



Protótipo de uma aplicação rica de internet para monitoramento de vídeo através de streaming e Silverlight

Thiago da Silva Negherbon – Acadêmico
Roosevelt dos Santos Júnior – Orientador

Roteiro

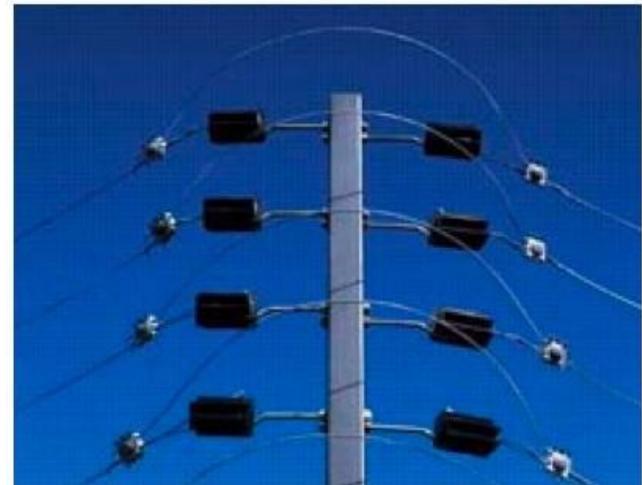


- Introdução
 - Objetivos do trabalho
- Fundamentação teórica
 - RIA, Silverlight, XAML, *streaming*, compressão, DirectShow, trabalhos correlatos
- Desenvolvimento do protótipo
 - Requisitos principais, especificação, implementação, resultados e discussão
- Conclusão
 - Extensões

Introdução



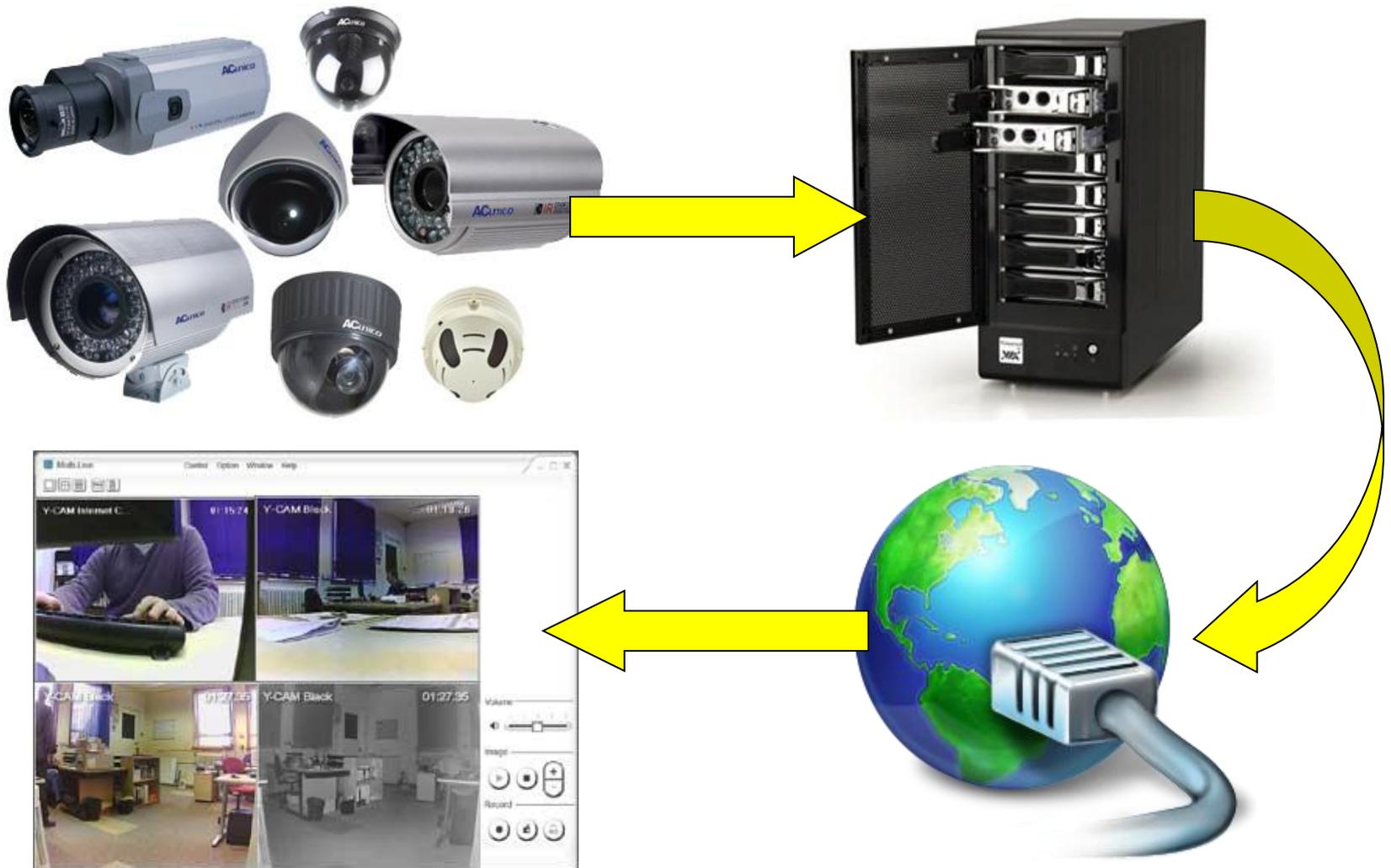
Introdução



Introdução



Objetivos do trabalho



Fundamentação teórica

Rich Internet Application



[Search the repository](#) | [Browse for research](#) | [About this repository](#) | [Repository policies](#) | [FAQ](#) | [Contact](#)

Patterns in HCI: A critical review

Full text available: [Patterns in HCI: A critical review](#)

Abstract:	This article presents a critical review of patterns and pattern languages in human-computer interaction (HCI). In recent years, patterns and pattern languages have received considerable attention in HCI for their potential as a means for developing and communicating information and knowledge to support good design. This review examines the background to patterns and pattern languages in HCI, and seeks to locate pattern languages in relation to other approaches to interaction design. The review explores four key issues: What is a pattern? What is a pattern language? How are patterns and pattern languages used? and How are values reflected in the pattern-based approach to design? Following on from the review, a future research agenda is proposed for patterns and pattern languages in HCI.
ISSN:	1532-7051
DOI:	10.1207/s15327051hci2101_3
Reference:	Dearden, A., Finlay, J. (2006) Patterns in HCI: A critical review. Human Computer Interaction Journal , 21 (1) March, pp.49-102.
Item type:	Journal Item
Copyright:	Taylor & Francis (2006) Uploaded in accordance with the publisher's self archiving policy.
Published status:	Published
Published URL:	http://www.informaworld.com/smpp/content~content=a795059926~db=all
Refereed status:	Refereed
Classification:	T - Technology
Deposited on:	2nd July 2009, 8:26
Author(s):	Dearden, Andy , Finlay, Janet
Publisher:	Taylor & Francis
Deposited by:	Sheppard, Nick
Subject(s):	Human-computer interaction , Ergonomics design and management

[Leeds Met Home](#)

[intraLibrary Login](#)

Rich Internet Application



Rich Internet Application



Microsoft®
Silverlight™

Silverlight



The screenshot shows a Silverlight application running in a browser. The application has a dark purple header with the 'NetAdvantage' logo and navigation links for 'COMMUNITY', 'BLOGS', and 'DOCUMENTATION'. Below the header is a toolbar with various icons for navigation and actions. The main content area displays a user profile for 'Ana Trujillo' with contact information, a bar chart showing data for Jan, Apr, May, Jun, Sept, and Nov, and a list of other contacts. The interface is designed with a modern, clean aesthetic typical of Silverlight applications.

Home > Avee Talent Management

AVEE TALENT MANAGEMENT
Avee uses the XamTileView to make contact management engaging. Maximize a tile to see client details. The XamTileView supports Silverlight DataBinding, enabling layouts like Avee via templating.

Signed in as Sam Bulagira [Help](#) [Sign Out](#)

avee
Talent Management

CURRENT VIEW CONTACTS

Ana Trujillo
Ana Trujillo Emparedados y helados
phone: (5) 555-4729
fax: (5) 555-3745
atrujillo@example.com

SYSTEM

Last Payment:	10/14/2008
Last Shipment:	10/11/2008
Last Invoice:	10/5/2008
Last Order:	10/4/2008

CUSTOMER

07 Aug 2008
Customer called to update contact info; previous contact was let go due to incompetencia.

23 Sep 2008
Customer called back to update shipping information for order 10926.

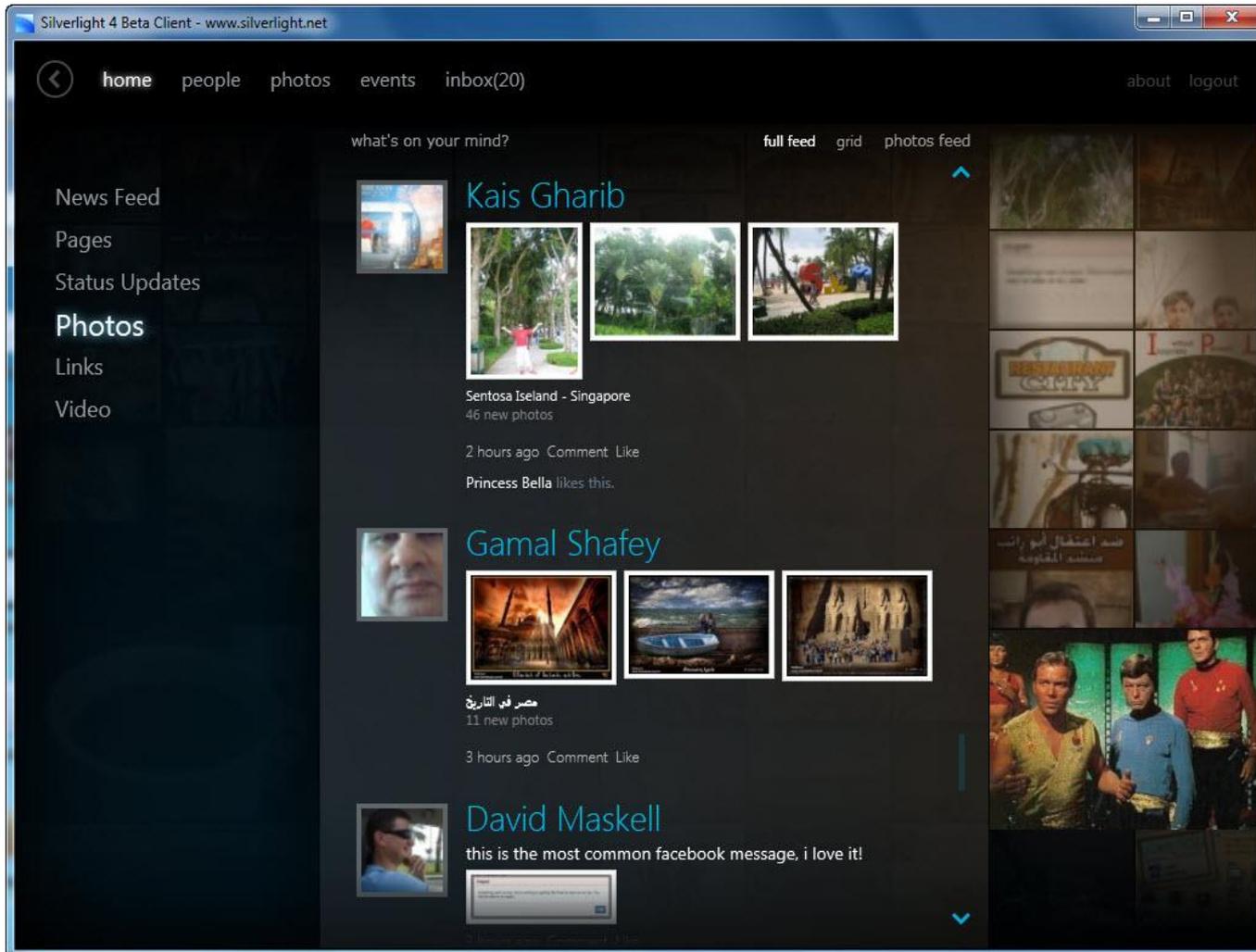
Bar Chart Data:

Month	Value
Jan	Low
Apr	Low
May	High
Jun	High
Sept	Medium
Nov	Low

Contacts List:

- Maria Anders
- Antonio Moreno
- Thomas Hardy
- Christina Berglund
- Hanna Moos
- Elizabeth Citeaux
- Martín Sommer
- Laurence Lebihan
- Jon Hammond

Silverlight



Silverlight



Media browser ^{Beta}

User: visitor
Last connection:
Level: 0
[Config](#) [Fullscreen](#)

Medias

Type: All Image Mp3 Streaming

Filter: File: Folder:

Current selection: [Open](#) [Edit](#) [Delete](#) [Download](#) [Refresh](#)

Filename: ferrari_aurea_gt_800.jpg
Date added: 12/14/2008 4:02:09 PM
Uploader: visitor2
Description:

A grid of 25 thumbnails showing various images of cars and airplanes. The first row contains five airplane images. The second row contains five more airplane images. The third row contains five car images, including a red Ferrari. The fourth row contains five car images. The fifth row contains five car images. The sixth row contains five car images.

Picture

A large, high-resolution image of a red Ferrari sports car, specifically a Ferrari F430, shown from a front-three-quarter view. The car is parked on a white surface against a dark background. A small copyright notice "© 2004 DGFdesign" is visible in the bottom right corner of the image.

Upload

Filename	State	Size	Progress

Files: 0 Uploaded: 0 KB

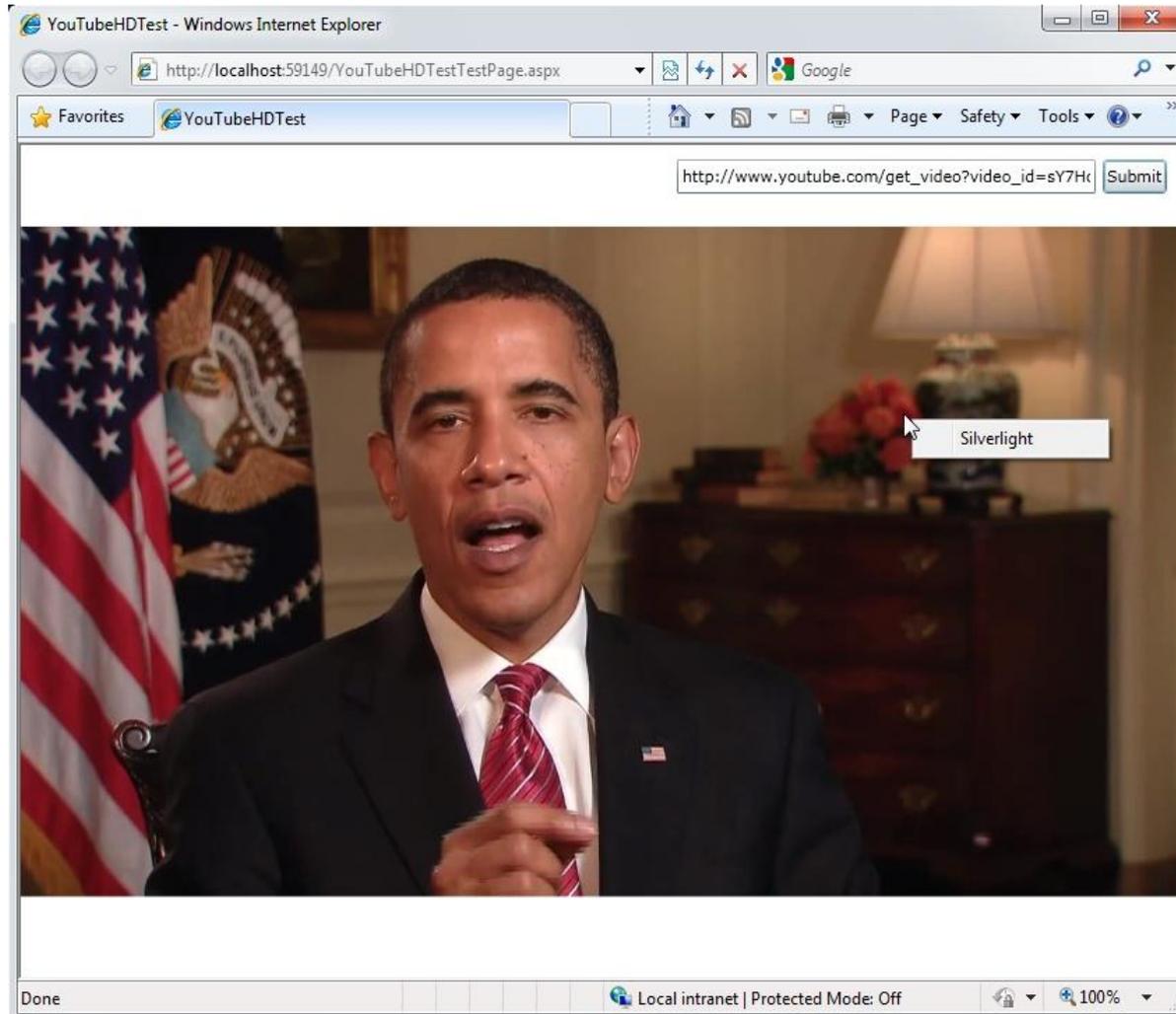
[Select Files](#) Folder name:

[Upload](#) Fill this information: Event date:

[Clear](#) File name prefix:

Level:

Silverlight



eXtensible Application Markup Language



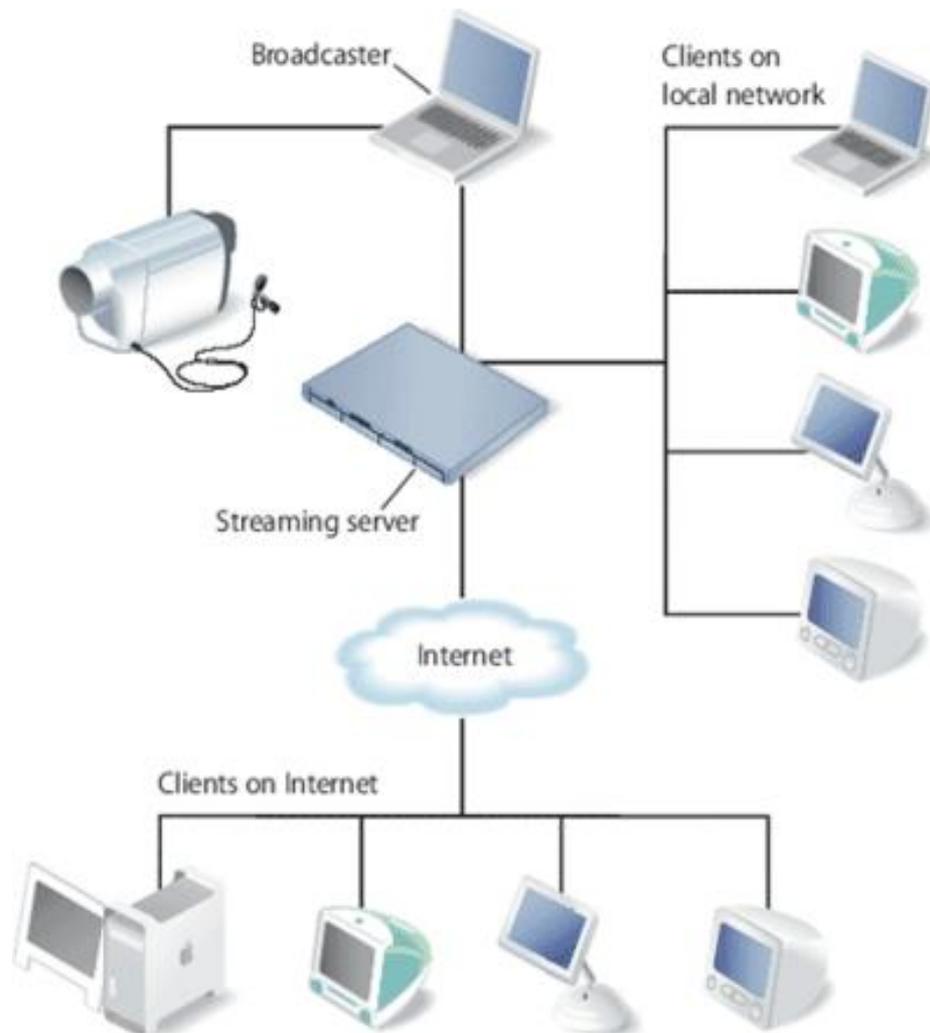
```
<UserControl ... >
  <StackPanel Background="White">
    <ContentControl ...
      <inputToolkit:TimePicker ...
    </inputToolkit:TimePicker>
  </ContentControl>
  <ContentControl ...
    Header}" />
    <inputToolkit:TimePicker ...
  </inputToolkit:TimePicker>
  <ContentControl ...
    Header}" />
    <inputToolkit:TimePicker ...
      <inputToolkit:TimePicker ...
    </inputToolkit:TimePicker>
  </inputToolkit:TimePicker>
  </ContentControl>
</StackPanel>
</UserControl>
```



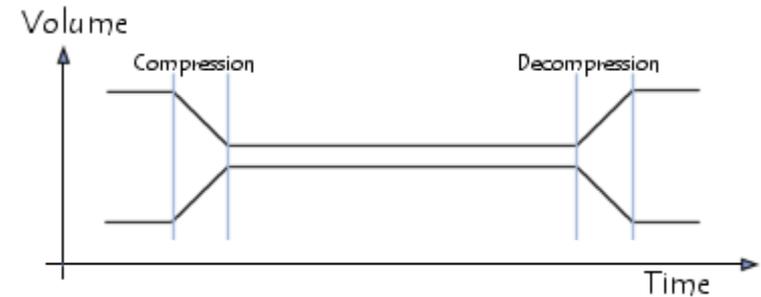
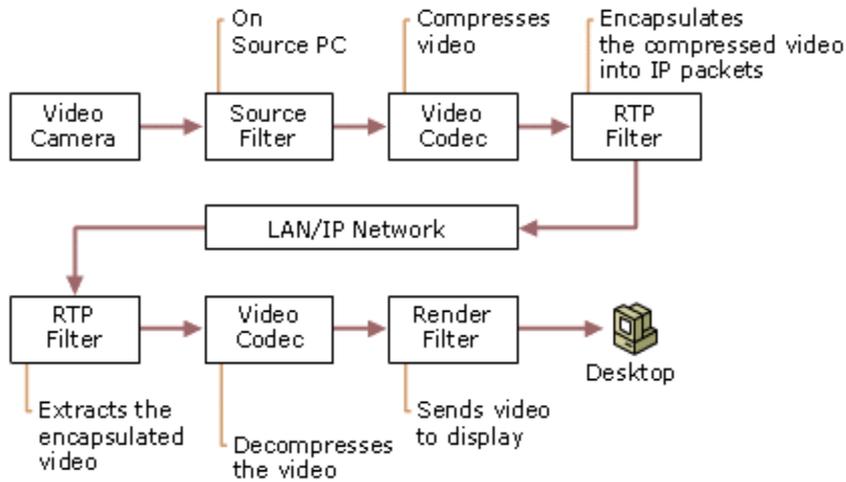
The image shows three examples of TimePicker controls. Each example consists of a header and a control. The first example, 'Default', shows a standard TimePicker with a text box, up/down arrows, and a clock icon. The second example, 'Open on Button Hover', shows a similar TimePicker but with a tooltip that appears when the up/down arrows are hovered over. The third example, 'Using the Range Popup', shows a TimePicker with a range popup that appears when the up/down arrows are clicked.

```
cResource Header}" />
Style="{StaticResource
"/>
Style="{StaticResource
:Popup />
```

Streaming



Compressão

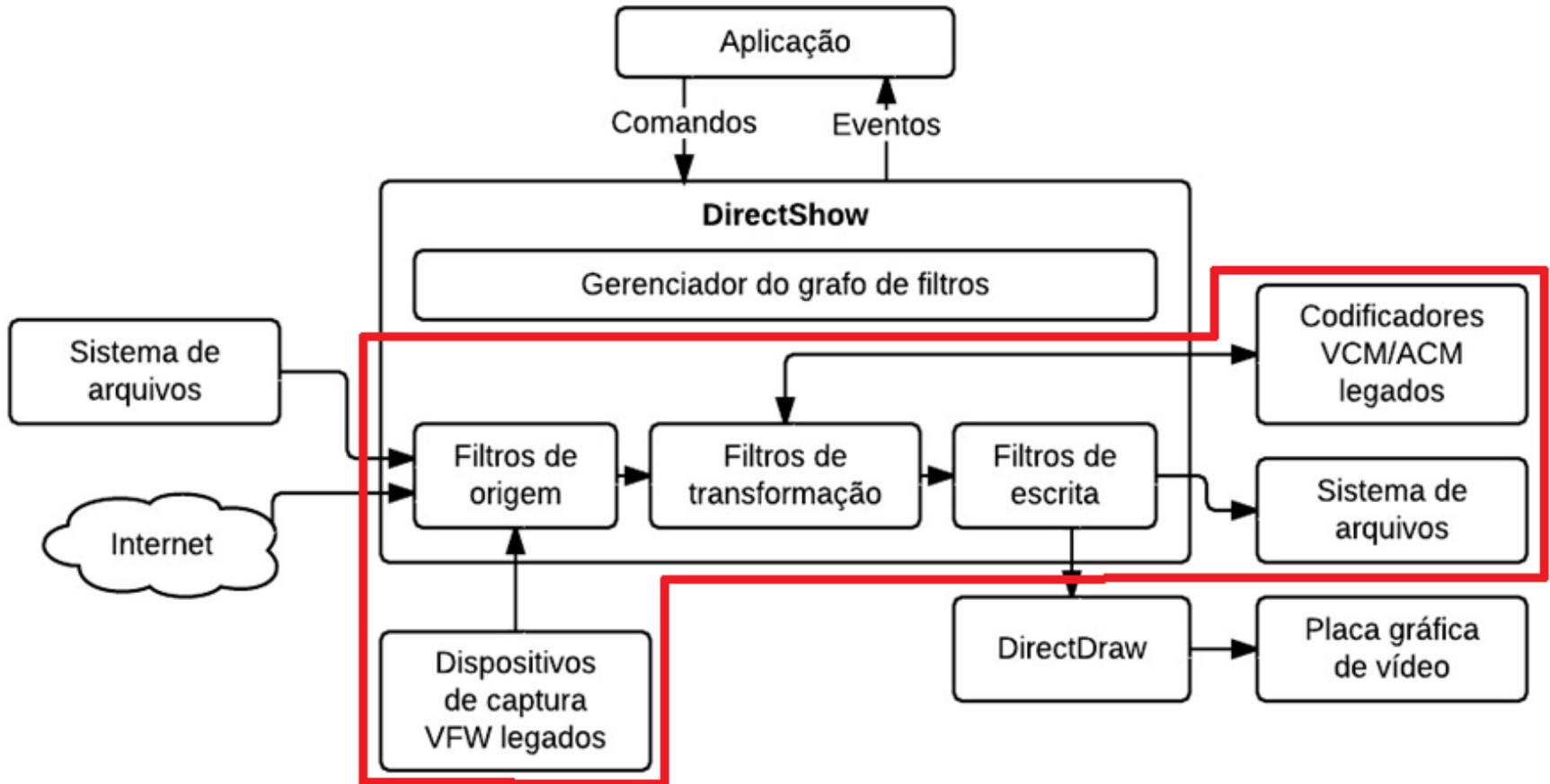


Compressão

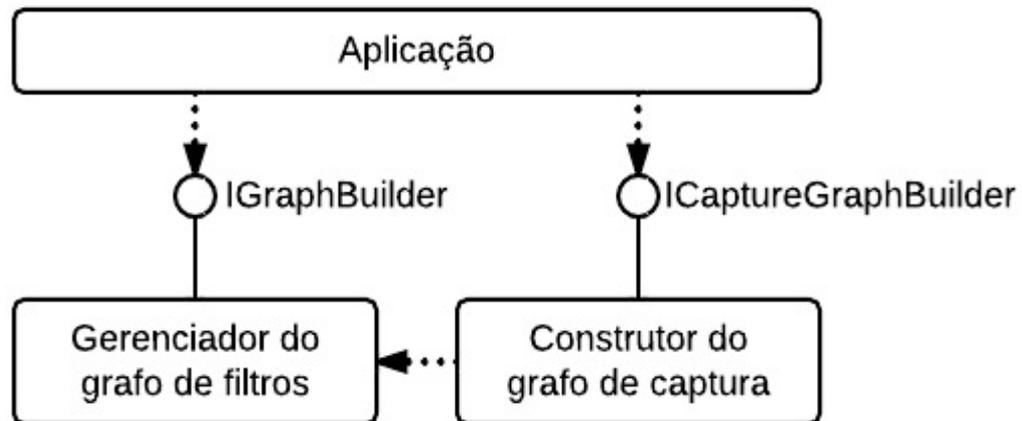


Aplicação	Taxa de dados	
	Descomprimido	Comprimido
Voz	64 kbps	2 – 4 kbps
Vídeo conferência (15 fps) tamanho do quadro 352 x 240, 24 <i>bits/pixel</i>	30,41 mbps	64 – 768 kbps
Transmissão de vídeo (30 fps) tamanho do quadro 720 x 480, 24 <i>bits/pixel</i>	248,83 mbps	3 – 8 mbps
Televisão de alta definição (59,94 fps) tamanho do quadro 1280 x 720, 24 <i>bits/pixel</i>	1,33 gbps	20 mbps

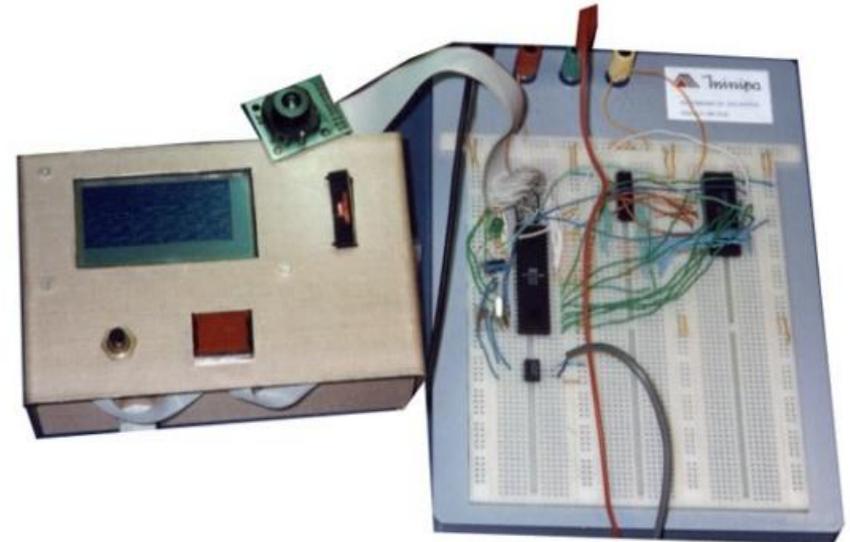
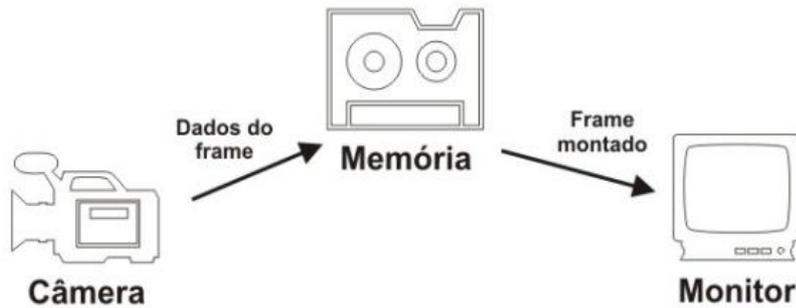
DirectShow



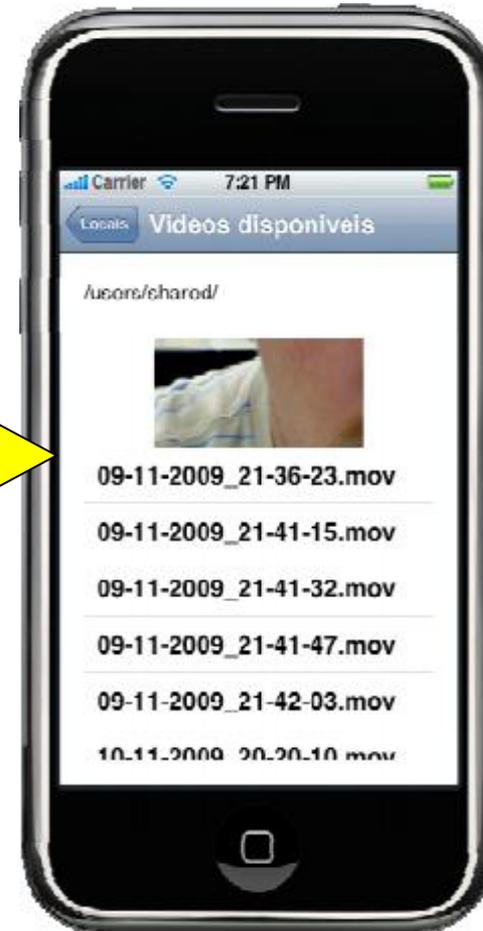
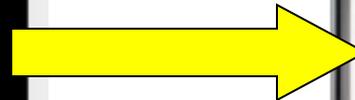
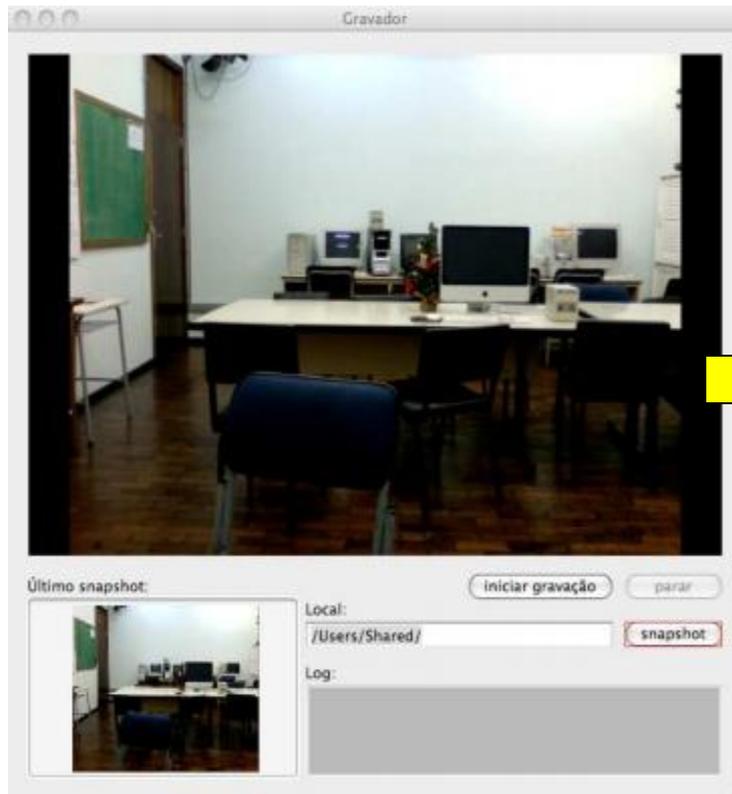
DirectShow



Trabalho de Mереge Neto (2004)



Trabalho de Carlassara (2009)



Desenvolvimento do protótipo

Requisitos funcionais



- RF01: A aplicação cliente deve disponibilizar um painel que ocupe o maior espaço possível da janela do navegador onde o usuário poderá interagir.

Requisitos funcionais



- RF02: A aplicação cliente deve disponibilizar um quadro para cada *streaming* de câmera que estiver recebendo.

Requisitos funcionais



- RF03: A aplicação cliente deve possuir um menu onde podem ser escolhidas as câmeras escondidas que se deseja restaurar.

Requisitos funcionais



- RF04: A aplicação cliente deve permitir redimensionamento e movimentação livres (limitado pelo tamanho do painel) de cada quadro de câmera.

Requisitos funcionais



- RF05: A aplicação cliente deve permitir fechar cada quadro de câmera.

Requisitos funcionais



- RF06: A aplicação cliente deve permitir parar e continuar a exibição das imagens de cada janela de câmera.

Requisitos funcionais



- RF07: A aplicação servidora deve ler a entrada de cada câmera conectada ao computador.

Requisitos funcionais



- RF08: A aplicação servidora deve processar os dados e criar o *streaming* de vídeo de cada câmera.

Requisitos funcionais



- RF09: A aplicação servidora deve disponibilizar os *streamings* para leitura na internet.

Diagrama de casos de uso

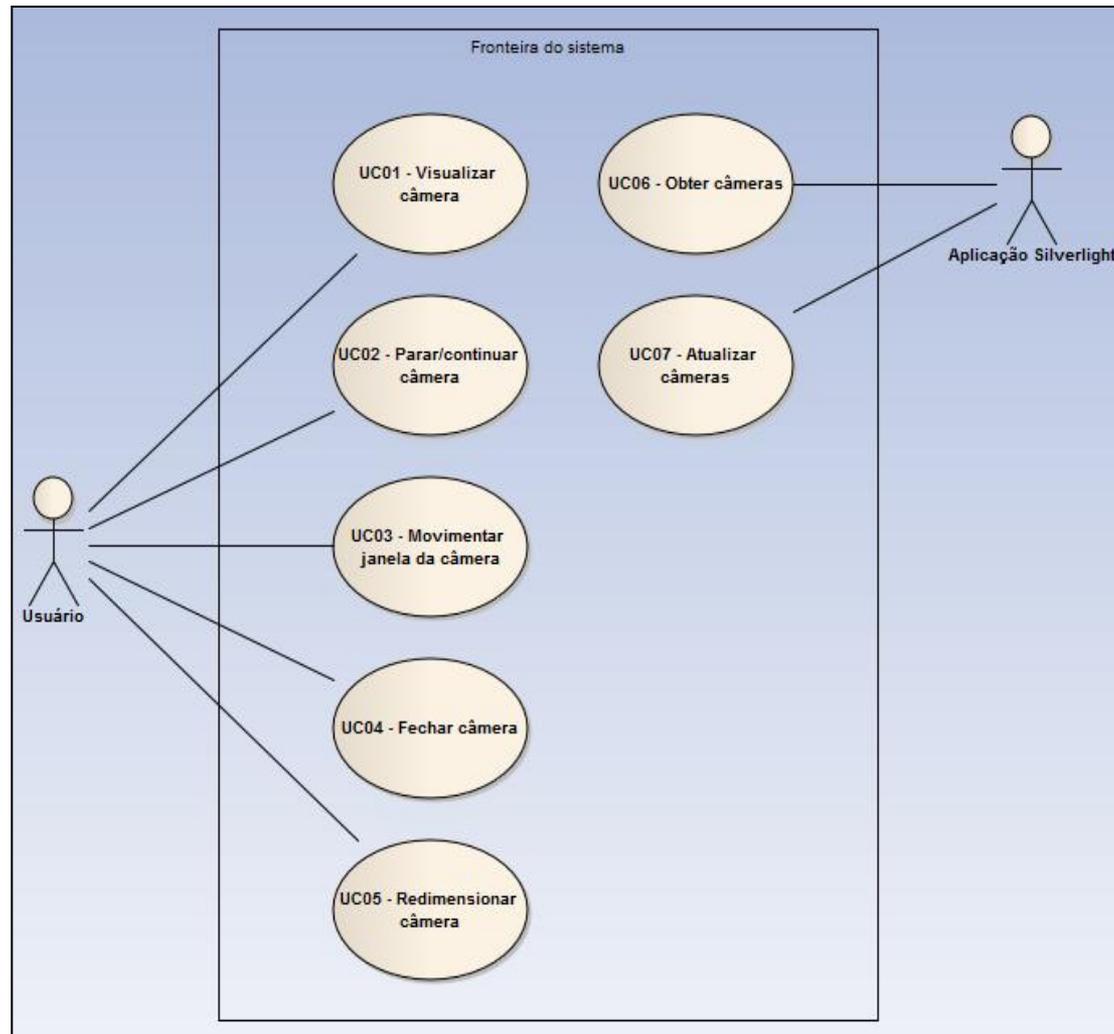


Diagrama de classes Captura

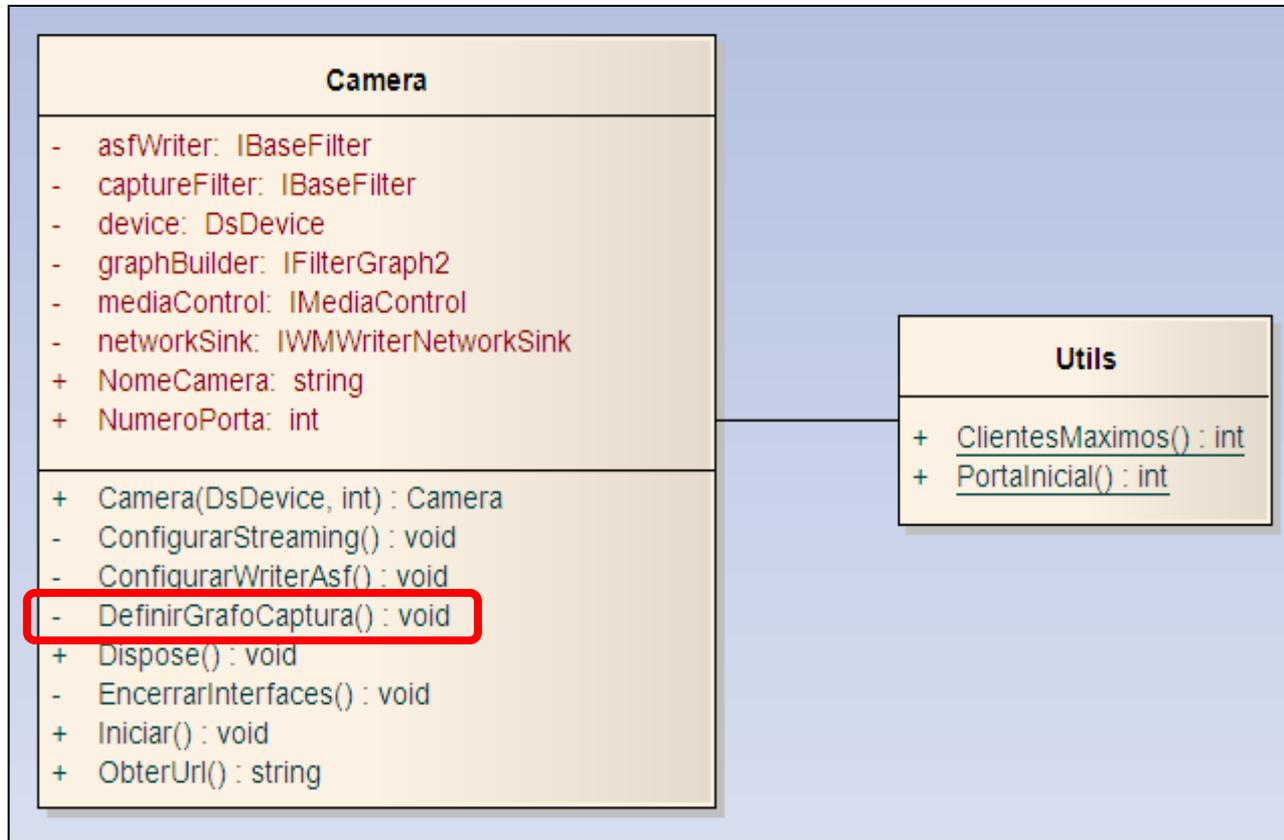


Diagrama de classes Cliente.Web



Diagrama de classes Cliente

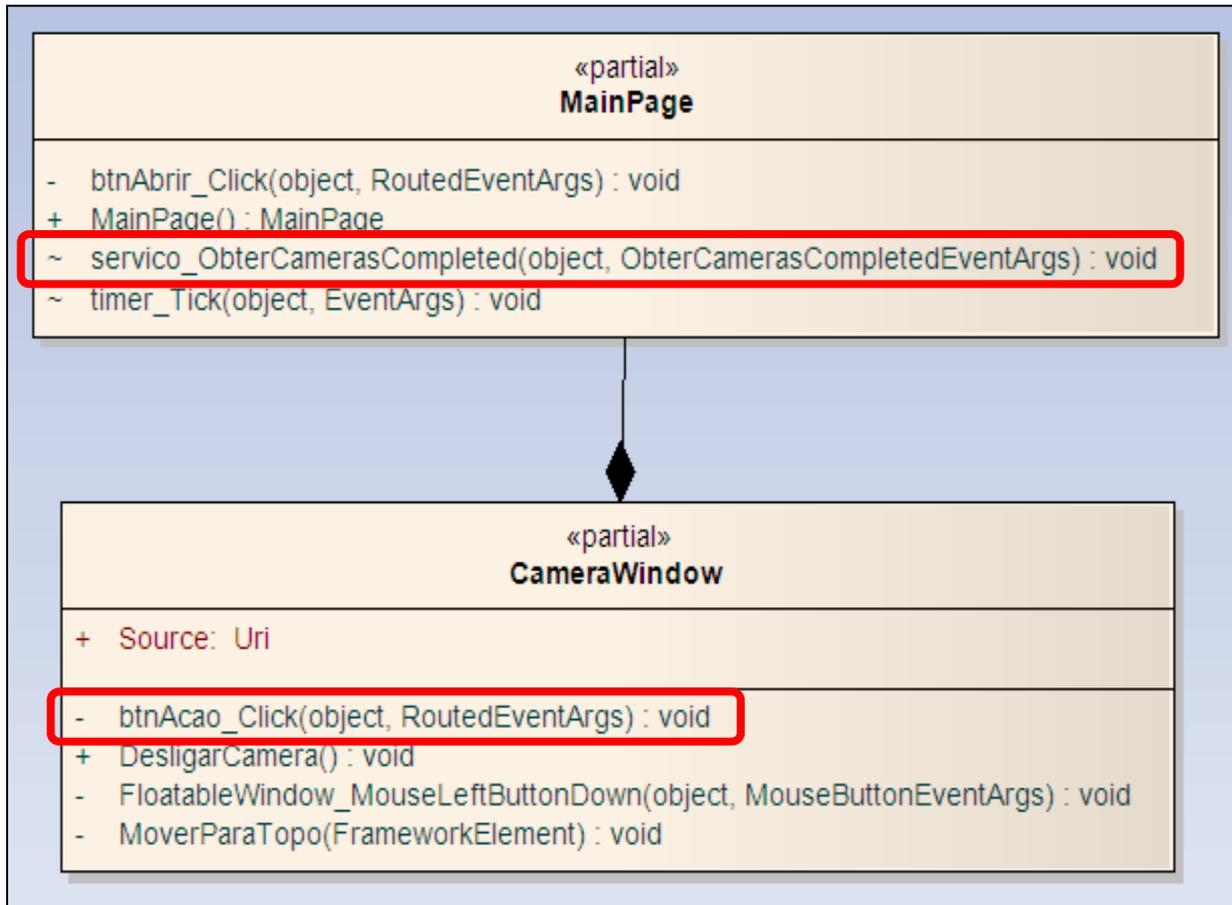
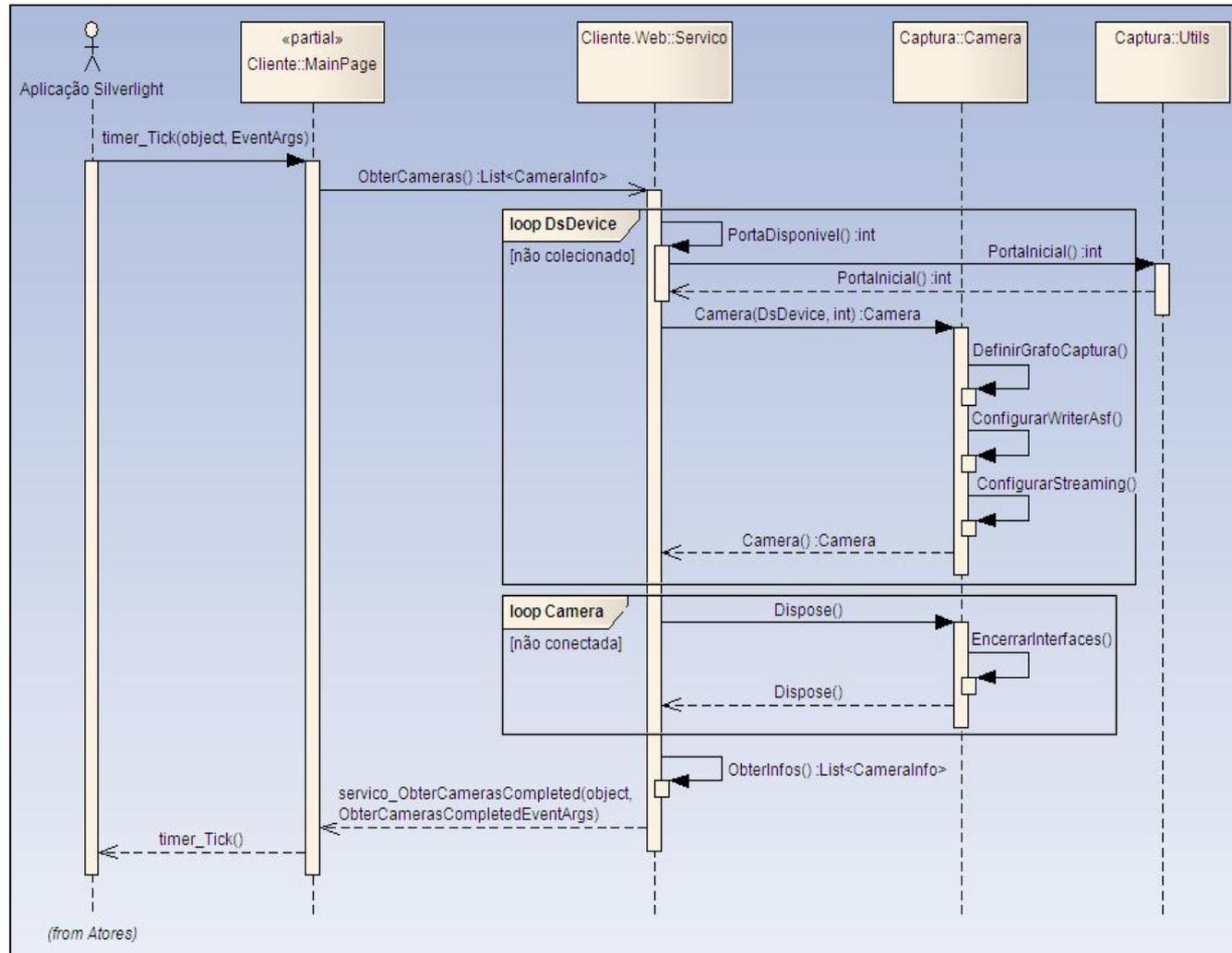
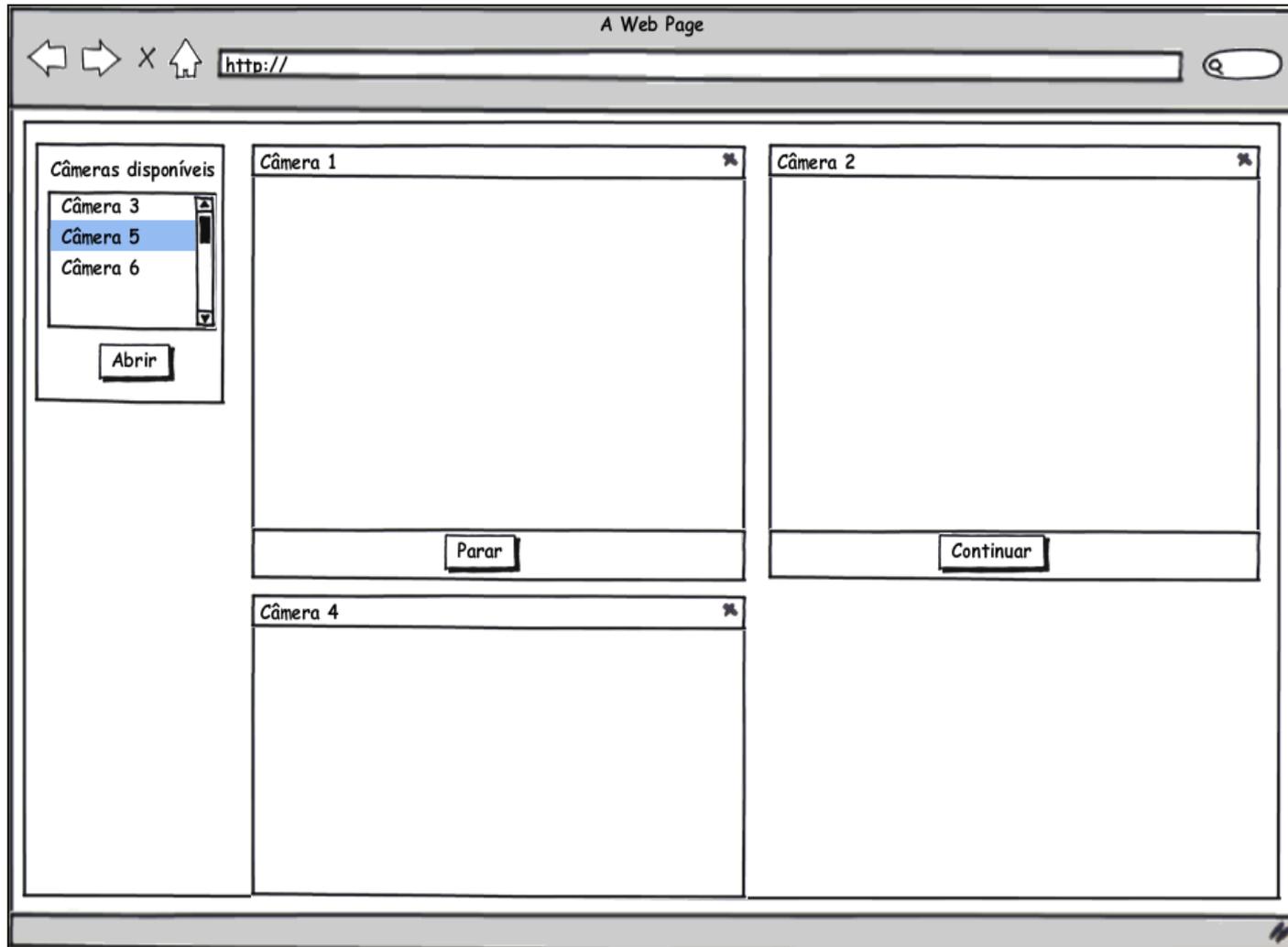


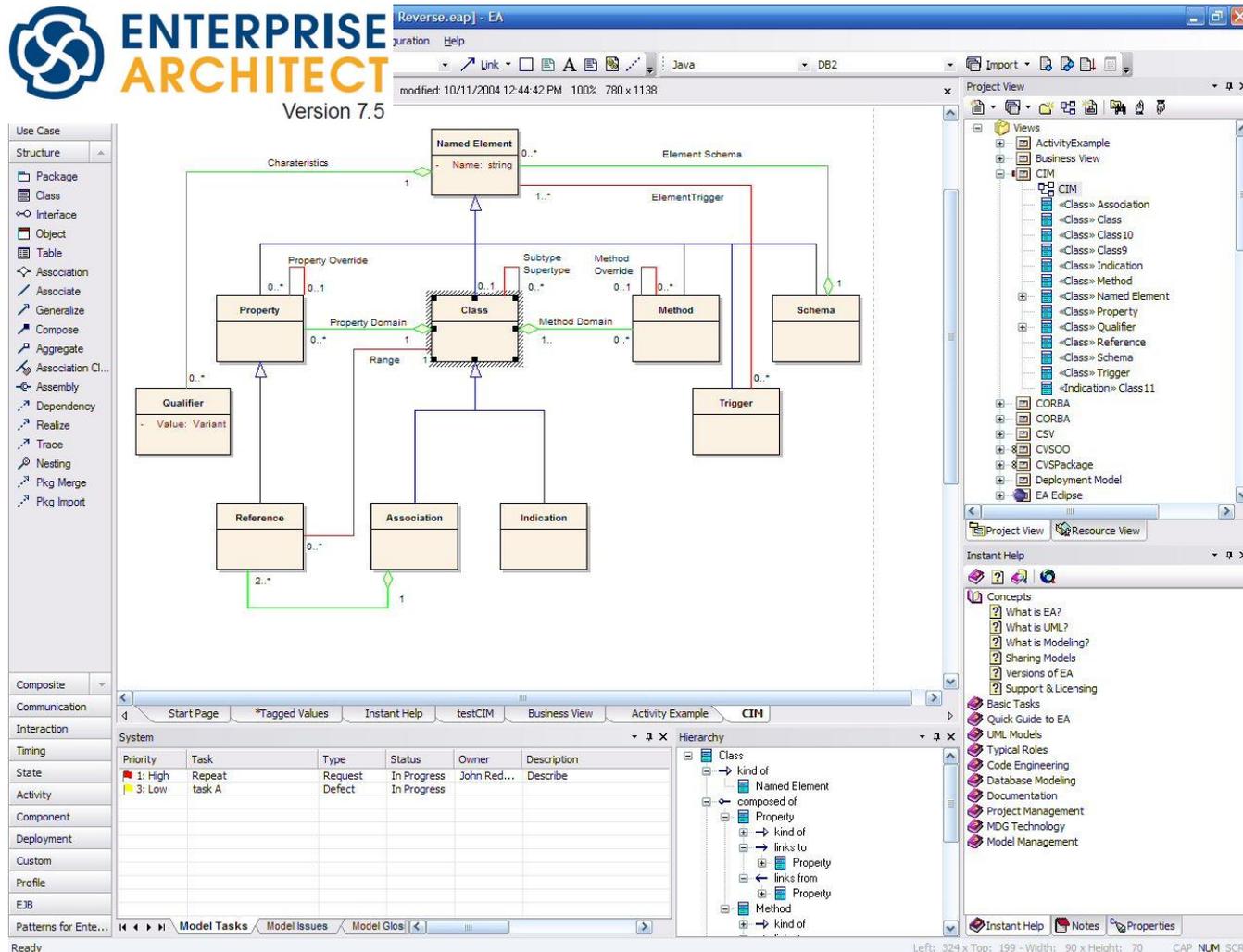
Diagrama de sequência



Leiaute da interface



Tecnologias e ferramentas utilizadas



Tecnologias e ferramentas utilizadas



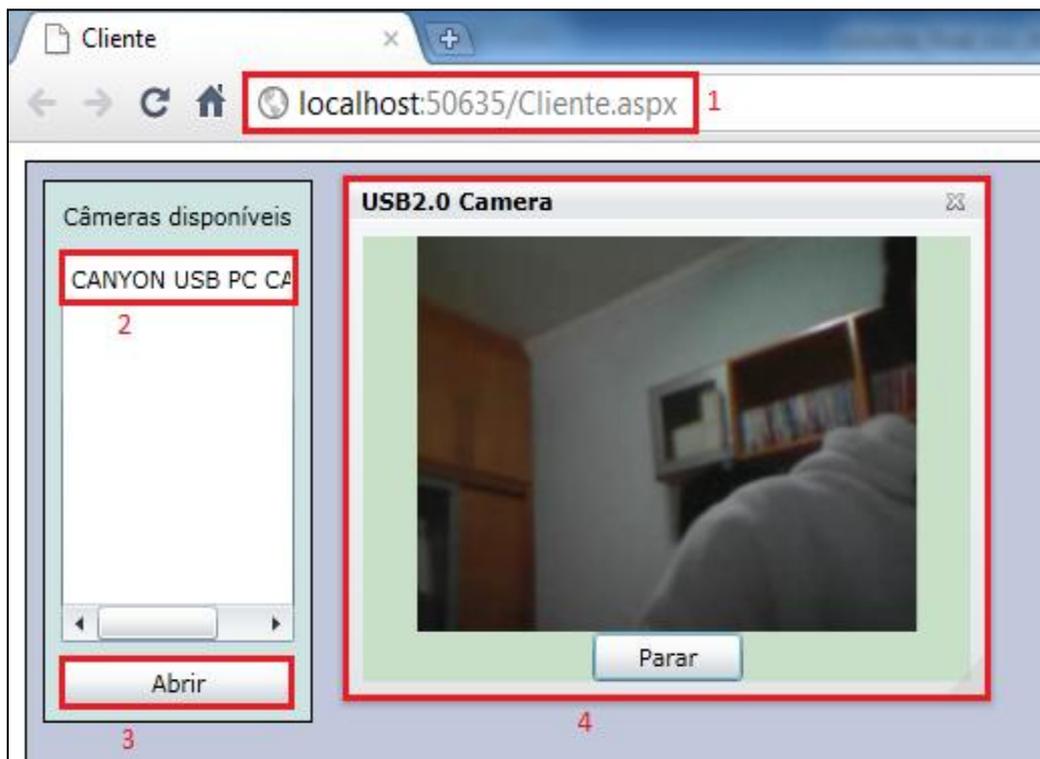
The screenshot displays the Microsoft Visual Studio 2008 environment. The main window shows a web application in design view. On the left, the 'Select CSS style to apply' pane is visible, showing a list of styles including 'Default.css' and 'Frame.css'. The central design canvas shows a 'LoginArea' with a 'Login to Site' form, a 'Welcome to My Site' header, and a 'whatsnew' section. The 'whatsnew' section contains a 'What's New' area and a 'Cool Links' area. The right side of the interface features property windows for 'whatsnew' and 'CSS Properties'. The code editor at the top shows the following ASP.NET markup:

```
<asp:image runat="Server" id="DownloadButton" AlternateText="Download" />
<p>See <a href="Albums.aspx">more photos </a></p>
<hr />
</ItemTemplate>
</asp:formview>
<h4>My Latest Piece of Work</h4>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
</div>
<div id="content">
<h3>Welcome to My Site</h3>
<p>This is my personal site. In it you will find lorem ipsum dolor si
<hr />
<div id="whatsnew">
<h4>What's New</h4>
<p>Lorem <a href="#">ipsum</a> dolor sit amet, consectetur adipi
<p>Lorem <a href="#">ipsum</a> dolor sit amet, consectetur adipi
</div>
<div id="coollinks">
```

Tecnologias e ferramentas utilizadas

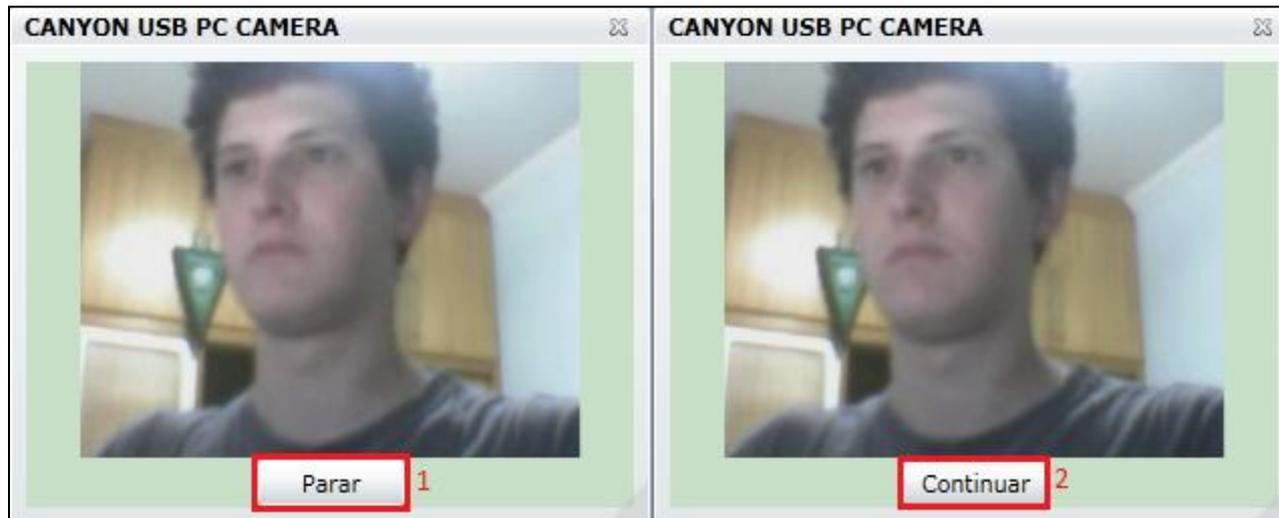


Operacionalidade da implementação



Operacionalidade do caso de uso UC01

Operacionalidade da implementação



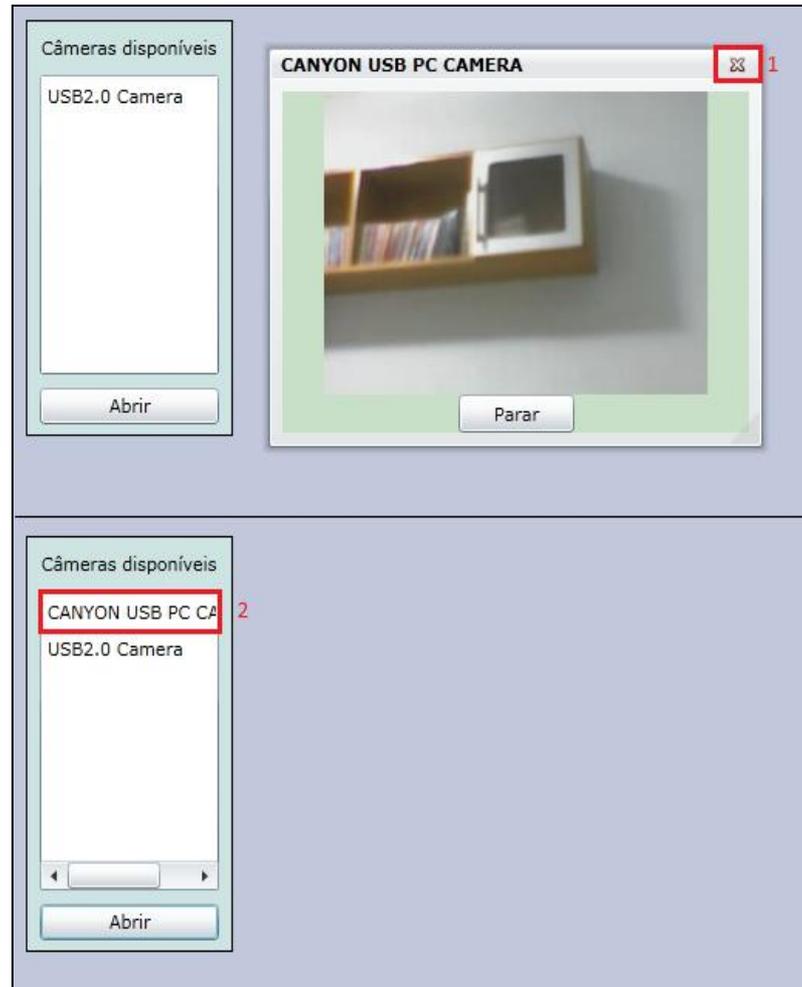
Operacionalidade do caso de uso UC02

Operacionalidade da implementação



Operacionalidade do caso de uso UC03

Operacionalidade da implementação



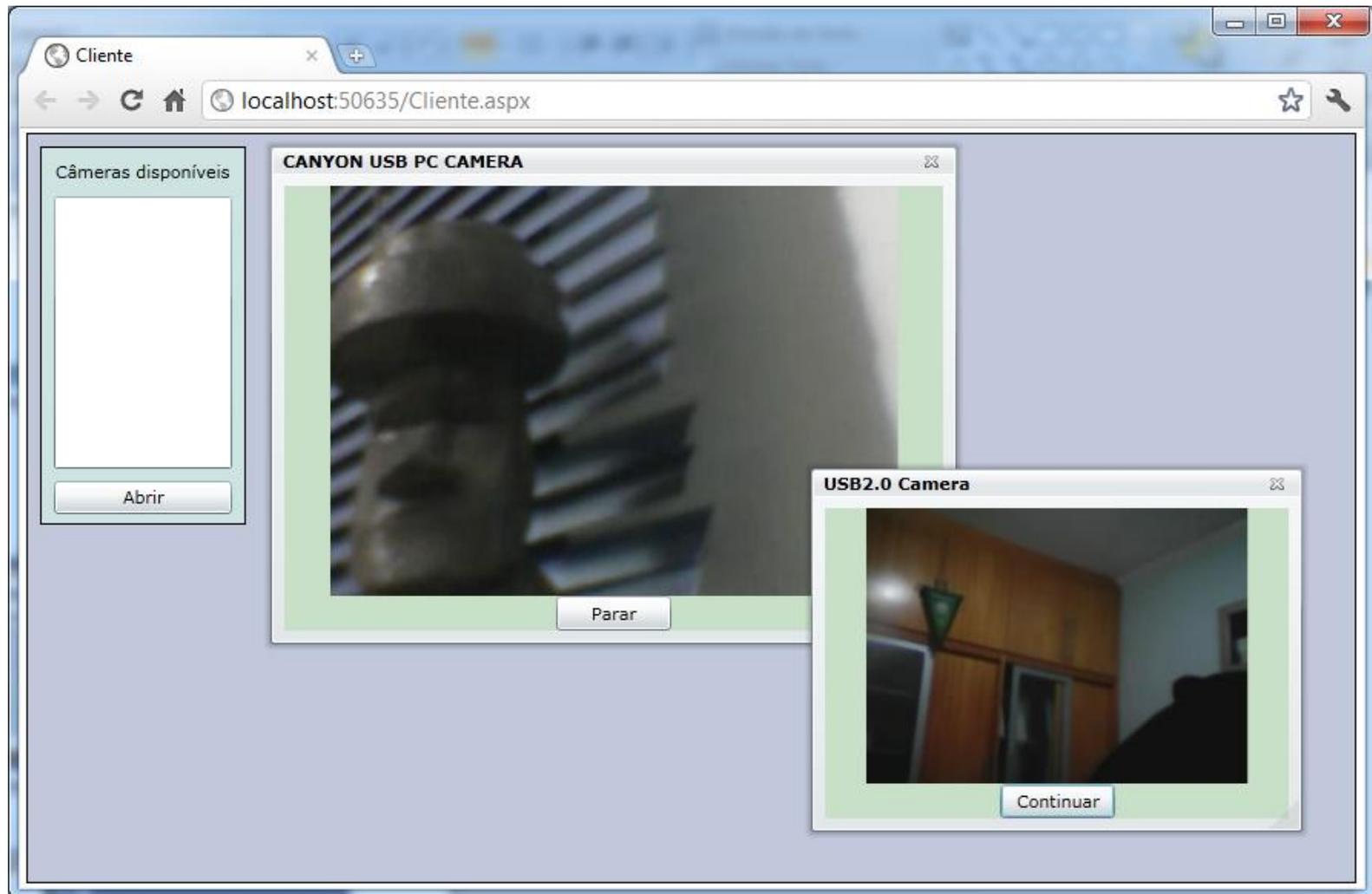
Operacionalidade do caso de uso UC04

Operacionalidade da implementação



Operacionalidade do caso de uso UC05

Resultados e discussão



Resultados e discussão



- ✓ RF01: A aplicação cliente deve disponibilizar um painel que ocupe o maior espaço possível da janela do navegador onde o usuário poderá interagir.

Resultados e discussão



✓ RF02: A aplicação cliente deve disponibilizar um quadro para cada *streaming* de câmera que estiver recebendo.

Resultados e discussão



- ✓ RF03: A aplicação cliente deve possuir um menu onde podem ser escolhidas as câmeras escondidas que se deseja restaurar.

Resultados e discussão



- ✓ RF04: A aplicação cliente deve permitir redimensionamento e movimentação livres (limitado pelo tamanho do painel) de cada quadro de câmera.

Resultados e discussão



✓ RF05: A aplicação cliente deve permitir fechar cada quadro de câmera.

Resultados e discussão



- ✓ RF06: A aplicação cliente deve permitir parar e continuar a exibição das imagens de cada janela de câmera.

Resultados e discussão



✓ RF07: A aplicação servidora deve ler a entrada de cada câmera conectada ao computador.

Resultados e discussão



✓ RF08: A aplicação servidora deve processar os dados e criar o *streaming* de vídeo de cada câmera.

Resultados e discussão



✓ RF09: A aplicação servidora deve disponibilizar os *streamings* para leitura na internet.

Resultados e discussão



Características técnicas dos equipamentos utilizados:

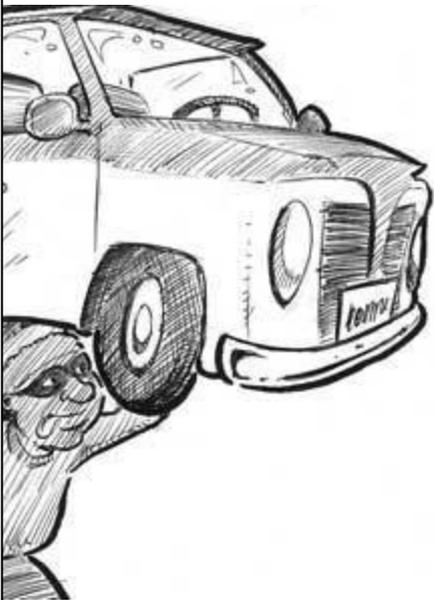
- computador *notebook* modelo Dell Vostro 1310;
- sistema operacional Microsoft Windows 7 Professional 64-bit Service Pack 1;
- processador Intel Core 2 Duo T5670 1.8 Ghz;
- memória RAM 4.0 gigabytes DDR 2 332 Mhz;
- placa de vídeo NVIDIA GeForce 8400M GS;
- resolução de vídeo 1280 x 800 *pixels* com taxa de atualização de 60 Hz e profundidade de cores de 32 *bits*;
- disco rígido de 80 Gb;
- câmera USB com resolução de vídeo 320 x 240 *pixels* e captura de 30 quadros por segundo;
- câmera USB com resolução de vídeo 640 x 480 *pixels* e captura de 30 quadros por segundo.

Resultados e discussão



Funcionalidade	Este protótipo	Trabalho de Carlassara	WebCam Monitor 5.24 de DeskShare Incorporated
Armazenamento das imagens		✓	✓
Compressão de vídeo	✓	✓	✓
Detecção automática de novas câmeras	✓		
Detecção de movimentos			✓
Movimentação remota das câmeras			✓
Transmissão através da internet	✓	✓	✓
Transmissão de áudio			✓
Transmissão em tempo real	✓		✓
Visualização de múltiplas câmeras simultaneamente	✓		✓
Visualização em dispositivo móvel		✓	
Visualização no navegador	✓		✓

Conclusões

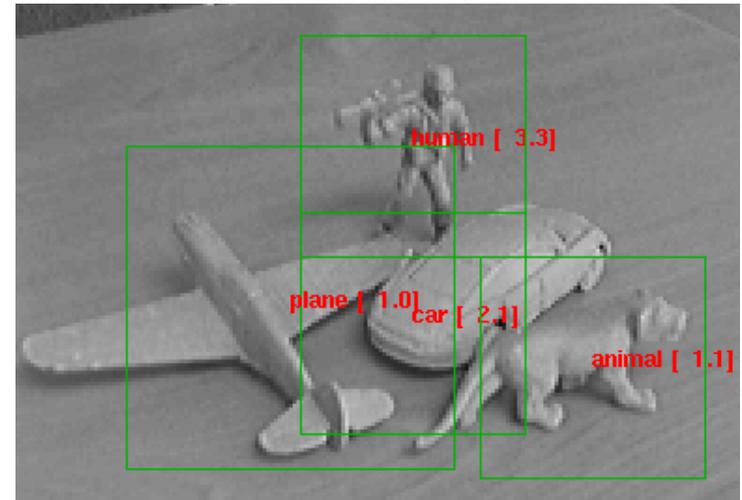
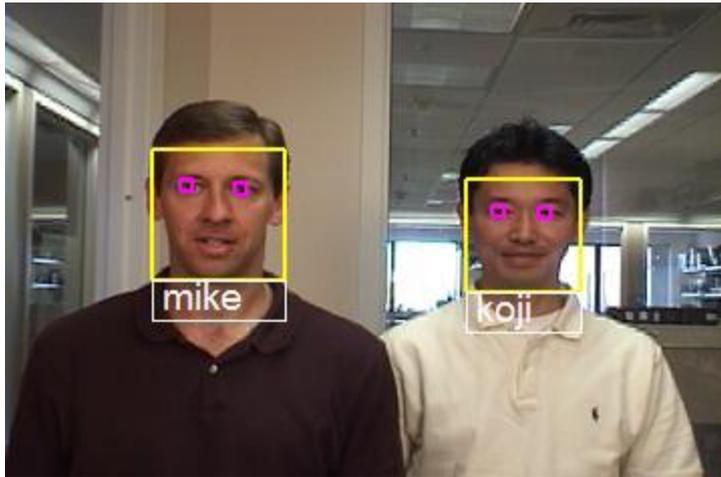


Conclusões



- Utilização de tecnologias RIA, compressão de vídeo e *streaming*
- Dificuldade em integrar os componentes do DirectShow.NET
- Documentação escassa a respeito das tecnologias de compressão e *streaming*
- Orientação a objetos e estratégia de divisão das tarefas do sistema em módulos

Extensões



Extensões



- Interface utilizando a tecnologia HTML 5
- Controlar remotamente o ângulo de visão de cada câmera
- Capturar e transmitir áudio
- Controlar o *zoom* da imagem produzida em cada câmera.
- Gravar as imagens obtidas de cada câmera.



Obrigado!