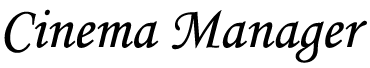
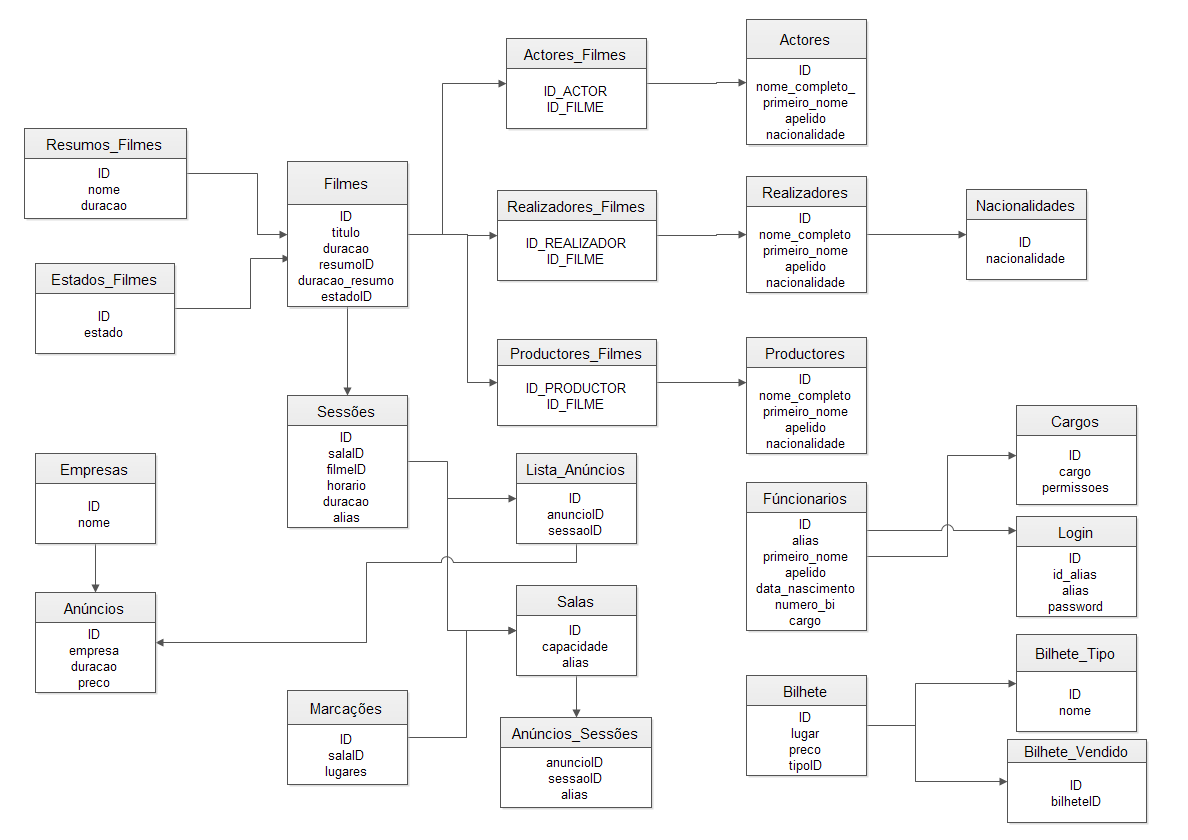
**Gestor de Cinema**

****

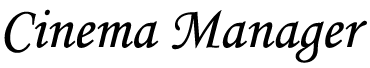
**Lisboa, 05 de Abril de 2014**

**Índice**

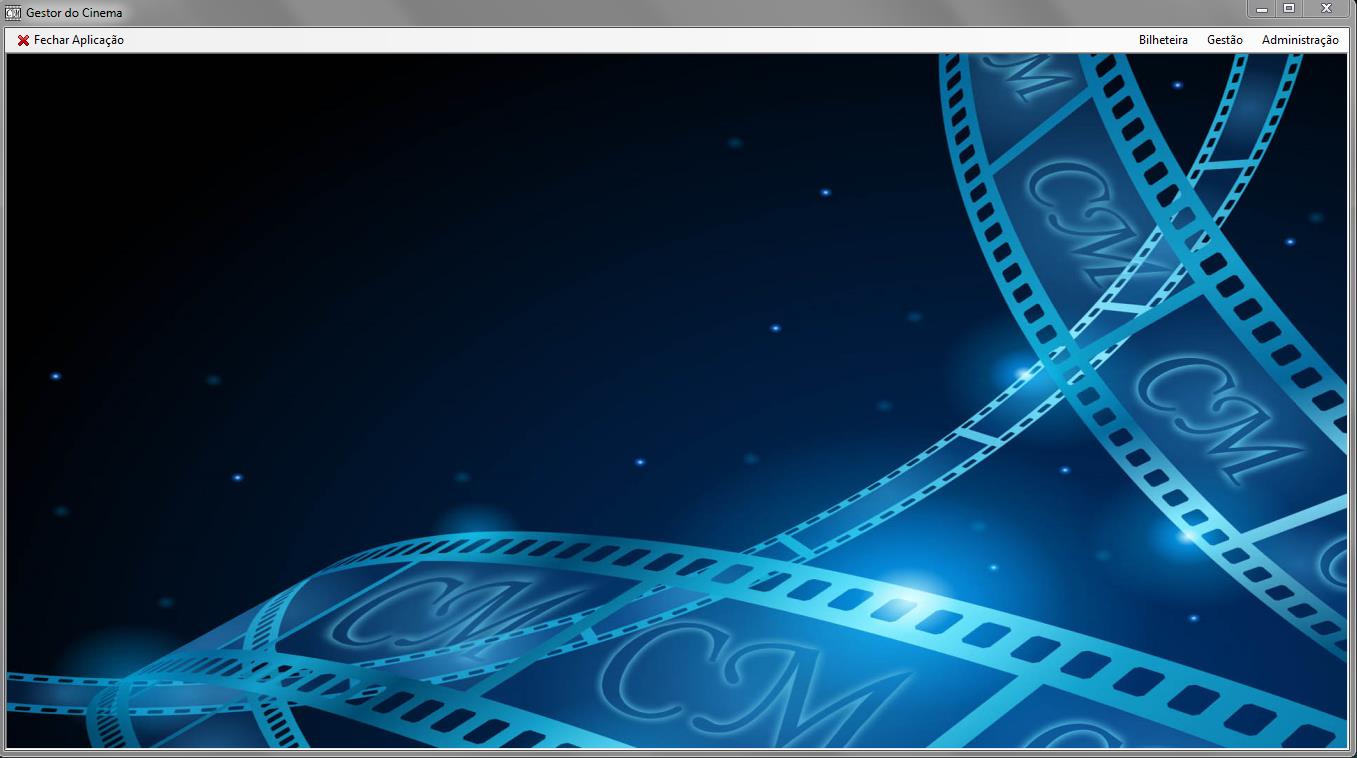
|  |  |
| --- | --- |
| Lista de Figuras | p.3 |
| Lista de Programas | p.8 |
| Introdução | p.9 |
| Fundamentação Técnica-Científica | p.10 |
| Descrição do Produto | p.11 |
| Condicionalismos | p.12 |
| Bibliografia | p.13 |
| Anexos | p.14 |

**Lista de Figuras**

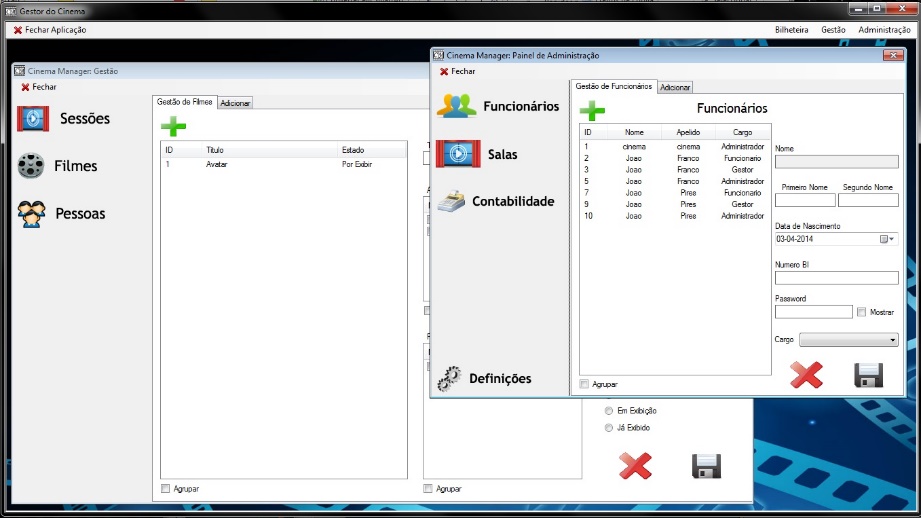
1. Base de dados

****

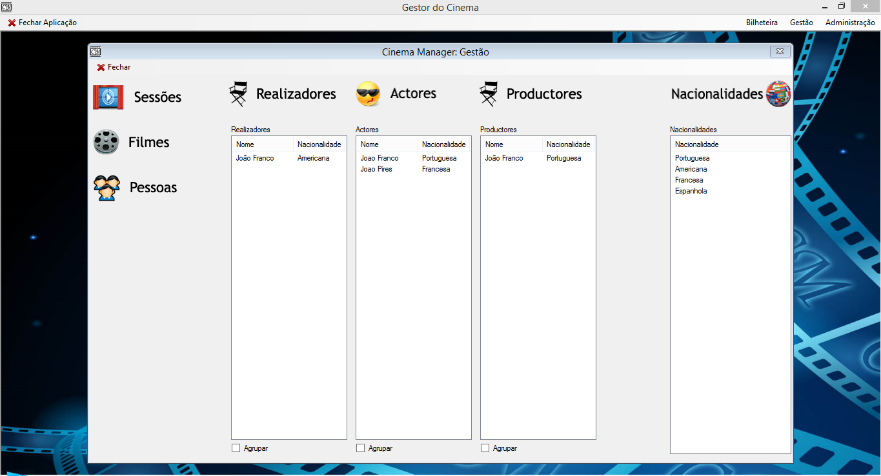
c) Logótipo.

****

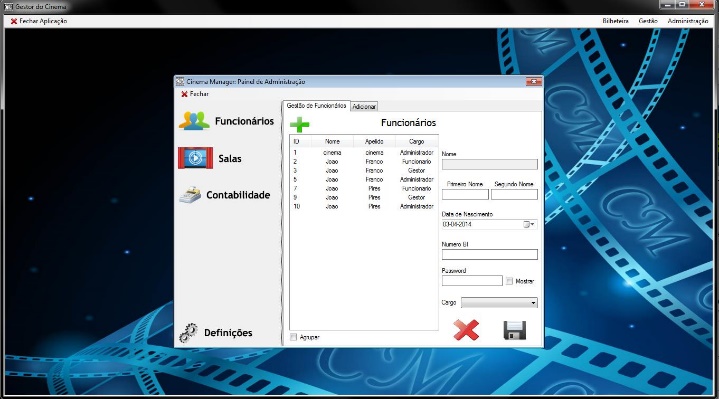
g) Form principal da aplicação.

****

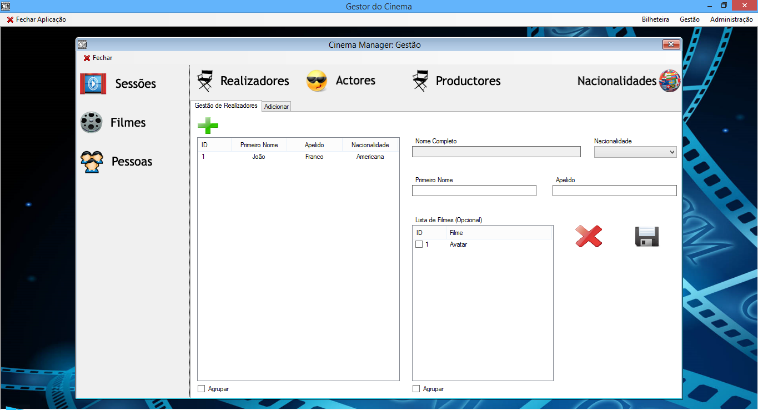
e) Form para adicionar funcionários.

****

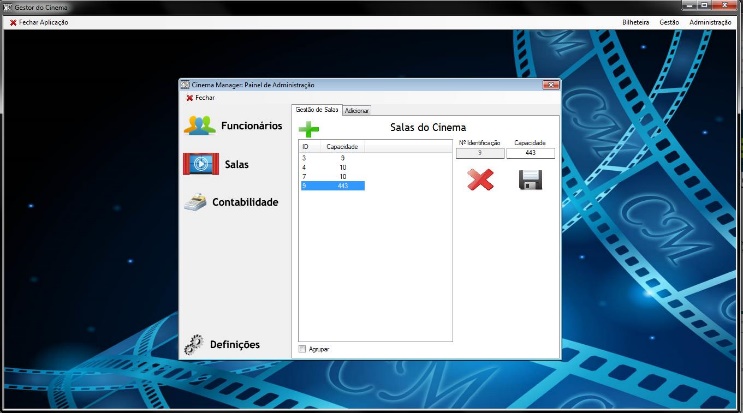
f) Listagem de Intervenientes e Nacionalidades

****

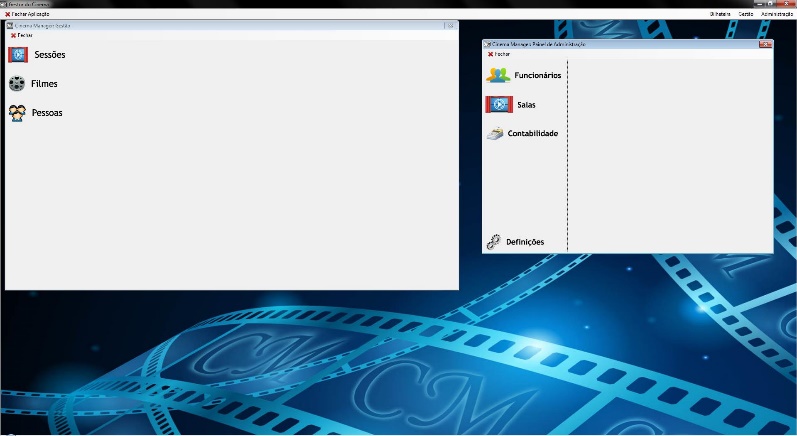
h) Edição de funcionários.

****

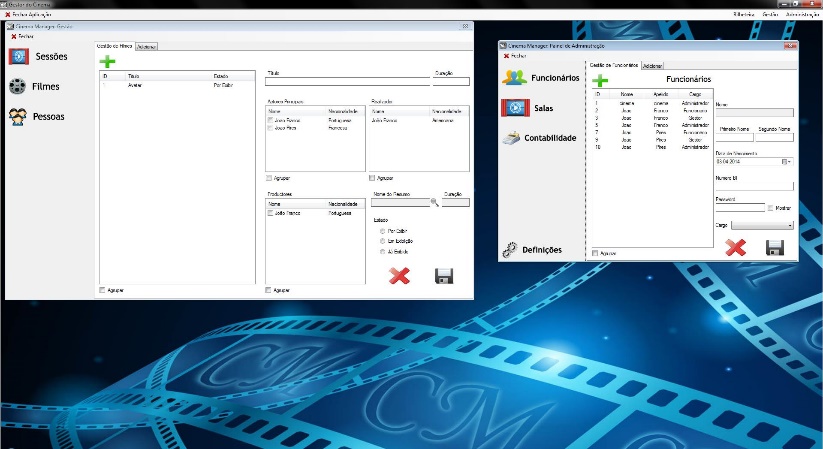
i) Form para gestão de Realizadores.

****

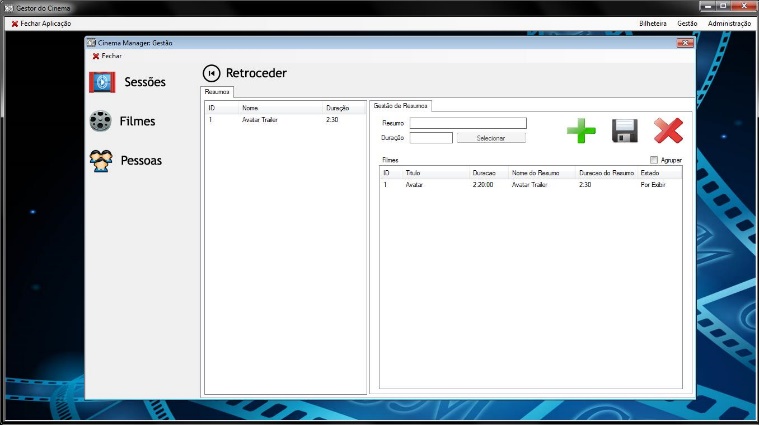
j) Gestão das Salas de Cinema.

****

l) Gestão do painel e Administração e do painel de gestão.

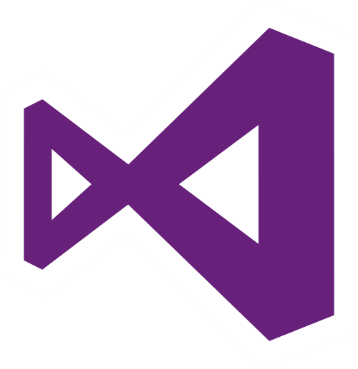
****

m) Gestão dos filmes.

****

n) Form para adicionar resumos.

**Lista de Programas**



**Microsoft Visual Studio 2013**  
O Visual Studio é um programa que abrange inúmeras ferramentas e serviços que ajudam a criar uma ampla variedade de aplicativos para a plataforma Microsoft e além. O Visual Studio tem a possibilidade também de se conectar a todos os seus projetos, equipa e participantes.



**HeidiSQL**

HeidiSQL, é um programa livre e cliente código-aberto para o MySQL.  
O programa permite que administradores conectem-se com múltiplos servidores de base de dados, criem e editem tabelas entre outras funcionalidades em SQL.

**Introdução**

Esta aplicação permite fazer a gestão das salas de cinema que automatize as suas atividades, que faça a gestão da lotação, marcação e a nível financeiro dos bilhetes já vendidos segundo o tipo de bilhete vendido.   
A gestão da publicidade antes do filme que é mostrada aos espetadores, também para a gestão dos filmes que já foram exibidos no cinema e os que futuramente também irão ser exibidos.

A ideia de desenvolver esta aplicação deu-se após dado o tema pelo professor e para uma melhor gestão de um cinema rapidamente concluímos que a melhor solução seria esta aplicação.

Após uma vasta pesquisa na Internet sobre a oferta de aplicações relacionadas com esta temática, quer em regime ‘shareware’, quer em comerciais, definiu-se que o ‘layout’ ideal seria este desenhado.

Os objetivos principais da aplicação são:

* Gestão das salas de cinema (Lotação/Marcação);
* Gestão de contabilidade dos bilhetes vendidos;
* Gestão da publicidade rodada;
* Gestão dos filmes já exibidos e por exibir;

Conjuntamente, serão desenvolvidas mais funcionalidades tais como:

* Marcação pela internet via website;
* Pagamento de imediato via internet;
* Código QR(SMS) para identificação válida do bilhete comprado via a app;

**Fundamentação Técnica-Científica**

Análise dos Controlos e Técnicas Utilizadas

Demos utilização ao C# ou C Sharp, que é uma linguagem de programação orientada a objetos, desenvolvida pela Microsoft como parte da plataforma .NET, esta mesma inclui muitas influências de outras linguagens de programação como Java e Pascal.

Em complemento ao C# também usámos o MySQL que é um sistema de gestão de base de dados e utiliza a linguagem SQL como interface. Este sistema de gestão é dos mais populares.

**Descrição do Produto**

O cinema manager permite a gestão de uma empresa de cinemas mais facilitada aos funcionários e administradores, esta aplicação serve para se poder receber reservas de lugares para a sala de cinema com o filme respetivo e consegue ter o maior controlo no que diz respeito a lotação como a contabilidade.

**Em modo administrador as funcionalidades são:**

Os administradores têm todas as permissões sem quaisquer restrições, e devem cingir-se a gerir todos os empregados actualmente no cinema;

Devem também fazer a gestão das salas do cinema indicando a capacidade de cada uma.

Podem visualizar toda a contabilidade semanal ou mensal dos custos, investimentos e lucros.

**Em modo gestor as funcionalidades são:**

Os gestores estão encarregados de fazer uma total gestão das colecções de Filmes o seus respectivos resumos (Trailer’s).

Gestão dos interveniente de cada filme, sejam estes os realizadores, actores e até mesmo productores.

Elaborar e gerir as sessões diárias para os filmes e as respectivas salas.

Não têm permissões de acesso à parte Administrativa.

**Em modo funcionário as funcionalidades são:**

Os funcionários têm a função de efectuar as reservas mediante as salas que os Administradores criam e os filmes que os Gestores disponibilizam.

São estes utilizadores que estarão no atendimento ao público e a tirarem proveito desta aplicação para que o mesmo serviço seja rápido e com qualidade

Não têm permissões de acesso à parte de Gestão nem à parte Administrativa.

**Condicionalismos**

**Administração:**Contabilidade.  
Definições (Connections Strings, Base de Dados).

**Gestão:**Sessões

Não existe painel de gestão de anúncios.

Filmes  
 Não é possível Associar os intervenientes (Realizadores, Actores, Produtores).  
Pessoas  
 Não é possível Associar os mesmos.

**Bilheteira:**  
Não está implementada a reserva de bilhetes (Lugar, Filme, Sala).  
Não interage com o gestor nem administrador.

**Anexos**

* **[A] Classe de Variáveis Globais**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Cinema\_Manager\_1.\_0

{

class CLASS\_GLOBAL\_VARIABLES

{

public static string Servidor = "datasource=bitcorp.servegame.com;port=3306;username=cinema;password=cinema";

public static string ServidorStoredProcedure = "Server= bitcorp.servegame.com;Database=cinema;Uid=cinema;Pwd=cinema;Use Procedure Bodies=false;";

public static string Endereco = "localhost";

public static string Porta = "3306";

public static string Utilizador = "cinema";

public static string Password = "cinema";

public static string BD = "cinema";

public static bool RankMatch = false;

public static bool PwdMatch = false;

public static int PermissionCheck;

public static string PwdCheck;

public static string GestaoResumos\_RequestSource = "";

public static bool NovoResumo = false;

public static string AUX\_RESUMO\_ID;

public static int INT\_AUX\_RESUMO\_ID;

public static string AUX\_RESUMO\_Titulo;

public static string AUX\_RESUMO\_Duracao;

public static string AUX\_RESUMO\_Estado;

public static string AUX\_RESUMO\_Resumo;

public static string AUX\_RESUMO\_DuracaoResumo;

}

}

* **[B] Form de Login**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using System.IO;

using System.Threading;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0

{

public partial class FORM\_LOGIN : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

DataTable TabelaDados = new DataTable();

public int UserNum;

public string UserPass;

public string UserRank;

public FORM\_LOGIN()

{

InitializeComponent();

}

private void BUTTON\_LOGIN\_Click(object sender, EventArgs e)

{

UserNum = Convert.ToInt32(TEXTBOX\_NOME.Text);

UserPass = TEXTBOX\_PASSWORD.Text;

Properties.Settings.Default.succes\_increment = 0;

if(RADIOBUTTON\_ADMIN.Checked == true)

{

Properties.Settings.Default.administrador = true;

Properties.Settings.Default.gestor = false;

Properties.Settings.Default.funcionario = false;

Properties.Settings.Default.permissoes = 3;

UserRank = "Administrador";

}

if(RADIOBUTTON\_GESTOR.Checked == true)

{

Properties.Settings.Default.administrador = false;

Properties.Settings.Default.gestor = true;

Properties.Settings.Default.funcionario = false;

Properties.Settings.Default.permissoes = 2;

UserRank = "Gestor";

}

if(RADIOBUTTON\_FUNCIONARIO.Checked == true)

{

Properties.Settings.Default.administrador = false;

Properties.Settings.Default.gestor = false;

Properties.Settings.Default.funcionario = true;

Properties.Settings.Default.permissoes = 1;

UserRank = "Funcionário";

}

if (UserExiste() == true)

{

if (VerificarCargo() == true)

{

if (VerificarPWD() == true)

{

IniciarSessao();

}

}

}

LigacaoDB.Close();

}

/\*

---------------------------------------------------------------------------------------------------

\*/

public bool UserExiste()

{

try

{

int AUX\_User;

LigacaoDB.Open();

string Query\_SelectUser = "SELECT ID FROM cinema.funcionarios WHERE ID = " + UserNum + "";

MySqlCommand SELECT\_User = new MySqlCommand(Query\_SelectUser, LigacaoDB);

AUX\_User = Convert.ToInt32(SELECT\_User.ExecuteScalar());

if(AUX\_User == UserNum)

{

LigacaoDB.Close();

Properties.Settings.Default.succes\_increment = Properties.Settings.Default.succes\_increment + 1;

return true;

}

else

{

LigacaoDB.Close();

MessageBox.Show("Ocorreu um erro ao verificar a existência do " + UserRank + ". Excepção: \n\n", "Verificar Utilizador", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

catch(Exception EX)

{

LigacaoDB.Close();

MessageBox.Show("Ocorreu um erro ao verificar a existência do " + UserRank + ".\n\nDescrição do Erro:\n" + EX.Message, "Verificar Utilizador", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

public bool VerificarCargo()

{

try

{

LigacaoDB.Open();

string Query\_SelectCargo = "SELECT cargo FROM cinema.funcionarios WHERE ID = " + UserNum + "";

MySqlCommand SELECT\_Cargo = new MySqlCommand(Query\_SelectCargo, LigacaoDB);

CLASS\_GLOBAL\_VARIABLES.PermissionCheck = Convert.ToInt32(SELECT\_Cargo.ExecuteScalar());

if (CLASS\_GLOBAL\_VARIABLES.PermissionCheck == Properties.Settings.Default.permissoes)

{

LigacaoDB.Close();

CLASS\_GLOBAL\_VARIABLES.RankMatch = true;

Properties.Settings.Default.succes\_increment = Properties.Settings.Default.succes\_increment + 1;

return true;

}

else

{

LigacaoDB.Close();

MessageBox.Show("Ocorreu um erro ao confirmar o cargo " + UserRank + "\n\n Não tem permissões de " + UserRank + "", "Verificar Cargo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

catch(Exception EX)

{

LigacaoDB.Close();

MessageBox.Show("Ocorreu um erro ao confirmar o cargo " + UserRank + ". Excepção: \n\n " + EX.Message + "", "Verificar Cargo", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

public bool VerificarPWD()

{

try

{

LigacaoDB.Open();

string Query\_SelectPWD = "SELECT password FROM cinema.login WHERE id\_alias = " + UserNum + "";

MySqlCommand SELECT\_PWD = new MySqlCommand(Query\_SelectPWD, LigacaoDB);

CLASS\_GLOBAL\_VARIABLES.PwdCheck = SELECT\_PWD.ExecuteScalar().ToString();

if (CLASS\_GLOBAL\_VARIABLES.PwdCheck == UserPass)

{

LigacaoDB.Close();

CLASS\_GLOBAL\_VARIABLES.PwdMatch = true;

Properties.Settings.Default.succes\_increment = Properties.Settings.Default.succes\_increment + 1;

return true;

}

else

{

LigacaoDB.Close();

MessageBox.Show("A Password está incorrecta.", "Password Incorrecta", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

catch(Exception EX)

{

LigacaoDB.Close();

MessageBox.Show("Ocorreu um erro ao confirmar a passsword " + UserRank + ". Excepção: \n\n " + EX.Message + "", "Verificar Password", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

public void IniciarSessao()

{

if(Properties.Settings.Default.succes\_increment == 3)

{

Properties.Settings.Default.credentials\_correct = true;

}

if(Properties.Settings.Default.credentials\_correct == true)

{

this.Hide();

FORM\_INICIO FormInicio = new FORM\_INICIO();

FormInicio.Show();

}

}

private void BUTTON\_LIGACAO\_Click(object sender, EventArgs e)

{

Properties.Settings.Default.connection\_request\_login = true;

this.Hide();

FORMS\_ADMIN.FORM\_DEFINICOES\_LIGACAO FormLigacoes = new FORMS\_ADMIN.FORM\_DEFINICOES\_LIGACAO();

FormLigacoes.ShowDialog();

}

private void FORM\_LOGIN\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

}

}

* **[C] Form da Janela MDI**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Cinema\_Manager\_1.\_0

{

public partial class FORM\_INICIO : Form

{

public FORM\_INICIO()

{

InitializeComponent();

if (Properties.Settings.Default.permissoes < 3)

{

foreach (ToolStripMenuItem BTN in MENUSTRIP\_FORM\_INICIO.Items)

{

if (BTN.Text == "Administração")

BTN.Visible = false;

}

}

if (Properties.Settings.Default.permissoes == 2)

{

foreach (ToolStripMenuItem BTN in MENUSTRIP\_FORM\_INICIO.Items)

{

if (BTN.Text == "Bilheteira")

BTN.Visible = false;

}

}

if (Properties.Settings.Default.permissoes == 1)

{

foreach (ToolStripMenuItem BTN in MENUSTRIP\_FORM\_INICIO.Items)

{

if (BTN.Text == "Gestão")

BTN.Visible = false;

}

}

foreach (Control cntrl in this.Controls)

{

if (cntrl is MdiClient)

{

this.BackgroundImageLayout = ImageLayout.Stretch;

cntrl.BackgroundImage = Properties.Resources.Cinema\_Background;

//cntrl.BackgroundImage = System.Drawing.Image.FromFile(@"C:\Desert.jpg");

}

}

}

private void MENUSTRIP\_BUTTON\_ADMIN\_Click(object sender, EventArgs e)

{

//FORM\_ADMIN FormAdmin = new FORM\_ADMIN();

//FormAdmin.MdiParent = this;

//if(FormAdmin != null)

// FormAdmin.Show();

FORM\_ADMIN FormAberto = new FORM\_ADMIN();

Verificar\_FormsAbertos(FormAberto);

}

private void FORM\_INICIO\_FormClosed(object sender, FormClosedEventArgs e)

{

FORM\_LOGIN FormLogin = new FORM\_LOGIN();

FormLogin.Visible = true;

}

private void MENUSTRIP\_FORM\_INICIO\_BUTTON\_FECHAR\_Click(object sender, EventArgs e)

{

this.Close();

}

public void OcultarAdmin()

{

MENUSTRIP\_BUTTON\_ADMIN.Visible = false;

}

public void Verificar\_FormsAbertos(Form FormAberto)

{

foreach (Form FRM in this.MdiChildren)

{

if (FRM.GetType() == FormAberto.GetType())

{

FRM.Focus();

return;

}

}

FormAberto.MdiParent = this;

FormAberto.Show();

}

private void MENUSTRIP\_BUTTON\_GESTAO\_Click(object sender, EventArgs e)

{

FORMS\_GESTAO.FORM\_GESTAO FormAberto = new FORMS\_GESTAO.FORM\_GESTAO();

Verificar\_FormsAbertos(FormAberto);

}

}

}

* **[D] Form do Painel de Aministração**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Cinema\_Manager\_1.\_0

{

public partial class FORM\_ADMIN : Form

{

public FORM\_ADMIN()

{

InitializeComponent();

}

private void MENUSTRIP\_FORM\_ADMIN\_BUTTON\_FUNCIONARIOS\_Click(object sender, EventArgs e)

{

}

private void PICTUREBOX\_FormAdmin\_FUNCIONARIOS\_Click(object sender, EventArgs e)

{

}

private void LABEL\_FormAdmin\_FUNCIONARIOS\_Click(object sender, EventArgs e)

{

}

private void PICTUREBOX\_ADMIN\_FUNCIONARIOS\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_ADMIN.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_ADMIN.Panel2.Controls.Clear();

FORMS\_ADMIN.FORM\_ADMIN\_FUNCIONARIOS FormAdminFuncionarios = new FORMS\_ADMIN.FORM\_ADMIN\_FUNCIONARIOS();

FormAdminFuncionarios.TopLevel = false;

FormAdminFuncionarios.AutoScroll = true;

SPLITCONTAINER\_ADMIN.Panel2.Controls.Add(FormAdminFuncionarios);

FormAdminFuncionarios.Visible = true;

return;

}

if(SPLITCONTAINER\_ADMIN.Panel2.Controls.Count == 1 || SPLITCONTAINER\_ADMIN.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_ADMIN.Panel2.Controls.Clear();

FORMS\_ADMIN.FORM\_ADMIN\_FUNCIONARIOS FormAdminFuncionarios = new FORMS\_ADMIN.FORM\_ADMIN\_FUNCIONARIOS();

FormAdminFuncionarios.TopLevel = false;

FormAdminFuncionarios.AutoScroll = true;

SPLITCONTAINER\_ADMIN.Panel2.Controls.Add(FormAdminFuncionarios);

FormAdminFuncionarios.Visible = true;

return;

}

}

private void PICTUREBOX\_ADMIN\_SALAS\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_ADMIN.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_ADMIN.Panel2.Controls.Clear();

FORMS\_ADMIN.FORM\_ADMIN\_SALAS FormAdminSalas = new FORMS\_ADMIN.FORM\_ADMIN\_SALAS();

FormAdminSalas.TopLevel = false;

FormAdminSalas.AutoScroll = true;

SPLITCONTAINER\_ADMIN.Panel2.Controls.Add(FormAdminSalas);

FormAdminSalas.Visible = true;

return;

}

if (SPLITCONTAINER\_ADMIN.Panel2.Controls.Count == 1 || SPLITCONTAINER\_ADMIN.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_ADMIN.Panel2.Controls.Clear();

FORMS\_ADMIN.FORM\_ADMIN\_SALAS FormAdminSalas = new FORMS\_ADMIN.FORM\_ADMIN\_SALAS();

FormAdminSalas.TopLevel = false;

FormAdminSalas.AutoScroll = true;

SPLITCONTAINER\_ADMIN.Panel2.Controls.Add(FormAdminSalas);

FormAdminSalas.Visible = true;

return;

}

}

private void PICTUREBOX\_ADMIN\_CONTABILIDADE\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_ADMIN.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_ADMIN.Panel2.Controls.Clear();

FORMS\_ADMIN.FORM\_ADMIN\_CONTABILIDADE FormAdminContabilidade = new FORMS\_ADMIN.FORM\_ADMIN\_CONTABILIDADE();

FormAdminContabilidade.TopLevel = false;

FormAdminContabilidade.AutoScroll = true;

SPLITCONTAINER\_ADMIN.Panel2.Controls.Add(FormAdminContabilidade);

FormAdminContabilidade.Visible = true;

return;

}

if (SPLITCONTAINER\_ADMIN.Panel2.Controls.Count == 1 || SPLITCONTAINER\_ADMIN.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_ADMIN.Panel2.Controls.Clear();

FORMS\_ADMIN.FORM\_ADMIN\_CONTABILIDADE FormAdminContabilidade = new FORMS\_ADMIN.FORM\_ADMIN\_CONTABILIDADE();

FormAdminContabilidade.TopLevel = false;

FormAdminContabilidade.AutoScroll = true;

SPLITCONTAINER\_ADMIN.Panel2.Controls.Add(FormAdminContabilidade);

FormAdminContabilidade.Visible = true;

return;

}

}

private void MENUSTRIP\_FORM\_ADMIN\_BUTTON\_FECHAR\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

* **[E] Form de Funcionários**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_ADMIN

{

public partial class FORM\_ADMIN\_FUNCIONARIOS : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_ADMIN\_FUNCIONARIOS()

{

InitializeComponent();

Preencher\_Listview\_Funcionarios();

}

private void PICTUREBOX\_REMOVER\_FUNCIONARIO\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_FUNCIONARIO\_NOME.Text == "")

{

MessageBox.Show("Não tem nenhum funcionário selecionado! Não é possivel remover.", "Nenhum funcionário selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_FUNCIONARIO\_NOME.Text != "")

{

DialogResult RemoverFuncionario = MessageBox.Show("Tem a certeza que deseja remover este funcionário?", "Remover Funcionário?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (RemoverFuncionario == DialogResult.Yes)

{

Remover\_Funcionario();

ClearListView();

Preencher\_Listview\_Funcionarios();

Limpar\_TextBoxs();

}

}

}

private void PICTUREBOX\_GRAVAR\_ALTERACOES\_FUNCIONARIO\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_FUNCIONARIO\_NOME.Text == "")

{

MessageBox.Show("Não tem nenhum funcionário selecionado! Não é possivel alterar.", "Nenhum funcionário selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_FUNCIONARIO\_NOME.Text != "")

{

DialogResult DefinirDados = MessageBox.Show("Tem a certeza que deseja editar as informações deste funcionário?", "Alterar Informações?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (DefinirDados == DialogResult.Yes)

{

Alterar\_Funcionario();

ClearListView();

Preencher\_Listview\_Funcionarios();

Limpar\_TextBoxs();

}

}

}

private void PICTUREBOX\_ADICIONAR\_FUNCIONARIO\_Click(object sender, EventArgs e)

{

TABCONTROL\_FUNCIONARIOS.SelectedIndex = 1;

}

private void LISTVIEW\_FUNCIONARIOS\_SelectedIndexChanged(object sender, EventArgs e)

{

Preencher\_TextBox\_Funcionario();

}

private void BUTTON\_ADICIONAR\_FUNCIONARIO\_Click(object sender, EventArgs e)

{

Adicionar\_Funcionario();

ClearListView();

Preencher\_Listview\_Funcionarios();

Limpar\_TextBoxs();

TABCONTROL\_FUNCIONARIOS.SelectedIndex = 0;

}

private void CHECKBOX\_ADICIONARFUNCIONARIO\_MOSTRAR\_PASSWORD\_CheckedChanged(object sender, EventArgs e)

{

if (CHECKBOX\_ADICIONARFUNCIONARIO\_MOSTRAR\_PASSWORD.Checked == true)

TEXTBOX\_ADICIONARFUNCIONARIO\_PASSWORD.PasswordChar = '\0';

if (CHECKBOX\_ADICIONARFUNCIONARIO\_MOSTRAR\_PASSWORD.Checked == false)

TEXTBOX\_ADICIONARFUNCIONARIO\_PASSWORD.PasswordChar = '●';

}

private void CHECKBOX\_FUNCIONARIO\_PASSWORD\_CheckedChanged(object sender, EventArgs e)

{

if (CHECKBOX\_FUNCIONARIO\_PASSWORD.Checked == true)

TEXTBOX\_FUNCIONARIO\_PASSWORD.PasswordChar = '\0';

if (CHECKBOX\_FUNCIONARIO\_PASSWORD.Checked == false)

TEXTBOX\_FUNCIONARIO\_PASSWORD.PasswordChar = '●';

}

private void TEXTBOX\_FUNCIONARIO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_FUNCIONARIO\_NOME.Text = Regex.Replace(TEXTBOX\_FUNCIONARIO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("\"", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("|", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("!", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace(@"\", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("@", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("#", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("£", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("$", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("§", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("%", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("&", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("/", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("{", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("(", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("[", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace(")", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("]", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("=", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("}", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("?", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("'", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("»", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("«", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("+", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("-", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("\*", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("¨", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("ª", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("º", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("<", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace(">", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace(",", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace(";", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace(".", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace(":", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("-", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("\_", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_NOME.Text.Replace("€", string.Empty);

}

private void TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text;

}

private void TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text = Regex.Replace(TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text = Regex.Replace(TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_FUNCIONARIO\_NOME.Text = TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text;

}

private void TEXTBOX\_FUNCIONARIO\_NUMERO\_BI\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_FUNCIONARIO\_NUMERO\_BI.Text = Regex.Replace(TEXTBOX\_FUNCIONARIO\_NUMERO\_BI.Text, @"[^\d]", string.Empty);

}

private void TEXTBOX\_FUNCIONARIO\_PASSWORD\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_FUNCIONARIO\_PASSWORD.Text = Regex.Replace(TEXTBOX\_FUNCIONARIO\_PASSWORD.Text, @"[^ A-Z a-z 0-9]", string.Empty);

}

private void TEXTBOX\_ADICIONARFUNCIONARIO\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_ADICIONARFUNCIONARIO\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_ADICIONARFUNCIONARIO\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_ADICIONARFUNCIONARIO\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_ADICIONARFUNCIONARIO\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

private void TEXTBOX\_ADICIONARFUNCIONARIO\_ULTIMO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_ADICIONARFUNCIONARIO\_ULTIMO\_NOME.Text = Regex.Replace(TEXTBOX\_ADICIONARFUNCIONARIO\_ULTIMO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_ADICIONARFUNCIONARIO\_ULTIMO\_NOME.Text = Regex.Replace(TEXTBOX\_ADICIONARFUNCIONARIO\_ULTIMO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

private void TEXTBOX\_ADICIONARFUNCIONARIO\_NUMERO\_BI\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_ADICIONARFUNCIONARIO\_NUMERO\_BI.Text = Regex.Replace(TEXTBOX\_ADICIONARFUNCIONARIO\_NUMERO\_BI.Text, @"[^\d]", string.Empty);

}

private void TEXTBOX\_ADICIONARFUNCIONARIO\_PASSWORD\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_ADICIONARFUNCIONARIO\_PASSWORD.Text = Regex.Replace(TEXTBOX\_ADICIONARFUNCIONARIO\_PASSWORD.Text, @"[^ A-Z a-z 0-9]", string.Empty);

}

private void LISTVIEW\_FUNCIONARIOS\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar();

}

private void CHECKBOX\_AGRUPAR\_FUNCIONARIOS\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_FUNCIONARIOS.ShowGroups = CHECKBOX\_AGRUPAR\_FUNCIONARIOS.Checked;

Ordenar();

}

public void Preencher\_Listview\_Funcionarios()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT ID,primeiro\_nome,segundo\_nome,cargo FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".funcionarios";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["primeiro\_nome"].ToString());

ROW.SubItems.Add(Reader["segundo\_nome"].ToString());

if(Reader["cargo"].ToString() == "1")

ROW.SubItems.Add("Funcionario");

if(Reader["cargo"].ToString() == "2")

ROW.SubItems.Add("Gestor");

if (Reader["cargo"].ToString() == "3")

ROW.SubItems.Add("Administrador");

LISTVIEW\_FUNCIONARIOS.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void ClearListView()

{

LISTVIEW\_FUNCIONARIOS.Items.Clear();

}

public void Preencher\_TextBox\_Funcionario()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_FUNCIONARIOS.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

LigacaoDB.Open();

string Nome;

string PrimeiroNome;

string SegundoNome;

string DataNascimento;

string NumeroBI;

string Cargo;

string Password;

string Query\_PrimeiroNome = "SELECT primeiro\_nome FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".funcionarios WHERE ID = " + ID + "";

string Query\_SegundoNome = "SELECT segundo\_nome FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".funcionarios WHERE ID = " + ID + "";

string Query\_DataNascimento = "SELECT data\_nascimento FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".funcionarios WHERE ID = " + ID + "";

string Query\_NumeroBI = "SELECT numero\_bi FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".funcionarios WHERE ID = " + ID + "";

string Query\_Cargo = "SELECT cargo FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".funcionarios WHERE ID = " + ID + "";

string Query\_Password = "SELECT password FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".login WHERE id\_alias = " + ID + "";

MySqlCommand Comando\_PrimeiroNome = new MySqlCommand(Query\_PrimeiroNome, LigacaoDB);

MySqlCommand Comando\_SegundoNome = new MySqlCommand(Query\_SegundoNome, LigacaoDB);

MySqlCommand Comando\_DataNascimento = new MySqlCommand(Query\_DataNascimento, LigacaoDB);

MySqlCommand Comando\_NumeroBI = new MySqlCommand(Query\_NumeroBI, LigacaoDB);

MySqlCommand Comando\_Cargo = new MySqlCommand(Query\_Cargo, LigacaoDB);

MySqlCommand Comando\_Password = new MySqlCommand(Query\_Password, LigacaoDB);

PrimeiroNome = Comando\_PrimeiroNome.ExecuteScalar().ToString();

SegundoNome = Comando\_SegundoNome.ExecuteScalar().ToString();

Nome = PrimeiroNome + " " + SegundoNome;

DataNascimento = Comando\_DataNascimento.ExecuteScalar().ToString();

NumeroBI = Comando\_NumeroBI.ExecuteScalar().ToString();

Cargo = Comando\_Cargo.ExecuteScalar().ToString();

Password = Comando\_Password.ExecuteScalar().ToString();

TEXTBOX\_FUNCIONARIO\_NOME.Text = Nome;

TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text = PrimeiroNome;

TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text = SegundoNome;

DATETIMEPICKER\_FUNCIONARIO\_DATA\_NASCIMENTO.Value = DateTime.Parse(DataNascimento);

TEXTBOX\_FUNCIONARIO\_NUMERO\_BI.Text = NumeroBI;

TEXTBOX\_FUNCIONARIO\_PASSWORD.Text = Password;

if (Cargo == "1")

COMBOBOX\_FUNCIONARIO\_CARGO.SelectedIndex = 2;

if(Cargo == "2")

COMBOBOX\_FUNCIONARIO\_CARGO.SelectedIndex = 1;

if (Cargo == "3")

COMBOBOX\_FUNCIONARIO\_CARGO.SelectedIndex = 0;

LigacaoDB.Close();

}

catch(Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Adicionar\_Funcionario()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string Alias;

string PrimeiroNome = TEXTBOX\_ADICIONARFUNCIONARIO\_PRIMEIRO\_NOME.Text;

string UltimoNome = TEXTBOX\_ADICIONARFUNCIONARIO\_ULTIMO\_NOME.Text;

string DataNascimento = DATEPICKER\_ADICIONARFUNCIONARIO\_DATA\_NASCIMENTO.Value.ToShortDateString();

string NumeroBI = TEXTBOX\_ADICIONARFUNCIONARIO\_NUMERO\_BI.Text;

int INT\_NumeroBI = Convert.ToInt32(NumeroBI);

string Cargo = COMBOBOX\_ADICIONARFUNCIONARIO\_CARGO.Text;

int CargoID = 0;

if (Cargo == "Funcionário")

CargoID = 1;

if (Cargo == "Gestor")

CargoID = 2;

if (Cargo == "Administrador")

CargoID = 3;

string Password = TEXTBOX\_ADICIONARFUNCIONARIO\_PASSWORD.Text;

if(Password == "")

{

string Password\_Part1 = NumeroBI.Substring(0, 3);

string Password\_Part2 = PrimeiroNome.ToLower();

string Password\_Part3 = UltimoNome.ToLower();

Password = Password\_Part1 + Password\_Part2 + Password\_Part3;

}

Alias = CargoID + "-" + PrimeiroNome.ToLower() + "-" + UltimoNome.ToLower();

MySqlCommand Comando\_FuncionarioInserir = new MySqlCommand("SP\_FUNCIONARIO\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_FuncionarioInserir.CommandType = CommandType.StoredProcedure;

Comando\_FuncionarioInserir.Parameters.AddWithValue("\_alias", Alias);

Comando\_FuncionarioInserir.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando\_FuncionarioInserir.Parameters.AddWithValue("\_ultimo\_nome", UltimoNome);

Comando\_FuncionarioInserir.Parameters.AddWithValue("\_data\_nascimento", DataNascimento);

Comando\_FuncionarioInserir.Parameters.AddWithValue("\_numero\_bi", INT\_NumeroBI);

Comando\_FuncionarioInserir.Parameters.AddWithValue("\_cargo", CargoID);

Reader = Comando\_FuncionarioInserir.ExecuteReader();

Reader.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

{

LigacaoDB\_StoredProcedures.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Closed)

{

System.Threading.Thread.Sleep(2000);

LigacaoDB.Open();

string Query\_GetAlias = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".funcionarios WHERE alias='" + Alias + "'";

string GetID;

int INT\_GetID;

MySqlCommand Comando\_GetAlias = new MySqlCommand(Query\_GetAlias, LigacaoDB);

GetID = Comando\_GetAlias.ExecuteScalar().ToString();

INT\_GetID = Convert.ToInt32(GetID);

if (LigacaoDB.State == ConnectionState.Open)

{

LigacaoDB.Close();

if (LigacaoDB.State == ConnectionState.Closed)

{

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_LoginInserir = new MySqlCommand("SP\_LOGIN\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_LoginInserir.CommandType = CommandType.StoredProcedure;

Comando\_LoginInserir.Parameters.AddWithValue("\_id\_alias", INT\_GetID);

Comando\_LoginInserir.Parameters.AddWithValue("\_alias", Alias);

Comando\_LoginInserir.Parameters.AddWithValue("\_password", Password);

Reader = Comando\_LoginInserir.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Funcionario Inserido com Sucesso", "Funcionario Inserido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

}

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Remover\_Funcionario()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_FUNCIONARIOS.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

MySqlCommand Command\_LoginRemover = new MySqlCommand("SP\_LOGIN\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_LoginRemover.CommandType = CommandType.StoredProcedure;

Command\_LoginRemover.Parameters.AddWithValue("\_id\_alias", ID);

Reader = Command\_LoginRemover.ExecuteReader();

System.Threading.Thread.Sleep(2000);

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

{

LigacaoDB\_StoredProcedures.Close();

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Command = new MySqlCommand("SP\_FUNCIONARIO\_REMOVER", LigacaoDB\_StoredProcedures);

Command.CommandType = CommandType.StoredProcedure;

Command.Parameters.AddWithValue("\_id", ID);

Reader = Command.ExecuteReader();

Reader.Close();

MessageBox.Show("Funcionário removido com sucesso", "Funcionario Removido", MessageBoxButtons.OK, MessageBoxIcon.Information);

LigacaoDB\_StoredProcedures.Close();

}

}

catch(Exception EX)

{ }

}

public void Alterar\_Funcionario()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_FUNCIONARIOS.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

string Alias;

string PrimeiroNome = TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text;

string SegundoNome = TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text;

string DataNascimento = DATETIMEPICKER\_FUNCIONARIO\_DATA\_NASCIMENTO.Value.ToShortDateString();

int NumeroBI = Convert.ToInt32(TEXTBOX\_FUNCIONARIO\_NUMERO\_BI.Text);

int Cargo = 0;

string Password = TEXTBOX\_FUNCIONARIO\_PASSWORD.Text;

if (COMBOBOX\_FUNCIONARIO\_CARGO.Text == "Funcionário")

Cargo = 1;

if (COMBOBOX\_FUNCIONARIO\_CARGO.Text == "Gestor")

Cargo = 2;

if (COMBOBOX\_FUNCIONARIO\_CARGO.Text == "Administrador")

Cargo = 3;

Alias = Cargo + "-" + PrimeiroNome.ToLower() + "-" + SegundoNome.ToLower();

MySqlCommand Comando = new MySqlCommand("SP\_FUNCIONARIO\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando.CommandType = CommandType.StoredProcedure;

Comando.Parameters.AddWithValue("\_id", ID);

Comando.Parameters.AddWithValue("\_alias", Alias);

Comando.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando.Parameters.AddWithValue("\_segundo\_nome", SegundoNome);

Comando.Parameters.AddWithValue("\_data\_nascimento", DataNascimento);

Comando.Parameters.AddWithValue("\_numero\_bi", NumeroBI);

Comando.Parameters.AddWithValue("\_cargo", Cargo);

Comando.ExecuteNonQuery();

MySqlCommand Comando\_Login = new MySqlCommand("SP\_LOGIN\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando\_Login.CommandType = CommandType.StoredProcedure;

Comando\_Login.Parameters.AddWithValue("\_id\_alias", ID);

Comando\_Login.Parameters.AddWithValue("\_alias", Alias);

Comando\_Login.Parameters.AddWithValue("\_password", Password);

Comando\_Login.ExecuteNonQuery();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Funcionario Alterado com Sucesso", "Funcionario Alterado", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Limpar\_TextBoxs()

{

TEXTBOX\_FUNCIONARIO\_NOME.Text = "";

TEXTBOX\_FUNCIONARIO\_PRIMEIRO\_NOME.Text = "";

TEXTBOX\_FUNCIONARIO\_SEGUNDO\_NOME.Text = "";

TEXTBOX\_FUNCIONARIO\_NUMERO\_BI.Text = "";

COMBOBOX\_FUNCIONARIO\_CARGO.SelectedItem = null;

TEXTBOX\_FUNCIONARIO\_PASSWORD.Text = "";

TEXTBOX\_ADICIONARFUNCIONARIO\_PRIMEIRO\_NOME.Text = "";

TEXTBOX\_ADICIONARFUNCIONARIO\_ULTIMO\_NOME.Text = "";

TEXTBOX\_ADICIONARFUNCIONARIO\_NUMERO\_BI.Text = "";

COMBOBOX\_ADICIONARFUNCIONARIO\_CARGO.SelectedItem = null;

TEXTBOX\_ADICIONARFUNCIONARIO\_PASSWORD.Text = "";

}

public void Ordenar()

{

if (CHECKBOX\_AGRUPAR\_FUNCIONARIOS.Checked == true)

{

if (LISTVIEW\_FUNCIONARIOS.ShowGroups)

BuildGroups(UltimaColunaOrdenada);

}

else

LISTVIEW\_FUNCIONARIOS.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups(int coluna)

{

LISTVIEW\_FUNCIONARIOS.Groups.Clear();

int Contagem = LISTVIEW\_FUNCIONARIOS.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_FUNCIONARIOS.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_FUNCIONARIOS.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}

* **[F] Form de Salas**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_ADMIN

{

public partial class FORM\_ADMIN\_SALAS : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_ADMIN\_SALAS()

{

InitializeComponent();

Preencher\_ListView();

}

private void LISTVIEW\_ADMIN\_SALAS\_SelectedIndexChanged(object sender, EventArgs e)

{

Preencher\_TextBoxs();

}

private void PICTUREBOX\_ADICIONAR\_SALA\_Click(object sender, EventArgs e)

{

TABCONTROL\_ADMIN\_SALAS.SelectedIndex = 1;

}

private void BUTTON\_\_ADICIONAR\_SALA\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE.Text == "")

{

MessageBox.Show("Não indicou a capacidade para a nova sala! Não é possivel inserir.", "Capacidade não Indicada", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE.Text != "")

{

DialogResult DialogResult\_InserirSala = MessageBox.Show("Tem a certeza que deseja inserir uma nova sala com Capacidade de: " + TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE.Text + " Lugares?", "Inserir Sala?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (DialogResult\_InserirSala == DialogResult.Yes)

{

Inserir\_Sala();

Limpar\_TextBoxs();

ClearListView();

Preencher\_ListView();

TABCONTROL\_ADMIN\_SALAS.SelectedIndex = 0;

}

}

}

private void PICTUREBOX\_SALAS\_GRAVAR\_ALTERACOES\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text == "")

{

MessageBox.Show("Não tem nenhuma sala selecionada! Não é possivel alterar.", "Nenhuma Sala Selecionada", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text != "")

{

DialogResult DialogResult\_AlterarSala = MessageBox.Show("Tem a certeza que deseja alterar a sala selecionada?", "Alterar Sala?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (DialogResult\_AlterarSala == DialogResult.Yes)

{

Alterar\_Sala();

Limpar\_TextBoxs();

ClearListView();

Preencher\_ListView();

}

}

}

private void PICTUREBOX\_SALAS\_REMOVER\_SALA\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text == "")

{

MessageBox.Show("Não tem nenhuma sala selecionada! Não é possivel remover.", "Nenhuma Sala Selecionada", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text != "")

{

DialogResult DialogResult\_RemoverSala = MessageBox.Show("Tem a certeza que deseja remover a sala selecionada?", "Remover Sala?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (DialogResult\_RemoverSala == DialogResult.Yes)

{

Remover\_Sala();

Limpar\_TextBoxs();

ClearListView();

Preencher\_ListView();

}

}

}

private void TEXTBOX\_SALAS\_CAPACIDADE\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_SALAS\_CAPACIDADE.Text = Regex.Replace(TEXTBOX\_SALAS\_CAPACIDADE.Text, @"[^\d]", string.Empty);

}

private void TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE.Text = Regex.Replace(TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE.Text, @"[^\d]", string.Empty);

}

private void LISTVIEW\_ADMIN\_SALAS\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar();

}

private void CHECKBOX\_AGRUPAR\_SALAS\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_ADMIN\_SALAS.ShowGroups = CHECKBOX\_AGRUPAR\_SALAS.Checked;

Ordenar();

}

public void Preencher\_ListView()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT ID,capacidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".salas";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["capacidade"].ToString());

LISTVIEW\_ADMIN\_SALAS.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void ClearListView()

{

LISTVIEW\_ADMIN\_SALAS.Items.Clear();

}

public void Preencher\_TextBoxs()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_ADMIN\_SALAS.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

int Capacidade;

LigacaoDB.Open();

string Query\_Capacidade = "SELECT capacidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".salas WHERE ID = " + ID + "";

MySqlCommand Comando\_Capacidade = new MySqlCommand(Query\_Capacidade, LigacaoDB);

Capacidade = Convert.ToInt32(Comando\_Capacidade.ExecuteScalar());

TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text = ID.ToString();

TEXTBOX\_SALAS\_CAPACIDADE.Text = Capacidade.ToString();

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Limpar\_TextBoxs()

{

TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text = "";

TEXTBOX\_SALAS\_CAPACIDADE.Text =

TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE.Text = "";

}

//public void Inserir\_IdSala() { }

public void Inserir\_Sala()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

int Capacidade = Convert.ToInt32(TEXTBOX\_ADICIONAR\_SALA\_CAPACIDADE.Text);

MySqlCommand Comando\_SalaInserir = new MySqlCommand("SP\_SALAS\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_SalaInserir.CommandType = CommandType.StoredProcedure;

Comando\_SalaInserir.Parameters.AddWithValue("\_capacidade", Capacidade);

Reader = Comando\_SalaInserir.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Sala inserida com Sucesso", "Sala Inserida", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Alterar\_Sala()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

int ID = Convert.ToInt32(TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text);

int Capacidade = Convert.ToInt32(TEXTBOX\_SALAS\_CAPACIDADE.Text);

MySqlCommand Comando = new MySqlCommand("SP\_SALAS\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando.CommandType = CommandType.StoredProcedure;

Comando.Parameters.AddWithValue("\_id", ID);

Comando.Parameters.AddWithValue("\_capacidade", Capacidade);

Comando.ExecuteNonQuery();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Sala alterada com Sucesso", "Sala Alterada", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Remover\_Sala()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

int ID = Convert.ToInt32(TEXTBOX\_SALAS\_N\_IDENTIFICACAO.Text);

MySqlCommand Comando = new MySqlCommand("SP\_SALAS\_REMOVER", LigacaoDB\_StoredProcedures);

Comando.CommandType = CommandType.StoredProcedure;

Comando.Parameters.AddWithValue("\_id", ID);

Comando.ExecuteNonQuery();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Sala removida com Sucesso", "Sala Removida", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Ordenar()

{

if (CHECKBOX\_AGRUPAR\_SALAS.Checked == true)

{

if (LISTVIEW\_ADMIN\_SALAS.ShowGroups)

BuildGroups(UltimaColunaOrdenada);

}

else

LISTVIEW\_ADMIN\_SALAS.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups(int coluna)

{

LISTVIEW\_ADMIN\_SALAS.Groups.Clear();

int Contagem = LISTVIEW\_ADMIN\_SALAS.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_ADMIN\_SALAS.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_ADMIN\_SALAS.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}

* **[G] Form do Painel de Gestão**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO : Form

{

public FORM\_GESTAO()

{

InitializeComponent();

}

private void PICTUREBOX\_GESTAO\_SESSOES\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_SESSOES FormGestaoSessoes = new FORM\_GESTAO\_SESSOES();

FormGestaoSessoes.TopLevel = false;

FormGestaoSessoes.AutoScroll = true;

SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoSessoes);

FormGestaoSessoes.Visible = true;

return;

}

if (SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_SESSOES FormGestaoSessoes = new FORM\_GESTAO\_SESSOES();

FormGestaoSessoes.TopLevel = false;

FormGestaoSessoes.AutoScroll = true;

SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoSessoes);

FormGestaoSessoes.Visible = true;

return;

}

}

private void PICTUREBOX\_GESTAO\_FILMES\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

return;

}

if (SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

return;

}

}

private void PICTUREBOX\_GESTAO\_PESSOAS\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS FormGestaoPessoas = new FORM\_GESTAO\_PESSOAS();

FormGestaoPessoas.TopLevel = false;

FormGestaoPessoas.AutoScroll = true;

SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoPessoas);

FormGestaoPessoas.Visible = true;

return;

}

if (SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS FormGestaoPessoas = new FORM\_GESTAO\_PESSOAS();

FormGestaoPessoas.TopLevel = false;

FormGestaoPessoas.AutoScroll = true;

SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoPessoas);

FormGestaoPessoas.Visible = true;

return;

}

}

}

}

* **[H] Form de Gestão de Filmes**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO\_FILMES : Form

{

FORM\_GESTAO FormGestao = (FORM\_GESTAO)Application.OpenForms["FORM\_GESTAO"];

FORM\_GESTAO\_FILMES\_RESUMOS FormGestaoResumos = (FORM\_GESTAO\_FILMES\_RESUMOS)Application.OpenForms["FORM\_GESTAO\_FILMES\_RESUMOS"];

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_GESTAO\_FILMES()

{

InitializeComponent();

Preencher\_ListView\_Filmes();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Productores();

Preencher\_ListView\_Actores();

}

private void PICTUREBOX\_ADICIONAR\_FILME\_Click(object sender, EventArgs e)

{

TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

}

private void PICTUREBOX\_GestaoFilmes\_PROCURAR\_RESUMO\_Click(object sender, EventArgs e)

{

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "GestaoFilmes";

try

{

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_ID = LISTVIEW\_GestaoFilmes\_FILMES.SelectedIndices[0].ToString();

CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID = Convert.ToInt32(CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_ID);

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo = TEXTBOX\_GestaoFilmes\_TITULO.Text;

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao = TEXTBOX\_GestaoFilmes\_DURACAO.Text;

if (RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked == true)

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado = "Por Exibir";

if (RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked == true)

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado = "Em Exibicao";

if (RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked == true)

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado = "Ja Exibido";

}

catch (Exception EX)

{ }

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORM\_GESTAO\_FILMES\_RESUMOS FormGestaoFilmesResumos = new FORM\_GESTAO\_FILMES\_RESUMOS();

FormGestaoFilmesResumos.TopLevel = false;

FormGestaoFilmesResumos.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmesResumos);

FormGestaoFilmesResumos.Visible = true;

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORM\_GESTAO\_FILMES\_RESUMOS FormGestaoFilmesResumos = new FORM\_GESTAO\_FILMES\_RESUMOS();

FormGestaoFilmesResumos.TopLevel = false;

FormGestaoFilmesResumos.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmesResumos);

FormGestaoFilmesResumos.Visible = true;

return;

}

}

catch(Exception EX)

{ }

}

private void PICTUREBOX\_AdicionarFilme\_PROCURAR\_RESUMO\_Click(object sender, EventArgs e)

{

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "AdicionarFilme";

try

{

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo = TEXTBOX\_AdicionarFilme\_TITULO.Text;

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao = TEXTBOX\_AdicionarFilme\_DURACAO.Text;

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo = TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text;

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo = TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text;

if(RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked == true)

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado = "Por Exibir";

if(RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked == true)

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado = "Em Exibicao";

if(RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked == true)

CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado = "Ja Exibido";

}

catch (Exception EX)

{ }

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORM\_GESTAO\_FILMES\_RESUMOS FormGestaoFilmesResumos = new FORM\_GESTAO\_FILMES\_RESUMOS();

FormGestaoFilmesResumos.TopLevel = false;

FormGestaoFilmesResumos.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmesResumos);

FormGestaoFilmesResumos.Visible = true;

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORM\_GESTAO\_FILMES\_RESUMOS FormGestaoFilmesResumos = new FORM\_GESTAO\_FILMES\_RESUMOS();

FormGestaoFilmesResumos.TopLevel = false;

FormGestaoFilmesResumos.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmesResumos);

FormGestaoFilmesResumos.Visible = true;

return;

}

}

catch (Exception EX)

{ }

}

private void LISTVIEW\_GestaoFilmes\_FILMES\_SelectedIndexChanged(object sender, EventArgs e)

{

Preencher\_TextBox\_Filme();

if (LISTVIEW\_GestaoFilmes\_FILMES.SelectedItems == null)

PICTUREBOX\_GestaoFilmes\_PROCURAR\_RESUMO.Enabled = false;

else

PICTUREBOX\_GestaoFilmes\_PROCURAR\_RESUMO.Enabled = true;

}

private void BUTTON\_AdicionarFilme\_ADICIONAR\_FILME\_Click(object sender, EventArgs e)

{

Adicionar\_Filme();

ClearListView();

Preencher\_ListView\_Filmes();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Productores();

Preencher\_ListView\_Actores();

LimparTudo();

TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 0;

}

private void PICTUREBOX\_REMOVER\_FILME\_Click(object sender, EventArgs e)

{

Remover\_Filme();

ClearListView();

Preencher\_ListView\_Filmes();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Productores();

Preencher\_ListView\_Actores();

LimparTudo();

}

private void PICTUREBOX\_ALTERAR\_FILME\_Click(object sender, EventArgs e)

{

Alterar\_Filme();

ClearListView();

Preencher\_ListView\_Filmes();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Productores();

Preencher\_ListView\_Actores();

LimparTudo();

}

private void LISTVIEW\_GestaoFilmes\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

private void CHECKBOX\_GestaoFilmes\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoFilmes\_FILMES.ShowGroups = CHECKBOX\_GestaoFilmes\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

private void LISTVIEW\_GestaoFilmes\_ACTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Actores\_TabGestao();

}

private void CHECKBOX\_GestaoFilmes\_AGRUPAR\_ACTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoFilmes\_ACTORES.ShowGroups = CHECKBOX\_GestaoFilmes\_AGRUPAR\_ACTORES.Checked;

Ordenar\_Actores\_TabGestao();

}

private void LISTVIEW\_GestaoFilmes\_REALIZADORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Realizadores\_TabGestao();

}

private void CHECKBOX\_GestaoFilmes\_AGRUPAR\_REALIZADORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoFilmes\_REALIZADORES.ShowGroups = CHECKBOX\_GestaoFilmes\_AGRUPAR\_REALIZADORES.Checked;

Ordenar\_Realizadores\_TabGestao();

}

private void LISTVIEW\_GestaoFilmes\_PRODUCTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Productores\_TabGestao();

}

private void CHECKBOX\_GestaoFilmes\_AGRUPAR\_PRODUCTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoFilmes\_PRODUCTORES.ShowGroups = CHECKBOX\_GestaoFilmes\_AGRUPAR\_PRODUCTORES.Checked;

Ordenar\_Productores\_TabGestao();

}

private void LISTVIEW\_AdicionarFilme\_ACTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Actores\_TabAdicionar();

}

private void CHECKBOX\_AdicionarFilme\_AGRUPAR\_ACTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_AdicionarFilme\_ACTORES.ShowGroups = CHECKBOX\_AdicionarFilme\_AGRUPAR\_ACTORES.Checked;

Ordenar\_Actores\_TabAdicionar();

}

private void LISTVIEW\_AdicionarFilme\_REALIZADORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Realizadores\_TabAdicionar();

}

private void CHECKBOX\_AdicionarFilme\_AGRUPAR\_REALIZADORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_AdicionarFilme\_REALIZADORES.ShowGroups = CHECKBOX\_AdicionarFilme\_AGRUPAR\_REALIZADORES.Checked;

Ordenar\_Realizadores\_TabAdicionar();

}

private void LISTVIEW\_AdicionarFilme\_PRODUCTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Productores\_TabAdicionar();

}

private void CHECKBOX\_AdicionarFilme\_AGRUPAR\_PRODUCTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_AdicionarFilme\_PRODUCTORES.ShowGroups = CHECKBOX\_AdicionarFilme\_AGRUPAR\_PRODUCTORES.Checked;

Ordenar\_Productores\_TabAdicionar();

}

public void Preencher\_ListView\_Filmes()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes.ID," + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes.titulo," + CLASS\_GLOBAL\_VARIABLES.BD + ".estados\_filmes.estado FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".estados\_filmes ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes.estadoID = " + CLASS\_GLOBAL\_VARIABLES.BD + ".estados\_filmes.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["titulo"].ToString());

ROW.SubItems.Add(Reader["estado"].ToString());

LISTVIEW\_GestaoFilmes\_FILMES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Realizadores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

var ROW2 = new ListViewItem();

ROW2.Text = Reader["nome\_completo"].ToString();

ROW2.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoFilmes\_REALIZADORES.Items.Add(ROW);

LISTVIEW\_AdicionarFilme\_REALIZADORES.Items.Add(ROW2);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Productores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

var ROW2 = new ListViewItem();

ROW2.Text = Reader["nome\_completo"].ToString();

ROW2.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoFilmes\_PRODUCTORES.Items.Add(ROW);

LISTVIEW\_AdicionarFilme\_PRODUCTORES.Items.Add(ROW2);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Actores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

var ROW2 = new ListViewItem();

ROW2.Text = Reader["nome\_completo"].ToString();

ROW2.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoFilmes\_ACTORES.Items.Add(ROW);

LISTVIEW\_AdicionarFilme\_ACTORES.Items.Add(ROW2);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void ClearListView()

{

LISTVIEW\_GestaoFilmes\_FILMES.Items.Clear();

LISTVIEW\_GestaoFilmes\_ACTORES.Items.Clear();

LISTVIEW\_GestaoFilmes\_REALIZADORES.Items.Clear();

LISTVIEW\_GestaoFilmes\_PRODUCTORES.Items.Clear();

LISTVIEW\_AdicionarFilme\_ACTORES.Items.Clear();

LISTVIEW\_AdicionarFilme\_REALIZADORES.Items.Clear();

LISTVIEW\_AdicionarFilme\_PRODUCTORES.Items.Clear();

}

public void LimparTudo()

{

TEXTBOX\_AdicionarFilme\_TITULO.Text = "";

TEXTBOX\_AdicionarFilme\_DURACAO.Text = "";

TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = "";

TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = "";

RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = false;

RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = false;

RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = false;

TEXTBOX\_GestaoFilmes\_TITULO.Text = "";

TEXTBOX\_GestaoFilmes\_DURACAO.Text = "";

TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = "";

TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = "";

RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = false;

RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = false;

RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = false;

LISTVIEW\_GestaoFilmes\_ACTORES.SelectedItems.Clear();

LISTVIEW\_GestaoFilmes\_REALIZADORES.SelectedItems.Clear();

LISTVIEW\_GestaoFilmes\_PRODUCTORES.SelectedItems.Clear();

}

public void Preencher\_TextBox\_Filme()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_GestaoFilmes\_FILMES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

LigacaoDB.Open();

string Titulo;

string Duracao;

string Estado;

string NomeResumo;

string DuracaoResumo;

string Query\_Titulo = "SELECT titulo FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes WHERE ID = " + ID + "";

string Query\_Duracao = "SELECT duracao FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes WHERE ID = " + ID + "";

string Query\_Estado = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".estados\_filmes.estado FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".estados\_filmes ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes.estadoID = " + CLASS\_GLOBAL\_VARIABLES.BD + ".estados\_filmes.ID WHERE filmes.ID = " + ID + "";

string Query\_NomeResumo = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes.nome FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes.resumoID = " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes.ID WHERE filmes.ID = " + ID + "";

string Query\_DuracaoResumo = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes.duracao FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes.duracao\_resumo = " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes.ID WHERE filmes.ID = " + ID + "";

MySqlCommand Comando\_Titulo = new MySqlCommand(Query\_Titulo, LigacaoDB);

MySqlCommand Comando\_Duracao = new MySqlCommand(Query\_Duracao, LigacaoDB);

MySqlCommand Comando\_Estado = new MySqlCommand(Query\_Estado, LigacaoDB);

MySqlCommand Comando\_NomeResumo = new MySqlCommand(Query\_NomeResumo, LigacaoDB);

MySqlCommand Comando\_DuracaoResumo = new MySqlCommand(Query\_DuracaoResumo, LigacaoDB);

Titulo = Comando\_Titulo.ExecuteScalar().ToString();

Duracao = Comando\_Duracao.ExecuteScalar().ToString();

Estado = Comando\_Estado.ExecuteScalar().ToString();

NomeResumo = Comando\_NomeResumo.ExecuteScalar().ToString();

DuracaoResumo = Comando\_DuracaoResumo.ExecuteScalar().ToString();

TEXTBOX\_GestaoFilmes\_TITULO.Text = Titulo;

TEXTBOX\_GestaoFilmes\_DURACAO.Text = Duracao;

if (Estado == "Por Exibir")

RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = true;

if (Estado == "Em Exibição")

RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = true;

if (Estado == "Já Exibido")

RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = true;

TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = NomeResumo;

TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = DuracaoResumo;

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

LISTVIEW\_GestaoFilmes\_FILMES.SelectedItems.Clear();

TEXTBOX\_GestaoFilmes\_TITULO.Text = "";

TEXTBOX\_GestaoFilmes\_DURACAO.Text = "";

RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = false;

RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = false;

RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = false;

TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = "";

TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = "";

}

}

public void Adicionar\_Filme()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Titulo = TEXTBOX\_AdicionarFilme\_TITULO.Text;

string Duracao = TEXTBOX\_AdicionarFilme\_DURACAO.Text;

string NomeResumo = TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text;

int ResumoID;

int Estado = 0;

if (RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked == true)

Estado = 1;

if (RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked == true)

Estado = 2;

if (RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked == true)

Estado = 3;

string Query\_ObterResumoID = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes WHERE nome ='" + NomeResumo + "'";

MySqlCommand Comando\_ObterResumoID = new MySqlCommand(Query\_ObterResumoID, LigacaoDB);

ResumoID = Convert.ToInt32(Comando\_ObterResumoID.ExecuteScalar());

LigacaoDB.Close();

System.Threading.Thread.Sleep(1000);

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_InserirFilme = new MySqlCommand("SP\_FILME\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_InserirFilme.CommandType = CommandType.StoredProcedure;

Comando\_InserirFilme.Parameters.AddWithValue("\_titulo", Titulo);

Comando\_InserirFilme.Parameters.AddWithValue("\_duracao", Duracao);

Comando\_InserirFilme.Parameters.AddWithValue("\_resumo", ResumoID);

Comando\_InserirFilme.Parameters.AddWithValue("\_duracao\_resumo", ResumoID);

Comando\_InserirFilme.Parameters.AddWithValue("\_estado", Estado);

Reader = Comando\_InserirFilme.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Filme Inserido com Sucesso", "Filme Inserido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Remover\_Filme()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoFilmes\_FILMES.SelectedItems[0].Text;

string Resumo = TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text;

int ID = Convert.ToInt32(ListView\_ID);

MySqlCommand Command\_RemoverFilme = new MySqlCommand("SP\_FILME\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_RemoverFilme.CommandType = CommandType.StoredProcedure;

Command\_RemoverFilme.Parameters.AddWithValue("\_id", ID);

Reader = Command\_RemoverFilme.ExecuteReader();

Reader.Close();

DialogResult RemoverResumo = MessageBox.Show("Deseja remover o resumo associado a este filme?","Remover Resumo?",MessageBoxButtons.YesNo,MessageBoxIcon.Question);

if(RemoverResumo == DialogResult.Yes)

{

MySqlCommand Command\_RemoverResumoFilme = new MySqlCommand("SP\_ResumoFilme\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_RemoverResumoFilme.CommandType = CommandType.StoredProcedure;

Command\_RemoverResumoFilme.Parameters.AddWithValue("\_nome", Resumo);

Reader = Command\_RemoverResumoFilme.ExecuteReader();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Resumo removido com sucesso", "Resumo Removido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

if(LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Filme removido com sucesso", "Filme Removido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Alterar\_Filme()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string ListView\_ID = LISTVIEW\_GestaoFilmes\_FILMES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

string Titulo = TEXTBOX\_GestaoFilmes\_TITULO.Text;

string Duracao = TEXTBOX\_GestaoFilmes\_DURACAO.Text;

string NomeResumo = TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text;

int ResumoID;

int Estado = 0;

if (RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked == true)

Estado = 1;

if (RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked == true)

Estado = 2;

if (RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked == true)

Estado = 3;

string Query\_ObterResumoID = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes WHERE nome ='" + NomeResumo + "'";

MySqlCommand Comando\_ObterResumoID = new MySqlCommand(Query\_ObterResumoID, LigacaoDB);

ResumoID = Convert.ToInt32(Comando\_ObterResumoID.ExecuteScalar());

Reader.Close();

LigacaoDB.Close();

System.Threading.Thread.Sleep(1000);

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_AlterarFilme = new MySqlCommand("SP\_FILME\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando\_AlterarFilme.CommandType = CommandType.StoredProcedure;

Comando\_AlterarFilme.Parameters.AddWithValue("\_id", ID);

Comando\_AlterarFilme.Parameters.AddWithValue("\_titulo", Titulo);

Comando\_AlterarFilme.Parameters.AddWithValue("\_duracao", Duracao);

Comando\_AlterarFilme.Parameters.AddWithValue("\_resumo", ResumoID);

Comando\_AlterarFilme.Parameters.AddWithValue("\_duracao\_resumo", ResumoID);

Comando\_AlterarFilme.Parameters.AddWithValue("\_estado", Estado);

Reader = Comando\_AlterarFilme.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Filme Alterado com Sucesso", "Filme Alterado", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Ordenar\_Filmes()

{

if (CHECKBOX\_GestaoFilmes\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_GestaoFilmes\_FILMES.ShowGroups)

BuildGroups\_Filmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoFilmes\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Realizadores\_TabGestao()

{

if (CHECKBOX\_GestaoFilmes\_AGRUPAR\_REALIZADORES.Checked == true)

{

if (LISTVIEW\_GestaoFilmes\_REALIZADORES.ShowGroups)

BuildGroups\_Realizadores\_TabGestao(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoFilmes\_REALIZADORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Actores\_TabGestao()

{

if (CHECKBOX\_GestaoFilmes\_AGRUPAR\_ACTORES.Checked == true)

{

if (LISTVIEW\_GestaoFilmes\_ACTORES.ShowGroups)

BuildGroups\_Actores\_TabGestao(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoFilmes\_ACTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Productores\_TabGestao()

{

if (CHECKBOX\_GestaoFilmes\_AGRUPAR\_PRODUCTORES.Checked == true)

{

if (LISTVIEW\_GestaoFilmes\_PRODUCTORES.ShowGroups)

BuildGroups\_Productores\_TabGestao(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoFilmes\_PRODUCTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Realizadores\_TabAdicionar()

{

if (CHECKBOX\_AdicionarFilme\_AGRUPAR\_REALIZADORES.Checked == true)

{

if (LISTVIEW\_AdicionarFilme\_REALIZADORES.ShowGroups)

BuildGroups\_Realizadores\_TabAdicionar(UltimaColunaOrdenada);

}

else

LISTVIEW\_AdicionarFilme\_REALIZADORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Actores\_TabAdicionar()

{

if (CHECKBOX\_AdicionarFilme\_AGRUPAR\_ACTORES.Checked == true)

{

if (LISTVIEW\_AdicionarFilme\_ACTORES.ShowGroups)

BuildGroups\_Actores\_TabAdicionar(UltimaColunaOrdenada);

}

else

LISTVIEW\_AdicionarFilme\_ACTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Productores\_TabAdicionar()

{

if (CHECKBOX\_AdicionarFilme\_AGRUPAR\_PRODUCTORES.Checked == true)

{

if (LISTVIEW\_AdicionarFilme\_PRODUCTORES.ShowGroups)

BuildGroups\_Productores\_TabAdicionar(UltimaColunaOrdenada);

}

else

LISTVIEW\_AdicionarFilme\_PRODUCTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups\_Filmes(int coluna)

{

LISTVIEW\_GestaoFilmes\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoFilmes\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoFilmes\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoFilmes\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Realizadores\_TabGestao(int coluna)

{

LISTVIEW\_GestaoFilmes\_REALIZADORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoFilmes\_REALIZADORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoFilmes\_REALIZADORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoFilmes\_REALIZADORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Actores\_TabGestao(int coluna)

{

LISTVIEW\_GestaoFilmes\_ACTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoFilmes\_ACTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoFilmes\_ACTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoFilmes\_ACTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Productores\_TabGestao(int coluna)

{

LISTVIEW\_GestaoFilmes\_PRODUCTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoFilmes\_PRODUCTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoFilmes\_PRODUCTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoFilmes\_PRODUCTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Realizadores\_TabAdicionar(int coluna)

{

LISTVIEW\_AdicionarFilme\_REALIZADORES.Groups.Clear();

int Contagem = LISTVIEW\_AdicionarFilme\_REALIZADORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_AdicionarFilme\_REALIZADORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_AdicionarFilme\_REALIZADORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Actores\_TabAdicionar(int coluna)

{

LISTVIEW\_AdicionarFilme\_ACTORES.Groups.Clear();

int Contagem = LISTVIEW\_AdicionarFilme\_ACTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_AdicionarFilme\_ACTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_AdicionarFilme\_ACTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Productores\_TabAdicionar(int coluna)

{

LISTVIEW\_AdicionarFilme\_PRODUCTORES.Groups.Clear();

int Contagem = LISTVIEW\_AdicionarFilme\_PRODUCTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_AdicionarFilme\_PRODUCTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_AdicionarFilme\_PRODUCTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}

* **[I] Form de Gestão de Resumos**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO\_FILMES\_RESUMOS : Form

{

FORM\_GESTAO FormGestao = (FORM\_GESTAO)Application.OpenForms["FORM\_GESTAO"];

FORM\_GESTAO\_FILMES FormGestaoFilmes = (FORM\_GESTAO\_FILMES)Application.OpenForms["FORM\_GESTAO\_FILMES"];

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_GESTAO\_FILMES\_RESUMOS()

{

InitializeComponent();

Preencher\_ListView\_Resumos();

Preencher\_ListView\_Filmes();

}

private void PICTUREBOX\_GestaoResumos\_RETROCEDER\_Click(object sender, EventArgs e)

{

if(CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "GestaoFilmes")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

if (CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "AdicionarFilme")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

}

private void BUTTON\_GestaoResumos\_SELECIONAR\_Click(object sender, EventArgs e)

{

string ConfirmarExistencia = TEXTBOX\_GestaoResumos\_RESUMO.Text;

string ConfirmarDuracao = TEXTBOX\_GestaoResumos\_DURACAO.Text;

bool Existe = false;;

bool ResumoInserido = false;

bool EfectuarCopia = false;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems[0].SubItems[1].ToString() && TEXTBOX\_GestaoResumos\_DURACAO.Text == LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems[0].SubItems[2].ToString())

{

if (CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "GestaoFilmes")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

return;

}

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

return;

}

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

if (CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "AdicionarFilme")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

return;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems[0].SubItems[1].ToString() || TEXTBOX\_GestaoResumos\_DURACAO.Text != LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems[0].SubItems[2].ToString())

{

foreach (ListViewItem ItemExiste in LISTVIEW\_GestaoResumos\_RESUMOS.Items)

{

for (int i = 0; i < ItemExiste.SubItems.Count; i++)

{

if (ItemExiste.SubItems[i].Text.ToLower() == ConfirmarExistencia.ToLower() && ItemExiste.SubItems[2].Text == ConfirmarDuracao)

{

Existe = true;

}

}

}

if (Existe == true)

{

if (CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "GestaoFilmes")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

return;

}

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

return;

}

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

if (CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "AdicionarFilme")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

return;

}

if (Existe == false)

{

DialogResult NovoResumo = MessageBox.Show("O resumo que indicou ainda não existe. Pretende Adiciona-lo?", "Novo Resumo?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (NovoResumo == DialogResult.Yes)

{

bool JaExiste = false;

foreach (ListViewItem ItemExiste in LISTVIEW\_GestaoResumos\_RESUMOS.Items)

{

for (int i = 0; i < ItemExiste.SubItems.Count; i++)

{

if (ItemExiste.SubItems[i].Text.ToLower() == ConfirmarExistencia.ToLower())

{

MessageBox.Show("O resumo que está a tentar inserir já existe. Não é possivel adicionar resumos com o mesmo nome", "Resumo Já Existe", MessageBoxButtons.OK, MessageBoxIcon.Error);

JaExiste = true;

return;

}

}

}

if (JaExiste == false)

{

Adicionar\_Resumo();

ClearListView();

Preencher\_ListView\_Resumos();

Preencher\_ListView\_Filmes();

DialogResult Copiar = MessageBox.Show("Resumo Inserido com Sucesso. Deseja continuar a cópia?", "Continuar Cópia", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (Copiar == DialogResult.Yes)

{

EfectuarCopia = true;

}

if (Copiar == DialogResult.No)

{

EfectuarCopia = false;

LimparTudo();

}

}

}

if (NovoResumo == DialogResult.No)

{

MessageBox.Show("Não é possivel copiar um resumo que não tenha sido inserido. Terá de adiciona-lo para que possa copiar.","Impossivel Copiar!",MessageBoxButtons.OK,MessageBoxIcon.Error);

return;

}

if(EfectuarCopia == true)

{

if (CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "GestaoFilmes")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

return;

}

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Focus();

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Selected = true;

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.EnsureVisible(CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID);

FormGestaoFilmes.LISTVIEW\_GestaoFilmes\_FILMES.Items[CLASS\_GLOBAL\_VARIABLES.INT\_AUX\_RESUMO\_ID].Focused = true;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

return;

}

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_GestaoFilmes\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_GestaoFilmes\_JA\_EXIBIDO.Checked = true;

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

if (CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource == "AdicionarFilme")

{

try

{

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 0)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

if (FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count == 1 || FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Count > 1)

{

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_FILMES FormGestaoFilmes = new FORM\_GESTAO\_FILMES();

FormGestaoFilmes.TopLevel = false;

FormGestaoFilmes.AutoScroll = true;

FormGestao.SPLITCONTAINER\_GESTAO.Panel2.Controls.Add(FormGestaoFilmes);

FormGestaoFilmes.Visible = true;

FormGestaoFilmes.TABCONTROL\_GESTAO\_FILMES.SelectedIndex = 1;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = TEXTBOX\_GestaoResumos\_RESUMO.Text;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = TEXTBOX\_GestaoResumos\_DURACAO.Text;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" && TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_TITULO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Titulo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Duracao;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_NOME\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Resumo;

FormGestaoFilmes.TEXTBOX\_AdicionarFilme\_DURACAO\_RESUMO.Text = CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_DuracaoResumo;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Por Exibir")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_POR\_EXIBIR.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Em Exibicao")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_EM\_EXIBICAO.Checked = true;

if (CLASS\_GLOBAL\_VARIABLES.AUX\_RESUMO\_Estado == "Ja Exibido")

FormGestaoFilmes.RADIOBUTTON\_AdicionarFilme\_JA\_EXIBIDO.Checked = true;

}

CLASS\_GLOBAL\_VARIABLES.GestaoResumos\_RequestSource = "";

return;

}

}

catch (Exception EX)

{ }

}

}

}

}

}

private void TEXTBOX\_GestaoResumos\_RESUMO\_TextChanged(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

BUTTON\_GestaoResumos\_SELECIONAR.Enabled = true;

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "")

BUTTON\_GestaoResumos\_SELECIONAR.Enabled = false;

if (LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems == null)

BUTTON\_GestaoResumos\_SELECIONAR.Enabled = false;

}

private void TEXTBOX\_GestaoResumos\_DURACAO\_TextChanged(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

BUTTON\_GestaoResumos\_SELECIONAR.Enabled = true;

if (TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

BUTTON\_GestaoResumos\_SELECIONAR.Enabled = false;

if (LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems == null)

BUTTON\_GestaoResumos\_SELECIONAR.Enabled = false;

}

private void LISTVIEW\_GestaoResumos\_RESUMOS\_SelectedIndexChanged(object sender, EventArgs e)

{

if (LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems == null)

BUTTON\_GestaoResumos\_SELECIONAR.Enabled = false;

Preencher\_TextBox\_Resumo();

}

public void Preencher\_ListView\_Resumos()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT ID,nome,duracao FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["nome"].ToString());

ROW.SubItems.Add(Reader["duracao"].ToString());

LISTVIEW\_GestaoResumos\_RESUMOS.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Filmes()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Command = new MySqlCommand("SP\_FILMES\_SELECIONAR", LigacaoDB\_StoredProcedures);

Command.CommandType = CommandType.StoredProcedure;

Reader = Command.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["titulo"].ToString());

ROW.SubItems.Add(Reader["duracao"].ToString());

ROW.SubItems.Add(Reader["nome"].ToString());

ROW.SubItems.Add(Reader["duracao\_resumo"].ToString());

ROW.SubItems.Add(Reader["estado"].ToString());

LISTVIEW\_GestaoResumos\_FILMES.Items.Add(ROW);

}

LigacaoDB\_StoredProcedures.Close();

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Preencher\_TextBox\_Resumo()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

LigacaoDB.Open();

string Resumo;

string Duracao;

string Query\_Resumo = "SELECT nome FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes WHERE ID = " + ID + "";

string Query\_Duracao = "SELECT duracao FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".resumos\_filmes WHERE ID = " + ID + "";

MySqlCommand Comando\_Resumo = new MySqlCommand(Query\_Resumo, LigacaoDB);

MySqlCommand Comando\_Duracao = new MySqlCommand(Query\_Duracao, LigacaoDB);

Resumo = Comando\_Resumo.ExecuteScalar().ToString();

Duracao = Comando\_Duracao.ExecuteScalar().ToString();

TEXTBOX\_GestaoResumos\_RESUMO.Text = Resumo;

TEXTBOX\_GestaoResumos\_DURACAO.Text = Duracao;

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems.Clear();

TEXTBOX\_GestaoResumos\_RESUMO.Text = "";

TEXTBOX\_GestaoResumos\_DURACAO.Text = "";

}

}

public void ClearListView()

{

LISTVIEW\_GestaoResumos\_RESUMOS.Items.Clear();

LISTVIEW\_GestaoResumos\_FILMES.Items.Clear();

}

public void LimparTudo()

{

TEXTBOX\_GestaoResumos\_RESUMO.Text = "";

TEXTBOX\_GestaoResumos\_DURACAO.Text = "";

}

public void Adicionar\_Resumo()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string NomeResumo = TEXTBOX\_GestaoResumos\_RESUMO.Text;

string DuracaoResumo = TEXTBOX\_GestaoResumos\_DURACAO.Text;

MySqlCommand Comando\_InserirResumo = new MySqlCommand("SP\_RESUMO\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_InserirResumo.CommandType = CommandType.StoredProcedure;

Comando\_InserirResumo.Parameters.AddWithValue("\_nome", NomeResumo);

Comando\_InserirResumo.Parameters.AddWithValue("\_duracao", DuracaoResumo);

Reader = Comando\_InserirResumo.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Resumo inserido com Sucesso", "Resumo Inserido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Remover\_Resumo()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

MySqlCommand Command\_RemoverResumo = new MySqlCommand("SP\_RESUMO\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_RemoverResumo.CommandType = CommandType.StoredProcedure;

Command\_RemoverResumo.Parameters.AddWithValue("\_id", ID);

Reader = Command\_RemoverResumo.ExecuteReader();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Resumo removido com sucesso", "Resumo Removido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Alterar\_Resumo()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

string NomeResumo = TEXTBOX\_GestaoResumos\_RESUMO.Text;

string DuracaoResumo = TEXTBOX\_GestaoResumos\_DURACAO.Text;

MySqlCommand Comando\_AlterarResumo = new MySqlCommand("SP\_RESUMO\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando\_AlterarResumo.CommandType = CommandType.StoredProcedure;

Comando\_AlterarResumo.Parameters.AddWithValue("\_ID", ID);

Comando\_AlterarResumo.Parameters.AddWithValue("\_nome", NomeResumo);

Comando\_AlterarResumo.Parameters.AddWithValue("\_duracao", DuracaoResumo);

Reader = Comando\_AlterarResumo.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Resumo Alterado com Sucesso", "Resumo Alterado", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

private void PICTUREBOX\_GestaoResumos\_ADICIONAR\_RESUMO\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoResumos\_RESUMO.Text != "" && TEXTBOX\_GestaoResumos\_DURACAO.Text != "")

{

string ConfirmarExistencia = TEXTBOX\_GestaoResumos\_RESUMO.Text;

bool JaExiste = false;

foreach(ListViewItem ItemExiste in LISTVIEW\_GestaoResumos\_RESUMOS.Items)

{

for (int i = 0; i < ItemExiste.SubItems.Count; i++)

{

if (ItemExiste.SubItems[i].Text.ToLower() == ConfirmarExistencia.ToLower())

{

MessageBox.Show("O resumo que está a tentar inserir já existe. Não é possivel adicionar resumos com o mesmo nome", "Resumo Já Existe", MessageBoxButtons.OK, MessageBoxIcon.Error);

JaExiste = true;

}

}

}

if (JaExiste == false)

{

Adicionar\_Resumo();

ClearListView();

Preencher\_ListView\_Resumos();

Preencher\_ListView\_Filmes();

LimparTudo();

}

}

else

{

MessageBox.Show("Os campos não estão totalmente preenchidos.", "Campos Inválidos", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

}

private void PICTUREBOX\_GestaoResumos\_GRAVAR\_ALTERACOES\_Click(object sender, EventArgs e)

{

if(LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems == null)

{

MessageBox.Show("Não tem nenhum resumo selecionado! Não é possivel alterar.", "Nenhum Resumo Selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems != null)

{

if (TEXTBOX\_GestaoResumos\_RESUMO.Text == "" || TEXTBOX\_GestaoResumos\_DURACAO.Text == "")

{

MessageBox.Show("Não tem nenhum resumo selecionado! Não é possivel alterar.", "Nenhum Resumo Selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

else

{

DialogResult AlterarResumo = MessageBox.Show("Tem a certeza que deseja alterar este resumo?", "Alterar Resumo?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (AlterarResumo == DialogResult.Yes)

{

Alterar\_Resumo();

ClearListView();

Preencher\_ListView\_Resumos();

Preencher\_ListView\_Filmes();

LimparTudo();

}

}

}

}

private void PICTUREBOX\_GestaoResumos\_REMOVER\_RESUMO\_Click(object sender, EventArgs e)

{

if (LISTVIEW\_GestaoResumos\_RESUMOS.SelectedItems == null)

{

MessageBox.Show("Não tem nenhum resumo selecionado! Não é possivel remover.", "Nenhum Resumo Selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

else

{

DialogResult RemoverResumo = MessageBox.Show("Tem a certeza que deseja remover este resumo?", "Remover Resumo?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (RemoverResumo == DialogResult.Yes)

{

Remover\_Resumo();

ClearListView();

Preencher\_ListView\_Resumos();

Preencher\_ListView\_Filmes();

LimparTudo();

}

}

}

private void CHECKBOX\_GestaoResumos\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoResumos\_FILMES.ShowGroups = CHECKBOX\_GestaoResumos\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

private void LISTVIEW\_GestaoResumos\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

public void Ordenar\_Filmes()

{

if (CHECKBOX\_GestaoResumos\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_GestaoResumos\_FILMES.ShowGroups)

BuildGroups\_Filmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoResumos\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups\_Filmes(int coluna)

{

LISTVIEW\_GestaoResumos\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoResumos\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoResumos\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoResumos\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}

* **[J] Form de Gestão de Nacionalidades**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO\_NACIONALIDADES : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_GESTAO\_NACIONALIDADES()

{

InitializeComponent();

Preencher\_ListView\_Nacionalidades();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Actores();

Preencher\_ListView\_Productores();

}

private void LISTVIEW\_GestaoNacionalidades\_NACIONALIDADES\_SelectedIndexChanged(object sender, EventArgs e)

{

Preencher\_TextBox\_Nacionalidade();

}

private void BUTTON\_AdicionarNacionalidade\_ADICIONAR\_NACIONALIDADE\_Click(object sender, EventArgs e)

{

Adicionar\_Nacionalidade();

ClearListView();

Preencher\_ListView\_Nacionalidades();

LimparTudo();

TABCONTROL\_GestaoNacionalidades.SelectedIndex = 0;

}

private void PICTUREBOX\_GestaoNacionalidades\_ADICIONAR\_NACIONALIDADE\_Click(object sender, EventArgs e)

{

TABCONTROL\_GestaoNacionalidades.SelectedIndex = 1;

}

private void PICTUREBOX\_GestaoNacionalidades\_GRAVAR\_ALTERACOES\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoNacionalidades\_ID.Text == "" && TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text == "")

{

MessageBox.Show("Não tem nenhuma nacionalidade selecionada! Não é possivel alterar.", "Nenhuma Nacionalidade Selecionada", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text == "" && TEXTBOX\_GestaoNacionalidades\_ID.Text != "")

{

MessageBox.Show("O campo Nacionalidade está vazio. Não é possivel alterar.", "Campo Nacionalidade está Vazio", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

else

{

DialogResult AlterarNacionalidade = MessageBox.Show("Tem a certeza que deseja alterar esta nacionalidade?", "Alterar Nacionalidade?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (AlterarNacionalidade == DialogResult.Yes)

{

Alterar\_Nacionalidade();

ClearListView();

Preencher\_ListView\_Nacionalidades();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Actores();

Preencher\_ListView\_Productores();

LimparTudo();

}

}

}

private void PICTUREBOX\_GestaoNacionalidades\_REMOVER\_NACIONALIDADE\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoNacionalidades\_ID.Text == "" && TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text == "")

{

MessageBox.Show("Não tem nenhuma nacionalidade selecionada! Não é possivel remover.", "Nenhuma Nacionalidade Selecionada", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

else

{

DialogResult RemoverNacionalidade = MessageBox.Show("Tem a certeza que deseja remover esta nacionalidade?", "Remover Nacionalidade?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (RemoverNacionalidade == DialogResult.Yes)

{

Remover\_Nacionalidade();

ClearListView();

Preencher\_ListView\_Nacionalidades();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Actores();

Preencher\_ListView\_Productores();

LimparTudo();

}

}

}

private void LISTVIEW\_GestaoNacionalidades\_REALIZADORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Realizadores();

}

private void CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_REALIZADORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoNacionalidades\_REALIZADORES.ShowGroups = CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_REALIZADORES.Checked;

Ordenar\_Realizadores();

}

private void LISTVIEW\_GestaoNacionalidades\_ACTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Actores();

}

private void CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_ACTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoNacionalidades\_ACTORES.ShowGroups = CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_ACTORES.Checked;

Ordenar\_Actores();

}

private void LISTVIEW\_GestaoNacionalidades\_PRODUCTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Productores();

}

private void CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_PRODUCTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.ShowGroups = CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_PRODUCTORES.Checked;

Ordenar\_Productores();

}

private void TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text = Regex.Replace(TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text, @"[0-9]", string.Empty);

TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text = Regex.Replace(TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

private void TEXTBOX\_AdicionarNacionalidade\_NACIONALIDADE\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_AdicionarNacionalidade\_NACIONALIDADE.Text = Regex.Replace(TEXTBOX\_AdicionarNacionalidade\_NACIONALIDADE.Text, @"[0-9]", string.Empty);

TEXTBOX\_AdicionarNacionalidade\_NACIONALIDADE.Text = Regex.Replace(TEXTBOX\_AdicionarNacionalidade\_NACIONALIDADE.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

public void Preencher\_ListView\_Nacionalidades()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT ID,nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoNacionalidades\_NACIONALIDADES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Realizadores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoNacionalidades\_REALIZADORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Actores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoNacionalidades\_ACTORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Productores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void ClearListView()

{

LISTVIEW\_GestaoNacionalidades\_NACIONALIDADES.Items.Clear();

LISTVIEW\_GestaoNacionalidades\_REALIZADORES.Items.Clear();

LISTVIEW\_GestaoNacionalidades\_ACTORES.Items.Clear();

LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.Items.Clear();

}

public void LimparTudo()

{

TEXTBOX\_AdicionarNacionalidade\_NACIONALIDADE.Text = "";

TEXTBOX\_GestaoNacionalidades\_ID.Text = "";

TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text = "";

}

public void Preencher\_TextBox\_Nacionalidade()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_GestaoNacionalidades\_NACIONALIDADES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

LigacaoDB.Open();

string IDNacionalidade;

string Nacionalidade;

string Query\_ID = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades WHERE ID = " + ID + "";

string Query\_Nacionalidade = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades";

MySqlCommand Comando\_ID = new MySqlCommand(Query\_ID, LigacaoDB);

MySqlCommand Comando\_Nacionalidade = new MySqlCommand(Query\_Nacionalidade, LigacaoDB);

IDNacionalidade = Comando\_ID.ExecuteScalar().ToString();

Nacionalidade = Comando\_Nacionalidade.ExecuteScalar().ToString();

TEXTBOX\_GestaoNacionalidades\_ID.Text = IDNacionalidade;

TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text = Nacionalidade;

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

LISTVIEW\_GestaoNacionalidades\_NACIONALIDADES.SelectedItems.Clear();

TEXTBOX\_GestaoNacionalidades\_ID.Text = "";

TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text = "";

}

}

public void Adicionar\_Nacionalidade()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string Nacionalidade = TEXTBOX\_AdicionarNacionalidade\_NACIONALIDADE.Text;

MySqlCommand Comando\_InserirNacionalidade = new MySqlCommand("SP\_NACIONALIDADE\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_InserirNacionalidade.CommandType = CommandType.StoredProcedure;

Comando\_InserirNacionalidade.Parameters.AddWithValue("\_nacionalidade", Nacionalidade);

Reader = Comando\_InserirNacionalidade.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Nacionalidade Inserida com Sucesso", "Nacionalidade Inserida", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Remover\_Nacionalidade()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoNacionalidades\_NACIONALIDADES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

MySqlCommand Command\_RemoverNacionalidade = new MySqlCommand("SP\_NACIONALIDADE\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_RemoverNacionalidade.CommandType = CommandType.StoredProcedure;

Command\_RemoverNacionalidade.Parameters.AddWithValue("\_id", ID);

Reader = Command\_RemoverNacionalidade.ExecuteReader();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Nacionalidade removida com sucesso", "Nacionalidade Removida", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Alterar\_Nacionalidade()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoNacionalidades\_NACIONALIDADES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

string Nacionalidade = TEXTBOX\_GestaoNacionalidades\_NACIONALIDADE.Text;

MySqlCommand Comando\_AlterarNacionalidade = new MySqlCommand("SP\_NACIONALIDADE\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando\_AlterarNacionalidade.CommandType = CommandType.StoredProcedure;

Comando\_AlterarNacionalidade.Parameters.AddWithValue("\_id", ID);

Comando\_AlterarNacionalidade.Parameters.AddWithValue("\_nacionalidade", Nacionalidade);

Reader = Comando\_AlterarNacionalidade.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Nacionalidade alterada com Sucesso", "Nacionalidade Alterada", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Ordenar\_Realizadores()

{

if (CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_REALIZADORES.Checked == true)

{

if (LISTVIEW\_GestaoNacionalidades\_REALIZADORES.ShowGroups)

BuildGroups\_Realizadores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoNacionalidades\_REALIZADORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Actores()

{

if (CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_ACTORES.Checked == true)

{

if (LISTVIEW\_GestaoNacionalidades\_ACTORES.ShowGroups)

BuildGroups\_Actores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoNacionalidades\_ACTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Productores()

{

if (CHECKBOX\_GestaoNacionalidades\_AGRUPAR\_PRODUCTORES.Checked == true)

{

if (LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.ShowGroups)

BuildGroups\_Productores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups\_Realizadores(int coluna)

{

LISTVIEW\_GestaoNacionalidades\_REALIZADORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoNacionalidades\_REALIZADORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoNacionalidades\_REALIZADORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoNacionalidades\_REALIZADORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Actores(int coluna)

{

LISTVIEW\_GestaoNacionalidades\_ACTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoNacionalidades\_ACTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoNacionalidades\_ACTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoNacionalidades\_ACTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Productores(int coluna)

{

LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoNacionalidades\_PRODUCTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}

* **[L] Form do Painel de Intervenientes**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO\_PESSOAS : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_GESTAO\_PESSOAS()

{

InitializeComponent();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Actores();

Preencher\_ListView\_Productores();

Preencher\_ListView\_Nacionalidade();

}

private void PICTUREBOX\_GESTAO\_PESSOAS\_REALIZADORES\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS\_REALIZADORES FormGestaoRealizadores = new FORM\_GESTAO\_PESSOAS\_REALIZADORES();

FormGestaoRealizadores.TopLevel = false;

FormGestaoRealizadores.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoRealizadores);

FormGestaoRealizadores.Visible = true;

return;

}

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 1 || SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS\_REALIZADORES FormGestaoRealizadores = new FORM\_GESTAO\_PESSOAS\_REALIZADORES();

FormGestaoRealizadores.TopLevel = false;

FormGestaoRealizadores.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoRealizadores);

FormGestaoRealizadores.Visible = true;

return;

}

}

private void PICTUREBOX\_GESTAO\_PESSOAS\_ACTORES\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS\_ACTORES FormGestaoActores = new FORM\_GESTAO\_PESSOAS\_ACTORES();

FormGestaoActores.TopLevel = false;

FormGestaoActores.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoActores);

FormGestaoActores.Visible = true;

return;

}

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 1 || SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS\_ACTORES FormGestaoActores = new FORM\_GESTAO\_PESSOAS\_ACTORES();

FormGestaoActores.TopLevel = false;

FormGestaoActores.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoActores);

FormGestaoActores.Visible = true;

return;

}

}

private void PICTUREBOX\_GESTAO\_PESSOAS\_PRODUCTORES\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS\_PRODUCTORES FormGestaoProductores = new FORM\_GESTAO\_PESSOAS\_PRODUCTORES();

FormGestaoProductores.TopLevel = false;

FormGestaoProductores.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoProductores);

FormGestaoProductores.Visible = true;

return;

}

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 1 || SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_PESSOAS\_PRODUCTORES FormGestaoProductores = new FORM\_GESTAO\_PESSOAS\_PRODUCTORES();

FormGestaoProductores.TopLevel = false;

FormGestaoProductores.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoProductores);

FormGestaoProductores.Visible = true;

return;

}

}

private void PICTUREBOX\_GESTAO\_PESSOAS\_NACIONALIDADES\_Click(object sender, EventArgs e)

{

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 0)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_NACIONALIDADES FormGestaoNacionalidades = new FORM\_GESTAO\_NACIONALIDADES();

FormGestaoNacionalidades.TopLevel = false;

FormGestaoNacionalidades.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoNacionalidades);

FormGestaoNacionalidades.Visible = true;

return;

}

if (SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count == 1 || SPLITCONTAINER\_PESSOAS.Panel2.Controls.Count > 1)

{

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Clear();

FORMS\_GESTAO.FORM\_GESTAO\_NACIONALIDADES FormGestaoNacionalidades = new FORM\_GESTAO\_NACIONALIDADES();

FormGestaoNacionalidades.TopLevel = false;

FormGestaoNacionalidades.AutoScroll = true;

SPLITCONTAINER\_PESSOAS.Panel2.Controls.Add(FormGestaoNacionalidades);

FormGestaoNacionalidades.Visible = true;

return;

}

}

private void LISTVIEW\_GestaoPessoas\_REALIZADORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Realizadores();

}

private void CHECKBOX\_GestaoPessoas\_AGRUPAR\_REALIZADORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoPessoas\_REALIZADORES.ShowGroups = CHECKBOX\_GestaoPessoas\_AGRUPAR\_REALIZADORES.Checked;

Ordenar\_Realizadores();

}

private void LISTVIEW\_GestaoPessoas\_ACTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Actores();

}

private void CHECKBOX\_GestaoPessoas\_AGRUPAR\_ACTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoPessoas\_ACTORES.ShowGroups = CHECKBOX\_GestaoPessoas\_AGRUPAR\_ACTORES.Checked;

Ordenar\_Actores();

}

private void LISTVIEW\_GestaoPessoas\_PRODUCTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Productores();

}

private void CHECKBOX\_GestaoPessoas\_AGRUPAR\_PRODUCTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoPessoas\_PRODUCTORES.ShowGroups = CHECKBOX\_GestaoPessoas\_AGRUPAR\_PRODUCTORES.Checked;

Ordenar\_Productores();

}

public void Preencher\_ListView\_Realizadores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoPessoas\_REALIZADORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Actores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoPessoas\_ACTORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Productores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.nome\_completo," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nome\_completo"].ToString();

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoPessoas\_PRODUCTORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Nacionalidade()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["nacionalidade"].ToString();

LISTVIEW\_GestaoPessoas\_NACIONALIDADES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Ordenar\_Realizadores()

{

if (CHECKBOX\_GestaoPessoas\_AGRUPAR\_REALIZADORES.Checked == true)

{

if (LISTVIEW\_GestaoPessoas\_REALIZADORES.ShowGroups)

BuildGroups\_Realizadores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoPessoas\_REALIZADORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Actores()

{

if (CHECKBOX\_GestaoPessoas\_AGRUPAR\_ACTORES.Checked == true)

{

if (LISTVIEW\_GestaoPessoas\_ACTORES.ShowGroups)

BuildGroups\_Actores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoPessoas\_ACTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Productores()

{

if (CHECKBOX\_GestaoPessoas\_AGRUPAR\_PRODUCTORES.Checked == true)

{

if (LISTVIEW\_GestaoPessoas\_PRODUCTORES.ShowGroups)

BuildGroups\_Productores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoPessoas\_PRODUCTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups\_Realizadores(int coluna)

{

LISTVIEW\_GestaoPessoas\_REALIZADORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoPessoas\_REALIZADORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoPessoas\_REALIZADORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoPessoas\_REALIZADORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Actores(int coluna)

{

LISTVIEW\_GestaoPessoas\_ACTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoPessoas\_ACTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoPessoas\_ACTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoPessoas\_ACTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_Productores(int coluna)

{

LISTVIEW\_GestaoPessoas\_PRODUCTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoPessoas\_PRODUCTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoPessoas\_PRODUCTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoPessoas\_PRODUCTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}

* **[M] Form de Gestão de Actores**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO\_PESSOAS\_ACTORES : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_GESTAO\_PESSOAS\_ACTORES()

{

InitializeComponent();

Preencher\_ListView\_Actores();

Preencher\_ListView\_Filmes();

Preencher\_ComboBox\_Nacionalidades();

}

private void PICTUREBOX\_GestaoActores\_ADICIONAR\_ACTOR\_Click(object sender, EventArgs e)

{

TABCONTROL\_GestaoActores.SelectedIndex = 1;

}

private void LISTVIEW\_GestaoActores\_ACTORES\_SelectedIndexChanged(object sender, EventArgs e)

{

Preencher\_TextBox\_Actor();

}

private void LISTVIEW\_GestaoActores\_ACTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Actores();

}

private void CHECKBOX\_GestaoActores\_AGRUPAR\_ACTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoActores\_ACTORES.ShowGroups = CHECKBOX\_GestaoActores\_AGRUPAR\_ACTORES.Checked;

Ordenar\_Actores();

}

private void LISTVIEW\_GestaoActores\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

private void CHECKBOX\_GestaoActores\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoActores\_FILMES.ShowGroups = CHECKBOX\_GestaoActores\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

private void LISTVIEW\_AdicionarActor\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

private void CHECKBOX\_AdicionarActor\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_AdicionarActor\_FILMES.ShowGroups = CHECKBOX\_AdicionarActor\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

private void BUTTON\_AdicionarActor\_ADICIONAR\_ACTOR\_Click(object sender, EventArgs e)

{

Adicionar\_Actor();

ClearListView();

Preencher\_ListView\_Actores();

LimparTudo();

TABCONTROL\_GestaoActores.SelectedIndex = 0;

}

private void PICTUREBOX\_GestaoActores\_REMOVER\_ACTOR\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text == "")

{

MessageBox.Show("Não tem nenhum actor selecionado! Não é possivel remover.", "Nenhum actor selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text != "")

{

DialogResult RemoverActor = MessageBox.Show("Tem a certeza que deseja remover este actor?", "Remover Actor?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (RemoverActor == DialogResult.Yes)

{

Remover\_Actor();

ClearListView();

Preencher\_ListView\_Actores();

LimparTudo();

}

}

}

private void PICTUREBOX\_GestaoActores\_GRAVAR\_ALTERACOES\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text == "")

{

MessageBox.Show("Não tem nenhum actor selecionado! Não é possivel alterar.", "Nenhum Actor Selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text != "")

{

DialogResult AlterarActor = MessageBox.Show("Tem a certeza que deseja alterar este actor?", "Alterar Actor?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (AlterarActor == DialogResult.Yes)

{

Alterar\_Actor();

ClearListView();

Preencher\_ListView\_Actores();

LimparTudo();

}

}

}

private void TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text = TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoActores\_APELIDO.Text;

}

private void TEXTBOX\_GestaoActores\_APELIDO\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_GestaoActores\_APELIDO.Text = Regex.Replace(TEXTBOX\_GestaoActores\_APELIDO.Text, @"[0-9]", string.Empty);

TEXTBOX\_GestaoActores\_APELIDO.Text = Regex.Replace(TEXTBOX\_GestaoActores\_APELIDO.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text = TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoActores\_APELIDO.Text;

}

private void TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

private void TEXTBOX\_AdicionarActor\_APELIDO\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_AdicionarActor\_APELIDO.Text = Regex.Replace(TEXTBOX\_AdicionarActor\_APELIDO.Text, @"[0-9]", string.Empty);

TEXTBOX\_AdicionarActor\_APELIDO.Text = Regex.Replace(TEXTBOX\_AdicionarActor\_APELIDO.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

public void Preencher\_ListView\_Actores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.ID," + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.primeiro\_nome," + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.apelido," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["primeiro\_nome"].ToString());

ROW.SubItems.Add(Reader["apelido"].ToString());

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoActores\_ACTORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Filmes()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT ID,titulo FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["titulo"].ToString());

var ROW2 = new ListViewItem();

ROW2.Text = Reader["ID"].ToString();

ROW2.SubItems.Add(Reader["titulo"].ToString());

LISTVIEW\_GestaoActores\_FILMES.Items.Add(ROW);

LISTVIEW\_AdicionarActor\_FILMES.Items.Add(ROW2);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ComboBox\_Nacionalidades()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

COMBOBOX\_GestaoActores\_NACIONALIDADE.Items.Add(Reader.GetString("nacionalidade"));

COMBOBOX\_AdicionarActor\_NACIONALIDADE.Items.Add(Reader.GetString("nacionalidade"));

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void ClearListView()

{

LISTVIEW\_GestaoActores\_ACTORES.Items.Clear();

}

public void LimparTudo()

{

TEXTBOX\_AdicionarActor\_APELIDO.Text = "";

TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME.Text = "";

COMBOBOX\_AdicionarActor\_NACIONALIDADE.SelectedItem = null;

TEXTBOX\_GestaoActores\_APELIDO.Text = "";

TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text = "";

TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text = "";

COMBOBOX\_GestaoActores\_NACIONALIDADE.SelectedItem = null;

}

public void Preencher\_TextBox\_Actor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_GestaoActores\_ACTORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

LigacaoDB.Open();

string Nome;

string PrimeiroNome;

string Apelido;

int IDNacionalidade;

string Query\_PrimeiroNome = "SELECT primeiro\_nome FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores WHERE ID = " + ID + "";

string Query\_SegundoNome = "SELECT apelido FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores WHERE ID = " + ID + "";

string Query\_IDNacionalidade = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".actores WHERE ID = " + ID + "";

MySqlCommand Comando\_PrimeiroNome = new MySqlCommand(Query\_PrimeiroNome, LigacaoDB);

MySqlCommand Comando\_SegundoNome = new MySqlCommand(Query\_SegundoNome, LigacaoDB);

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

PrimeiroNome = Comando\_PrimeiroNome.ExecuteScalar().ToString();

Apelido = Comando\_SegundoNome.ExecuteScalar().ToString();

Nome = PrimeiroNome + " " + Apelido;

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text = Nome;

TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text = PrimeiroNome;

TEXTBOX\_GestaoActores\_APELIDO.Text = Apelido;

if(IDNacionalidade == 1)

COMBOBOX\_GestaoActores\_NACIONALIDADE.SelectedIndex = 0;

else

COMBOBOX\_GestaoActores\_NACIONALIDADE.SelectedIndex = IDNacionalidade - 1;

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

LISTVIEW\_GestaoActores\_FILMES.SelectedItems.Clear();

TEXTBOX\_GestaoActores\_NOME\_COMPLETO.Text = "";

TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text = "";

TEXTBOX\_GestaoActores\_APELIDO.Text = "";

COMBOBOX\_GestaoActores\_NACIONALIDADE.SelectedItem = null;

}

}

public void Adicionar\_Actor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Nome = TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_AdicionarActor\_APELIDO.Text;

string PrimeiroNome = TEXTBOX\_AdicionarActor\_PRIMEIRO\_NOME.Text;

string Apelido = TEXTBOX\_AdicionarActor\_APELIDO.Text;

string Nacionalidade = COMBOBOX\_AdicionarActor\_NACIONALIDADE.Text;

int IDNacionalidade;

string Query\_IDNacionalidade = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades WHERE nacionalidade='" + Nacionalidade + "'";

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

LigacaoDB.Close();

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_InserirActor = new MySqlCommand("SP\_ACTOR\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_InserirActor.CommandType = CommandType.StoredProcedure;

Comando\_InserirActor.Parameters.AddWithValue("\_nome\_completo", Nome);

Comando\_InserirActor.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando\_InserirActor.Parameters.AddWithValue("\_apelido", Apelido);

Comando\_InserirActor.Parameters.AddWithValue("\_nacionalidade", IDNacionalidade);

Reader = Comando\_InserirActor.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Actor Inserido com Sucesso", "Actor Inserido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Remover\_Actor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoActores\_ACTORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

MySqlCommand Command\_RemoverActor = new MySqlCommand("SP\_ACTOR\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_RemoverActor.CommandType = CommandType.StoredProcedure;

Command\_RemoverActor.Parameters.AddWithValue("\_id", ID);

Reader = Command\_RemoverActor.ExecuteReader();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Actor removido com sucesso", "Actor Removido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Alterar\_Actor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string ListView\_ID = LISTVIEW\_GestaoActores\_ACTORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

string Nome = TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoActores\_APELIDO.Text;

string PrimeiroNome = TEXTBOX\_GestaoActores\_PRIMEIRO\_NOME.Text;

string Apelido = TEXTBOX\_GestaoActores\_APELIDO.Text;

string Nacionalidade = COMBOBOX\_GestaoActores\_NACIONALIDADE.Text;

int IDNacionalidade;

string Query\_IDNacionalidade = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades WHERE nacionalidade='" + Nacionalidade + "'";

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

LigacaoDB.Close();

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_AlterarActor = new MySqlCommand("SP\_ACTOR\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando\_AlterarActor.CommandType = CommandType.StoredProcedure;

Comando\_AlterarActor.Parameters.AddWithValue("\_id", ID);

Comando\_AlterarActor.Parameters.AddWithValue("\_nome\_completo", Nome);

Comando\_AlterarActor.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando\_AlterarActor.Parameters.AddWithValue("\_apelido", Apelido);

Comando\_AlterarActor.Parameters.AddWithValue("\_nacionalidade", IDNacionalidade);

Reader = Comando\_AlterarActor.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Actor Alterado com Sucesso", "Actor Alterado", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Ordenar\_Actores()

{

if (CHECKBOX\_GestaoActores\_AGRUPAR\_ACTORES.Checked == true)

{

if (LISTVIEW\_GestaoActores\_ACTORES.ShowGroups)

BuildGroups\_Actores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoActores\_ACTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Filmes()

{

if (CHECKBOX\_GestaoActores\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_GestaoActores\_FILMES.ShowGroups)

BuildGroups\_GestaoFilmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoActores\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

if (CHECKBOX\_AdicionarActor\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_AdicionarActor\_FILMES.ShowGroups)

BuildGroups\_AdicionarFilmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_AdicionarActor\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups\_Actores(int coluna)

{

LISTVIEW\_GestaoActores\_ACTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoActores\_ACTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoActores\_ACTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoActores\_ACTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_GestaoFilmes(int coluna)

{

LISTVIEW\_GestaoActores\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoActores\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoActores\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoActores\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_AdicionarFilmes(int coluna)

{

LISTVIEW\_AdicionarActor\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_AdicionarActor\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_AdicionarActor\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_AdicionarActor\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}

* **[N] Form de Gestão de Productores**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO\_PESSOAS\_PRODUCTORES : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_GESTAO\_PESSOAS\_PRODUCTORES()

{

InitializeComponent();

Preencher\_ListView\_Productores();

Preencher\_ListView\_Filmes();

Preencher\_ComboBox\_Nacionalidades();

}

private void PICTUREBOX\_GestaoProductores\_ADICIONAR\_PRODUCTOR\_Click(object sender, EventArgs e)

{

TABCONTROL\_GestaoProductores.SelectedIndex = 1;

}

private void PICTUREBOX\_GestaoProductores\_REMOVER\_PRODUCTOR\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text == "")

{

MessageBox.Show("Não tem nenhum productor selecionado! Não é possivel remover.", "Nenhum productor selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text != "")

{

DialogResult RemoverProductor = MessageBox.Show("Tem a certeza que deseja remover este productor?", "Remover Actor?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (RemoverProductor == DialogResult.Yes)

{

Remover\_Productor();

ClearListView();

Preencher\_ListView\_Productores();

LimparTudo();

}

}

}

private void PICTUREBOX\_GestaoProductores\_GRAVAR\_ALTERACOES\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text == "")

{

MessageBox.Show("Não tem nenhum productor selecionado! Não é possivel remover.", "Nenhum productor selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text != "")

{

DialogResult AlterarActor = MessageBox.Show("Tem a certeza que deseja alterar este productor?", "Alterar Productor?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (AlterarActor == DialogResult.Yes)

{

Alterar\_Productor();

ClearListView();

Preencher\_ListView\_Productores();

LimparTudo();

}

}

}

private void BUTTON\_AdicionarProductor\_ADICIONAR\_PRODUCTOR\_Click(object sender, EventArgs e)

{

Adicionar\_Productor();

ClearListView();

Preencher\_ListView\_Productores();

LimparTudo();

TABCONTROL\_GestaoProductores.SelectedIndex = 0;

}

private void LISTVIEW\_GestaoProductores\_PRODUCTORES\_SelectedIndexChanged(object sender, EventArgs e)

{

Preencher\_TextBox\_Productor();

}

private void TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text = TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoProductores\_APELIDO.Text;

}

private void TEXTBOX\_GestaoProductores\_APELIDO\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_GestaoProductores\_APELIDO.Text = Regex.Replace(TEXTBOX\_GestaoProductores\_APELIDO.Text, @"[0-9]", string.Empty);

TEXTBOX\_GestaoProductores\_APELIDO.Text = Regex.Replace(TEXTBOX\_GestaoProductores\_APELIDO.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text = TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoProductores\_APELIDO.Text;

}

private void TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

private void TEXTBOX\_AdicionarProductor\_APELIDO\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_AdicionarProductor\_APELIDO.Text = Regex.Replace(TEXTBOX\_AdicionarProductor\_APELIDO.Text, @"[0-9]", string.Empty);

TEXTBOX\_AdicionarProductor\_APELIDO.Text = Regex.Replace(TEXTBOX\_AdicionarProductor\_APELIDO.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

public void Preencher\_ListView\_Productores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.ID," + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.primeiro\_nome," + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.apelido," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["primeiro\_nome"].ToString());

ROW.SubItems.Add(Reader["apelido"].ToString());

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoProductores\_PRODUCTORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Filmes()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT ID,titulo FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["titulo"].ToString());

var ROW2 = new ListViewItem();

ROW2.Text = Reader["ID"].ToString();

ROW2.SubItems.Add(Reader["titulo"].ToString());

LISTVIEW\_GestaoProductores\_FILMES.Items.Add(ROW);

LISTVIEW\_AdicionarProductor\_FILMES.Items.Add(ROW2);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ComboBox\_Nacionalidades()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

COMBOBOX\_GestaoProductores\_NACIONALIDADE.Items.Add(Reader.GetString("nacionalidade"));

COMBOBOX\_AdicionarProductor\_NACIONALIDADE.Items.Add(Reader.GetString("nacionalidade"));

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void ClearListView()

{

LISTVIEW\_GestaoProductores\_PRODUCTORES.Items.Clear();

}

public void LimparTudo()

{

TEXTBOX\_AdicionarProductor\_APELIDO.Text = "";

TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME.Text = "";

COMBOBOX\_AdicionarProductor\_NACIONALIDADE.SelectedItem = null;

TEXTBOX\_GestaoProductores\_APELIDO.Text = "";

TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text = "";

TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text = "";

COMBOBOX\_GestaoProductores\_NACIONALIDADE.SelectedItem = null;

}

public void Preencher\_TextBox\_Productor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_GestaoProductores\_PRODUCTORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

LigacaoDB.Open();

string Nome;

string PrimeiroNome;

string Apelido;

int IDNacionalidade;

string Query\_PrimeiroNome = "SELECT primeiro\_nome FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores WHERE ID = " + ID + "";

string Query\_SegundoNome = "SELECT apelido FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores WHERE ID = " + ID + "";

string Query\_IDNacionalidade = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".productores WHERE ID = " + ID + "";

MySqlCommand Comando\_PrimeiroNome = new MySqlCommand(Query\_PrimeiroNome, LigacaoDB);

MySqlCommand Comando\_SegundoNome = new MySqlCommand(Query\_SegundoNome, LigacaoDB);

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

PrimeiroNome = Comando\_PrimeiroNome.ExecuteScalar().ToString();

Apelido = Comando\_SegundoNome.ExecuteScalar().ToString();

Nome = PrimeiroNome + " " + Apelido;

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text = Nome;

TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text = PrimeiroNome;

TEXTBOX\_GestaoProductores\_APELIDO.Text = Apelido;

if (IDNacionalidade == 1)

COMBOBOX\_GestaoProductores\_NACIONALIDADE.SelectedIndex = 0;

else

COMBOBOX\_GestaoProductores\_NACIONALIDADE.SelectedIndex = IDNacionalidade - 1;

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

LISTVIEW\_GestaoProductores\_PRODUCTORES.SelectedItems.Clear();

TEXTBOX\_GestaoProductores\_NOME\_COMPLETO.Text = "";

TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text = "";

TEXTBOX\_GestaoProductores\_APELIDO.Text = "";

COMBOBOX\_GestaoProductores\_NACIONALIDADE.SelectedItem = null;

}

}

public void Adicionar\_Productor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Nome = TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_AdicionarProductor\_APELIDO.Text;

string PrimeiroNome = TEXTBOX\_AdicionarProductor\_PRIMEIRO\_NOME.Text;

string Apelido = TEXTBOX\_AdicionarProductor\_APELIDO.Text;

string Nacionalidade = COMBOBOX\_AdicionarProductor\_NACIONALIDADE.Text;

int IDNacionalidade;

string Query\_IDNacionalidade = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades WHERE nacionalidade='" + Nacionalidade + "'";

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

LigacaoDB.Close();

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_InserirProductor = new MySqlCommand("SP\_PRODUCTOR\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_InserirProductor.CommandType = CommandType.StoredProcedure;

Comando\_InserirProductor.Parameters.AddWithValue("\_nome\_completo", Nome);

Comando\_InserirProductor.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando\_InserirProductor.Parameters.AddWithValue("\_apelido", Apelido);

Comando\_InserirProductor.Parameters.AddWithValue("\_nacionalidade", IDNacionalidade);

Reader = Comando\_InserirProductor.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Productor Inserido com Sucesso", "Productor Inserido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Remover\_Productor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoProductores\_PRODUCTORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

MySqlCommand Command\_RemoverProductor = new MySqlCommand("SP\_PRODUCTOR\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_RemoverProductor.CommandType = CommandType.StoredProcedure;

Command\_RemoverProductor.Parameters.AddWithValue("\_id", ID);

Reader = Command\_RemoverProductor.ExecuteReader();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Productor removido com sucesso", "Productor Removido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Alterar\_Productor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string ListView\_ID = LISTVIEW\_GestaoProductores\_PRODUCTORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

string Nome = TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoProductores\_APELIDO.Text;

string PrimeiroNome = TEXTBOX\_GestaoProductores\_PRIMEIRO\_NOME.Text;

string Apelido = TEXTBOX\_GestaoProductores\_APELIDO.Text;

string Nacionalidade = COMBOBOX\_GestaoProductores\_NACIONALIDADE.Text;

int IDNacionalidade;

string Query\_IDNacionalidade = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades WHERE nacionalidade='" + Nacionalidade + "'";

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

LigacaoDB.Close();

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_AlterarProductor = new MySqlCommand("SP\_PRODUCTOR\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando\_AlterarProductor.CommandType = CommandType.StoredProcedure;

Comando\_AlterarProductor.Parameters.AddWithValue("\_id", ID);

Comando\_AlterarProductor.Parameters.AddWithValue("\_nome\_completo", Nome);

Comando\_AlterarProductor.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando\_AlterarProductor.Parameters.AddWithValue("\_apelido", Apelido);

Comando\_AlterarProductor.Parameters.AddWithValue("\_nacionalidade", IDNacionalidade);

Reader = Comando\_AlterarProductor.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Productor Alterado com Sucesso", "Productor Alterado", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Ordenar\_Productores()

{

if (CHECKBOX\_GestaoProductores\_AGRUPAR\_PRODUCTORES.Checked == true)

{

if (LISTVIEW\_GestaoProductores\_PRODUCTORES.ShowGroups)

BuildGroups\_Productores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoProductores\_PRODUCTORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Filmes()

{

if (CHECKBOX\_GestaoProductores\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_GestaoProductores\_FILMES.ShowGroups)

BuildGroups\_GestaoFilmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoProductores\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

if (CHECKBOX\_AdicionarProductor\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_AdicionarProductor\_FILMES.ShowGroups)

BuildGroups\_AdicionarFilmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_AdicionarProductor\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups\_Productores(int coluna)

{

LISTVIEW\_GestaoProductores\_PRODUCTORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoProductores\_PRODUCTORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoProductores\_PRODUCTORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoProductores\_PRODUCTORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_GestaoFilmes(int coluna)

{

LISTVIEW\_GestaoProductores\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoProductores\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoProductores\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoProductores\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_AdicionarFilmes(int coluna)

{

LISTVIEW\_AdicionarProductor\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_AdicionarProductor\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_AdicionarProductor\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_AdicionarProductor\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

private void CHECKBOX\_GestaoProductores\_AGRUPAR\_PRODUCTORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoProductores\_PRODUCTORES.ShowGroups = CHECKBOX\_GestaoProductores\_AGRUPAR\_PRODUCTORES.Checked;

Ordenar\_Productores();

}

private void CHECKBOX\_GestaoProductores\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoProductores\_FILMES.ShowGroups = CHECKBOX\_GestaoProductores\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

private void LISTVIEW\_GestaoProductores\_PRODUCTORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Productores();

}

private void LISTVIEW\_GestaoProductores\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

private void CHECKBOX\_AdicionarProductor\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_AdicionarProductor\_FILMES.ShowGroups = CHECKBOX\_AdicionarProductor\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

private void LISTVIEW\_AdicionarProductor\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

}

}

* **[O] Form de Gestão de Realizadores**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Globalization;

using System.Text.RegularExpressions;

namespace Cinema\_Manager\_1.\_0.FORMS\_GESTAO

{

public partial class FORM\_GESTAO\_PESSOAS\_REALIZADORES : Form

{

MySqlConnection LigacaoDB = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.Servidor);

MySqlConnection LigacaoDB\_StoredProcedures = new MySqlConnection(CLASS\_GLOBAL\_VARIABLES.ServidorStoredProcedure);

MySqlDataAdapter Adapter = new MySqlDataAdapter();

MySqlDataReader Reader;

MySqlDataReader Reader2;

DataTable TabelaDados = new DataTable();

DataSet DS = new DataSet();

private int UltimaColunaOrdenada = 0;

private SortOrder UltimaOrdem = SortOrder.Ascending;

public FORM\_GESTAO\_PESSOAS\_REALIZADORES()

{

InitializeComponent();

Preencher\_ListView\_Realizadores();

Preencher\_ListView\_Filmes();

Preencher\_ComboBox\_Nacionalidades();

}

private void PICTUREBOX\_GestaoRealizadores\_ADICIONAR\_REALIZADOR\_Click(object sender, EventArgs e)

{

TABCONTROL\_GestaoRealizadores.SelectedIndex = 1;

}

private void PICTUREBOX\_GestaoRealizadores\_REMOVER\_REALIZADOR\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text == "")

{

MessageBox.Show("Não tem nenhum realizador selecionado! Não é possivel remover.", "Nenhum realizador selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text != "")

{

DialogResult RemoverRealizador = MessageBox.Show("Tem a certeza que deseja remover este realizador?", "Remover Realizador?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (RemoverRealizador == DialogResult.Yes)

{

Remover\_Realizador();

ClearListView();

Preencher\_ListView\_Realizadores();

LimparTudo();

}

}

}

private void PICTUREBOX\_GestaoRealizadores\_GRAVAR\_ALTERACOES\_Click(object sender, EventArgs e)

{

if (TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text == "")

{

MessageBox.Show("Não tem nenhum realizador selecionado! Não é possivel remover.", "Nenhum realizador selecionado", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

if (TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text != "")

{

DialogResult AlterarRealizador = MessageBox.Show("Tem a certeza que deseja alterar este realizador?", "Alterar Realizador?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (AlterarRealizador == DialogResult.Yes)

{

Alterar\_Actor();

ClearListView();

Preencher\_ListView\_Realizadores();

LimparTudo();

}

}

}

private void BUTTON\_AdicionarRealizador\_ADICIONAR\_REALIZADOR\_Click(object sender, EventArgs e)

{

Adicionar\_Realizador();

ClearListView();

Preencher\_ListView\_Realizadores();

LimparTudo();

TABCONTROL\_GestaoRealizadores.SelectedIndex = 0;

}

private void LISTVIEW\_GestaoRealizadores\_REALIZADORES\_SelectedIndexChanged(object sender, EventArgs e)

{

Preencher\_TextBox\_Realizador();

}

private void TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text = TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoRealizadores\_APELIDO.Text;

}

private void TEXTBOX\_GestaoRealizadores\_APELIDO\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_GestaoRealizadores\_APELIDO.Text = Regex.Replace(TEXTBOX\_GestaoRealizadores\_APELIDO.Text, @"[0-9]", string.Empty);

TEXTBOX\_GestaoRealizadores\_APELIDO.Text = Regex.Replace(TEXTBOX\_GestaoRealizadores\_APELIDO.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text = TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoRealizadores\_APELIDO.Text;

}

private void TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME.Text, @"[0-9]", string.Empty);

TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME.Text = Regex.Replace(TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

private void TEXTBOX\_AdicionarRealizador\_APELIDO\_TextChanged(object sender, EventArgs e)

{

TEXTBOX\_AdicionarRealizador\_APELIDO.Text = Regex.Replace(TEXTBOX\_AdicionarRealizador\_APELIDO.Text, @"[0-9]", string.Empty);

TEXTBOX\_AdicionarRealizador\_APELIDO.Text = Regex.Replace(TEXTBOX\_AdicionarRealizador\_APELIDO.Text, @"[| \ ! @ # £ $ § % & / { } ( ) [ ] = ? ' » « \* + ¨ ª º - \_ . : , ; > <]", string.Empty);

}

private void LISTVIEW\_GestaoRealizadores\_REALIZADORES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Realizadores();

}

private void CHECKBOX\_GestaoRealizadores\_AGRUPAR\_REALIZADORES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoRealizadores\_REALIZADORES.ShowGroups = CHECKBOX\_GestaoRealizadores\_AGRUPAR\_REALIZADORES.Checked;

Ordenar\_Realizadores();

}

private void LISTVIEW\_GestaoRealizadores\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

private void CHECKBOX\_GestaoRealizadores\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_GestaoRealizadores\_FILMES.ShowGroups = CHECKBOX\_GestaoRealizadores\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

private void LISTVIEW\_AdicionarRealizador\_FILMES\_ColumnClick(object sender, ColumnClickEventArgs e)

{

if (e.Column == UltimaColunaOrdenada)

{

if (UltimaOrdem == SortOrder.Ascending)

UltimaOrdem = SortOrder.Descending;

else

UltimaOrdem = SortOrder.Ascending;

}

else

{

UltimaOrdem = SortOrder.Ascending;

UltimaColunaOrdenada = e.Column;

}

Ordenar\_Filmes();

}

private void CHECKBOX\_AdicionarRealizador\_AGRUPAR\_FILMES\_CheckedChanged(object sender, EventArgs e)

{

LISTVIEW\_AdicionarRealizador\_FILMES.ShowGroups = CHECKBOX\_AdicionarRealizador\_AGRUPAR\_FILMES.Checked;

Ordenar\_Filmes();

}

public void Preencher\_ListView\_Realizadores()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.ID," + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.primeiro\_nome," + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.apelido," + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores INNER JOIN " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades ON " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores.nacionalidade = " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades.ID";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["primeiro\_nome"].ToString());

ROW.SubItems.Add(Reader["apelido"].ToString());

ROW.SubItems.Add(Reader["nacionalidade"].ToString());

LISTVIEW\_GestaoRealizadores\_REALIZADORES.Items.Add(ROW);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ListView\_Filmes()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT ID,titulo FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".filmes";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

var ROW = new ListViewItem();

ROW.Text = Reader["ID"].ToString();

ROW.SubItems.Add(Reader["titulo"].ToString());

var ROW2 = new ListViewItem();

ROW2.Text = Reader["ID"].ToString();

ROW2.SubItems.Add(Reader["titulo"].ToString());

LISTVIEW\_GestaoRealizadores\_FILMES.Items.Add(ROW);

LISTVIEW\_AdicionarRealizador\_FILMES.Items.Add(ROW2);

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void Preencher\_ComboBox\_Nacionalidades()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Query = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades";

MySqlCommand Comando = new MySqlCommand(Query, LigacaoDB);

Reader = Comando.ExecuteReader();

while (Reader.Read())

{

COMBOBOX\_GestaoRealizadores\_NACIONALIDADE.Items.Add(Reader.GetString("nacionalidade"));

COMBOBOX\_AdicionarRealizador\_NACIONALIDADE.Items.Add(Reader.GetString("nacionalidade"));

}

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

}

}

public void ClearListView()

{

LISTVIEW\_GestaoRealizadores\_REALIZADORES.Items.Clear();

}

public void LimparTudo()

{

TEXTBOX\_AdicionarRealizador\_APELIDO.Text = "";

TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME.Text = "";

COMBOBOX\_AdicionarRealizador\_NACIONALIDADE.SelectedItem = null;

TEXTBOX\_GestaoRealizadores\_APELIDO.Text = "";

TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text = "";

TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text = "";

COMBOBOX\_GestaoRealizadores\_NACIONALIDADE.SelectedItem = null;

}

public void Preencher\_TextBox\_Realizador()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

string ListView\_ID = LISTVIEW\_GestaoRealizadores\_REALIZADORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

LigacaoDB.Open();

string Nome;

string PrimeiroNome;

string Apelido;

int IDNacionalidade;

string Query\_PrimeiroNome = "SELECT primeiro\_nome FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores WHERE ID = " + ID + "";

string Query\_SegundoNome = "SELECT apelido FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores WHERE ID = " + ID + "";

string Query\_IDNacionalidade = "SELECT nacionalidade FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".realizadores WHERE ID = " + ID + "";

MySqlCommand Comando\_PrimeiroNome = new MySqlCommand(Query\_PrimeiroNome, LigacaoDB);

MySqlCommand Comando\_SegundoNome = new MySqlCommand(Query\_SegundoNome, LigacaoDB);

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

PrimeiroNome = Comando\_PrimeiroNome.ExecuteScalar().ToString();

Apelido = Comando\_SegundoNome.ExecuteScalar().ToString();

Nome = PrimeiroNome + " " + Apelido;

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text = Nome;

TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text = PrimeiroNome;

TEXTBOX\_GestaoRealizadores\_APELIDO.Text = Apelido;

if (IDNacionalidade == 1)

COMBOBOX\_GestaoRealizadores\_NACIONALIDADE.SelectedIndex = 0;

else

COMBOBOX\_GestaoRealizadores\_NACIONALIDADE.SelectedIndex = IDNacionalidade - 1;

LigacaoDB.Close();

}

catch (Exception EX)

{

//MessageBox.Show(EX.Message);

LISTVIEW\_GestaoRealizadores\_REALIZADORES.SelectedItems.Clear();

TEXTBOX\_GestaoRealizadores\_NOME\_COMPLETO.Text = "";

TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text = "";

TEXTBOX\_GestaoRealizadores\_APELIDO.Text = "";

COMBOBOX\_GestaoRealizadores\_NACIONALIDADE.SelectedItem = null;

}

}

public void Adicionar\_Realizador()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string Nome = TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_AdicionarRealizador\_APELIDO.Text;

string PrimeiroNome = TEXTBOX\_AdicionarRealizador\_PRIMEIRO\_NOME.Text;

string Apelido = TEXTBOX\_AdicionarRealizador\_APELIDO.Text;

string Nacionalidade = COMBOBOX\_AdicionarRealizador\_NACIONALIDADE.Text;

int IDNacionalidade;

string Query\_IDNacionalidade = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades WHERE nacionalidade='" + Nacionalidade + "'";

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

LigacaoDB.Close();

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_InserirRealizador = new MySqlCommand("SP\_REALIZADOR\_INSERIR", LigacaoDB\_StoredProcedures);

Comando\_InserirRealizador.CommandType = CommandType.StoredProcedure;

Comando\_InserirRealizador.Parameters.AddWithValue("\_nome\_completo", Nome);

Comando\_InserirRealizador.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando\_InserirRealizador.Parameters.AddWithValue("\_apelido", Apelido);

Comando\_InserirRealizador.Parameters.AddWithValue("\_nacionalidade", IDNacionalidade);

Reader = Comando\_InserirRealizador.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Realizador Inserido com Sucesso", "Realizador Inserido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Remover\_Realizador()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB\_StoredProcedures.Open();

string ListView\_ID = LISTVIEW\_GestaoRealizadores\_REALIZADORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

MySqlCommand Command\_RemoverRealizador = new MySqlCommand("SP\_REALIZADOR\_REMOVER", LigacaoDB\_StoredProcedures);

Command\_RemoverRealizador.CommandType = CommandType.StoredProcedure;

Command\_RemoverRealizador.Parameters.AddWithValue("\_id", ID);

Reader = Command\_RemoverRealizador.ExecuteReader();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Realizador removido com sucesso", "Realizador Removido", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{ }

}

public void Alterar\_Actor()

{

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

if (LigacaoDB\_StoredProcedures.State == ConnectionState.Open)

LigacaoDB\_StoredProcedures.Close();

try

{

LigacaoDB.Open();

string ListView\_ID = LISTVIEW\_GestaoRealizadores\_REALIZADORES.SelectedItems[0].Text;

int ID = Convert.ToInt32(ListView\_ID);

string Nome = TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text + " " + TEXTBOX\_GestaoRealizadores\_APELIDO.Text;

string PrimeiroNome = TEXTBOX\_GestaoRealizadores\_PRIMEIRO\_NOME.Text;

string Apelido = TEXTBOX\_GestaoRealizadores\_APELIDO.Text;

string Nacionalidade = COMBOBOX\_GestaoRealizadores\_NACIONALIDADE.Text;

int IDNacionalidade;

string Query\_IDNacionalidade = "SELECT ID FROM " + CLASS\_GLOBAL\_VARIABLES.BD + ".nacionalidades WHERE nacionalidade='" + Nacionalidade + "'";

MySqlCommand Comando\_IDNacionalidade = new MySqlCommand(Query\_IDNacionalidade, LigacaoDB);

IDNacionalidade = Convert.ToInt32(Comando\_IDNacionalidade.ExecuteScalar());

LigacaoDB.Close();

if (LigacaoDB.State == ConnectionState.Open)

LigacaoDB.Close();

LigacaoDB\_StoredProcedures.Open();

MySqlCommand Comando\_AlterarRealizador = new MySqlCommand("SP\_REALIZADOR\_ALTERAR", LigacaoDB\_StoredProcedures);

Comando\_AlterarRealizador.CommandType = CommandType.StoredProcedure;

Comando\_AlterarRealizador.Parameters.AddWithValue("\_id", ID);

Comando\_AlterarRealizador.Parameters.AddWithValue("\_nome\_completo", Nome);

Comando\_AlterarRealizador.Parameters.AddWithValue("\_primeiro\_nome", PrimeiroNome);

Comando\_AlterarRealizador.Parameters.AddWithValue("\_apelido", Apelido);

Comando\_AlterarRealizador.Parameters.AddWithValue("\_nacionalidade", IDNacionalidade);

Reader = Comando\_AlterarRealizador.ExecuteReader();

Reader.Close();

LigacaoDB\_StoredProcedures.Close();

MessageBox.Show("Realizador Alterado com Sucesso", "Realizador Alterado", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message);

}

}

public void Ordenar\_Realizadores()

{

if (CHECKBOX\_GestaoRealizadores\_AGRUPAR\_REALIZADORES.Checked == true)

{

if (LISTVIEW\_GestaoRealizadores\_REALIZADORES.ShowGroups)

BuildGroups\_Realizadores(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoRealizadores\_REALIZADORES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void Ordenar\_Filmes()

{

if (CHECKBOX\_GestaoRealizadores\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_GestaoRealizadores\_FILMES.ShowGroups)

BuildGroups\_GestaoFilmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_GestaoRealizadores\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

if (CHECKBOX\_AdicionarRealizador\_AGRUPAR\_FILMES.Checked == true)

{

if (LISTVIEW\_AdicionarRealizador\_FILMES.ShowGroups)

BuildGroups\_AdicionarFilmes(UltimaColunaOrdenada);

}

else

LISTVIEW\_AdicionarRealizador\_FILMES.ListViewItemSorter = new CompararColuna(UltimaColunaOrdenada, UltimaOrdem);

}

public void BuildGroups\_Realizadores(int coluna)

{

LISTVIEW\_GestaoRealizadores\_REALIZADORES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoRealizadores\_REALIZADORES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoRealizadores\_REALIZADORES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoRealizadores\_REALIZADORES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_GestaoFilmes(int coluna)

{

LISTVIEW\_GestaoRealizadores\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_GestaoRealizadores\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_GestaoRealizadores\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_GestaoRealizadores\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

public void BuildGroups\_AdicionarFilmes(int coluna)

{

LISTVIEW\_AdicionarRealizador\_FILMES.Groups.Clear();

int Contagem = LISTVIEW\_AdicionarRealizador\_FILMES.Items.Count;

Dictionary<String, List<ListViewItem>> Mapear = new Dictionary<String, List<ListViewItem>>();

foreach (ListViewItem LVI in LISTVIEW\_AdicionarRealizador\_FILMES.Items)

{

String Item = LVI.SubItems[coluna].Text;

if (coluna == 0 && Item.Length > 0)

Item = Item.Substring(0, 1);

if (!Mapear.ContainsKey(Item))

Mapear[Item] = new List<ListViewItem>();

Mapear[Item].Add(LVI);

}

List<ListViewGroup> Grupos = new List<ListViewGroup>();

foreach (String Item in Mapear.Keys)

{

Grupos.Add(new ListViewGroup(Item));

}

Grupos.Sort(new ListViewCompararGrupo(UltimaOrdem));

CompararColuna OrdenarItems = new CompararColuna(coluna, UltimaOrdem);

foreach (ListViewGroup Grupo in Grupos)

{

LISTVIEW\_AdicionarRealizador\_FILMES.Groups.Add(Grupo);

Mapear[Grupo.Header].Sort(OrdenarItems);

Grupo.Items.AddRange(Mapear[Grupo.Header].ToArray());

}

}

internal class ListViewCompararGrupo : IComparer<ListViewGroup>

{

public ListViewCompararGrupo(SortOrder ordem)

{

this.Ordem = ordem;

}

public int Compare(ListViewGroup x, ListViewGroup y)

{

int Resultado = String.Compare(x.Header, y.Header, true);

if (this.Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

private SortOrder Ordem;

}

internal class CompararColuna : IComparer, IComparer<ListViewItem>

{

private int Coluna;

private SortOrder Ordem;

public CompararColuna(int coluna, SortOrder ordem)

{

Coluna = coluna;

Ordem = ordem;

}

public int Compare(object x, object y)

{

return Compare((ListViewItem)x, (ListViewItem)y);

}

public int Compare(ListViewItem x, ListViewItem y)

{

int Resultado = String.Compare(x.SubItems[Coluna].Text, y.SubItems[Coluna].Text, true);

if (Ordem == SortOrder.Descending)

Resultado = 0 - Resultado;

return Resultado;

}

}

}

}