

# Prospecção e Implantação de Tecnologia de Identificação Biométrica

José David<sup>1</sup>, Marcio Belo<sup>2</sup>

<sup>1</sup>IST-Rio - Instituto Superior de Tecnologia de Ciências da Computação  
Rio de Janeiro – RJ – Brazil  
jdavidsilv@gmail.com

<sup>2</sup>IST-Rio - Instituto Superior de Tecnologia de Ciências da Computação  
Rio de Janeiro – RJ  
mbelo.ist-rio@faetec.rj.gov.br

**Resumo.** *As tecnologias de identificação biométrica vêm acrescentando maior segurança na utilização de sistemas informatizados, uma vez que permitem à aplicação o reconhecimento unívoco do indivíduo. Essas tecnologias, embora já estabelecidas em sistemas embutidos, ainda não são largamente utilizadas em aplicações WEB. Este trabalho aborda especificamente a importância da utilização da foto do indivíduo como elemento de reconhecimento biométrico, e procura demonstrar sua viabilidade através de uma implementação prática em um ambiente escolar – o WebCamBiblio.*

**Abstract.** *The biometric identification technologies have added more security in the use of computerized systems, since they allow the application unequivocal recognition of the individual. These technologies are not yet widely used in web applications, though already established in embedded systems,. This paper specifically addresses the importance of using the photo of the individual as a biometric recognition, and attempts to demonstrate its feasibility through a practical implementation in an educational environment - the WebCamBiblio.*

## **1. Introdução**

Os sistemas informatizados vêm embutindo recursos de reconhecimento biométrico para aumentar a confiabilidade no seu uso, evitando acessos indevidos à informação. Como exemplo, podemos citar as instituições bancárias que, preocupadas em realizar uma autenticação segura do seu cliente, embutem nos seus sistemas recursos de coleta de informação biométrica para combinar com outros elementos de identificação, além das tradicionais conta e senha. A autenticação de três meios – algo que se sabe, algo que se tem, e algo que se é – confere um grande incremento de segurança para sistemas de informação críticos [Pinheiro, 2008].

Neste trabalho, buscamos valorizar a visualização da foto do usuário como importante mecanismo biométrico para confiança no uso e para a responsabilização do usuário sobre sua sessão de acesso ao sistema (após a autenticação).

Percebe-se que, embora a maioria das aplicações de janela (ou aplicações *desktop*, em inglês) já esteja embutindo recursos de visualização de imagem e demais recursos biométricos, as aplicações WEB ainda engatinham nesse sentido, devido a um ambiente de programação mais restrito no que tange a recursos da máquina cliente.

Este trabalho busca averiguar o estado da arte em interação de aplicações WEB com dispositivos de captura de vídeo, que permitam embutir em sistemas de uso geral recursos de entrada de visualização da imagem do usuário ou do ambiente, com a intenção de facilitar a identificação do usuário e aumentar o nível de segurança do sistema. Como forma de demonstrar a viabilidade de uso das tecnologias pesquisadas, foi implementado um recurso numa aplicação existente, chamada WebCamBiblio, implantada em um ambiente escolar público.

## **2. Uso de Foto como Elemento Biométrico**

O uso da foto de um indivíduo em página web é normalmente observado em sites de relacionamento como, por exemplo, Orkut e o antigo 1Grau. No entanto, atualmente isso já é comum em sites como GMail [GMail], MySpace , Yahoo! Respostas [Yahoo] e Last.Fm [LAST.FM]. Estes sites, por sua vez, não são voltados para o relacionamento entre indivíduos, porém utilizam a foto dos mesmos, como elemento de reconhecimento biométrico.

Para exemplificar a importância da utilização da foto do indivíduo como elemento de reconhecimento biométrico, abaixo segue um exemplo baseado no site GMail.

A figura 1 apresenta uma página web com o perfil da usuária Jéssica Rodrigues. Nesta figura, este perfil não possui uma foto para identificar a usuária. Com isso, este perfil pode se adequar a qualquer pessoa chamada Jéssica Rodrigues.

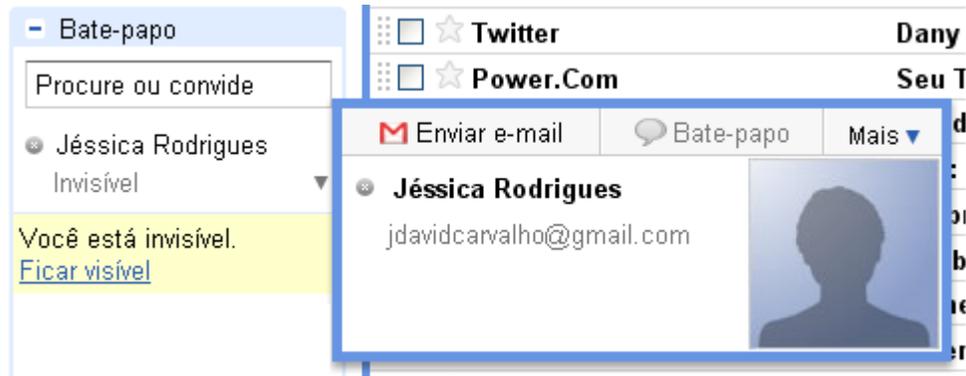


Figura 1: Página web sem foto

No entanto, quando uma foto é utilizada, é possível identificar o indivíduo. Isto pode ser observado na figura 2 abaixo. Utilizando um elemento biométrico – a foto – fica evidente reconhecer quem é a Jéssica Rodrigues.

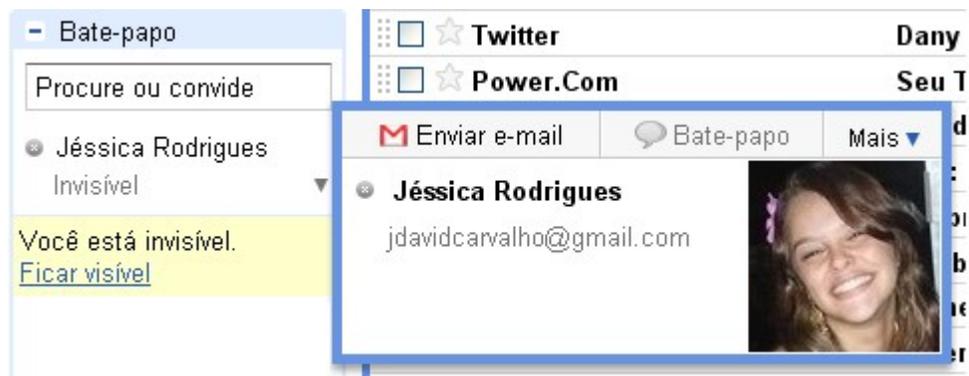


Figura 2: Página web com foto

### 3. Tecnologias Envolvidas

Antes de dar início às pesquisas com o objetivo de criar o aplicativo WebCamBiblio, foram identificadas as tecnologias envolvidas. São elas:

- WebCams;
- Banco de Dados;
- Linguagens de Programação.

Abaixo seguem em detalhe as pesquisas e seus resultados para cada tecnologia.

### 3.1 WebCams

Foram realizadas pesquisas nos principais sites de venda e informação sobre WebCams com o objetivo de encontrar a melhor tendo em vista a relação custo/benefício. Para isso, foram balizados os seguintes parâmetros que uma WebCam deve possuir para se adequar às nossas necessidades. São eles:

- Resolução;
- Captura (x quadros por segundo);
- Flexibilidade;
- Custo.

De acordo com os parâmetros identificados, o quadro abaixo apresentada as três WebCams que se destacaram na pesquisa.

Imagem	Fabricante	Modelo	Especificação	Preço
	Creaty	CV-1150	WebCam resolução 350 Kpixels. Captura até 30 quadros/segundo. Foco ajustável. Ajusta automático de exposição e do branco. Windows 2000, XP(Professional, home, SP1, SP2)	R\$ 28,90
	Clone	11089	WebCam resolução 350 Kpixels. Captura até 35 quadros/segundo. Inclinação e Rotação Windows 2000 / ME / XP Vista	R\$ 55,90
	Microsoft	NX-6000 LifeCam	WebCam resolução 2 Mpixels (2.000 Kpixels). Captura até 30 quadros/segundo. Microsoft Windows XP (SP2) / Vista	R\$ 259,00

**Tabela 1. Três WebCams que obtiveram destaques na pesquisa.**

Depois de identificadas estas três, a WebCam de Modelo 11089 Fabricante Clone, foi a escolhida pois a sua resolução de 350 Kpixels é adequada às nossas necessidades, e a captura de 35 quadros por segundos é aceitável, visto que a WebCam servirá para armazenar fotos e não vídeos, e a sua flexibilidade se torna um diferencial pois é possível inclinar e rotacionar esta WebCam, o que é fundamental para o usuário que a utilizará pois este irá posicioná-la de forma correta para tirar as fotos.

### 3.2 Banco de Dados

Antes mesmo de iniciar as pesquisas, foi levantado um pré-requisito para o Banco de Dados que será utilizado: deve possuir o armazenamento do tipo Blob. Para tal, segue abaixo uma breve explicação sobre este tipo de armazenamento.

O BLOB (Binary Large Object - Grande Objeto Binário) é um campo criado para o armazenamento de qualquer tipo de informação em formato binário. Com ele, é possível armazenar as fotos diretamente no Banco de Dados. Para demonstrar tipos de BLOB, segue na tabela 2, por exemplo, os tipos de BLOB utilizados no banco de dados MySQL [MYSQL].

<b>Tipo de Dado</b>	<b>Limite de Armazenamento</b>
TINYBLOB	L + 1 bytes onde L menor 2 elevado à 8 = 255 caracteres
BLOB	L + 2 bytes onde L menor 2 elevado à 16
MEDIUMBLOB	L + 3 bytes onde L menor 2 elevado à 24
LOB	L + 4 bytes onde L menor 2 elevado à 32 = 4GB

**Tabela 2. Tipos de armazenamento Blob.**

Depois de pesquisas realizadas, foi identificado que sistemas que armazenam as fotos em formato Blob no Banco de Dados possuem menos eficiência quando comparado à sistemas que optam por armazenar as fotos em pastas no servidor. Porém, por estimativa, espera-se que essa questão de performance não afete o WebCamBiblio. E, além disso, a forma de armazenamento das fotos de forma persistente no Banco de Dados, aumenta a segurança pois, para acessá-las, é necessário possuir as informações de login e senha do Banco de Dados, ao contrário do que ocorreria se as fotos estivessem armazenadas em uma pasta no servidor.

De acordo com os pesquisadores Sérgio R. Delfino, Fátima L.S. Nunes, Edmundo S. Spoto

[Delfino 2005], que realizaram testes de inserção e recuperação de dados em formato binário utilizando as linguagens de programação PHP e Java com os Banco de Dados MySQL e PostgreSQL, foi constatado que a combinação da linguagem de programação PHP com o Banco de Dados MySQL é a mais eficiente tanto na inserção quanto na recuperação de dados em formato binário em Banco de Dados.

Então, em função das pesquisas realizadas e do resultados dos testes citados no parágrafo anterior, será utilizado o Banco de Dados MySQL para a persistência das fotos no aplicativo WebCamBiblio.

### **3.3. Linguagens de Programação**

O aplicativo WebCamBiblio será executado parcialmente no browser. As imagens tiradas pela WebCam do cliente serão armazenadas no Banco de Dados que estará no servidor. Para tal, necessitamos de três linguagens de programação (tecnologias): uma para rodar no browser do cliente que iremos chamar de "Front-End"; outra para rodar no servidor que iremos chamar de "Back-End"; e uma para fazer a comunicação entre estas duas camadas, que iremos chamar de "Communication". Abaixo seguem três tópicos descrevendo as três tecnologias escolhidas.

#### **3.2.1. Front-End**

Para esta camada, pesquisas foram realizadas para encontrar algum plugin para browser de Internet que possuísse as funcionalidades de tirar fotos de WebCam do cliente e armazenar estas em Banco de Dados no servidor. Alguns plugins foram encontrados, tais como: DeskCam, Adult WebCams e WebCam 2.0. Porém, dentre as funcionalidades que precisamos, estes plugins se limitam a somente se conectarem à WebCam do cliente e exibir as imagens coletadas.

Então, decidimos elaborar a camada Front-End. Adotamos a tecnologia Flash para isto pois, em função de testes realizados, verificamos que podemos utilizá-la para conectar a uma WebCam e obter imagens. Neste momento ainda não possuímos uma tecnologia para transmitir as imagens da WebCam para os servidor. No entanto, o Flash nos fornece as funcionalidades iniciais que precisamos.

#### **3.2.2. Back-End**

Para esta camada, pesquisas foram realizadas para encontrar a melhor linguagem de programação que se adequasse de forma eficiente ao Banco de Dados MySQL persistindo arquivo binário (fotos) em formato Blob. Duas linguagens foram encontradas: Java e PHP.

Com a linguagem Java, é possível inserir e recuperar dado binário em formato Blob utilizando JDBC, porém estas transações no Banco de Dados são menos eficientes quando comparadas com as transações realizadas com a linguagem PHP. Além disso, de acordo com os testes realizados em [Delfino 2005], a integração PHP e MySQL, realizando estas transações com dado binário, é a mais eficiente. Com isso, para realizar estas transações no Back-End, utilizamos o linguagem PHP.

### **3.2.3. Communication**

Para esta camada, pesquisas foram realizadas para encontrar a melhor tecnologia para integrar a camada Front-End (executada no browser do cliente) com a camada Back-End (executada no servidor).

Como está sendo utilizado a linguagem Actionscript 3.0 (Flash) na camada Front-End e a linguagem PHP na camada Back-End, o pré-requisito da tecnologia para a camada Communication é que esta possua a funcionalidade de transformar objeto Actionscript em objeto PHP. Sabendo disto, foram realizadas pesquisas com o objetivo de encontrar uma tecnologia que possua esta funcionalidade.

Com os resultados dessas análises, a tecnologia que mais de adequou as nossas necessidades foi a AMFPHP [AMFPHP].

De modo geral, o AMFPHP traz a possibilidade de uma aplicação Flash se comunicar com o servidor, transformando objeto Actionscript em objeto PHP. Esta funcionalidade é o necessário para existir a comunicação entre as camadas Front-End e Back-End.

## **4. O WebCamBiblio**

O WebCamBiblio é um pacote de alterações introduzido num sistema aberto para informatização de bibliotecas, chamado OpenBiblio [Openbiblio]. O OpenBiblio permite catalogar o acervo bibliográfico e controlar os processos de empréstimos e devolução de exemplares para usuários de uma biblioteca.

O WebCamBiblio possui como principais funcionalidades:

- Tirar fotos da WebCam do cliente;
- Armazená-las em banco de dados no servidor;
- Exibir as mesmas no browser do cliente.

O WebCamBiblio foi projetado com três camadas: Front-End, Back-End e Communication. A figura 3 sintetiza a arquitetura elaborada neste trabalho. A seguir detalhamos cada uma das camadas:

- Front-End: Esta camada corresponde à parte da aplicação que é executada no browser do cliente. Esta foi desenvolvida em Flash (Actionscript 3.0).
- Back-End: Esta camada corresponde à parte da aplicação que é executada no servidor. Esta foi desenvolvida em PHP utilizando o banco de dados MySQL para persistência das imagens.
- Communication: Camada que corresponde à parte de comunicação entre o Front-End e o Back-End. Esta foi desenvolvida utilizando a biblioteca open-source AMFPHP.

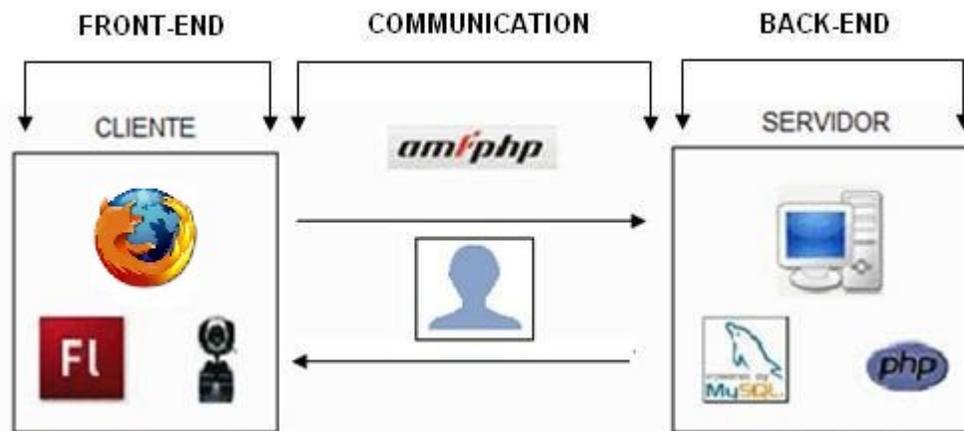


Figura 3. As três camadas do WebCamBiblio

## 5. Desenvolvimento do WebCamBiblio

Esta seção descreve, de forma detalhada, como foram desenvolvidas as três camadas, bem como as tecnologias utilizadas, no WebCamBiblio.

## 5.1. Front-End

Esta camada corresponde à parte visual do WebCamBiblio que foi codificada em Actionscript, versão 3.0, utilizando a tecnologia Flash. Esta camada é exibida na figura 4.



**Figura 4. Front-End do WebCamBiblio**

O ActionScript [Actionscript] é executado com a AVM (ActionScript Virtual Machine), que faz parte do Flash Player. O código do ActionScript em geral é compilado no *formato de código de bytes* por um compilador, como o criado no Adobe® Flash® CS4 Professional (utilizado na criação do WebCamBiblio). O código de bytes é incorporado aos arquivos SWF, que são executados pelo Flash Player. Este, por sua vez, é um plug-in utilizado para executar aplicações criadas no Adobe® Flash® em browsers como, por exemplo, Firefox e Internet Explorer.

O ActionScript, versão 3.0, oferece um modelo de programação robusto que parecerá familiar aos desenvolvedores com um conhecimento básico de programação orientada a objetos. Nos próximos tópicos será apresentado o funcionamento do WebCamBiblio e em paralelo os códigos que são executados.

Na página web onde o WebCamBiblio é executado, está sendo utilizado JavaScript para chamar o arquivo compilado (.swf) do mesmo. Abaixo segue o código 1 exibindo a parte deste código que é importante ressaltar

```
18 <script type="text/javascript" src="WebCamBiblio.js"></script>
19 <script language="JavaScript" type="text/javascript">
20 AC_FL_RunContent(
21   'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
22   'width', '480',
23   'height', '240',
24   'src', 'WebCamBiblio',
25   'quality', 'high',
26   'pluginspage', 'http://www.adobe.com/go/getflashplayer',
27   'align', 'middle',
28   'play', 'true',
29   'loop', 'true',
30   'scale', 'showall',
31   'wmode', 'window',
32   'devicefont', 'false',
33   'id', 'WebCamBiblio',
34   'bgcolor', '#ffffff',
35   'name', 'WebCamBiblio',
36   'menu', 'true',
37   'allowFullScreen', 'false',
38   'allowScriptAccess', 'sameDomain',
39   // UrlAmfGateway: variável com o caminho de onde está o gateway para a comunicação com o AMFPHP
40   'flashvars', 'UrlAmfGateway=http://localhost/openbiblio/webcambiblio/web_service/amf/gateway.php' +
41   '&mbrid=?php echo $mbrid ?>',
42   'movie', 'WebCamBiblio',
43   'salign', ''
44 ); //end AC code
45 </script>
```

**Código 1. Código JavaScript com os parâmetros que são passados ao arquivo compilado (.swf) do WebCamBiblio.**

O Código 1 é importante pois apresenta o parâmetro (“UrlAmfGateway”) que é transferido para o arquivo WebCamBiblio.swf. Este deve ser setado com o caminho de onde se encontra o Gateway do AMFPHP no servidor.

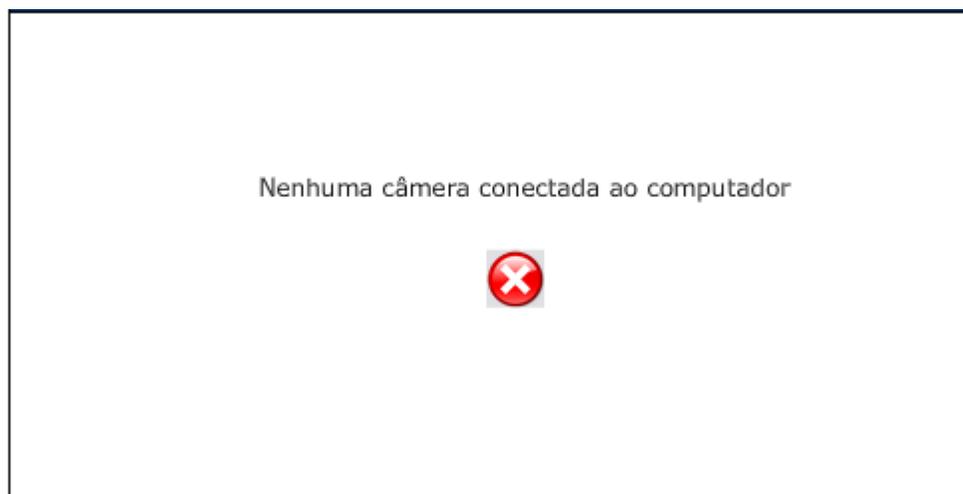
A função AC\_FL\_RunContent é responsável por apresentar o WebCamBiblio no browser. O código desta está no arquivo “WebCamBiblio.js” importado na linha 18.

No código 2 em ActionScript, percebemos os parâmetros UrlAmfGateway e o mbrid, ambos recebidos pelo código 1 em JavaScript. O parâmetro mbrid é utilizado no sistema OpenBiblio para indicar um usuário deste aplicativo.

```
5 import lib.com.adobe.images.PNGEncoder;
6
7 var _objService:NetConnection;
8 var _objResponder:Responder;
9
10 var parameters:Object = LoaderInfo(this.root.loaderInfo).parameters;
11 var _amfGateway:String = String(parameters["UrlAmfGateway"]);
12 var _mbrid:String = String(parameters["mbrid"]);
13 var bmdCapturado:Bitmap;
14
15 var cam:WebCam = new WebCam(stage);
16
17 if (!cam.start()) {
18     erro_info.visible = true;
19     erro_img.visible = true;
20     btnCapturar.visible = false;
21 } else {
22     erro_info.visible = false;
23     erro_img.visible = false;
24 }
25 |
26 btnArmazenar.visible = false;
```

**Código 2.** Código Actionscript apresentando como é recebido o parâmetro “UrlAmfGateway” no Flash.

Caso não exista uma WebCam conectada ao computador, será apresentada a mensagem da figura 5.



**Figura 5. Mensagem de erro quando é executado o WebCamBiblio e não existe WebCam conectada ao computador.**

Depois de conectada uma WebCam e executar novamente o WebCamBiblio, o mesmo será executado conforme a figura 6.



**Figura 6. O WebCamBiblio em execução com a WebCam conectada ao computador**

Neste momento, os parâmetros transferidos na chamada JavaScript já foram lidos pelo Flash e foi conectado à WebCam.

Para realizar comunicação com a WebCam, foi criada uma classe Actionscript 3.0 chamada WebCam. Nesta foi criado o método start() com o objetivo de conectar à WebCam do cliente, capturar o vídeo e armazená-lo no palco<sup>1</sup>.

<sup>1</sup> Palco: é a parte visual de uma aplicação Flash.

```
7      import flash.display.MovieClip;
8      import flash.geom.Matrix;
9      import flash.media.Camera;
10     import flash.media.Video;
11     import flash.events.Event;
12
13     public class WebCam extends MovieClip {
14
15         public function start():Boolean {
16             camera = Camera.getCamera();
17
18             if (camera) {
19                 // cria o video
20                 video = new Video();
21
22                 // setando as dimensões
23                 video.width = this.stageP.stageWidth-280;
24                 video.height = this.stageP.stageHeight-72;
25
26                 // setando as coordenadas
27                 video.x = 15;
28                 video.y = 15;
29
30                 // adicionando a camera
31                 video.attachCamera(camera);
32
33                 // adicionando o video ao container (movieclip empty no stage)
34                 this.stageP.addChild(video);
35
36                 started = true;
37                 return true;
38             } else {
39                 trace("Não existe camera conectada");
40                 return false;
41             }
42         }
43     }
```

**Código 3.** Como é realizada a conexão e captura do vídeo da WebCam do cliente.

Para capturar o vídeo da WebCam conectada ao computador, na linha 16 está sendo utilizada a classe “Camera” [Camera] e o seu método “getCamera()”. Para utilizar a classe “Camera”, deve-se importar a mesma conforme apresentado na linha 9.

Depois da função start() do código 3 ser executada e existir WebCam conectada ao computador do cliente, o mesmo poderá tirar uma foto clicando no botão “Capturar”. No momento em que é clicado neste botão, é obtido a imagem do vídeo da WebCam e apresentada no palco (tela). Esta etapa é realizada utilizando o método getPicture() da classe WebCam conforme apresentado no código 4.

```
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
--  
  
public function getPicture():Bitmap {  
    // setando parâmetro matrix  
    var matrixPreview:Matrix = new Matrix();  
    matrixPreview.scale(0.6,0.7);  
  
    // obtendo a imagem da Webcam do cliente  
    bmdDataPicturePreview = new BitmapData(this.stageP.stageWidth - 280,  
                                           this.stageP.stageHeight-60);  
    bmdDataPicturePreview.draw(this.video, matrixPreview);  
    bmdPicturePreview = new Bitmap(bmdDataPicturePreview);  
  
    return bmdPicturePreview;  
}
```

**Código 4. Como é realizada a obtenção da imagem do cliente e apresentada a mesma no palco.**

Quando obtida e apresentada a imagem no palco, o WebCamBilio é alterado como na figura 7.



**Figura 7. Como o WemCamBiblio apresenta a imagem do cliente.**

Caso seja clicado no botão “Voltar”, o WebCamBiblio ficará de acordo com a figura 6. Caso seja clicado no botão “Armazenar”, a foto será armazenada em banco de dados no servidor conforme apresentado no código 5.

```
40 btnArmazenar.addEventListener(MouseEvent.CLICK, armazenar);
41 function armazenar(event:MouseEvent) {
42     var bmd:BitmapData = new BitmapData(320, 240);
43     bmd.draw(cam.getObjectVideo());
44     var ba:ByteArray = PNGEncoder.encode(cam.getBmdDataPicture());
45
46     _objService = new NetConnection();
47     _objResponder = new Responder(onResultEvent, onStatusEvent);
48     _objService.connect(_amfGateway);
49     _objService.call("WebCamBiblio.Image.saveDataToFile", _objResponder, ba, _mbrid);
50
51     // para o palco voltar ao seu status inicial
52     voltar_();
53 }
```

**Código 5. Como é enviada a imagem capturada para o servidor**

Nesta função “armazenar”, a imagem que o usuário solicitou para armazenar é renderizada e armazenada na variável “ba” (linha 44 – código 5). Depois disso, é realizada uma conexão com o AMFPHP (linha 48 - a variável “\_amfGateway” possui o caminho onde está localizado o gateway do AMFPHP enviado por parâmetro pelo JavaScript - códigos 1 e 2) e chamada a função “saveDataToFile” da classe Image.

A partir daí, a camada Back-End recebe a imagem solicitada e realiza os seus procedimentos que estão detalhados na sua seção.

## 5.2. Communication

Esta camada é representada pelo Gateway do AMFPHP. Em resumo, o Gateway receberá a chamada invocada pela camada Front-End e encaminhará a mesma para o seu destinatário. No nosso caso, o destinatário é WebCamBiblio.Image.saveDataToFile (mais detalhes na seção posterior 5.3.Back-End) que representa a função “saveDataToFile” da classe “Image” no pacote “WebCamBiblio”. O destinatário e o caminho até ele é apresentado a seguir.

## 5.3. Back-End

Abaixo segue a estrutura de pastas no servidor representando onde se encontra o script PHP (destinatário) que persiste a imagem no banco de dados conforme a figura 8 abaixo.

```
'flashvars', 'UrlAmfGateway=http://localhost/openbiblio/webcambiblio/web_service/amf/gateway.php ' +  
'&mbrid=<?php echo $mbrid; ?>' +  
'&msg=<?php echo $msg; ?>' +  
'&reset=Y',
```



**Figura 8. Caminho do Gateway do AMFPHP e, representado nas pastas, o caminho para o destinatário.**

Dentro da pasta “WebCamBiblio” se encontra o script PHP chamado “Image.php”. Conforme apresentado no código 6 à seguir, este script possui o método “saveDataToFile()” que insere o arquivo de imagem recebido do cliente no banco de dados em formato Blob.

```

1 <?php
2 // obtém as variáveis para conexão com o banco de dados
3 require_once("../../../../../database_constants.php");
4
5 class Image
6 {
7     // *****
8     /* Função
9     - Nome: saveDataToFile
10    - Descrição: Recebi um arquivo em formato ByteArray e inseri o
11                mesmo no banco de dados em formato Blob.
12
13    - Parâmetros:
14    -> $byteArray: Arquivo em formato ByteArray contendo a imagem.
15    -> $mbrid: Chave primária de um aluno no IcaroBiblio.
16    */
17    // *****
18    public function saveDataToFile($byteArray, $mbrid)
19    {
20        $conteudo = addslashes($byteArray->data);
21
22        $db = mysql_connect(OBIB_HOST,OBIB_USERNAME,OBIB_PWD);
23        mysql_select_db(OBIB_DATABASE,$db);
24
25        // verifica se o aluno já possui imagem
26        $query = "SELECT mbrid FROM imagem WHERE mbrid = " . $mbrid;
27        $res = mysql_query($query,$db);
28
29        // se aluno ainda não possui imagem
30        if (mysql_num_rows($res) == 0) {
31            $query = "INSERT INTO imagem (mbrid, conteudo) VALUES (" . $mbrid . ", '" . $conteudo . "')";
32            mysql_query($query,$db);
33        } else {
34            $query = "UPDATE imagem SET conteudo = '" . $conteudo . "' WHERE mbrid = " . $mbrid ;
35            mysql_query($query,$db);
36        }
37    }
38 }
39 ?>

```

**Código 6. Classe Image para inserção da imagem no banco de dados.**

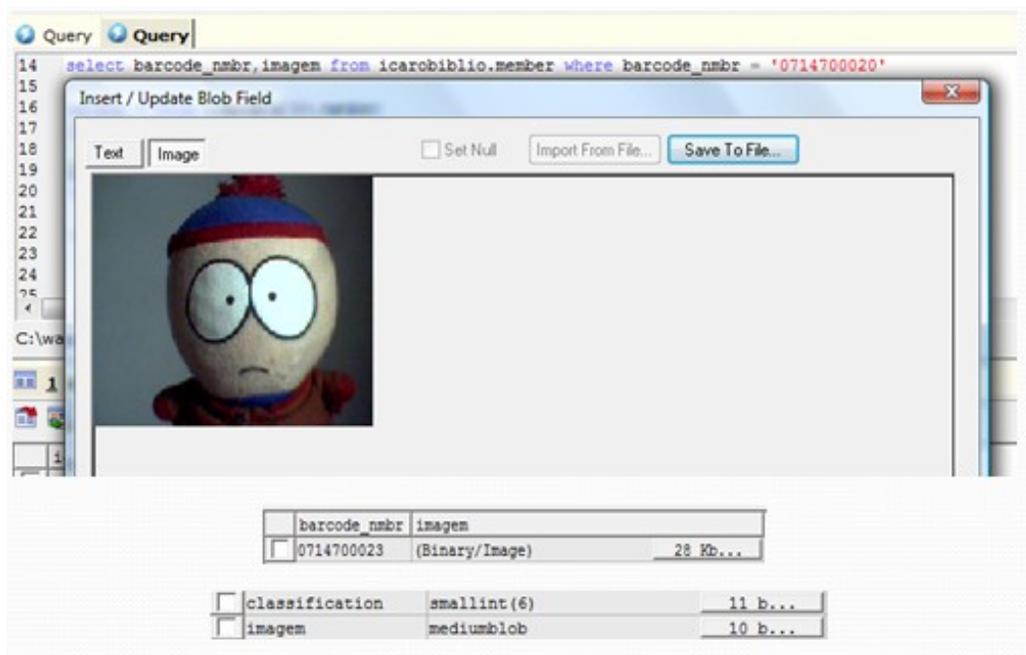


Figura 9. Demonstrativo da foto salva no banco de dados MySQL em formato Blob.

#### 5.4. Exibição da Imagem

Este tópico apresenta a implementação realizada para exibir a imagem em formato Blob no browser.

Para isso, foi criado o script apresentado no código 7 abaixo.

```

2  require_once("../database_constants.php");
3
4  $matricula = $_GET['matricula'];
5
6  $link = mysql_connect(OBIB_HOST_CORUJA,OBIB_USERNAME_CORUJA,OBIB_PWD_CORUJA);
7  mysql_select_db(OBIB_DATABASE_CORUJA,$link);
8
9  $query = "SELECT l.foto
10           FROM matriculaAluno m, login l
11           WHERE m.matriculaAluno = '$matricula'
12                 AND m.idPessoa      = l.idPessoa;";
13
14  $res  = mysql_query($query,$link);
15  $linha = mysql_fetch_array($res);
16
17  // informando que será enviada uma imagem jpeg
18  header( "Content-type: image/jpeg" );
19
20  // se aluno possui foto
21  if (mysql_num_rows($res) > 0) {
22
23      // exibe a foto no browser
24      echo $linha['foto'];
25
26  } else {
27      // nome da foto no servidor
28      $foto = "../images/sem_foto.jpg";
29
30      // cria foto a partir de um arquivo
31      $img = imagecreatefromjpeg($foto);
32
33      // exibe a foto no browser
34      imagejpeg($img, NULL, 100);
35
36      // libera o espaço na memória utilizada pela foto
37      imagedestroy($img);
38  }

```

**Código 7. Script para exibição da imagem salva em formato Blob**

Conforme apresentado no código 7, após ter realizado uma consulta no banco de dados para buscar a imagem, na linha 18 é preenchido o header HTML informando que o script enviará uma imagem de formato jpeg.

Na linha 21 é verificado se a consulta retornou valor. Caso positivo, exibe a imagem. Caso negativo é exibida uma imagem armazenada no servidor informando que o usuário não possui foto.

Para a imagem ser exibida de fato no browser, é embutido no código HTML da página que será carregada no browser o código 8 abaixo.

```

200
201     <?php $matricula = H($mbr->getBarcodeNmbr()); ?>
202     
203

```

### Código 8. Script para exibição da imagem salva em formato Blob

O arquivo “mbr\_foto.php” exibido no código 8 acima chama o script PHP apresentado no código 7 anteriormente. Como o servidor interpreta este arquivo PHP (código 7) como uma imagem, devido a este estar preenchendo o header HTML informando que será enviada uma imagem, a foto do usuário é exibida compondo com a página do código 8. Isto pode ser observado na figura 10 abaixo.

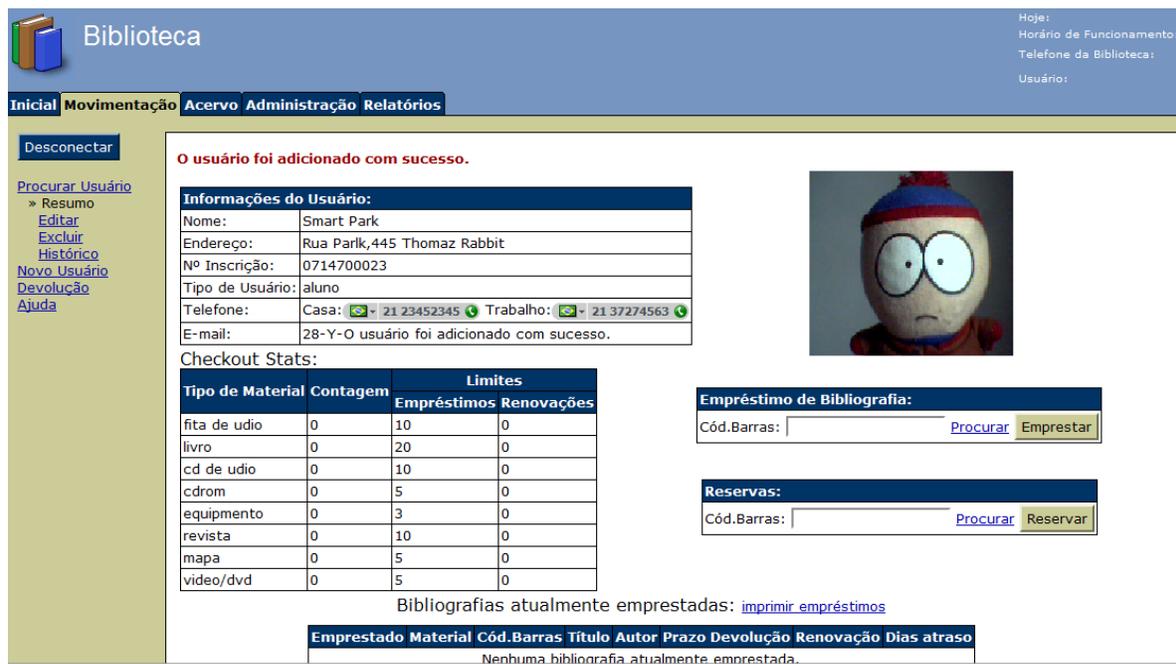


Figura 10. Sistema ÍcaroBiblio – Página de aluno com a imagem capturada do aplicativo WebCamBiblio.

## 6. Trabalhos Similares

Entre alguns trabalhos semelhantes ao WebCamBiblio podemos destacar dois: DeskCam e Cam Watcher.

O DeskCam [DESKCAM] é um plugin para o Firefox. Com ele é possível obter a filmagem da Webcam em apenas um clique. Além disso, a instalação é realizada de forma facilitada pelo mecanismo gerenciador de extensões do Firefox.

O Cam Watcher [CAM WATCHER], também um plugin para o Firefox, . Com ele, além de obter a filmagem da Webcam, o mesmo apresenta um alerta quando um amigo, usando o mesmo plugin, conecta-se à internet.

## 7. Conclusão

O sistema OpenBiblio possuirá um aplicativo que fará com que os usuários deste sistema utilizem identificação biométrica para reconhecer os alunos. Assim, a identificação dos mesmos será realizada de forma mais precisa, rápida e segura.

Além disso, um mecanismo de captura de imagens originadas da WebCam de um usuário na Web faz com que o Site que possua este mecanismo tenha a possibilidade de receber imagens em tempo real da WebCam, de qualquer usuário conectado à Internet, e o mesmo possa visualizar estas imagens no seu browser. Várias classes de aplicações podem se beneficiar desse recurso, a citar: educação, comercial, serviços, etc.

## 8. Referências

Adobe, Actionscript, Disponível em:

<[http://help.adobe.com/pt\\_BR/ActionScript/3.0\\_ProgrammingAS3](http://help.adobe.com/pt_BR/ActionScript/3.0_ProgrammingAS3)>. Acesso em: 03/07/2009.

AMFPHP. Flash Remoting for PHP: A responsive Client-Server Architecture for the Web.

Disponível em: <<http://www.amfphp.org>>, Acesso em: 13/05/2009.

CAM WATCHER. Disponível em:

<<https://addons.mozilla.org/pt-BR/firefox/addon/13498>>, Acesso em: 27/10/2009.

Camera, Classe Camera Actionscript, Disponível em:

<[http://help.adobe.com/pt\\_BR/AS3LCR/Flash\\_10.0/flash/media/Camera.html](http://help.adobe.com/pt_BR/AS3LCR/Flash_10.0/flash/media/Camera.html)>, Acesso em: 4/11/2009.

CREATY, Web Site, Disponível em: <<http://www.creaty.com.br>>. Acesso em: 12/04/2009.

DESKCAM, Disponível em: <<https://addons.mozilla.org/pt-BR/firefox/addon/4564>>, Acesso em 27/10/2009.

Delfino, S.R., Nunes, F.L.S., Spoto E.S. (2005). Avaliação de Desempenho de SGBDs de Código Aberto para Construção de Bases de Imagens Médicas. Disponível em: <[http://galileu.fundanet.br/jornada/artigos/computacao/Sergio\\_Delfino.pdf](http://galileu.fundanet.br/jornada/artigos/computacao/Sergio_Delfino.pdf)>, Acesso em: 12/04/2009.

FIREFOX Add-ons, Disponível em: <<https://addons.mozilla.org/pt-BR/firefox/browse/type:7>>, Acesso em: 05/05/2009.

EMAIL. Web Site, Disponível em: <<http://gmail.com>>, Acesso em 26 jun 2009.

JAVA FREE. Manipulando campos BLOB e CLOB com JDBC, Disponível em: <<http://www.javafree.org/artigo/851262/Manipulando-campos-BLOB-e-CLOB-com-JDBC.html>>, Acesso em: 13/05/2009

LAST.FM, Disponível em: <<http://www.lastfm.com.br>>, Acesso em 26 jun 2009.

LOJA CTIS, Disponível em: <<http://www.lojactis.com.br>>, Acesso em 12/04/2009.

MICROSOFT Downloads, Disponível em: <<http://www.microsoft.com/downloadS/en/default.aspx>>, Acesso em 05/05/2009.

MYSQL. The BLOB and Text DataTypes. Disponível em: <<http://dev.mysql.com/doc/refman/5.0/en/blob.html>>, Acesso em 22 abr 2009.

Openbiblio. Web site. Disponível em: <<http://obiblio.sourceforge.net>>. Acesso em 06 jul 2009.

ORKUT, Disponível em: <<http://www.orkut.com>>, Acesso em: 26/06/2009.

Pinheiro (2008), Biometria nos Sistemas Computacionais Você é a Senha., Editora Ciência Moderna.

PHP, <http://br.php.net/>, Acesso em 22 abr 2009

PHP Manual, Disponível em: <[http://www.php.net/manual/pt\\_BR/book.image.php](http://www.php.net/manual/pt_BR/book.image.php)>, Acesso em 06/08/2009

WAZ, Web Site. Disponível em: <<http://www.waz.com.br>>, Acesso em: 12/04/2009.

YAHOO, Yahoo! Respostas, Web Site, Disponível em: <<http://br.answers.yahoo.com>>, Acesso em: 26/06/2009.

1GRAU, Web Site, Disponível em: <<http://pt.wikipedia.org/wiki/1Grau>>., Acesso em: 26/06/2009.