Licenciatura em Engenharia Informática

**Relatório De Trabalho Final de Curso**

Aplicação para Windows Phone:

Vê e Classifica!



Trabalho realizado por : André Correia nº20063401

Professor Orientador: Professora Inês Oliveira

Índice

**Índice**…………...………………………………………………………………………….2

**Abstract**………..………………………………………………………………………….4

**1. Introdução**……..………………………………………………...…………………….5

**2. Enquadramento teórico**……………………………………….…………………….6

**3. Análise e desenho**……………..……………………………………………………..7

**4. Plataforma técnica**……….…………………………………….…………………...10

**5. Resultados**..………………………………………………………..………………...11

**6. Conclusões e trabalhos futuros**…………...……………………………………..16

**Bibliografia**….…………………………………………………………………………..17

**Anexos**…………...………………………………………………………………………19

Resumo

Neste Trabalho Final de curso pretendeu-se juntar as duas maiores paixões do autor, Informática e Cinema. Assim querendo acompanhar as novas tendências criou uma aplicação para Windows Phone em que o utilizador visualiza e classifica um Trailer de Cinema.

Trailer consiste num vídeo editado (entre 1min e 30s a 4min), com excertos de um filme que tem como objetivo promovê-lo. Os trailers costumam surgir no cinema antes da visualização dum filme para promover filmes que vão estrear ou filmes que já estejam em cartaz.

Com o aparecimento de *smartphones* (telemóveis com funcionalidades avançadas que podem ser estendidas por meio de programas executados pelo seu sistema operativo) é possível ver vídeos a partir do telemóvel assim tem todo o sentido existir uma aplicação onde se possa visualizar e classificar trailers. Classificando um trailer não se está a classificar o filme em si, mas sim a sua promoção, isto é, se o filme promete ser bom ou não.

Abstract

In this final project of the course the student wanted to join his two biggest passions, Computer and Cinema. Just wanting to follow the new trends has created an app for Windows Phone where the user view and rates a Movie Trailer.

Trailer is a video edited (between 1min to 4min), with excerpts from a film that aims to promote it. The trailers usually appear in the cinemas before viewing a movie to promote films that will debut or movies that are already on display.

With the emergence of smartphones (phones with advanced functionality that can be extended through programs run by his operating system) you can watch videos from your phone the it just makes sense if there is an application where you can view and classify trailers. Classifying a trailer is not classifying the film itself, but its promotion, that is if the movie promises to be good or not.

1.Introdução

O objetivo deste trabalho consiste numa aplicação móvel para Windows Phone. A aplicação é gerida por um site que está ligado a uma base de dados que contem todas as informações.

As plataformas utilizadas são as seguintes:

* Microsoft Visual Web Developer 2010 Express
* Microsoft Visual Studio 2010 Express for Windows Phone
* Microsoft Expression Blend 4
* SQL Server Management Studio

Como tecnologia utilizaram-se as seguintes linguagens:

* UML, para especificar as funcionalidades
* C# para a programação
* ASP.NET para a programação
* Silverlight (XAML)
* WCF Service, para fazer a interligação dos dados entre o site centraliza de e o dispositivo móvel
* XHTML e CSS para obter a estrutura do site

Neste Projeto o Utilizador poderá classificar um Trailer desde 1 estrela a 5 estrelas.

A organização do documento é a seguinte:

* Na secção 1, apresenta-se a introdução.
* Na secção 2, apresenta-se o Enquadramento Teórico
* Na secção 3, apresenta-se a Análise e Desenho
* Na secção 4, apresenta-se a Plataforma Técnica
* Na secção 5, apresenta-se os Resultados
* Na secção 6. apresenta-se as Conclusões e Trabalhos Futuros

2.Enquadramento Teórico

O Windows Phone 7 é um sistema operativo móvel, desenvolvido pela Microsoft, sucessor da plataforma Windows Mobile, (apesar de totalmente incompatível com esta) que, ao contrário do predecessor, é focado no mercado consumidor, em vez do mercado empresarial.

O UML ( Unified Modeling Language ) é uma linguagem de modelos em que nos permite ver o desenho do projeto e a comunicação entre objetos. Dentro dos diagramas disponibilizados, os casos de uso definem a estrutura do Projeto.

O Silverlight é uma ferramenta de desenvolvimento para a criação de experiencias atrativas e interativas para a web e aplicações móveis. O Silverlight é um Plug-in gratuito, cedido por .Net e compatível com vários navegadores, dispositivos e sistemas operativos, trazendo um novo nível de interatividade onde a Web Funciona.

Além do Windows Phone, as plataformas mais utilizadas para Mobile são: \* Android - sistema operativo baseado no Linux para dispositivos móveis, como o próprio Google Nexus One, Nexus S ou Galaxy Nexus, bem como por outros fabricantes de celulares como a HTC, Samsung, entre outros. É desenvolvido pela Open Handset Alliance, liderada pelo Google e outras empresas.

\* IOS - anteriomente chamado de iPhone OS, é um sistema operativo móvel da Apple Inc. desenvolvido originalmente para o iPhone, também é usado em iPod Touch, iPad e Apple TV.

Classificar todo o tipo de variadas coisas está em voga nestes últimos tempos havendo sites em que se podem classificar filmes, livros, álbuns de música.

3.Análise e Desenho

**Casos de Uso**

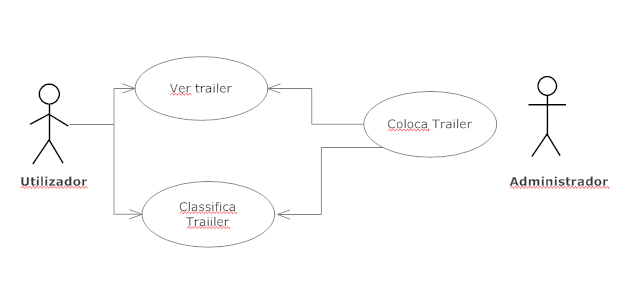


Ilustração 1 - Use Case Diagram

**Coloca Trailer** – O Administrador coloca o Trailer na Aplicação através de um formulário de upload de ficheiro. Neste formulário, existem também outros campos que devem ser introduzidos, como, por exemplo o Ano e Nome (do filme), etc.

**Ver Trailer** – O Utilizador visualiza o Trailer, a partir da aplicação móvel.

**Classifica Trailer** – O Utilizador Classifica o Trailer, escolhendo de uma a cinco estrelas.

**Diagrama de Classes**

As tabelas existentes na base de dados são as seguintes:

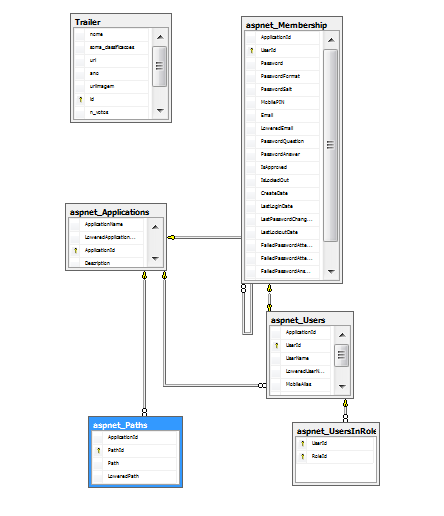


Ilustração 2 - Diagrama de Classes

A Tabela **Trailer** contém:

* Nome – nome do trailer
* Soma­\_classificacoes – A soma das classificações faz a soma para de pois se fazer a média.
* url – Url do trailer
* ano – Ano do Trailer
* urlimagem – Url da Imagem
* id – Id do Trailer
* n\_votos – Número de votos

A **classificação** é determinada pela divisão da **soma das classificações** e o **número de votos**.

As Tabelas **Membership** são geradas pela aplicação e servem para dar suporte de autenticação (login) ao site.

4.Plataforma Técnica

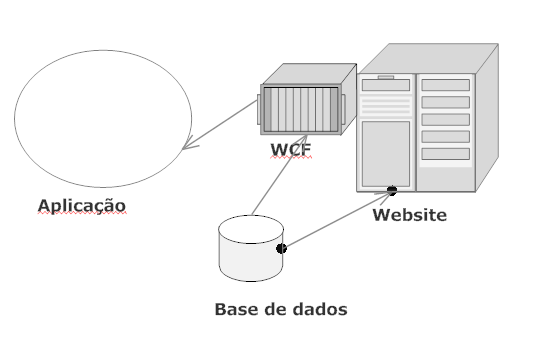


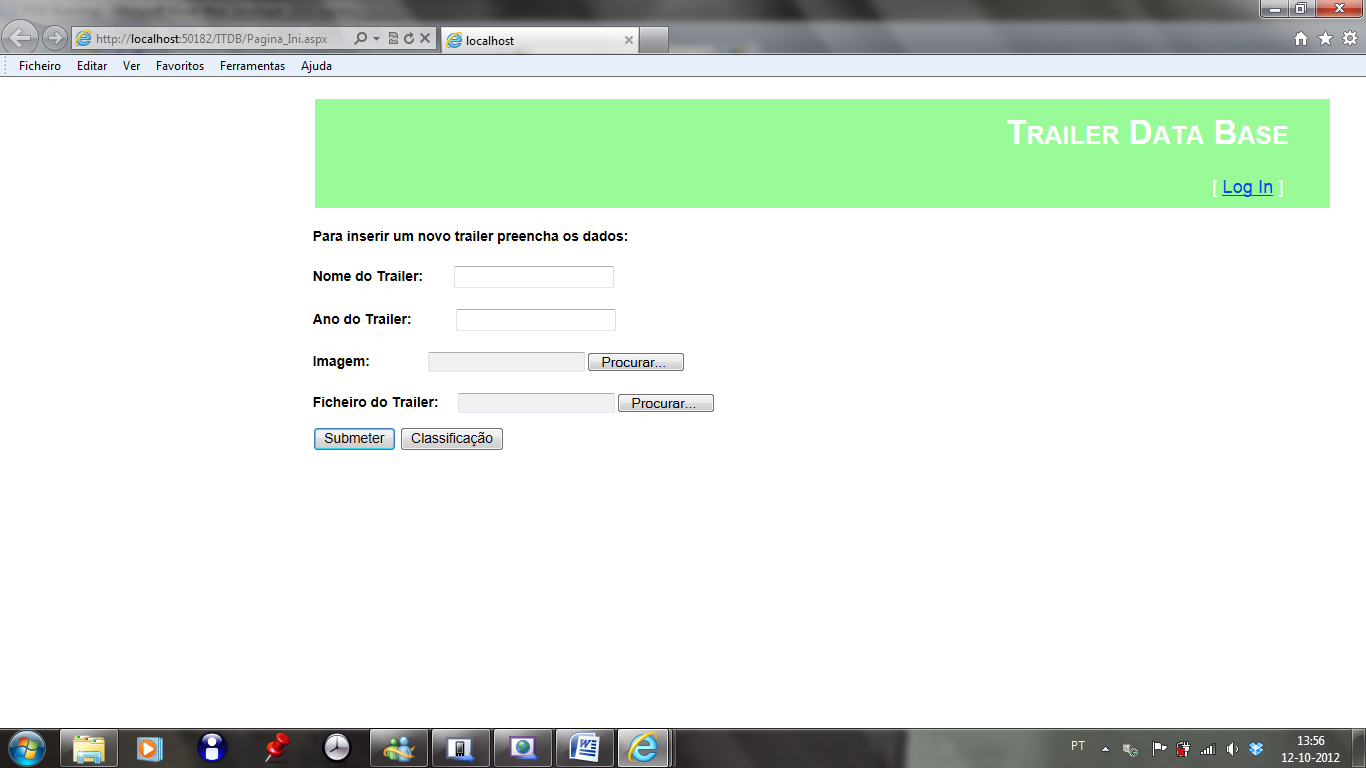
Ilustração 3 – Plataforma Técnica

Como se vê no esquema anterior a Aplicação retira as suas informações ao WCF ( WEbservice ) que está ligado ao Website. O Website está ligado á Base de Dados no SQL que está também ligado ao WCF.

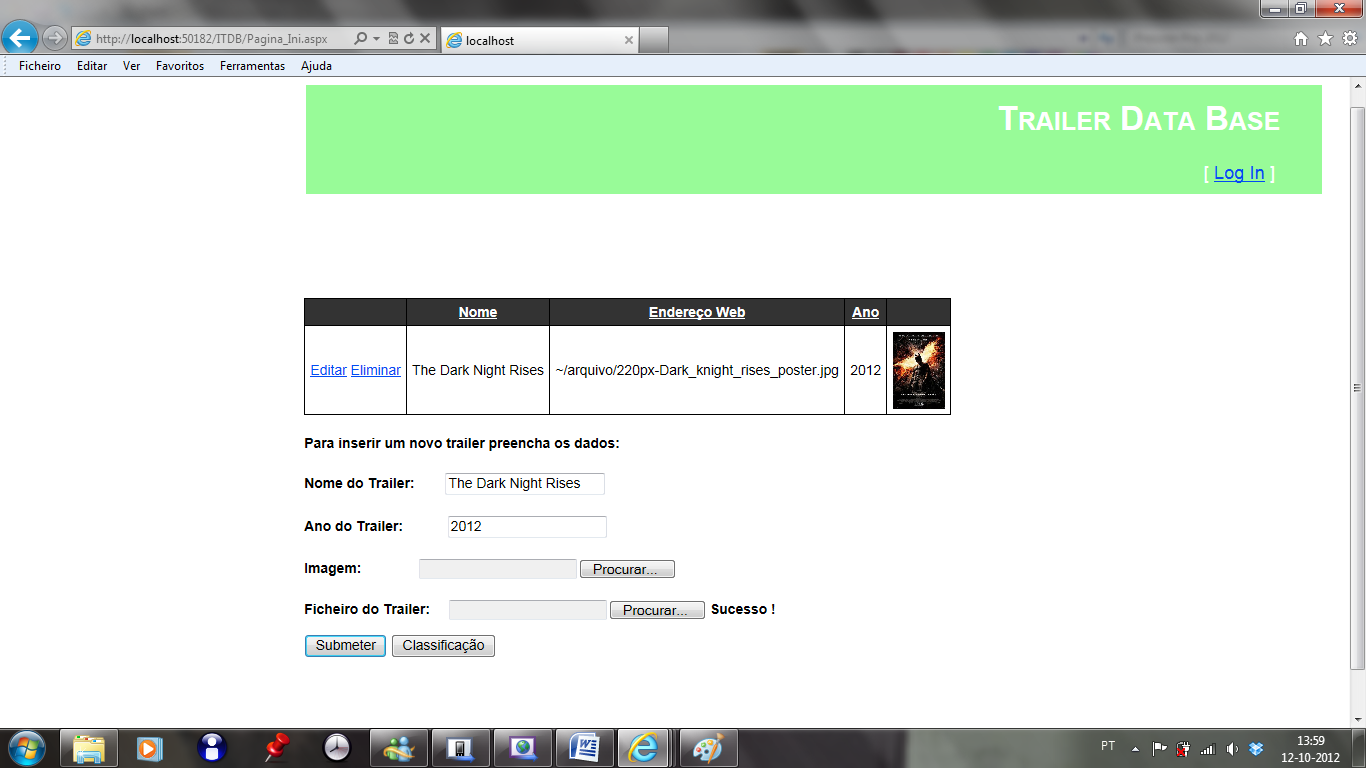
5.Resultados

**Website**

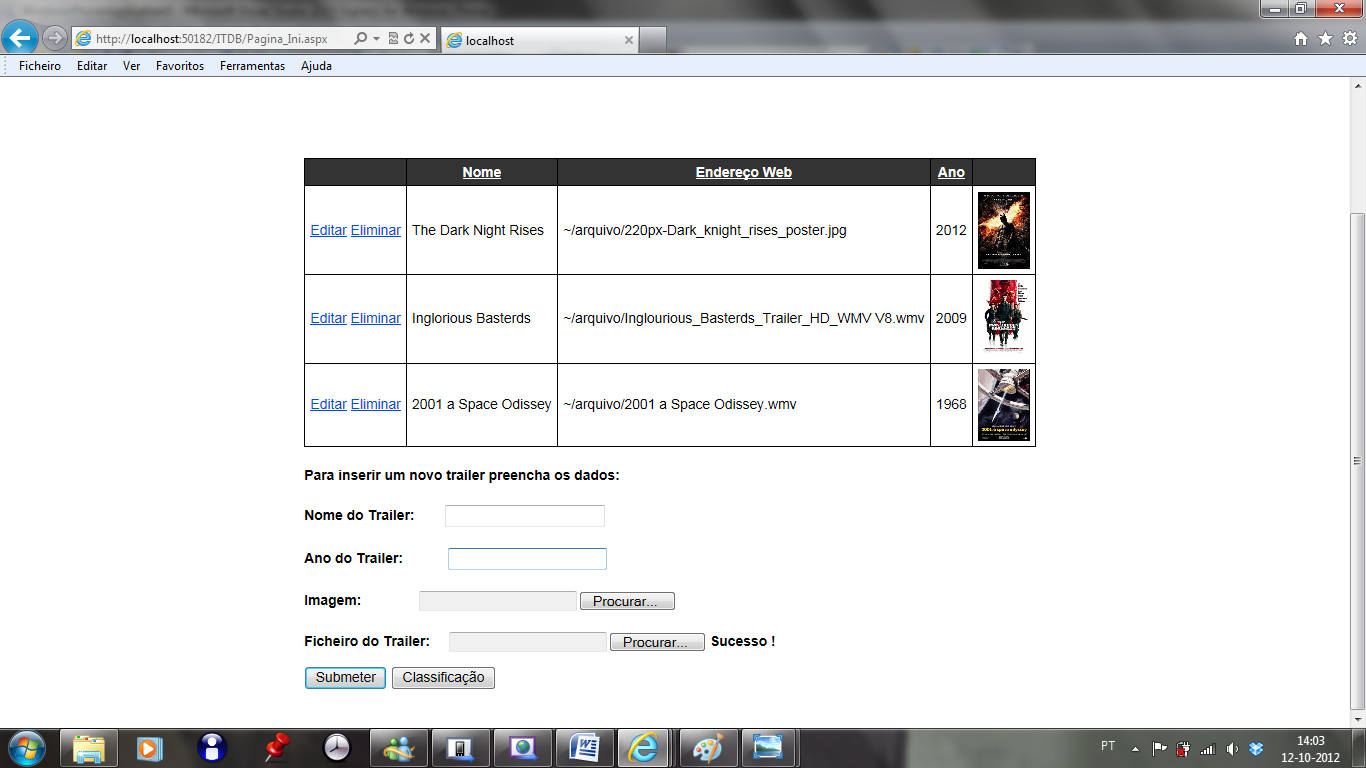
O Administrador cede ao site onde coloca os trailers e seus dados referentes



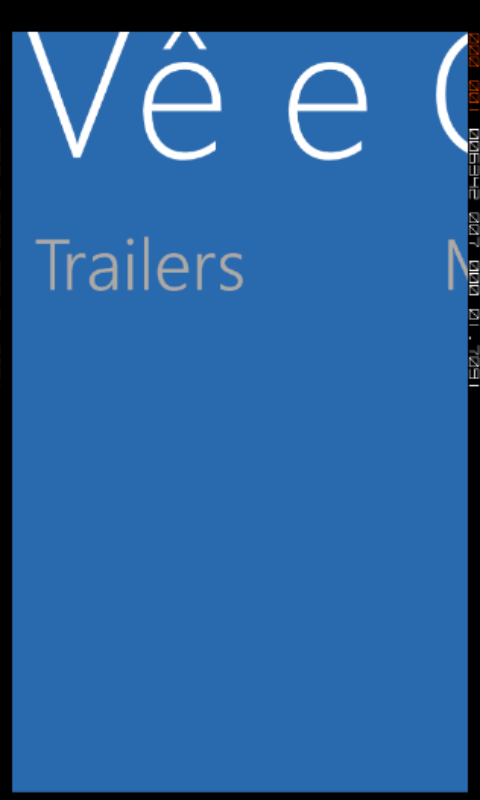
O Trailer já colocado no site



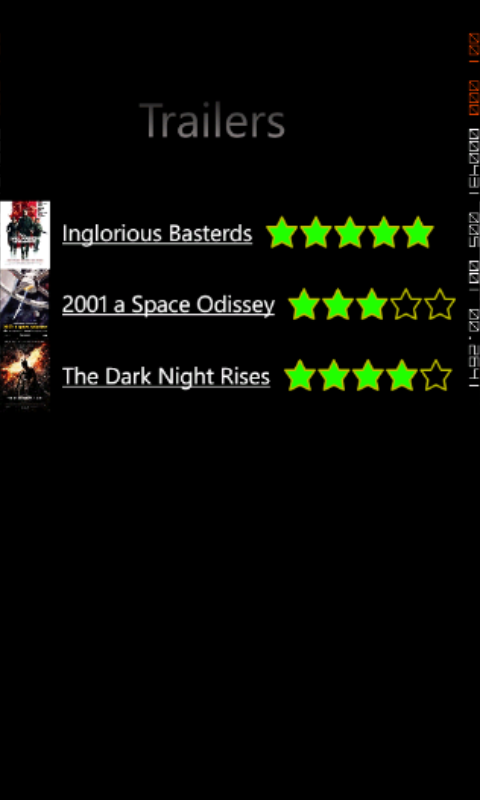
Vários trailers já colocados

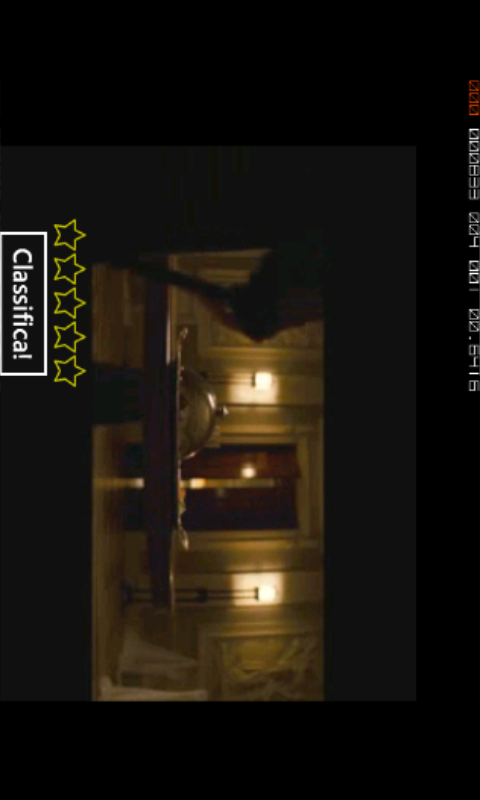


**Aplicação Móvel**

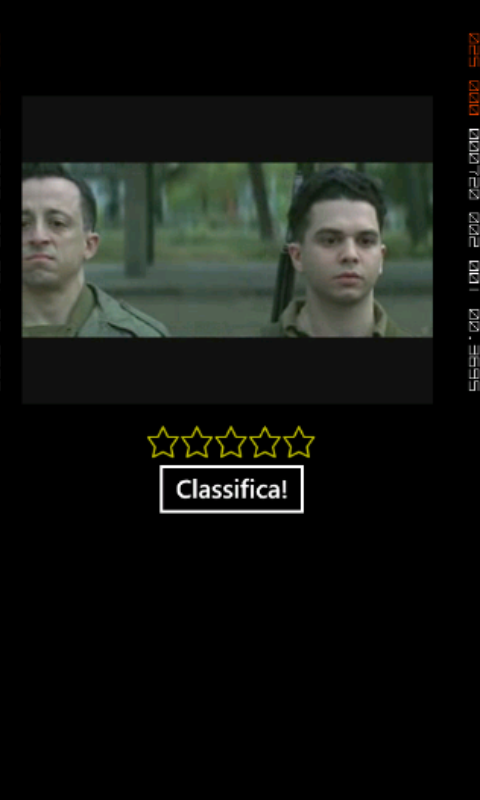
Início - surge no ecrã uma página em panorama onde se pode escolher os trailers ou os mais vistos (ver Trabalhos Futuros).

Listagem de Trailers – Surge a Lista com os trailers e a sua respectiva classificação. Aqui o Utilizador pode escolher o Trailer.



Visualização e Classificação – Aqui o utilizador pode ver e classificar o trailer.

Modo Portrait/Landscape (dispositivo na vertical/horizontal)



6.Conclusões e Trabalhos Futuros

Com este trabalho o aluno melhorou os seus conhecimentos em programação, conheceu novas plataformas aumentando o seu gosto por desenvolver aplicações móveis.

O desenvolvimento de uma aplicação móvel foi um desafio para o aluno mas bastante compensador conhecer ferramentas como Microsoft Visual Studio para Windows Phone e Expression Blend.

A aplicação foi criada de forma a que seja expandida facilmente:

* Criar um Log in no Website do administrador
* Melhor design na Aplicação
* A Aplicação conter uma página onde o utilizador vê uma lista com os trailers mais bem classificados.

Bibliografia

* SQL Server 2008 Curso Completo, Alberto Magalhães, FCA
* Beginning Silverlight, Robert Lair, Apress
* Pro Business Aplications with Silverlight 4, Chris Andersson, Apress
* <http://msdn.microsoft.com>
* <http://www.microsoft.com/expression/products/Blend_Overview.aspx>

Agradecimentos

O aluno agradece à sua família por todo o apoio recebido, à sua namorada por sempre confiar e especialmente à Professora Orientadora Inês Oliveira por toda a ajuda e apoio.

Anexos

**Website**

Código fonte da Página Inicial

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Model;

public partial class Default2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

if (Nome.Text == "" || Ano.Text == "")

{

Erro.Text = "Tem de introduzir o nome e o ano...";

}

else

{

int tam = 52428800;

if (Upload.HasFile && UploadImg.HasFile )

{

if (Upload.PostedFile.ContentLength < tam)

{

string caminho = Server.MapPath("~/arquivo/") + Upload.FileName;

Upload.SaveAs(caminho);

string caminhoi = Server.MapPath("~/arquivo/") + UploadImg.FileName;

UploadImg.SaveAs(caminhoi);

using (ModelContainer db = new ModelContainer())

{

Trailer novo = new Model.Trailer();

novo.nome = Nome.Text;

novo.ano = Ano.Text;

novo.url = "~/arquivo/" + Upload.FileName;

novo.urlimagem = "~/arquivo/" + UploadImg.FileName;

db.AddToTrailer(novo);

db.SaveChanges();

}

Erro.Text = "Sucesso !";

GridView1.DataBind();

}

else Erro.Text = "Excede o tamanho!";

}

}

}

protected void Ano\_TextChanged(object sender, EventArgs e)

{

}

protected void EntityDataSource1\_Deleting(object sender, EntityDataSourceChangingEventArgs e)

{

string url = (e.Entity as Trailer).url;

string caminhoCompleto = Server.MapPath(url);

Erro.Text = "CAMINHO CONVERTIDO = " + caminhoCompleto;

try

{

System.IO.File.Delete(caminhoCompleto);

Erro.Text += " -- Delete OK";

}

catch (System.IO.FileNotFoundException exception)

{

Erro.Text = "Ficheiro não encontrado";

}

e.Cancel = false;

}

protected void Button2\_Click(object sender, EventArgs e)

{

}

}

**Aplicação**

Código Fonte da Página de Listagem

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Animation;

using System.Windows.Shapes;

using Microsoft.Phone.Controls;

using WindowsPhoneApplication3.TrailerReference;

namespace WindowsPhoneApplication3

{

public partial class Data : PhoneApplicationPage

{

private string prefixo = "http://localhost:50182/ITDB";

public Data()

{

InitializeComponent();

}

void client\_GetTrailerCompleted(object sender, GetTrailerCompletedEventArgs e)

{

List<Trailer> res\_alterado = (from t in e.Result select

new Trailer { id = t.id,

nome = t.nome,

classificacao = t.n\_votos==0 ? 0 : (t.soma\_classificacoes/t.n\_votos)\*2,

url = prefixo+t.url.Substring(1),

urlimagem = prefixo + t.urlimagem.Substring(1)

}).ToList();

Trailer i = e.Result.ElementAt(0);

this.lista\_trailers.DataContext = res\_alterado;

}

private void HyperlinkButton\_Click(object sender, RoutedEventArgs e)

{

try

{

HyperlinkButton button = (HyperlinkButton)sender;

string url = string.Format("/Phonepage1.xaml?ID={0}&V={1}", button.TabIndex, button.Tag);

NavigationService.Navigate(new Uri(url, UriKind.Relative));

}catch(Exception x) {

int i=0;

}

}

private void PhoneApplicationPage\_Loaded(object sender, RoutedEventArgs e)

{

ServiceClient client = new ServiceClient();

client.GetTrailerCompleted += new EventHandler<GetTrailerCompletedEventArgs>(client\_GetTrailerCompleted);

client.GetTrailerAsync();

}

}

}

Código Fonte da Pagina da Classificação

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Animation;

using System.Windows.Shapes;

using Microsoft.Phone.Controls;

using WindowsPhoneApplication3.TrailerReference;

namespace WindowsPhoneApplication3

{

public partial class PhonePage1 : PhoneApplicationPage

{

int id;

public PhonePage1()

{

InitializeComponent();

}

protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)

{

}

private void MediaElement\_MediaOpened(object sender, RoutedEventArgs e)

{

int i = 0;

}

private void PhoneApplicationPage\_Loaded(object sender, RoutedEventArgs e)

{

string url = NavigationContext.QueryString["V"].ToString();

id = Convert.ToInt32(NavigationContext.QueryString["ID"]);

Player.Source = new Uri(url.ToString(), UriKind.Absolute);

}

private void Player\_MediaFailed(object sender, ExceptionRoutedEventArgs e)

{

int i = 0;

}

private void button1\_Click(object sender, RoutedEventArgs e)

{

int rating = starRatingControl1.Rating / 2;

ServiceClient client = new ServiceClient();

client.VotaTrailerCompleted+=new EventHandler<System.ComponentModel.AsyncCompletedEventArgs>(client\_VotaTrailerCompleted);

client.VotaTrailerAsync(id, rating);

}

void client\_VotaTrailerCompleted(object sender, System.ComponentModel.AsyncCompletedEventArgs e)

{

NavigationService.GoBack();

}

}

}