## ANHANGUERA EDUCACIONAL

**CAMPUS MARTE**

**SISTEMAS DE INFORMAÇÃO**

**4º SEMESTRE**

**YTALO MATOS**

# MARCADE FILMES

# APLICATIVO DE FILMES

## SÃO PAULO/SP

## 201

## LISTA DE ILUSTRAÇÕES

# FIGURA 1 – Padrão MVC 05

# FIGURA 2 – Menu/Tela Inicial do Programa 13

# FIGURA 3 – Tela de Cadastro 14

# FIGURA 4 – Tela de Pesquisa 16

# FIGURA 5 – Lista de Filmes Assistidos 19

# FIGURA 6 – Lista de Filmes Não Assistidos 20

# FIGURA 7 – Tela de Informação 20

# SUMÁRIO

1. **INTRODUÇÃO 04**
2. **DESENVOLVIMENTO 04**

**3 METODOLOGIA 05**

**4 FORMULÁRIOS / CÓDIGO FONTE 13**

**5 CONCLUSÃO 21**

**REFERÊNCIAS 22**

**1 INTRODUÇÃO**

Este trabalho demonstra toda a funcionalidade de um aplicativo/programa criado em grupo, para controlar filmes, é um programa onde podemos cadastrar os filmes, com informações básicas e marcar se foi assistido.

Fazer pesquisas, por código e título. E visualizar todos os filmes assistidos e ‘não’ assistidos, para um controle melhor.

**2 DESENVOLVIMENTO**

Trata-se de um trabalho, pensado e realizado em grupo, parte em sala de aula e parte individualmente em casa, por cada integrante.

Em sala, aproveitamos a disponibilidade do professor para dúvidas e ajudas.

Em casa, através de exemplos e pesquisas na internet.

Foi desenvolvido na linguagem C#, com banco de dados MySQL.

IDE utilizada foi o Visual Studio 2015

O programa/app foi desenvolvido utilizando o padrão arquitetural MVC, separando model, view e controler.

Na pasta DTO tem somente a classe filme, com atributos e get/set.

Na pasta DAO tem a classe filmeDao, com os seguintes métodos: Incluir filme, Atualizar, Deletar, Lista de filmes, Selecionar filme, Filmes não assistidos, Selecionar filme pelo Id e Obter filme pelo título.

No Facade a classe filmeFacade, tem os mesmos métodos de DAO, com implementação diferente.

**3 METODOLOGIA**

O projeto foi desenvolvido seguindo o Padrão MVC, respeitando suas normas de separação e organização. Conforme imagem abaixo, e na sequência segue código fonte de DAO, DTO, FACADE, UTILS e SCRIPTS.

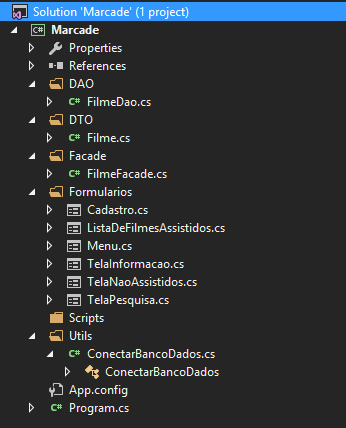


Figura 01 – Padrão MVC

DTO – Filme.cs

namespace Marcade.DTO

{

public class Filme

{

private int id;

private string titulo;

private string ano;

private string genero;

private string assistido;

private string ator;

private string diretor;

public int Id

{

get { return id; }

set { id = value; }

}

public string Titulo

{

get { return titulo; }

set { titulo = value; }

}

public string Ano

{

get { return ano; }

set { ano = value; }

}

public string Genero

{

get { return genero; }

set { genero = value; }

}

public string Assistido

{

get { return assistido; }

set { assistido = value; }

}

public string Ator

{

get { return ator; }

set { ator = value; }

}

public string Diretor

{

get { return diretor; }

set { diretor = value; }

}

}

}

DAO – FilmeDao.cs

namespace Marcade.DAO

{

public class FilmeDao

{

public void IncluirFilme(DTO.Filme filme)

{

using (MySqlConnection connection =

Utils.ConectarBancoDados.ObterConexao())

{

string queryString = @"insert into filme(id,titulo,ano,genero,assistido,ator,diretor)

values (@id, @titulo, @ano, @genero, @assistido, @ator, @diretor)";

MySqlCommand command = new MySqlCommand(queryString, connection);

command.Parameters.AddWithValue("@id", filme.Id);

command.Parameters.AddWithValue("@titulo", filme.Titulo);

command.Parameters.AddWithValue("@ano", filme.Ano);

command.Parameters.AddWithValue("@genero", filme.Genero);

command.Parameters.AddWithValue("@assistido", filme.Assistido);

command.Parameters.AddWithValue("@ator", filme.Ator);

command.Parameters.AddWithValue("@diretor", filme.Diretor);

try

{

connection.Open();

int valor = command.ExecuteNonQuery();

if (valor <= 0)

{

throw new Exception("Não foi possivel inserir o filme");

}

}

catch (Exception ex)

{

throw new Exception(ex.Message, ex);

}

}

}

public void AtualizarFilme(DTO.Filme filme)

{

using (MySqlConnection connection =

Utils.ConectarBancoDados.ObterConexao())

{

string mysql = "UPDATE Filme SET @id=id, @titulo=titulo, @ano=ano, @genero=genero, @assistido=assistido, @ator=ator, @diretor=diretor WHERE @id=id";

MySqlCommand cmd = new MySqlCommand(mysql, connection);

cmd.Parameters.AddWithValue("@id", filme.Id);

cmd.Parameters.AddWithValue("@titulo", filme.Titulo);

cmd.Parameters.AddWithValue("@ano", filme.Ano);

cmd.Parameters.AddWithValue("@genero", filme.Genero);

cmd.Parameters.AddWithValue("@assistido", filme.Assistido);

cmd.Parameters.AddWithValue("@ator", filme.Ator);

cmd.Parameters.AddWithValue("@diretor", filme.Diretor);

cmd.CommandType = CommandType.Text;

connection.Open();

try

{

int i = cmd.ExecuteNonQuery();

if (i > 0)

MessageBox.Show("Cadastro atualizado com sucesso!");

}

catch (Exception ex)

{

throw ex;

}

finally

{

connection.Close();

}

}

}

public void DeletarFilme(DTO.Filme filme)

{

using (MySqlConnection connection =

Utils.ConectarBancoDados.ObterConexao())

try

{

string mysql = "DELETE FROM Filme WHERE @id = id";

MySqlCommand cmd = new MySqlCommand(mysql, connection);

cmd.Parameters.AddWithValue("@id", filme.Id);

connection.Open();

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

throw ex;

}

finally

{

connection.Close();

}

}

public List<DTO.Filme> ListaDeFilmes()

{

using (MySqlConnection connection = Utils.ConectarBancoDados.ObterConexao())

using (MySqlCommand cmd = new MySqlCommand("Select \* from Filme", connection))

try

{

connection.Open();

List<DTO.Filme> listaFilmes = new List<DTO.Filme>();

using (MySqlDataReader leitor = cmd.ExecuteReader())

{

while (leitor.Read())

{

DTO.Filme filme = new DTO.Filme();

filme.Id = (int)leitor["id"];

filme.Titulo = leitor["titulo"].ToString();

filme.Ano = leitor["ano"].ToString();

filme.Genero = leitor["genero"].ToString();

filme.Assistido = leitor["assistido"].ToString();

filme.Ator = leitor["ator"].ToString();

filme.Diretor = leitor["diretor"].ToString();

}

}

return listaFilmes;

}

catch (Exception ex)

{

throw new Exception("Erro ao acessar o banco" + ex.Message);

}

}

public DataTable SelecionarFilmes()

{

using (MySqlConnection connection =

Utils.ConectarBancoDados.ObterConexao())

try

{

string mysql = "SELECT \* FROM Filme WHERE assistido = ‘Sim’";

MySqlCommand cmd = new MySqlCommand(mysql, connection);

MySqlDataAdapter da = new MySqlDataAdapter();

da.SelectCommand = cmd;

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

catch (Exception ex)

{

throw ex;

}

}

public DataTable FilmesNaoAssistidos()

{

using (MySqlConnection connection =

Utils.ConectarBancoDados.ObterConexao())

try

{

string mysql = "SELECT \* FROM Filme WHERE assistido = 'Não'";

MySqlCommand cmd = new MySqlCommand(mysql, connection);

MySqlDataAdapter da = new MySqlDataAdapter();

da.SelectCommand = cmd;

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

catch (Exception ex)

{

throw ex;

}

}

public DTO.Filme SelecionarFilmePeloId(int id)

{

using (MySqlConnection connection =

Utils.ConectarBancoDados.ObterConexao())

try

{

string mysql = "SELECT \* FROM Filme WHERE id = @id";

MySqlCommand cmd = new MySqlCommand(mysql, connection);

cmd.Parameters.AddWithValue("@id", id);

connection.Open();

MySqlDataReader leitor;

DTO.Filme filme = new DTO.Filme();

leitor = cmd.ExecuteReader(CommandBehavior.CloseConnection);

while (leitor.Read())

{

filme.Id = Convert.ToInt32(leitor["id"]);

filme.Titulo = leitor["titulo"].ToString();

filme.Ano = leitor["ano"].ToString();

filme.Genero = leitor["genero"].ToString();

filme.Assistido = leitor["assistido"].ToString();

filme.Ator = leitor["ator"].ToString();

filme.Diretor = leitor["diretor"].ToString();

}

return filme;

}

catch (Exception ex)

{

throw ex;

}

}

public DTO.Filme ObterFilmePeloTitulo(string titulo)

{

using (MySqlConnection connection =

Utils.ConectarBancoDados.ObterConexao())

{

string queryString = "SELECT \* FROM Filme WHERE titulo = @titulo";

MySqlCommand cmd = new MySqlCommand(queryString, connection);

cmd.Parameters.AddWithValue("@titulo", titulo);

connection.Open();

MySqlDataReader leitor;

DTO.Filme filme = new DTO.Filme();

leitor = cmd.ExecuteReader(CommandBehavior.CloseConnection);

while (leitor.Read())

{

filme.Id = Convert.ToInt32(leitor["id"]);

filme.Titulo = leitor["titulo"].ToString();

filme.Ano = leitor["ano"].ToString();

filme.Genero = leitor["genero"].ToString();

filme.Assistido = leitor["assistido"].ToString();

filme.Ator = leitor["ator"].ToString();

filme.Diretor = leitor["diretor"].ToString();

}

return filme;

}

}

}

}

FACADE – FilmeFacade.cs

namespace Marcade.Facade

{

public class FilmeFacade

{

private DAO.FilmeDao filmeDAO;

public FilmeFacade()

{

filmeDAO = new DAO.FilmeDao();

}

public void IncluirFilme(DTO.Filme filme)

{

filmeDAO.IncluirFilme(filme);

}

public void AtualizarFilme(DTO.Filme filme)

{

filmeDAO.AtualizarFilme(filme);

}

public void DeletarFilme(DTO.Filme filme)

{

try

{

filmeDAO.DeletarFilme(filme);

}

catch (Exception ex)

{

throw ex;

}

}

public List<DTO.Filme> ListaDeFilmes()

{

try

{

return filmeDAO.ListaDeFilmes();

}

catch (Exception ex)

{

throw ex;

}

}

public DataTable SelecionarFilmes()

{

DataTable tb = new DataTable();

try

{

tb = filmeDAO.SelecionarFilmes();

}

catch (Exception ex)

{

throw ex;

}

return tb;

}

public DataTable FilmesNaoAssistidos()

{

DataTable tb = new DataTable();

try

{

tb = filmeDAO.FilmesNaoAssistidos();

}

catch (Exception ex)

{

throw ex;

}

return tb;

}

public DTO.Filme SelecionarFilmePeloId(int id)

{

try

{

return filmeDAO.SelecionarFilmePeloId(id);

}

catch (Exception ex)

{

throw ex;

}

}

public DTO.Filme ObterFilmePeloTitulo(string titulo)

{

try

{

return filmeDAO.ObterFilmePeloTitulo(titulo);

}

catch (Exception ex)

{

throw ex;

}

}

}

}

UTILS – ConectarBancoDeDados.cs

namespace Marcade.Utils

{

public class ConectarBancoDados

{

private static string conexao = "Server=localhost;Database=filmes;Uid=root;Pwd=mysql";

public static MySqlConnection ObterConexao()

{

MySqlConnection mySqlConnection = new MySqlConnection(conexao);

return mySqlConnection;

}

}

}

SCRIPTS – filmes.txt

create database filmes;

use filmes;

create table filme (id int(4) not null auto\_increment,

titulo varchar(255) not null,

ano int(4) not null,

genero varchar(50) not null,

assistido varchar(5) not null,

ator varchar(255) not null,

diretor varchar(255) not null,

primary key (id))

ENGINE = innodb;

**4 FORMULÁRIOS / CÓDIGO FONTE**

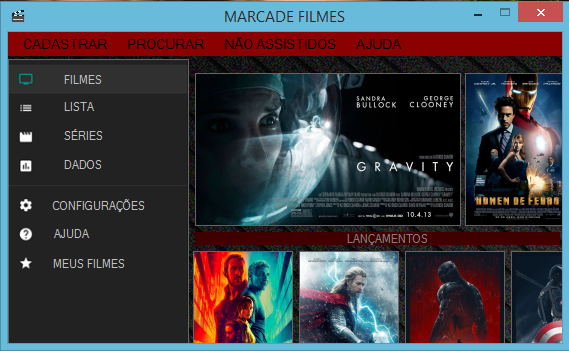


Figura 02 – Menu / Tela Inicial do programa

namespace Marcade

{

public partial class frmMenu : Form

{

public frmMenu()

{

InitializeComponent();

}

private void cADASTROToolStripMenuItem\_Click(object sender, EventArgs e)

{

frmCadastro telaCadastro = new frmCadastro ();

telaCadastro.ShowDialog();

}

private void pROCURARToolStripMenuItem\_Click(object sender, EventArgs e)

{

frmTelaPesquisa telaPesquisa = new frmTelaPesquisa();

telaPesquisa.ShowDialog();

}

private void aSSISTIDOSToolStripMenuItem\_Click(object sender, EventArgs e)

{

Marcade.Formularios.frmAssistidos telaAssistidos = new Marcade.Formularios.frmAssistidos();

telaAssistidos.ShowDialog();

}

private void iNFOToolStripMenuItem\_Click(object sender, EventArgs e)

{

Marcade.Formularios.frmInformacao telaInfo = new Marcade.Formularios.frmInformacao();

telaInfo.Show();

}

private void linkLabel2\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

Formularios.ListaDeFilmesAssistidos ListaDeFilmes = new Formularios.ListaDeFilmesAssistidos();

ListaDeFilmes.Show();

}

}

}



Figura 03 – Tela de Cadastro

namespace Marcade

{

public partial class frmCadastro : Form

{

public frmCadastro()

{

InitializeComponent();

}

public void LimparTela()

{

txtTitulo.Text = "";

cboAno.Text = "";

cboGenero.Text = "";

rbtSim.Checked = false;

rbtNao.Checked = false;

txtAtor.Text = "";

txtDiretor.Text = "";

}

private void frmCadastro\_Load(object sender, EventArgs e)

{

txtTitulo.Focus();

for (int data = 1940; data <= DateTime.Now.Year; data++)

cboAno.Items.Add(data);

rbtSim.Checked = true;

}

private void btnIncluir\_Click(object sender, EventArgs e)

{

DTO.Filme filme = new DTO.Filme();

filme.Titulo = txtTitulo.Text;

filme.Ano = cboAno.Text;

filme.Genero = cboGenero.Text;

if (rbtSim.Checked == true)

filme.Assistido = rbtSim.Text;

else if (rbtNao.Checked == true)

filme.Assistido = rbtNao.Text;

filme.Ator = txtAtor.Text;

filme.Diretor = txtDiretor.Text;

try

{

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

filmeFacade.IncluirFilme(filme);

MessageBox.Show("Incluido com sucesso");

LimparTela();

}

catch (Exception ex)

{

MessageBox.Show("Erro ao incluir" + ex.Message.ToString());

}

}

}

}

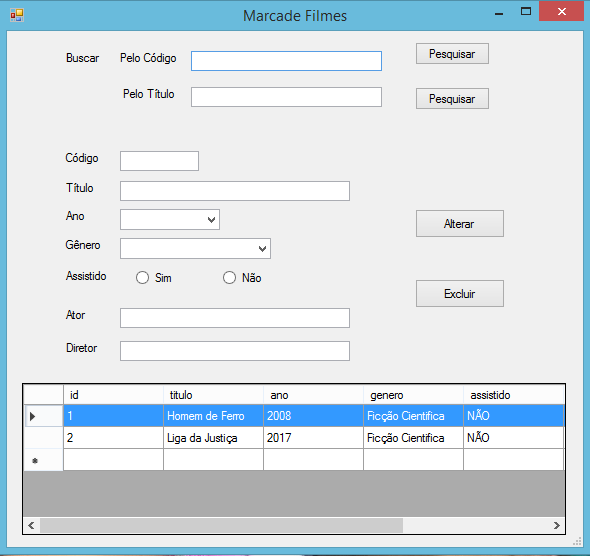


Figura 04 – Tela para pesquisar os filmes

namespace Marcade

{

public partial class frmTelaPesquisa : Form

{

public frmTelaPesquisa()

{

InitializeComponent();

}

public void LimparTela()

{

txtTitulo.Text = "";

cboAno.Text = "";

cboGenero.Text = "";

rbtSim.Checked = false;

rbtNao.Checked = false;

txtAtor.Text = "";

txtDiretor.Text = "";

}

private void frmTelaInicial\_Load(object sender, EventArgs e)

{

LimparTela();

txtBuscarId.Focus();

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

dgvFilmes.DataSource = filmeFacade.SelecionarFilmes();

}

private void btnPesquisarId\_Click(object sender, EventArgs e)

{

int id = Convert.ToInt32(txtBuscarId.Text);

DTO.Filme filme = new DTO.Filme();

try

{

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

filme = filmeFacade.SelecionarFilmePeloId(id);

txtCodigo.Text = filme.Id.ToString();

txtTitulo.Text = filme.Titulo;

cboAno.Text = filme.Ano;

cboGenero.Text = filme.Genero;

if (filme.Assistido == "SIM")

rbtSim.Checked = true;

else

rbtNao.Checked = true;

txtAtor.Text = filme.Ator;

txtDiretor.Text = filme.Diretor;

}

catch (Exception ex)

{

MessageBox.Show("Erro: " + ex.Message.ToString());

}

}

private void btnPesquisarTitulo\_Click(object sender, EventArgs e)

{

string titulo = txtBuscarTitulo.Text;

DTO.Filme filme = new DTO.Filme();

try

{

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

filme = filmeFacade.ObterFilmePeloTitulo(titulo);

txtCodigo.Text = filme.Id.ToString();

txtTitulo.Text = filme.Titulo;

cboAno.Text = filme.Ano;

cboGenero.Text = filme.Genero;

if (filme.Assistido == "SIM")

rbtSim.Checked = true;

else

rbtNao.Checked = true;

txtAtor.Text = filme.Ator;

txtDiretor.Text = filme.Diretor;

}

catch (Exception ex)

{

MessageBox.Show("Erro: " + ex.Message.ToString());

}

}

private void btnAlterar\_Click(object sender, EventArgs e)

{

DTO.Filme filme = new DTO.Filme();

filme.Titulo = txtTitulo.Text;

filme.Ano = cboAno.Text;

filme.Genero = cboGenero.Text;

if (rbtSim.Checked == true)

filme.Assistido = "Sim";

else if (rbtNao.Checked == true)

filme.Assistido = "Não";

filme.Ator = txtAtor.Text;

filme.Diretor = txtDiretor.Text;

try

{

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

filmeFacade.AtualizarFilme(filme);

LimparTela();

}

catch (Exception ex)

{

MessageBox.Show("Erro: " + ex.Message.ToString());

}

}

private void btnExcluir\_Click(object sender, EventArgs e)

{

DialogResult dialogo = MessageBox.Show("Você tem certeza!", "", MessageBoxButtons.YesNo , MessageBoxIcon.Exclamation);

if (dialogo == DialogResult.Yes)

{

DTO.Filme filme = new DTO.Filme();

filme.Id = Convert.ToInt32(txtBuscarId.Text);

try

{

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

filmeFacade.DeletarFilme(filme);

MessageBox.Show("Filme Excluído!");

}

catch (Exception ex)

{

MessageBox.Show("Erro ao alterar" + ex.Message.ToString());

}

}

}

private void dgvFilmes\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

txtCodigo.Text = dgvFilmes.SelectedRows[0].Cells[0].Value.ToString();

txtTitulo.Text = dgvFilmes.SelectedRows[0].Cells[1].Value.ToString();

cboAno.Text = dgvFilmes.SelectedRows[0].Cells[2].Value.ToString();

cboGenero.Text = dgvFilmes.SelectedRows[0].Cells[3].Value.ToString();

}

}

}

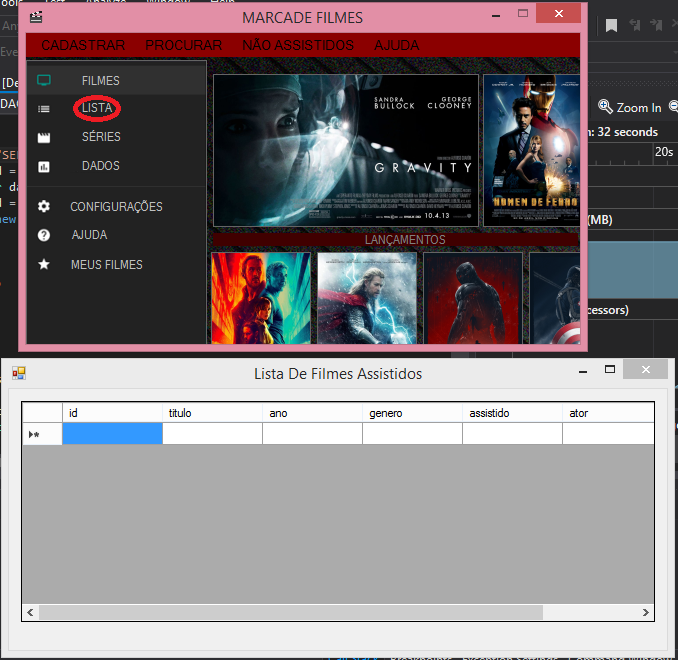


Figura 05 – Lista de Filmes Assistidos

namespace Marcade.Formularios

{

public partial class ListaDeFilmesAssistidos : Form

{

public ListaDeFilmesAssistidos()

{

InitializeComponent();

}

private void ListaDeFilmesAssistidos\_Load(object sender, EventArgs e)

{

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

dgvListaDeFilmes.DataSource = filmeFacade.SelecionarFilmes();

}

}

}

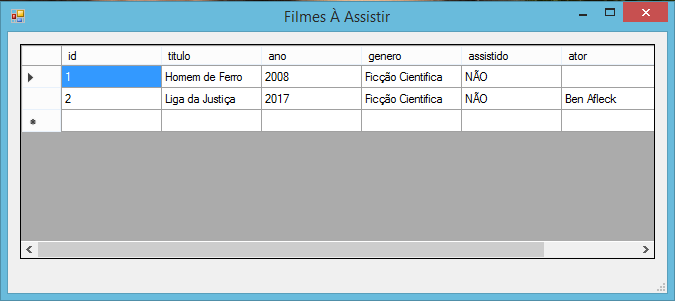


Figura 06 – Tela onde mostra todos os filmes não assistidos

namespace Marcade.Formularios

{

public partial class frmAssistidos : Form

{

public frmAssistidos()

{

InitializeComponent();

}

private void frmAssistidos\_Load(object sender, EventArgs e)

{

Facade.FilmeFacade filmeFacade = new Facade.FilmeFacade();

dgvAssistidos.DataSource = filmeFacade.FilmesNaoAssistidos();

}

}

}

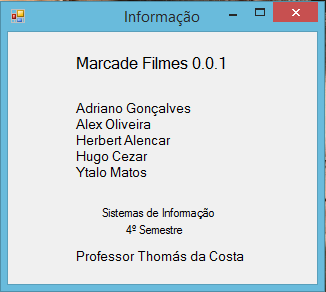


Figura 07 – Tela de Informação na Opção Ajuda do Menu

**5 CONCLUSÃO**

Ótimo trabalho proposto pelo Professor Thomás, onde tentamos colocar em prática todo o conteúdo visto nas aulas.

Agradecemos pelo empenho do professor em nos passar todo o seu conhecimento, e através deste, sem modéstia, ser bem avaliados.

E se possível, onde podemos e devemos melhorar.

**REFERÊNCIAS**

ANHANGUERA EDUCACIONAL. **Professor Thomás da Costa**.

Deitel, **C# Como programar.** Editora Pearson, 2002

**MSDN**. https://social.msdn.microsoft.com/