }

        private void BtnAgregarFoto_Click(object sender, EventArgs e)
        {
            try
            {
                vista.ofd_foto.Filter = "Archivos de
imagen|*.jpg;*.jpeg;*.png;*.gif;*.bmp|Todos los archivos|*.*";
                DialogResult result = vista.ofd_foto.ShowDialog();
                if (result == DialogResult.OK)
                {
                    string rutaImagenOriginal = vista.ofd_foto.FileName;
                    string nombreArchivoOriginal =
Path.GetFileName(rutaImagenOriginal);
                    nombreUnico = DateTime.Now.ToString("yyyyMMddHHmmss") +
".png";
                    string rutaDestino =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "foto_lista",
nombreUnico);
                    string directorioDestino =
Path.GetDirectoryName(rutaDestino);
                    if (!Directory.Exists(directorioDestino))
                    {
                        Directory.CreateDirectory(directorioDestino);
                    }

                    File.Copy(rutaImagenOriginal, rutaDestino, true);

                    vista.pbImagen.Image = Image.FromFile(rutaDestino);
                }
            }
            catch(OutOfMemoryException ex)
            {
```

```
                MessageBox.Show("Memoria insuficiente para cargar la imagen.
Por favor, elija una imagen más pequeña.", "Error de memoria");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error");
            }
        }

        private void CargarCarreras()
        {
            vista.cbCarrera.Items.Clear();
            List<Carrera> carreras = Carrera_con.Listar();
            for(int i=0; i<carreras.Count; i++)
            {
                vista.cbCarrera.Items.Add(carreras[i]);
            }
        }

        private string validarCamposVacios()
        {
            string msj = "";
            if (vista.txtNomLista.Text.Equals("")) msj += "Falta ingresar
nombre de lista\n";
            if (vista.txtNomPresidente.Text.Equals("")) msj += "Falta
ingresar nombre del presidente\n";
            if (vista.txtNomSecretario.Text.Equals("")) msj += "Falta
ingresar nombre del secretario\n";
            if (vista.txtNomVocal.Text.Equals("")) msj += "Falta ingresar
nombre del vocal\n";
            if (nombreUnico.Equals("")) msj += "Falta ingresar foto\n";
            if (vista.cbCarrera.SelectedIndex<0) msj += "Falta seleccionar
carrera\n";
            return msj;
        }

        private void Limpiar()
        {
            vista.listPropuestas.Items.Clear();
            vista.pbImagen.Image = null;
            vista.txtNomLista.Text = "";
            vista.txtNomPresidente.Text = "";
            vista.txtNomSecretario.Text = "";
            vista.txtNomVocal.Text = "";
            vista.txtDescripcion.Text = "";
            nombreUnico = "";
            vista.pbImagen.Image = null;
            if (vista.cbCarrera.SelectedIndex >= 0)
vista.cbCarrera.SelectedIndex = 0;
        }
    }
}
```

k. Resultado controlador

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmResultado_Controlador
    {
        private FrmResultado vista;
        private Lista lista;

        public FrmResultado_Controlador(FrmResultado vista, Lista lista)
        {
            this.vista = vista;
            this.lista = lista;
            CargarLista();

        }

        private void CargarLista()
        {
            // Crear un Panel con AutoScroll para contener el
TableLayoutPanel
            Panel panelScroll = new Panel
            {
                AutoScroll = true, // Habilitar el scroll automático
                Dock = DockStyle.Fill, // Hacer que el panel ocupe todo el
espacio disponible
            };

            vista.tableLayoutPanel1.AutoSize = true;
            vista.tableLayoutPanel1.AutoSizeMode =
AutoSizeMode.GrowAndShrink;
            vista.tableLayoutPanel1.ColumnCount = 4;

            List<Lista> listas = Lista_con.Listar();
            int total = 0;
            foreach (Lista c in listas)
            {
                total += c.CantVotos;
            }

            foreach (Lista c in listas)
            {
                Panel panelImagen = new Panel
                {
                    BorderStyle = BorderStyle.Fixed3D,
                    Margin = new Padding(15),
                    Size = new Size(200, 300),
                    BackColor = Color.FromArgb(25, 61, 111)
                };

                Label label1 = new Label
                {
                    Text = c.NomLista,
                    Size = new Size(180, 30),
                    Location = new Point(10, 10),
                    ForeColor = Color.White,
                    Font = new Font("Arial", 12, FontStyle.Regular),
                    TextAlign = ContentAlignment.MiddleCenter
                };

                PictureBox pictureBox = new PictureBox
```

```
                {
                    Image =
Image.FromFile(Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"foto_lista", c.FotoP)),
                    SizeMode = PictureBoxSizeMode.StretchImage,
                    Location = new Point(10, 50),
                    Size = new Size(180, 200),
                };

                Label label2 = new Label
                {
                    Text = "Puntuación: " + Math.Round(((c.CantVotos *
100.0) / total), 2) + " %",
                    Size = new Size(180, 30),
                    Location = new Point(10, 260),
                    ForeColor = Color.White,
                    Font = new Font("Arial", 12, FontStyle.Regular),
                    TextAlign = ContentAlignment.MiddleCenter
                };

                panelImagen.Controls.Add(label1);
                panelImagen.Controls.Add(pictureBox);
                panelImagen.Controls.Add(label2);

                vista.tableLayoutPanel1.Controls.Add(panelImagen);
            }

            // Añadir el TableLayoutPanel al Panel con scroll
            panelScroll.Controls.Add(vista.tableLayoutPanel1);

            // Añadir el Panel con scroll al formulario principal o al
contenedor adecuado
            vista.Controls.Add(panelScroll);
        }


    }
}
```

l. Visualizar lista

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.Utilidad;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmVisualizarLista_Controlador
    {
        private FrmVisualizarLista vista;
        private Lista lista;

        public FrmVisualizarLista_Controlador(FrmVisualizarLista vista,
Lista lista)
        {
            this.vista = vista;
```

```
            this.lista = lista;
            ListarPorcentaje();
            ListarEstudiantes_votaron();
            EstiloTabla.EstilizarDataGridView(this.vista.dgvEstudiantes);
            EstiloTabla.EstilizarDataGridView(this.vista.dgvPorcentaje);
        }

        public void ListarPorcentaje()
        {
            List<Lista> listas = Lista_con.Listar();
            int totalVotos = 0;
            for(int i = 0; i < listas.Count; i++)
            {
                totalVotos += listas[i].CantVotos;
            }

            for (int i = 0; i < listas.Count; i++)
            {
                double porcentaje = 0;
                if (totalVotos!=0) porcentaje= (listas[i].CantVotos * 100.0)
/ totalVotos;
                this.vista.dgvPorcentaje.Rows.Add(new string[] {
listas[i].NomLista, porcentaje+"%" });
            }
        }

        public void ListarEstudiantes_votaron()
        {
            List<Estudiante> estudiantes =
Estudiante_con.Estudiantes_votaron();

            vista.dgvEstudiantes.Rows.Clear();
            for (int i = 0; i < estudiantes.Count; i++)
            {
                vista.dgvEstudiantes.Rows.Add(new object[]
{estudiantes[i].IdEstudiante,estudiantes[i].Nombre,
estudiantes[i].Apellidos,
                estudiantes[i].Cedula, estudiantes[i].Correo,
estudiantes[i].Celular, estudiantes[i].NomSemestre});
            }

        }
    }
}
```

m. Votar lista

```
using SistemaVotacion.conectar;
using SistemaVotacion.modelo;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.controlador
{
    public class FrmVotarLista_Controlador
    {
```

```
        private FrmVotarLista vista;
        private Lista lista;
        private int idEstudiante;
```

3. Clases modelo

a. Administrador

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

namespace SistemaVotacion.modelo
{
    public class Administrador
    {
        private int idUsuario;
        private string nombre;
        private string apellidos;
        private string correo;
        private string contrasenia;

        public Administrador()
        {
        }

        public Administrador(string correo, string contrasenia)
        {
            this.correo = correo;
            this.contrasenia = contrasenia;
        }

        public int IdUsuario { get => idUsuario; set => idUsuario = value; }
        public string Nombre { get => nombre; set => nombre = value; }
        public string Apellidos { get => apellidos; set => apellidos =
value; }
        public string Correo { get => correo; set => correo = value; }
        public string Contrasenia { get => contrasenia; set => contrasenia =
value; }
    }
}
```

b. Carrera

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

namespace SistemaVotacion.modelo
{
    public class Carrera
    {
        private int idCarrera;
        private string nomCarrera;

        public int IdCarrera { get => idCarrera; set => idCarrera = value; }
        public string NomCarrera { get => nomCarrera; set => nomCarrera =
value; }
```

```
        public override string ToString()
        {
            return NomCarrera;
        }
    }
}
```

c. Estudiante

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

namespace SistemaVotacion.modelo
{
    public class Estudiante
    {
        private int idEstudiante;
        private string nombre;
        private string apellidos;
        private string correo;
        private string cedula;
        private string celular;
        private int idSemestre;
        private string nomSemestre;
        private string contrasenia;

        public Estudiante()
        {
        }

        public Estudiante(string correo, string contrasenia)
        {
            this.correo = correo;
            this.contrasenia = contrasenia;
        }

        public int IdEstudiante { get => idEstudiante; set => idEstudiante =
value; }
        public string Nombre { get => nombre; set => nombre = value; }
        public string Apellidos { get => apellidos; set => apellidos =
value; }
        public string Correo { get => correo; set => correo = value; }
        public string Cedula { get => cedula; set => cedula = value; }
        public string Celular { get => celular; set => celular = value; }
        public int IdSemestre { get => idSemestre; set => idSemestre =
value; }
        public string NomSemestre { get => nomSemestre; set => nomSemestre =
value; }
        public string Contrasenia { get => contrasenia; set => contrasenia =
value; }
    }
}
```

d. Invitado

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

```
namespace SistemaVotacion.modelo
{
    internal class Invitado
    {
    }
}
```

e. Lista

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

namespace SistemaVotacion.modelo
{
    public class Lista
    {
        private int idLista;
        private string nomLista;
        private string nomPresidente;
        private string nomSecretario;
        private string nomPrimerVocal;
        private int carrera;
        private string nomCarrera;
        private string fotoP;
        private string descripcion;
        private int cantVotos;

        public int IdLista { get => idLista; set => idLista = value; }
        public string NomLista { get => nomLista; set => nomLista = value; }
        public string NomPresidente { get => nomPresidente; set =>
nomPresidente = value; }
        public string NomSecretario { get => nomSecretario; set =>
nomSecretario = value; }
        public string NomPrimerVocal { get => nomPrimerVocal; set =>
nomPrimerVocal = value; }
        public int Carrera { get => carrera; set => carrera = value; }
        public string FotoP { get => fotoP; set => fotoP = value; }
        public string Descripcion { get => descripcion; set => descripcion =
value; }
        public string NomCarrera { get => nomCarrera; set => nomCarrera =
value; }
        public int CantVotos { get => cantVotos; set => cantVotos = value; }
    }
}
```

f. Propuesta

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

namespace SistemaVotacion.modelo
{
    public class Propuesta
    {
        private int idPropuesta;
        private int idLista;
```

```
        private string descripcion;

        public int IdPropuesta { get => idPropuesta; set => idPropuesta =
value; }
        public int IdLista { get => idLista; set => idLista = value; }
        public string Descripcion { get => descripcion; set => descripcion =
value; }
    }
}
```

g. Semestre

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

namespace SistemaVotacion.modelo
{
    public class Semestre
    {
        private int idSemestre;
        private string nomSemestre;

        public int IdSemestre { get => idSemestre; set => idSemestre =
value; }
        public string NomSemestre { get => nomSemestre; set => nomSemestre =
value; }

        public override string ToString()
        {
            return NomSemestre;
        }
    }
}
```

h. Votación

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

namespace SistemaVotacion.modelo
{
    public class Votacion
    {
        private int idVotacion;
        private int idCandidata;
        private string matricula;
        private int tipo;

        public int IdVotacion { get => idVotacion; set => idVotacion =
value; }
        public int IdCandidata { get => idCandidata; set => idCandidata =
value; }
        public string Matricula { get => matricula; set => matricula =
value; }
        public int Tipo { get => tipo; set => tipo = value; }
    }
}
```

4. Clases de utilidad

a. Archivo

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

namespace SistemaVotacion.Utilidad
{
    public class Archivo
    {
        private string rutaArchivo;

        public Archivo(string ruta)
        {
            rutaArchivo = ruta;
        }

        public bool Guardar(string contenido)
        {
            try
            {
                File.WriteAllText(rutaArchivo, contenido);
                return true;
            }
            catch (Exception)
            {
            }
            return false;
        }

        public string Obtener()
        {
            try
            {
                if (!File.Exists(rutaArchivo))
                {
                    File.Create(rutaArchivo).Close();
                }

                string contenido = File.ReadAllText(rutaArchivo);
                return contenido;
            }
            catch (Exception)
            {
                return string.Empty;
            }
        }


    }
}
```

b. Estilo de tabla

```
using System;
using System.Collections.Generic;
using System.Drawing;
```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion.Utilidad
{
    public class EstiloTabla
    {
        public static void EstilizarDataGridView(DataGridView dgv)
        {
            // Estilo de la cabecera de columna
            DataGridViewCellStyle columnHeaderStyle = new
DataGridViewCellStyle();
            columnHeaderStyle.BackColor = Color.Black; // Color de fondo
negro
            columnHeaderStyle.ForeColor = Color.White; // Color del texto
blanco
            columnHeaderStyle.Font = new Font("Verdana", 10,
FontStyle.Bold);
            columnHeaderStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            dgv.ColumnHeadersDefaultCellStyle = columnHeaderStyle;

            // Estilo de la cabecera de fila
            DataGridViewCellStyle rowHeaderStyle = new
DataGridViewCellStyle();
            rowHeaderStyle.BackColor = Color.Navy;
            rowHeaderStyle.ForeColor = Color.White;
            rowHeaderStyle.Font = new Font("Verdana", 10, FontStyle.Bold);
            rowHeaderStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
            dgv.RowHeadersDefaultCellStyle = rowHeaderStyle;

            // Estilo de las celdas
            DataGridViewCellStyle cellStyle = new DataGridViewCellStyle();
            cellStyle.BackColor = Color.LightGray;
            cellStyle.ForeColor = Color.Black;
            cellStyle.Font = new Font("Verdana", 9, FontStyle.Regular);
            cellStyle.SelectionBackColor = Color.DarkSlateBlue;
            cellStyle.SelectionForeColor = Color.White;
            dgv.DefaultCellStyle = cellStyle;

            // Alternar color de fila
            dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.White;

            // Color de fondo del área vacía
            dgv.BackgroundColor = Color.LightBlue;

            // Otras configuraciones del DataGridView
            dgv.EnableHeadersVisualStyles = false; // Deshabilitar estilos
visuales predeterminados
            dgv.BorderStyle = BorderStyle.Fixed3D;
            dgv.GridColor = Color.Black;
            dgv.RowHeadersWidthSizeMode =
DataGridViewRowHeadersWidthSizeMode.DisableResizing;
            dgv.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.DisableResizing;
            dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
// Ajustar columnas para llenar el control
            dgv.SelectionMode = DataGridViewSelectionMode.FullRowSelect; //
Selección de fila completa
```

```
            dgv.AllowUserToResizeColumns = false; // Deshabilitar
redimensionamiento de columnas
            dgv.AllowUserToResizeRows = false; // Deshabilitar
redimensionamiento de filas

            // Permitir scroll en el DataGridView
            dgv.ScrollBars = ScrollBars.Both; // Habilitar barras de scroll
vertical y horizontal

            // Configurar el ajuste de las columnas y filas
            dgv.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.DisplayedCells; // Ajustar columnas a las
celdas mostradas
            dgv.AutoSizeRowsMode =
DataGridViewAutoSizeRowsMode.DisplayedCells; // Ajustar filas a las celdas
mostradas

            // Otras configuraciones
            dgv.AllowUserToResizeColumns = true; // Permitir
redimensionamiento de columnas
            dgv.AllowUserToResizeRows = true; // Permitir redimensionamiento
de filas
            dgv.SelectionMode = DataGridViewSelectionMode.FullRowSelect; //
Selección de fila completa
        }
    }
}
```

5. Código de entrada principal a la aplicación

```
using SistemaVotacion.controlador;
using SistemaVotacion.modelo;
using SistemaVotacion.vista;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SistemaVotacion
{
    static class Program
    {
        /// <summary>
        /// Punto de entrada principal para la aplicación.
        /// </summary>
        [STAThread]
        static void Main()
        {
            FrmLogin form = new FrmLogin();
            FrmLogin_Controlador c = new FrmLogin_Controlador(form);
            c.Iniciar();

        }
    }
}
```