

## 9

## Referências

- ALY, S.; YOUSSEF, A. **Synchronization-Sensitive Frame Estimation: Video Quality Enhancement**. Multimedia Tools and Applications, 2002.
- AMIR, A.; PONCELEON, D.; BLANCHARD, B.; PETKOVIC, D.; SRINIVASAN, S.; Cohen, G. **Using Audio Time Scale Modification for Video Browsing**. Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.
- ARONS, B. **Techniques, Perception, and Applications of Time-Compressed Speech**. Proceedings of American Voice I/O Society, 1992.
- ARONS, B. **Efficient Listening with Two Ears: Dichotic Time Compression and Spatialization**. Proceedings of the 1994 International Conference of Auditory Display, 1994.
- ATSC. **Digital Audio Compression Standard (AC-3)**. 1995.
- AUDIO ENGINEERING SOCIETY. **Perceptual Audio Coders: What to Listen For**. AES Technical Council, 2001.
- BACHELET, B.; MAHEY, P.; RODRIGUES, R.F.; SOARES, L.F.G. **Elastic Time Computation in QoS-Driven Hypermedia Presentations**. Research Report RR-04-16, LIMOS, Blaise-Pascal University, Clermont-Ferrand, França, 2004.
- BERNSEE, S.M. **Time Stretching and Pitch Shifting of Audio Signals**. DSP Dimension, 2003. Disponível em <http://www.dspdimension.com>. Acesso em set. 05.
- BERNSEE, S.M. **C/C++ Library for High Quality Audio Time Stretching and Pitch Shifting**. DSP Dimension, 2005. Disponível em <http://www.dspdimension.com/>. Acesso em set. 05.

- CAVENDISH, S.A. **Ferramenta de Adaptação de Ajuste Elástico em Fluxos MPEG2**. Dissertação de Mestrado, Departamento de Informática – PUC-Rio, Rio de Janeiro, Brasil, 2005.
- COVELL, M.; SLANEY, M.; ROTHSTEIN, A. **FastMPEG: Time-Scale Modification of Bit-Compressed Audio Information**. Speech and Signal Processing (ICASSP), Proceedings of the 2001 International Conference on Acoustics (IEEE), Salt Lake City, UT, 2001.
- COVELL, M.; SLANEY, M.; ROTHSTEIN, A.; **FastMPEG: Time-Scale Modification of Bit-Compressed Audio Information**. 2001. Disponível em <http://www.slaney.org/covell/FastMPEG/>. Acesso em out. 2005.
- DOLBY. **Dolby Laboratories**. 2005. Disponível em <http://www.dolby.com/>. Acesso em set. 2005.
- DOLBY. **Dolby Solutions for Signal Processing**. 2005. Disponível em [http://www.dolby.com/professional/pro\\_audio\\_engineering/solutions\\_signalprocessing.html](http://www.dolby.com/professional/pro_audio_engineering/solutions_signalprocessing.html). Acesso em out. 2005.
- ENOUNCE. **Enounce 2xAV**. 2003. Disponível em <http://www.enounce.com/>. Acesso em jan. 05.
- FFMPEG. **ffmpeg**. Disponível em <http://sourceforge.net/projects/ffmpeg/>. Acesso em set. 05.
- GABOR, D. **Theory of Communication**. Journal of Institution of Electrical Engineers, 1946.
- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison Wesley, Primeira edição, 1995.
- GOLUBCHIK, L.; LUI, J.; MUNTZ, R. **Reducing I/O Demand in Video-On-Demand Storage Servers**. ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, 1995.
- HAMMER, F., **Time-scale Modification using the Phase Vocoder**, Dissertação de Mestrado, Institute for Electronic Music and Acoustics (IEM), Graz University of Music and Dramatic Arts, A-8010 Graz, Austria, 2001.

- ISO/IEC. **Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s - Part 3: Audio.** 11172-3, 1993.
- ISO/IEC. **Information technology - Generic coding of moving pictures and associated audio information - Part 7: Advanced Audio Coding (AAC).** 13818-7, 1997.
- ISO/IEC. **Information technology - Generic coding of moving pictures associated audio information - Part 3: Audio.** 13818-3, 1998.
- ISO/IEC. **Information technology - Generic coding of moving pictures and associated audio information: Systems.** 13818-1, 2000.
- ISO/IEC. **Information technology - Generic coding of moving pictures and associated audio information: Video.** 13818-2, 2000.
- ISO/IEC. **Information technology - Coding of audio-visual objects - Part 3: Systems.** 14496-1, 2001.
- ISO/IEC. **Information technology - Coding of audio-visual objects - Part 3: Audio.** 14496-3, 2001.
- ISO/IEC. **Information technology -- Generic coding of moving pictures and associated audio information -- Part 7: Advanced Audio Coding (AAC).** 13818-7, 2004.
- ITU-T. **Subjective audiovisual quality assessment methods for multimedia applications.** Recomendação P.911, 1998.
- JAIN, R. **The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.** John Wiley & Sons, 1991.
- JFFMPEG. **Codec pack for the Java Media Framework**, 2005. Disponível em <http://jffmpeg.sourceforge.net/>. Acesso em set. 2005.
- LAROCHE, J.; DOLSON, M. **Improved Phase Vocoder Time-scale Modification of Audio**, IEEE Transactions on Speech and Audio Processing, 1999.
- LEMAY, J.C. **Sample Calculators**, 1998. Disponível em [http://mp3.deepsound.net/eng/samples\\_calculs.php](http://mp3.deepsound.net/eng/samples_calculs.php). Acesso em set. 2005.

- LEE, E.; NAKRA, T. M.; BORCHERS, J. **You're The Conductor: a Realistic Interactive Conducting System for Children**, Proceedings of the NIME 2004 International Conference on New Interfaces for Musical Expression, Japão, 1998.
- LEVINE, S.; SMITH, J.O.III. **A Sines+Transients+Noise Audio Representation for Data Compression and Time/Pitch-Scale Modifications**. 105th Audio Engineering Society Convention, EUA, 1998.
- LEVINE, S. **Audio Representations for Data Compression and Compressed Domain Processing**. Tese de Doutorado, Stanford University, 1998. Disponível em <http://www-ccrma.stanford.edu/~scottl/thesis.html>. Acesso em out. 2005.
- LIN, C.; ZHOU, J.; YOUN, J.; SUN, M. **MPEG Video Streaming with VCR Functionality**. IEEE Transactions of Circuits and Systems for Video Technology, 2001.
- MICROSOFT CORPORATION. **Windows Media Player**. 2005. Disponível em <http://www.microsoft.com/windows/windowsmedia/>. Acesso em dez. 05.
- MOVING PICTURE EXPERTS GROUP. **Reference software ISO/IEC 14496-5**. 2001. Disponível em <http://www.chiariglione.org/mpeg/standards.htm>. Acesso em nov. 2005.
- MUCHALUAT-SAADE, D.C.; SILVA, H.V.O. **NCL 2.0: Exploiting and Integrating New Concepts to Hypermedia Declarative Languages**. Relatório técnico, Laboratório Telemídia, 2004. Disponível em <http://www.telemidia.puc-rio.br/>. Acesso em out. 2005.
- NILSSON, M. **ID3v2**. 2003. Disponível em <http://www.id3.org/>. Acesso em jun. 05.
- NOLL, P. **Digital Audio for Multimedia**. Proceedings Signal Processing for Multimedia. NATO Advanced Audio Institute, 1999.
- NOLL, P. **MPEG Digital Audio Coding Standards**. CRC Press LLC, 2000. Disponível em [http://www.ece.ucdavis.edu/~mihaela/mpeg\\_audio\\_coding.pdf](http://www.ece.ucdavis.edu/~mihaela/mpeg_audio_coding.pdf). Acesso em jan. 05.

- NULLSOFT. **Winamp 5.11 Player**. 2005. Disponível em <http://www.winamp.com/>. Acesso em nov 05.
- OMOIGUI, N.; HE, L.; GUPTA, A.; GRUDIN, J.; SANACKI, E. **Time-Compression: Systems Concerns, Usage, and Benefits**. Proceedings of the SIGCHI conference on Human factors in computing systems, EUA, 1999.
- PERKINS, C. **RTP: Audio and Video for the Internet**. Addison Wesley, 2003.
- PORTNOFF, M. R. **Time-scale Modification of Speech Based on Short-time Fourier Analysis**. IEEE Transactions on Acustics, Speech and Signal Processing, 1981.
- PROSONIQ MPEX. **Minimum Perceived Loss Time Compression/Expansion**. 2004. Disponível em <http://mpex.prosoniq.com/>. Acesso em set. 2005.
- PURNHAGEN, H. **MPEG Audio FAQ**. 2004. Disponível em <http://www.tnt.uni-hannover.de/project/mpeg/audio/faq/mpeg2.html>. Acesso em nov. 05.
- RODRIGUES, R.F. **Formatação e Controle de Apresentações Hiperídia com Mecanismos de Adaptação Temporal**. Tese de Doutorado, Departamento de Informática – PUC-Rio, Rio de Janeiro, Brasil, 2003.
- RONI MUSIC. **Amazing Slow Downer - an Easy Way to Slow Down the Music without Changing the Pitch**. 2005. Disponível em <http://www.ronimusic.com/>. Acesso em out. 05.
- SOARES, L.F.G. **Notas de Aula do Curso Fundamentos de Multimídia**. Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil, 2005.
- SONY. **The Sound Forge Product Family**. 2005. Disponível em <http://www.soundforge.com>. Acesso em ago. 2005.
- SUN MICROSYSTEMS. **Java Media Framework API Specification**. 1999. Disponível em <http://java.sun.com/jmf>. Acesso em set. 05.
- VIDEOLAN. **VLC media player**. 2005. Disponível em <http://www.videolan.org/vlc/>. Acesso em nov. 2005.
- WINKLER, S.; FALLER, C. **Audiovisual Quality Evaluation of Low-Bitrate Video**. Proc. SPIE/IS&T Human Vision and Electronic Imaging, 2005.

WOLTERS, M.; KJÖRLING, M.; HOMM, D.; PURNHAGEN, H. **A Closer Look into MPEG-4 High Efficiency AAC**. 2003. Disponível em [www.telos-systems.com/techtalk/hosted/HE-AAC.pdf](http://www.telos-systems.com/techtalk/hosted/HE-AAC.pdf). Acesso em nov. 2005.

## 10 Apêndice A: Exemplos de Código para Solicitar Ajuste Elástico

A ferramenta de ajuste foi desenvolvida para que outras aplicações possam facilmente solicitar ajuste elástico utilizando a API especificada na Subseção 4.1.

A Tabela 15 ilustra um trecho de código em Java simplificado para uma aplicação cliente solicitar a geração em tempo de compilação de um arquivo contendo o fluxo de áudio e a apresentação do fluxo processado no exibidor de conteúdo VLC (Videolan, 2005).

O código apresentado a seguir é dividido em quatro partes. A primeira delas é a configuração da ferramenta de ajuste. Para tal, é necessário chamar o método `config` passando o arquivo de entrada, o nome do arquivo de saída e o fator de ajuste. Além disso, a aplicação cliente se registra como observadora do evento de finalização de ajuste. A seguir, é necessário iniciar o processamento. Nessa parte, a aplicação cliente optou por criar uma nova *thread*. Na terceira parte, a aplicação cliente descobre o novo valor de um determinado instante da mídia original e imprime o valor encontrado. A última parte é o código da aplicação que deve ser executado quando esta foi avisada que o arquivo contendo o áudio processado já foi gerado. A aplicação cliente agora pode solicitar a apresentação do arquivo processado utilizando, no exemplo, o exibidor de conteúdo VLC.

```
//Para configurar a ferramenta de ajuste elástico
ICompileTimeTimescalePlayer    timescalePlayer    =    new
TimescalePlayer();
String inputFile = "C:/MediaFiles/FileAudio.mp3";
Double factor = 0.90;
String outputFile =
    Functions.getOutputFileName(inputFile,outputFile,factor);
int inputType = IOConstants.InputSource.LOCAL;
InputTools inputTools =
    InputToolsBuilder.getInputTools(inputFile, inputType);
timescalePlayer.config(inputTools, outputFile, factor);
timescalePlayer.addTimescaleListener(this);
```

```
//Para iniciar a geração do ajuste
Thread t = new Thread(timescalePlayer);
t.start();

//Para descobrir o valor de uma âncora
double originalInstant = 10;
double t = timescalePlayer.getTimescaleInstant(originalInstant);
String result = "Mudou instante " + originalInstant + " por " + t;
System.out.println(result);

//Quando é avisado do evento de finalização do ajuste
String vlcPath = IOConstants.VLC_PATH;
IThirdPartyPlayer thirdPartPlayer = new VLCFilePlayer(vlcPath);
String outputFile = timescalePlayer.getOutputTools().getName();
thirdPartPlayer.setInput(outputFile);
thirdPartPlayer.prepare();
thirdPartPlayer.play();
```

Tabela 15 - Trechos de código para solicitar a geração e exibição de um arquivo ajustado.

Um segundo exemplo é ilustrado na Tabela 16. Nesse caso, uma aplicação cliente está solicitando ajuste elástico em tempo de exibição com a apresentação do fluxo processado no exibidor de conteúdo JMF (Sun, 1999).

O código apresentado a seguir é dividido em quatro partes. A primeira delas é a configuração da ferramenta de ajuste. Para tal, é necessário chamar o método `config` passando o arquivo de entrada, o fator de ajuste e os instantes de tempo a serem monitorados. Além disso, a aplicação cliente se registra como observadora dos eventos de monitoramento de instantes de tempo. A seguir, é necessário iniciar a geração do ajuste. Nessa parte, a aplicação cliente optou por criar uma nova *thread*. A terceira parte é a configuração da ferramenta de exibição, que é executada na *thread* corrente. A última parte é o código da aplicação que deve ser executado quando esta foi avisada que o valor de um instante monitorado foi recalculado. O código de exemplo apenas exibe o novo valor encontrado.

```
//Para configurar a ferramenta de ajuste elástico
IExecutionTimeTimescalePlayer timescalePlayer = new
TimescalePlayer();
String inputFile = "C:/MediaFiles/FileAudio.mp3";
Double factor = 0.90;
int inputType = IOConstants.InputSource.LOCAL;
InputTools inputTools =
```



|                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> InputToolsBuilder.getInputTools(inputFile, inputType); IntantsSet instants = new InstantsSet(getInstants()); this.timescalePlayer.config(inputTools, factor, instants); this.timescalePlayer.addInstantListener(this.guiUpdater); </pre> |
| <pre> //Para iniciar a geração do ajuste e exibição de conteúdo Thread t = new Thread(timescalePlayer); t.start(); </pre>                                                                                                                      |
| <pre> IThirdPartyPlayer ap = new JMFDataSourcePlayer(); ByteOutputTools out = (ByteOutputTools) timescalePlayer.getOutputTools(); InputTools in = out.getInputTools(); ap.setInput(in); ap.prepare(); ap.play(); </pre>                        |
| <pre> //Quando é avisado de descoberta de valor de instante monitorado String result = "Mudou instante " + event.getOriginalInstant() + " por " + event.getUpdatedInstant()+ "\n"; System.out.println(result); </pre>                          |

Tabela 16 - Trechos de código para solicitar a exibição de um fluxo sendo ajustado.

## 11 Apêndice B:

### Aplicação de envio e recepção de fluxo de mídia

Uma aplicação foi desenvolvida com objetivo de transmitir dados de mídia, recebê-los e encaminhá-los à ferramenta de ajuste elástico para apresentação em tempo de exibição. Para tal, dois novos subsistemas, um transmissor e um receptor, foram desenvolvidos, como ilustrado na Figura 75.

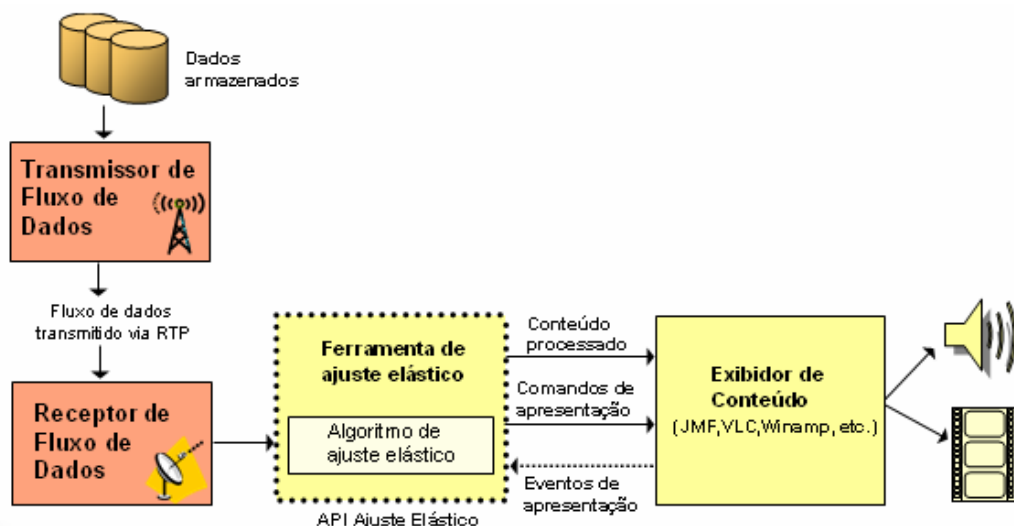


Figura 75 - Exibição de uma mídia com aplicação de ajuste elástico recebidas em tempo de exibição.

O subsistema **Transmissor** gera um fluxo de mídia a partir da leitura de um arquivo e o transmite utilizando o protocolo RTP. É esse subsistema quem inicia e controla a transmissão. O subsistema **Receptor** está sempre esperando receber fluxo em uma porta específica. Quando os dados chegam, o receptor os envia para a ferramenta de ajuste processá-lo e posteriormente exibi-lo utilizando a ferramenta de exibição JMF.

## 12 Apêndice C: Exemplo da Integração com Formatador HyperProp

A Tabela 17 ilustra um documento NCL (Muchaluat & Silva, 2004) que precisa de funções de ajuste para ser corretamente apresentado pelo formatador HyperProp. Nas linhas 39 a 61 do documento, é especificado que a apresentação hipermídia é formada pelas seguintes mídias: um documento HTML contendo o título da apresentação, uma música em MP3, sua letra em HTML e 20 imagens. Os relacionamentos que devem ocorrer durante a apresentação, apresentados nas linhas 62 a 155, são os seguintes:

- No início da apresentação, devem ser exibidas as mídias do título e da letra da música, a primeira imagem e começar a tocar o MP3 da música.
- As 20 imagens devem ser exibidas em seqüência, ou seja, inicialmente exibe a primeira durante um tempo, depois exibe a segunda também por um tempo, terceira até a vigésima. Desse modo, em nenhum momento duas imagens são exibidas simultaneamente.
- A apresentação do título, da letra da música, da vigésima imagem e do MP3 devem terminar simultaneamente.

As linhas 6 a 14 indicam a região do dispositivo de exibição onde cada mídia deve ser exibida. Todas as imagens devem ser exibidas na mesma região, só que em instantes diferentes. As linhas 26 a 35 especificam as características de exibição das mídias. Dentre elas, tem-se que o título da apresentação e a letra da música devem ser exibidos por 96 segundos, que a duração esperada de exibição de cada imagem é de 6 segundos, o nome da ferramenta de exibição do áudio e que a música e cada imagem possuem duração flexível especificadas por funções de custo. A duração esperada da música não é especificada no documento porque é calculada pela sua ferramenta de exibição.

As funções de custo são definidas nas linhas 14 a 25. Cada uma delas especifica os valores permitidos para o fator de ajuste (`deltaShrink` e

deltaStretch), o custo máximo para aumentar ou diminuir a duração da mídia (maxShrinkCost e maxStretchCost) e o tipo da função (type).

Para respeitar as restrições do documento, a apresentação deve durar 96 segundos. Considerando que o tempo esperado para exibir a sequência de imagens é de  $20 \times 6 = 120$  segundos, é necessário diminuir o tempo esperado de exibição de cada imagem. Em média, cada imagem deverá durar  $96/20 = 4.8$  segundos, que implica na utilização de um fator de ajuste igual a 0.8. O áudio MP3 pode ter duração entre  $96 \times 0.9 (=86.4)$  e  $96 \times 1.1 (=105.6)$  segundos. Supondo que o áudio MP3 dure 100 segundos, a ferramenta de ajuste deve ser utilizada com fator igual a 0.96.

Por fim, caberá ao formatador escolher se o ajuste deve ser realizado em tempo de compilação ou de exibição instanciando o adaptador para exibidor de conteúdo adequado.

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <ncl id="LaVieEnRose" xmlns="http://www.telemidia.puc-rio.br/specs/xml/NCL22/profiles"
03 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
04 xsi:schemaLocation="http://www.telemidia.puc-rio.br/specs/xml/NCL22/profiles/NCL22.xsd">
05   <head>
06     <regionBase>
07       <region id="menuWindow" title="Seminaire du Limos (R. F. Rodrigues)" width="1024"
08 height="740">
09         <region id="titleRegion" width="100%" height="100" />
10         <region id="lyricsRegion" top="100" width="40%" height="590" />
11         <region id="imageRegion" top="100" left="40%" width="60%" height="590" />
12       </region>
13     </regionBase>
14     <costFunctionBase>
15       <costFunction id="costFunction01" xsi:type="linear" deltaShrink="10%"
16 deltaStretch="10%">
17         <costFunctionParam name="maxShrinkCost" value="1000"/>
18         <costFunctionParam name="maxStretchCost" value="1000"/>
19       </costFunction>
20       <costFunction id="costFunction02" xsi:type="linear" deltaShrink="30%"
21 deltaStretch="30%">
22         <costFunctionParam name="maxShrinkCost" value="5000"/>
23         <costFunctionParam name="maxStretchCost" value="5000"/>
24       </costFunction>
25     </costFunctionBase>
26     <descriptorBase>
27       <descriptor id="titleDesc" region="titleRegion" explicitDur="96s"/>
28       <descriptor id="lyricsDesc" region="lyricsRegion" explicitDur="96s"/>
29       <descriptor id="audioDesc01" player="HYPF_FlexMP3AudioPlayer"
30 costFunction="costFunction01">
31         <descriptorParam name="soundLevel" value="0.3"/>
32       </descriptor>
33       <descriptor id="imageDesc" region="imageRegion" explicitDur="6s"
34 costFunction="costFunction02"/>
35     </descriptorBase>
36   <connectorBase><import alias="connBase" baseURI="limos.conn"/></connectorBase>
```

```

37 </head>
38 <body><port id="entryPort" component="title-lvr01"/>
39   <media type="audio" descriptor="audioDesc01" id="song-lvr01" src="laVieenRose.mp3"/>
40   <media type="text" descriptor="titleDesc" id="title-lvr01" src="lyrics/title.html"/>
41   <media type="text" descriptor="lyricsDesc" id="lvr01-lyrics" src="lyrics/lyr.html"/>
42   <media type="text" descriptor="imageDesc" id="lvr01-img1" src="images/image01.html"/>
43   <media type="text" descriptor="imageDesc" id="lvr01-img2" src="images/image02.html"/>
44   <media type="text" descriptor="imageDesc" id="lvr01-img3" src="images/image03.html"/>
45   <media type="text" descriptor="imageDesc" id="lvr01-img4" src="images/image04.html"/>
46   <media type="text" descriptor="imageDesc" id="lvr01-img5" src="images/image05.html"/>
47   <media type="text" descriptor="imageDesc" id="lvr01-img6" src="images/image06.html"/>
48   <media type="text" descriptor="imageDesc" id="lvr01-img7" src="images/image07.html"/>
49   <media type="text" descriptor="imageDesc" id="lvr01-img8" src="images/image08.html"/>
50   <media type="text" descriptor="imageDesc" id="lvr01-img9" src="images/image09.html"/>
51   <media type="text" descriptor="imageDesc" id="lvr01-img10" src="images/image10.html"/>
52   <media type="text" descriptor="imageDesc" id="lvr01-img11" src="images/image11.html"/>
53   <media type="text" descriptor="imageDesc" id="lvr01-img12" src="images/image12.html"/>
54   <media type="text" descriptor="imageDesc" id="lvr01-img13" src="images/image13.html"/>
55   <media type="text" descriptor="imageDesc" id="lvr01-img14" src="images/image14.html"/>
56   <media type="text" descriptor="imageDesc" id="lvr01-img15" src="images/image15.html"/>
57   <media type="text" descriptor="imageDesc" id="lvr01-img16" src="images/image16.html"/>
58   <media type="text" descriptor="imageDesc" id="lvr01-img17" src="images/image17.html"/>
59   <media type="text" descriptor="imageDesc" id="lvr01-img18" src="images/image18.html"/>
60   <media type="text" descriptor="imageDesc" id="lvr01-img19" src="images/image19.html"/>
61   <media type="text" descriptor="imageDesc" id="lvr01-img20" src="images/image20.html"/>
62   <link id="link1" xconnector="connBase#onBeginStart">
63     <bind component="title-lvr01" role="onBegin"/>
64     <bind component="song-lvr01" role="start"/>
65     <bind component="lvr01-img1" role="start"/>
66     <bind component="lvr01-lyrics" role="start"/>
67   </link>
68   <link id="link2" xconnector="connBase#endConstraint">
69     <bind component="song-lvr01" role="xEnd"/>
70     <bind component="title-lvr01" role="yEnd"/>
71   </link>
72   <link id="link3" xconnector="connBase#endConstraint">
73     <bind component="title-lvr01" role="xEnd"/>
74     <bind component="lvr01-img20" role="yEnd"/>
75   </link>
76   <link id="link4" xconnector="connBase#endConstraint">
77     <bind component="title-lvr01" role="xEnd"/>
78     <bind component="lvr01-lyrics" role="yEnd"/>
79   </link>
80   <link id="linkImage01" xconnector="connBase#onEndStart">
81     <bind component="lvr01-img1" role="onEnd"/>
82     <bind component="lvr01-img2" role="start"/>
83   </link>
84   <link id="linkImage02" xconnector="connBase#onEndStart">
85     <bind component="lvr01-img2" role="onEnd"/>
86     <bind component="lvr01-img3" role="start"/>
87   </link>
88   <link id="linkImage03" xconnector="connBase#onEndStart">
89     <bind component="lvr01-img3" role="onEnd"/>
90     <bind component="lvr01-img4" role="start"/>
91   </link>
92   <link id="linkImage04" xconnector="connBase#onEndStart">
93     <bind component="lvr01-img4" role="onEnd"/>
94     <bind component="lvr01-img5" role="start"/>

```

```
95     </link>
96     <link id="linkImage05" xconnector="connBase#onEndStart">
97         <bind component="lvr01-img5" role="onEnd"/>
98         <bind component="lvr01-img6" role="start"/>
99     </link>
100    <link id="linkImage06" xconnector="connBase#onEndStart">
101        <bind component="lvr01-img6" role="onEnd"/>
102        <bind component="lvr01-img7" role="start"/>
103    </link>
104    <link id="linkImage07" xconnector="connBase#onEndStart">
105        <bind component="lvr01-img7" role="onEnd"/>
106        <bind component="lvr01-img8" role="start"/>
107    </link>
108    <link id="linkImage08" xconnector="connBase#onEndStart">
109        <bind component="lvr01-img8" role="onEnd"/>
110        <bind component="lvr01-img9" role="start"/>
111    </link>
112    <link id="linkImage09" xconnector="connBase#onEndStart">
113        <bind component="lvr01-img9" role="onEnd"/>
114        <bind component="lvr01-img10" role="start"/>
115    </link>
116    <link id="linkImage10" xconnector="connBase#onEndStart">
117        <bind component="lvr01-img10" role="onEnd"/>
118        <bind component="lvr01-img11" role="start"/>
119    </link>
120    <link id="linkImage11" xconnector="connBase#onEndStart">
121        <bind component="lvr01-img11" role="onEnd"/>
122        <bind component="lvr01-img12" role="start"/>
123    </link>
124    <link id="linkImage12" xconnector="connBase#onEndStart">
125        <bind component="lvr01-img12" role="onEnd"/>
126        <bind component="lvr01-img13" role="start"/>
127    </link>
128    <link id="linkImage13" xconnector="connBase#onEndStart">
129        <bind component="lvr01-img13" role="onEnd"/>
130        <bind component="lvr01-img14" role="start"/>
131    </link>
132    <link id="linkImage14" xconnector="connBase#onEndStart">
133        <bind component="lvr01-img14" role="onEnd"/>
134        <bind component="lvr01-img15" role="start"/>
135    </link>
136    <link id="linkImage15" xconnector="connBase#onEndStart">
137        <bind component="lvr01-img15" role="onEnd"/>
138        <bind component="lvr01-img16" role="start"/>
139    </link>
140    <link id="linkImage16" xconnector="connBase#onEndStart">
141        <bind component="lvr01-img16" role="onEnd"/>
142        <bind component="lvr01-img17" role="start"/>
143    </link>
144    <link id="linkImage17" xconnector="connBase#onEndStart">
145        <bind component="lvr01-img17" role="onEnd"/>
146        <bind component="lvr01-img18" role="start"/>
147    </link>
148    <link id="linkImage18" xconnector="connBase#onEndStart">
149        <bind component="lvr01-img18" role="onEnd"/>
150        <bind component="lvr01-img19" role="start"/>
151    </link>
152    <link id="linkImage19" xconnector="connBase#onEndStart">
```

```
153     <bind component="lvr01-img19" role="onEnd"/>  
154     <bind component="lvr01-img20" role="start"/>  
155     </link>  
156 </body>  
157 </ncl>
```

Tabela 17 - Hiperdocumento escrito em NCL contendo especificação que precisa da aplicação de ajuste elástico para correta exibição.

## 13 Apêndice D: Aplicação de Teste da Ferramenta de Ajuste

Com o objetivo de utilizar a ferramenta de ajuste em diferentes condições e testar suas funcionalidades, uma aplicação de teste foi desenvolvida. A Figura 76 ilustra a interface gráfica dessa aplicação.

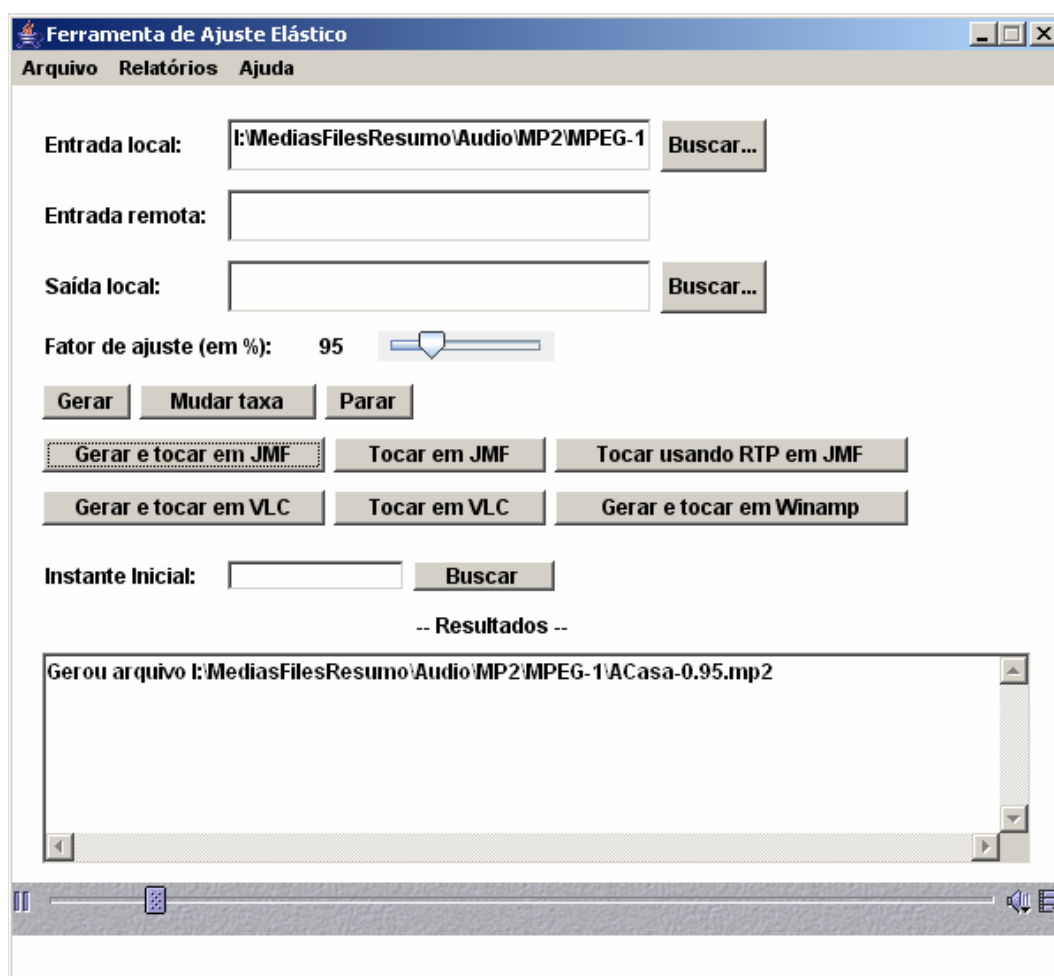


Figura 76 - Interface gráfica da aplicação de teste.

As funcionalidades da ferramenta de ajuste disponíveis através dessa interface desenvolvida são:

- Solicitar o ajuste de um arquivo de áudio, vídeo ou sistemas gerando como saída outro arquivo contendo a mídia processada (este arquivo pode ser posteriormente exibido ou não).



- Tocar um arquivo de mídia com a aplicação de ajuste elástico em tempo de exibição.
- Mudar o fator de ajuste utilizado no algoritmo de ajuste durante a geração de um novo arquivo ou a exibição de um fluxo que está sendo processado em tempo de exibição.
- Cancelar a aplicação de ajuste elástico e/ou a exibição de um fluxo de mídia.
- Visualizar o novo valor de um ou mais instantes de tempo durante a realização de um ajuste ou ao seu final.
- Visualizar um relatório contendo informações do último ajuste elástico realizado (quadros processados e tempo gasto para executar ajuste).

Os arquivos de mídia a serem processados podem estar locais na máquina ou remotos. Em qualquer dos casos, a aplicação de teste inicia e controla a transferência dos dados de mídia. Os exibidores de conteúdo utilizados foram: JMF, VLC e Winamp (ver Capítulo 5).