

**FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”  
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**CARLOS EDUARDO SANVIDO**

**GERADOR DE APRESENTAÇÕES**

**MARÍLIA  
2005**

**CARLOS EDUARDO SANVIDO**

**GERADOR DE APRESENTAÇÕES**

Monografia realizada no Curso de Bacharelado em  
Ciência da Computação da Fundação de Ensino  
“Eurípides Soares Da Rocha”, UNIVEM – Centro  
Universitário Eurípides de Marília, sob orientação do  
professor Dr. José Remo Ferreira Brega.

**MARÍLIA**  
**2005**

**CARLOS EDUARDO SANVIDO**

**GERADOR DE APRESENTAÇÕES**

Banca examinadora da monografia apresentada ao Curso de Ciência da Computação da UNIVEM,/F.E.E.S.R., para obtenção do Título bacharel em Ciência da Computação.

Resultado: \_\_\_\_\_

ORIENTADOR: Prof. Dr. José Remo Ferreira Brega.

1º EXAMINADOR: \_\_\_\_\_

2º EXAMINADOR: \_\_\_\_\_

Marília, 25 de novembro de 2005.

*Dedico esse trabalho a Deus pela paciência que nos  
deu nesses caminhos tão difíceis que trilhamos...*

## AGRADECIMENTOS

Em primeiro lugar agradeço aos meus pais José Vicente Sanvido e Sandra Aparecida Marques Sanvido, e a meus irmãos, Sergio e Andréia, pelo incentivo, mesmo estando à distância me proporcionaram todo o apoio.

A minha noiva Tathiane que me apoiou muito, mesmo nos momentos em que estive distante me dedicando a faculdade.

Aos meus colegas de classe, e um grande amigo Fabim, que esteve ao meu lado em todos os momentos de alegrias e tristezas que tivemos em Marília.

Ao Prof. Dr. José Remo Ferreira Brega, pelo seu apoio e confiança depositado em mim, para a execução dessa monografia.

A todos professores da Fundação de Ensino Eurípides Soares da Rocha.

SANVIDO, Carlos E. **GERADOR DE APRESENTAÇÕES: Sistema Gerador de Apresentações** 2005 , 75 f (Monografia do Curso de Bacharelado em Ciência da Computação) UNIVEM – Centro Universitário Eurípides de Marília , Fundação de Ensino Eurípides Soares da Rocha , Marília, 2005.

## RESUMO

Esse texto procura esclarecer o funcionamento de um sistema Gerador de Apresentações, aplicado a multi-plataformas, tanto o sistema gerador quanto os slides gerados por ele. Sua produção foi baseada a partir de livros ligados a sistemas multi-plataformas, contribuindo para que haja um melhor entendimento do assunto, formando assim um material de fácil entendimento e compreensão para auxiliar os estudantes de ciência da computação.

**Palavras-chaves:** slides, Java, xml, html, multiplataformas.

## LISTA DE FIGURAS

Figura 1 - Comandos básico do HTML.....	15
Figura 2 - Índice gerado pelo sistema.....	22
Figura 3 – Processo de transformação.....	22
Figura 4 – Código HMTL do índice.....	24
Figura 5 - Código XML slide Texto & Figura .....	25
Figura 6 – Tela inicial do Sistema.....	26
Figura 7 – Janela para criação de Slide Texto.....	27
Figura 8 – Janela para criação de Slide Texto com Figuras .....	28
Figura 9 – Janela procura de arquivos .....	29
Figura 10 – Visualização na criação do slide .....	30
Figura 11 – Janela para criação do Slide Figura.....	31
Figura 12 – Abrindo um Slide .....	32
Figura 13 – Slide tipo Texto gerado pelo sistema .....	33
Figura 14 – Slide tipo Texto & Figura gerado pelo sistema.....	34
Figura 15 – Slide tipo Figura gerado pelo sistema .....	35

## SUMÁRIO

1. INTRODUÇÃO.....	9
1.1 Objetivos.....	10
1.2 Estrutura do trabalho .....	10
2. A LINGUAGEM JAVA.....	11
2.1 A história de Java .....	11
2.2 Características da linguagem Java.....	12
2.2.1 Orientada a Objetos .....	12
2.2.2 Independente de Plataforma .....	13
2.2.3 Interface Gráfica .....	13
3. HTML.....	14
3.1 As Tags.....	14
3.2 Estrutura de um arquivo HTML .....	15
3.2.1 Elemento HTML .....	16
3.2.2 Elemento HEAD.....	16
3.2.3 Elemento BODY .....	17
4. XML – Uma linguagem de marcação estendida .....	18
4.1 JAVA E XML.....	19
4.1.1 A Biblioteca JDOM.....	20
5. GERADOR DE APRESENTAÇÕES .....	21
5.1 Estrutura .....	21
5.1.1 Código HTML gerado. ....	23
5.1.2 Código XML gerado.....	24
5.2 O funcionamento do Sistema.....	25
5.3 Exemplo do Sistema Gerador de Apresentações.....	26
5.4 Criando um Slide Texto.....	26
5.5 Criando um Slide Texto com Figura .....	27
5.6 Criando um Slide de Figura.....	30
5.7 Abrindo um Slide .....	31
5.8 Modelo Slide Texto Gerado .....	32
5.9 Modelo Slide Texto & Figura Gerado.....	33
5.10 Modelo Slide Figura Gerado .....	34



6. CONCLUSÕES .....	36
6.1 Aprendizado .....	36
6.2 Trabalhos futuros .....	36
7. BIBLIOGRAFIA .....	38
APÊNDICE A .....	39
APÊNDICE B .....	65
APÊNDICE C .....	67
APÊNDICE D .....	69
APÊNDICE E .....	70
APÊNDICE F .....	71
APÊNDICE G .....	73
APÊNDICE H .....	75

## 1. INTRODUÇÃO

Nos dias de hoje a apresentação em *slides* tem se tornado a principal forma de transmissão de informação nas faculdades, palestras e também aos poucos nas escolas de primeiro e segundo grau, pois a sua utilização faz com que as apresentações fiquem mais dinâmicas. A visualidade dos slides além de ajudar quem está transmitindo a informação faz também com que os espectadores observem mais facilmente as informações passadas.

Existem vários sistemas geradores de apresentações, portanto são limitados, por serem sistemas de plataformas únicas, tanto os sistemas geradores quanto os slides gerados.

Após visto essa carência de geradores de apresentação multi-plataformas, foi desenvolvido neste trabalho um gerador de apresentações que possa ser executado em qualquer sistema operacional, desde que tenha instalado a máquina virtual do Java.

Neste estudo, a principal característica foi que o sistema gerador além de poder ser executado em qualquer sistema operacional, os slides também podem ser visualizados em qualquer sistema, por esses slides serem um arquivo HTML, podendo ser até publicados para web.

Para armazenar as informações de configuração dos slides utilizou-se arquivos no formato XML, o que possibilita fazer alterações em slides já criados e ainda possibilita que os mesmos sejam exportados para outros aplicativos, assim o usuário desse sistema terá completa portabilidade de seus slides.

## **1.1 Objetivos**

Implementar um sistema gerador de apresentações de uso simples e prático facilitando ao usuário a criação de suas apresentações. Proporcionar ao usuário total portabilidade no sistema e em seus slides gerados, podendo ser levado a qualquer computador, sem ter a preocupação de não ter o sistema operacional específico.

## **1.2 Estrutura do trabalho**

Esta monografia está dividida em seis capítulos. Neste primeiro capítulo é introduzido o assunto tema da pesquisa e também abordando os objetivos relacionados ao trabalho.

No Capítulo 2 é apresentado um resumo sobre a linguagem Java, uma breve história, características e funcionamento.

No Capítulo 3 são abordados aspectos da linguagem de marcação HTML, que é o modelo escolhido para os slides gerados.

No Capítulo 4 é feito um estudo sobre XML, que foi o modelo adotado para o armazenamento das informações dos slides.

No Capítulo 5 está descrito todo o funcionamento do sistema gerador de apresentações, mostrando a sua forma simples de interação.

Já no Capítulo 6, Conclusão, são os resultados alcançados nesse trabalho.

E por último no Capítulo 7, encontra-se a bibliografia de todo material utilizado para a confecção da mesma.

## **2. A LINGUAGEM JAVA**

Java é uma linguagem gratuita, orientada a objetos, independente de plataformas e que possui um conjunto rico de bibliotecas que facilita muito o desenvolvimento de vários tipos de sistemas, inclusive o proposto neste estudo.

### **2.1 A história de Java**

A história do Java começou quando a Sun Microsystems financiou uma pesquisa corporativa interna com o codinome Green em 1991. O projeto resultou no desenvolvimento de uma linguagem baseada em C, C++ que seu criador, James Gosling, chamou de Oak (carvalho) em homenagem a uma árvore que dava para a janela de seu escritório na Sun. Descobriu-se mais tarde que já havia uma linguagem de computador chamada Oak. Quando uma equipe da Sun visitou uma cafeteria local, o nome Java (cidade de origem de um tipo de café importado) foi sugerido e pegou.

Mas o projeto Green passava por algumas dificuldades. O mercado para dispositivos eletrônicos inteligentes destinado ao consumidor final não estava se desenvolvendo tão rapidamente como a Sun havia previsto, e para agravar ainda mais a situação, um contrato importante pelo qual a Sun competia fora concedido à outra empresa, assim o projeto estava em risco de cancelamento. Por pura sorte, a World Wide Web explodiu em popularidade em 1993 e as pessoas da Sun viram o imediato potencial de utilizar Java para criar páginas da Web com o chamado conteúdo dinâmico. Isso deu nova vida ao projeto.

Em maio de 1995, a Sun anunciou Java formalmente em uma conferência importante. Normalmente, um evento como esse não teria gerado muita atenção. Entretanto, Java gerou interesse imediato na comunidade comercial por causa do fenomenal interesse pela World Wide Web. Java é agora utilizado para criar páginas da Web com conteúdos interativo e

dinâmico, para desenvolver aplicativos corporativos de larga escala, para aprimorar a funcionalidade de servidores da Word Wide Web (os computadores que fornecem o conteúdo que vemos em nossos navegadores da Web), fornecer aplicativos para dispositivos destinados ao consumidor final (como telefones celulares, pagers e PDAs) e assim por diante. (DEITEL, H. M. e DEITEL, P. J, 1998).

## **2.2 Características da linguagem Java**

As principais características da linguagem Java adotadas nesse estudo são: Primeiro uma característica que Java herdou do C++ é o paradigma de Programação Orientada a Objetos, segundo é uma linguagem interpretada, o que a torna uma linguagem independente de plataformas, ou seja, o mesmo programa pode ser executado em qualquer computador que possua seu interpretador, terceiro, Java oferece, dentre as funcionalidades incorporadas a sua API padrão um imenso conjunto de classes e interfaces para o desenvolvimento de aplicações gráficas. Esse conjunto facilita o desenvolvimento de interfaces gráficas com usuários as GUIs, as aplicações gráficas são criadas através do pacote javax.swing.

### **2.2.1 Orientada a Objetos**

Java é uma linguagem puramente orientada a objetos pois, com exceção de seus tipos primitivos de dados, tudo em Java são classes ou instâncias de uma classe. Java atende todos os requisitos necessários para uma linguagem ser denominada orientada a objetos.

### **2.2.2 Independente de Plataforma**

Java é uma linguagem independente de plataforma pois os programas Java são compilados para uma forma intermediária de código denominada bytecodes que utiliza instruções e tipos primitivos de tamanho fixo. Os bytecodes são como uma linguagem de máquina destinada a uma única plataforma, a máquina virtual Java (JVM – Java Virtual Machine), um interpretador de bytecodes.

### **2.2.3 Interface Gráfica**

Uma GUI fornece ao programa uma “aparência” e “funcionamento” distintos. Fornecendo a diferentes aplicativos um conjunto consistente de componentes intuitivos de interface com o usuário, as GUIs permitem ao usuário gastar menos tempo ao usar o sistema, não tendo sempre que lembrar a sequência de pressionamento de teclas, e assim deixando o sistema com uma aparência muito mais amigável.

### 3. HTML

HyperText Markup Language (HTML), uma linguagem de marcação é uma forma de descrever, usando instruções incorporadas a um documento, o que o texto do documento significa, ou como deve aparecer. Atualmente esta linguagem é usada em praticamente todas as páginas da Web, o seu código fonte deve ser escrito seguindo a sintaxe da linguagem, podendo ser editado com um simples editor, por exemplo, (Notepad, Bloco de Notas), posteriormente um browser fará a interpretação lendo o texto e procurando pelas tags que indicam como é o formato do texto para então exibí-lo.

#### 3.1 As Tags

Quando um browser exibe uma página web, ele lê o texto da página e procura símbolos especiais denominados tags, que dizem como o texto, ou informação, deve ser exibido. Por exemplo, uma tag pode dizer que um texto deve ser exibido em negrito, itálico, com determinado tipo de fonte etc. Além do aspecto físico do texto, uma tag pode indicar que determinado texto é na verdade o endereço de outras páginas Web que devem ser acessadas no caso de o texto ser selecionado.

As tags normalmente são especificadas em partes, delimitando um texto que sofrerá algum tipo de formatação. Contudo, exibem várias tags individuais.

As tags são identificadas por serem envolvidas pelos sinais “<>” ou “</>”.

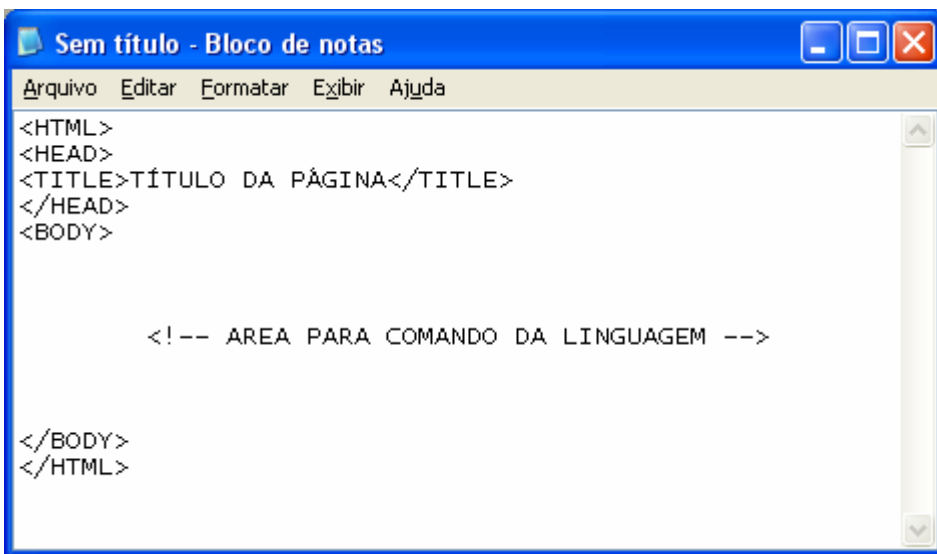
Entre os sinais “< >” são especificados os comando propriamente ditos. No caso de tags que necessitam envolver um texto, a sua finalização deve ser feita usando a barra de divisão “/”, indicando que a tag está finalizando a marcação de um texto.

O formato genérico de uma tag é :

<Nome da tag>Texto</Nome da tag>

### 3.2 Estrutura de um arquivo HTML

Assim como outras linguagens, existem algumas estruturas básicas que indicam partes de um programa HTML. Na Figura 1 é mostrado os comando básicos de um arquivo HTML.

A screenshot of a Notepad window titled "Sem título - Bloco de notas". The window has a menu bar with "Arquivo", "Editar", "Formatar", "Exibir", and "Ajuda". The text area contains the following HTML code:

```
<HTML>
<HEAD>
<TITLE>TÍTULO DA PÁGINA</TITLE>
</HEAD>
<BODY>

        <!-- AREA PARA COMANDO DA LINGUAGEM -->

</BODY>
</HTML>
```

**Figura 1 - Comandos básico do HTML**

Esses comandos não são obrigatórios nos browsers mais modernos. Dependendo da página, eles não farão falta. Os comando `<HTML>` é usado em par com o comando `</HTML>`. Ele delimita a área dos comandos da linguagem HTML. O par de comandos `<HEAD></HEAD>` é usado para especificar poucos comandos da linguagem. O mais importante destes comandos serve para criar um título que aparece na linha de título da janela do browser. O par de comandos `<BODY></BODY>` deve ser usado para envolver todos os comandos de formatação da página. Já no HTML para exibir comentário em seus arquivos é usado para iniciar o comentário “`<!--`” assim você estará estabelecendo o começo de seu comentário, podendo ter mais de uma linha, fechando com “`-->`”, assim estará finalizando o comentário. (RAMALHO, J.A, 1998).



### 3.2.1 Elemento HTML

Uso: <HTML>...</HTML>

Pode Conter: HEAD, BODY

Pode estar dentro: nada

Atributos: nenhum

O elemento HTML declara o texto contido como um documento HTML. Ele pode conter diretamente apenas dois elementos: HEAD e BODY.

### 3.2.2 Elemento HEAD

Uso: <HEAD>...</HEAD>

Pode Conter: BASE, ISINDEX, LINK, META, SCRIPT, STYLE, TITLE

Pode estar dentro: HTML

Atributos: HREF

O elemento HEAD contém informações gerais sobre o documento. Essas informações não são exibidas como parte do texto documento; e, conseqüentemente, apenas certos elementos são apropriados dentro do HEAD. Esses elementos são: BASE, ISINDEX, LINK, META, SCRIPT, STYLE e TITLE.

### 3.2.3 Elemento BODY

Uso: <BODY>...</BODY>

Pode Conter: caracteres, destaque de caracteres, A, APPLET, BR, IMG, BASEFONT, MAP, SCRIPT, INPUT, SELECT, TEXTAREA, DIR, DL, MENU, OL, UL, P, HR, Hn, ADDRESS, BLOCKQUOTE, CENTER, DIV, FORM, PRE, TABLE

Pode estar dentro: HTML

Atributos: ALINK, BACKGROUND, BGCOLOR, LINK, TEXT, VLINK

O elemento BODY contém o texto propriamente dito, ao contrário das metainformações encontradas no HEAD. Formalmente o elemento BODY não deve conter texto diretamente; em vez disso, ele deve conter elementos que por sua vez contêm o texto.

#### **4. XML – Uma linguagem de marcação estendida**

Uma linguagem de marcação é um mecanismo para identificar estruturas em um documento. A especificação da linguagem XML (*Extensible Markup Language*) define uma forma padronizada para a adição de marcações a documentos contendo informações estruturadas.

A linguagem XML, um dialeto mais simples da SGML (*Standard Generalized Markup Language*), foi projetada para facilitar publicações de informações em ambientes distribuídos e heterogêneos.

Diferente da HTML que possui um conjunto fixo e pré-definido de tags, XML é uma linguagem de marcação extensível designada para a descrição de dados em ambientes heterogêneos, como a Internet. Na verdade, XML é uma meta-linguagem (uma linguagem para a descrição de outras linguagens) que permite definir marcações específicas para um determinado domínio de aplicação.

É importante ressaltar que XML não veio para substituir o HTML, uma vez que HTML tem o foco na exibição da informação e XML na descrição da informação. Devido ao fato de prover separação entre apresentação e conteúdo, XML possibilita que o mesmo documento seja apresentado de maneiras diferentes, simplesmente pela aplicação de folhas de estilos destinadas. Desta forma, a linguagem XML permite o desenvolvimento flexível de documentos definidos pelo usuário, provendo um meio não-proprietário, robusto, verificável e padronizado para armazenamento e transmissão de dados através de uma rede de comunicação.

Os documentos XML podem ser classificados em duas categorias: os bem formados e os válidos. Cada instância de documento precisa estar de acordo com a gramática dos documentos XML. Em particular, algumas marcações são permitidas em lugares específicos,

logo o documento não é bem formatado se estas marcações ocorrem em outro lugar. Por definição, se um documento não é bem formatado, ele não é um documento XML. Um documento bem-formatado é também válido apenas se ele contém a declaração de tipo de documento apropriada e se o documento obedece às regras desta declaração. (ALMEIDA, M. 2002)

#### **4.1 JAVA E XML**

As tecnologias Java e XML permitem a criação de aplicações portáteis, distribuídas e completamente independentes de plataforma. Estas tecnologias se tornam companheiras ideais para o desenvolvimento de aplicações que exploram o consumo e a geração de informações por diversas classes de clientes.

Para a manipulação de documentos XML é necessário a existência de um analisador (*parser*) que suporte todas as funcionalidades necessárias para percorrer as estruturas destes documentos, permitindo o acesso aos seus elementos e atributos.

O conjunto de APIs Java para processamento XML inclui três tecnologias que são a base para a manipulação de documentos XML. A primeira delas implementada o modelo de processamento baseado na interface SAX (*Simple API for XML*). A interface SAX trabalha lendo o documento XML por partes, enquanto aciona um método para cada elemento encontrado. A segunda tecnologia utiliza a interface DOM (*Document Object Model*). Neste caso, a leitura do documento XML é feita de uma só vez para a criação de uma representação na memória em forma de árvore. A terceira tecnologia, XSLT (*Extensible Stylesheet Language for Transformations*) é uma linguagem para a transformação de documentos XML em outros documentos XML ou qualquer outro formato textual.

### **4.1.1 A Biblioteca JDOM**

A biblioteca JDOM é uma API Java, baseada em árvore, de código fonte aberto para transformação, criação, manipulação de documentos XML. JDOM foi inventado por Brett McLaughlin e Jason Hunter em 2000 quando ambos encontravam-se frustrados com a forma complexa, não intuitiva e trabalhosa de manipulação de documentos XML em Java.

A utilização do JDOM é mais amigável do que SAX e DOM para a maioria das tarefas. JDOM é mais abrangente do que SAX, que não oferece uma forma padrão de escrever um documento XML, e do que o DOM, que pode manipular documentos XML mas não permite análise gramatical. Ele possui ainda a convivência de uma árvore baseada na API DOM e a familiaridade de convenções do padrão Java para SAX. Entretanto, JDOM não é SAX e não é DOM.

JDOM usa classes ao invés de interfaces, isto significa que você pode criar instâncias da maioria dos tipos de nó apenas passando um argumento ou dois a um construtor.

A API JDOM possui, entretanto, algumas restrições. JDOM não pode manipular documentos cujo tamanho exceda a memória disponível. Ele apresenta apenas um modelo lógico de um documento XML, não disponibilizando detalhes do modelo físico.

## **5. GERADOR DE APRESENTAÇÕES**

Para facilitar a manipulação do sistema gerador de apresentações pelo usuário, o sistema possui uma interface gráfica muito simples e agradável, para que os usuários não tenham nenhuma dificuldade na criação de suas apresentações. Irei descrever passo a passo as funcionalidades do sistema, podendo assim, demonstrar a facilidade na manipulação do sistema gerador de apresentações.

### **5.1 Estrutura**

Na geração dos slides o aplicativo utiliza arquivo com formatação HTML e XML.

É no arquivo HTML que será feita as apresentações, por ser uma linguagem multi-plataforma, essas apresentações possuem uma imensa portabilidade, pois não ficam limitados a nenhum sistema operacional, logo o usuário pode criar um slide em um computador com um sistema operacional Linux e o mesmo pode ser apresentado em um computador com um sistema operacional Windows.

Nos arquivos XML ficam armazenados as informações passadas pelo usuário na criação do slide, como os textos e as imagens inseridas.

O armazenamento em arquivo XML possibilita que as informações possam ser manipuladas no futuro por outros aplicativos.

Ao gerar os slides, o aplicativo gera automaticamente um índice para a apresentação, contendo um link para cada slide gerado, sendo este índice o slide inicial, como mostra a Figura 2.

Ao abrir um slide, o sistema automaticamente identifica qual o tipo de slide que está sendo aberto e o carrega, assim mostrando a janela específica do tipo escolhido.

O processo de transformação está ilustrado na Figura 3.

ÍNDICE DEMONSTRAÇÃO MONOGRAFIA	
- INTRODUÇÃO	
- OBJETIVO	
- APÊNDICE	
- APÊNDICE	
- APÊNDICE	
- ESTRUTURA	
- FUNCIONAMENTO	
- JAVA	
- XML	
- HTML	

Figura 2 - Índice gerado pelo sistema

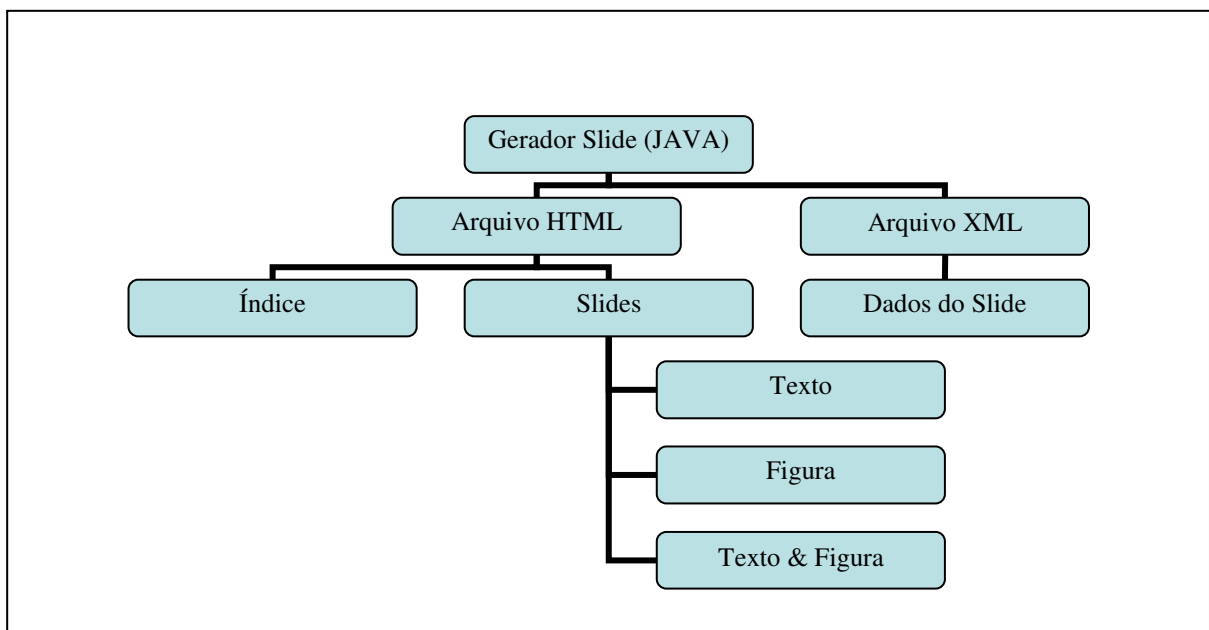


Figura 3 - Processo de transformação

### 5.1.1 Código HTML gerado.

A Figura 4 mostra o código HTML do índice gerado automaticamente pelo sistema.

```
<html>
<HEAD>

<style type=text/css>

.ativolinki {font-family: verdana; font-size : 9pt; margin-
left: 2pt; color: #003366; text-decoration:NONE}
.ativolinki:hover {font-family: verdana; font-size : 9pt;
margin-left: 2pt; color: #CC0000; text-decoration:NONE}

</STYLE>
<title>Gerador de Slide</title>
</head>
<body topmargin='0' leftmargin='0' cellpadding='0'
cellspacing='0' bgcolor='#E1E1E1'>
<table border='1' align='center' cellpadding='0'
bgcolor='#6262FF' cellspacing='0' style='border-width:0;
border-collapse: collapse' bordercolor='#111111' width='780'
id='AutoNumber1'>
<tr>
<td width='100%' style='border-style: none; border-width:
medium' bgcolor='#CBD5D9'>

<p align='center'>

<br><b>
<font color='#0072A8' face='Arial'
size='4'>ÍNDICE<BR>DEMONSTRAÇÃO MONOGRAFIA</font>
<font face='Arial'><br>&nbsp;   </font>
</b>
</td>
</tr>
```



```

<tr>
<td width='100%' style='border-style: none; border-width:
medium' bgcolor='#EBEFF0' height='500' valign='top'>
<BR>
<br><a href=slide1.htm class=ativolinki>- INTRODUÇÃO</a>
<br><a href=slide2.htm class=ativolinki>- OBJETIVO</a>
<br><a href=slide3.htm class=ativolinki>- APÊNDICE |</a>
<br><a href=slide3.htm class=ativolinki>- APÊNDICE ||</a>
<br><a href=slide3.htm class=ativolinki>- APÊNDICE ||</a>
<br><a href=slide3.htm class=ativolinki>- ESTRUTURA</a>
<br><a href=slide3.htm class=ativolinki>- FUNCIONAMENTO</a>
<br><a href=slide3.htm class=ativolinki>- JAVA</a>
<br><a href=slide3.htm class=ativolinki>- XML</a>
<br><a href=slide3.htm class=ativolinki>- HTML</a>
<br>
</td>
</tr>
</table>
</body>
</html>

```

**Figura 4 – Código HTML do índice**

### 5.1.2 Código XML gerado

A Figura 5 mostra o código XML de um slide do tipo Texto & Figura gerado automaticamente pelo sistema.

```

<?xml version='1.0' encoding='UTF-8'?>
<slide nome='GeradorSlide'>
<sld>
<tipo>TextosFiguras</tipo>
<nome>slide1.htm</nome>
<titulo>DEMONSTRAÇÃO SLIDE TEXTO COM FIGURA</titulo>

```

```
<texto1>DEMONSTRAÇÃO</texto1>
<texto2>DEMONSTRAÇÃO</texto2>
<path_img1>DEMONSTRACAO.jpg</path_img1>
<path_img2>DEMONSTRACAO.gif</path_img2>
<contador>1</contador>
</sld>
</slide>
```

**Figura 5 - Código XML slide Texto & Figura**

## 5.2 O funcionamento do Sistema

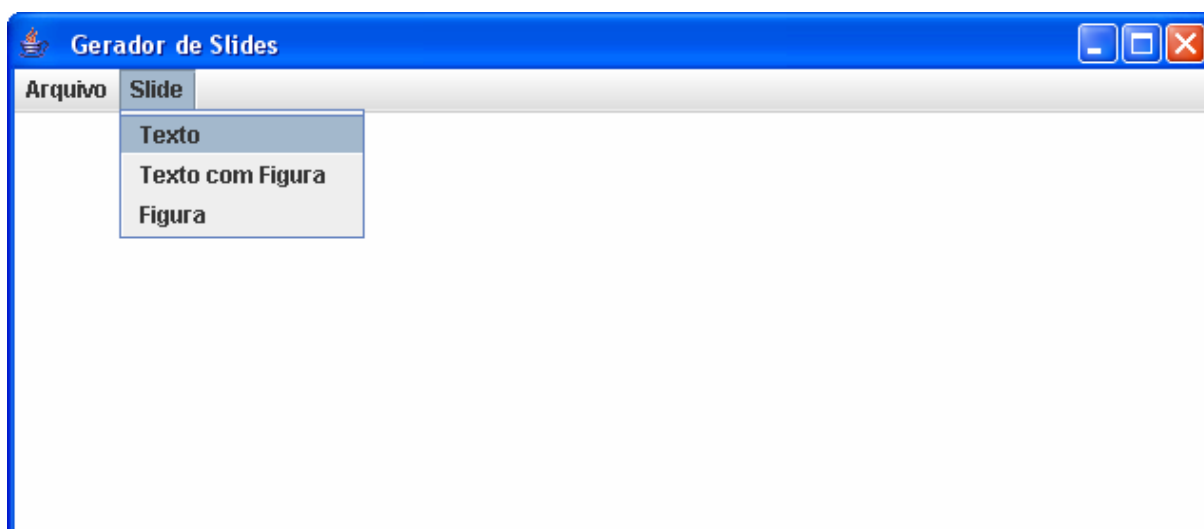
O sistema gerador de apresentações está dividido em três partes, que são os tipos de slides que poderão ser gerados, estes tipos são:

- **Slide Texto** : O slide de Texto, é caracterizado por conter o título e o corpo, onde o usuário poderá configurar o estilo e tamanho da fonte e a cor de fundo do slide.
- **Slide Figura** : O slide de Figura, é caracterizado por conter o título uma figura e a descrição desta figura.
- **Slide Texto & Figura** : O slide de Texto & Figura, é caracterizado por conter duas figuras e dois corpos de notícias.

O funcionamento do sistema é baseado em uma interface GUI, interface gráfica com o usuário. Através desta interface gráfica o usuário pode interagir com todas as funcionalidades do sistema, na qual o usuário fornece as informações sobre como serão feitas as apresentações.

### 5.3 Exemplo do Sistema Gerador de Apresentações

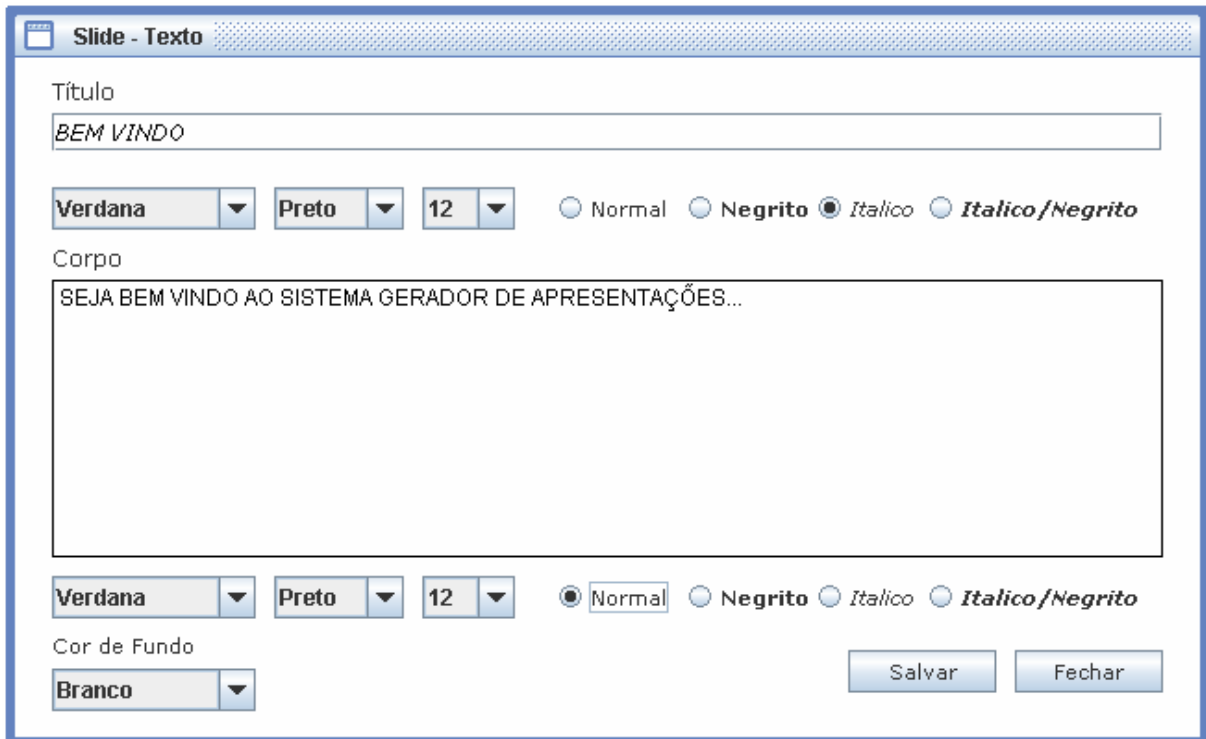
A tela inicial do sistema é onde o usuário irá escolher o tipo de slide que deseja, tipos esses citados anteriormente. Com esses tipos disponíveis para o usuário fica muito mais simples a interação usuário sistema, pois os modelos mais comuns na criação de uma apresentação são os que estão disponíveis no Sistema Gerador de Apresentações. A tela inicial do Sistema Gerador de Apresentação possui um layout muito agradável e de fácil interação como mostra a Figura 6.



**Figura 6 – Tela inicial do Sistema**

### 5.4 Criando um Slide Texto

Ao clicar em slide o usuário tem a opção dos slides desejados, ao escolher a opção Texto abrirá uma janela onde será confeccionado o slide de texto, como mostra a Figura 7.

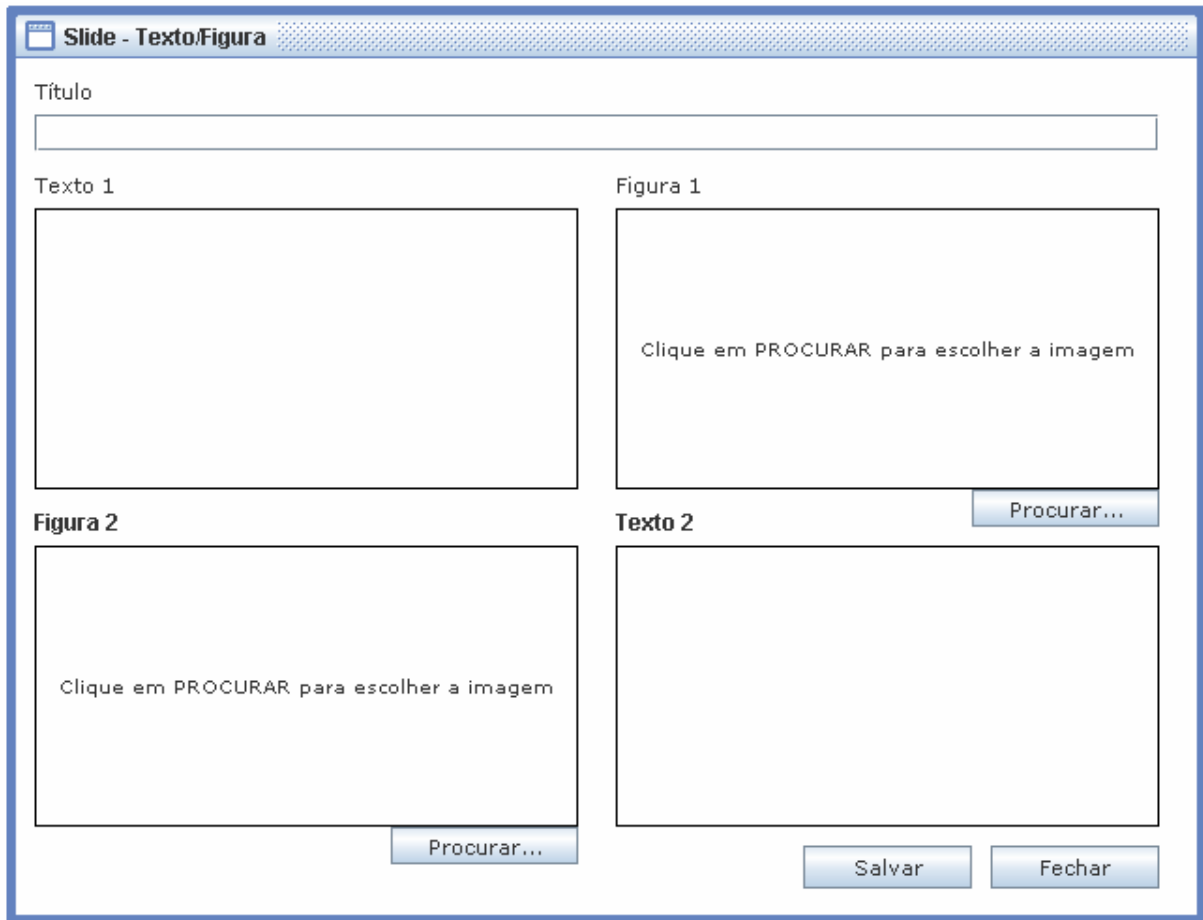


**Figura 7 – Janela para criação de Slide Texto**

Ao aparecer a janela de criação do slide do tipo texto, o usuário poderá configurar várias opções do seu slide, proporcionando assim a oportunidade do usuário criar o slide com as suas características desejadas, não limitando aos modelos já definidos. Logo após ter escolhido todas as características do slide, basta clicar em salvar que automaticamente o sistema irá gerar o slide e arquivo de índice.

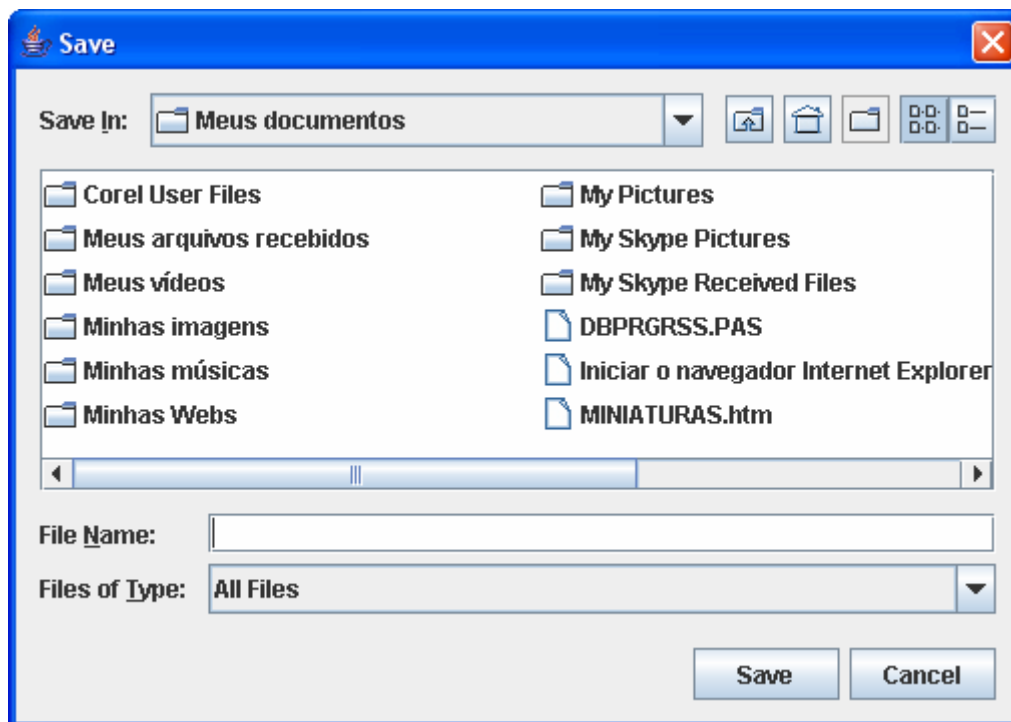
### **5.5 Criando um Slide Texto com Figura**

Após escolher a opção Texto com Figura abrirá uma outra janela para a confecção da mesma, nesta janela o usuário terá as opções de criar um slide com dois tipos de figuras, e com textos ao lado das figuras, como ilustrado na Figura 8.



**Figura 8 – Janela para criação de Slide Texto com Figuras**

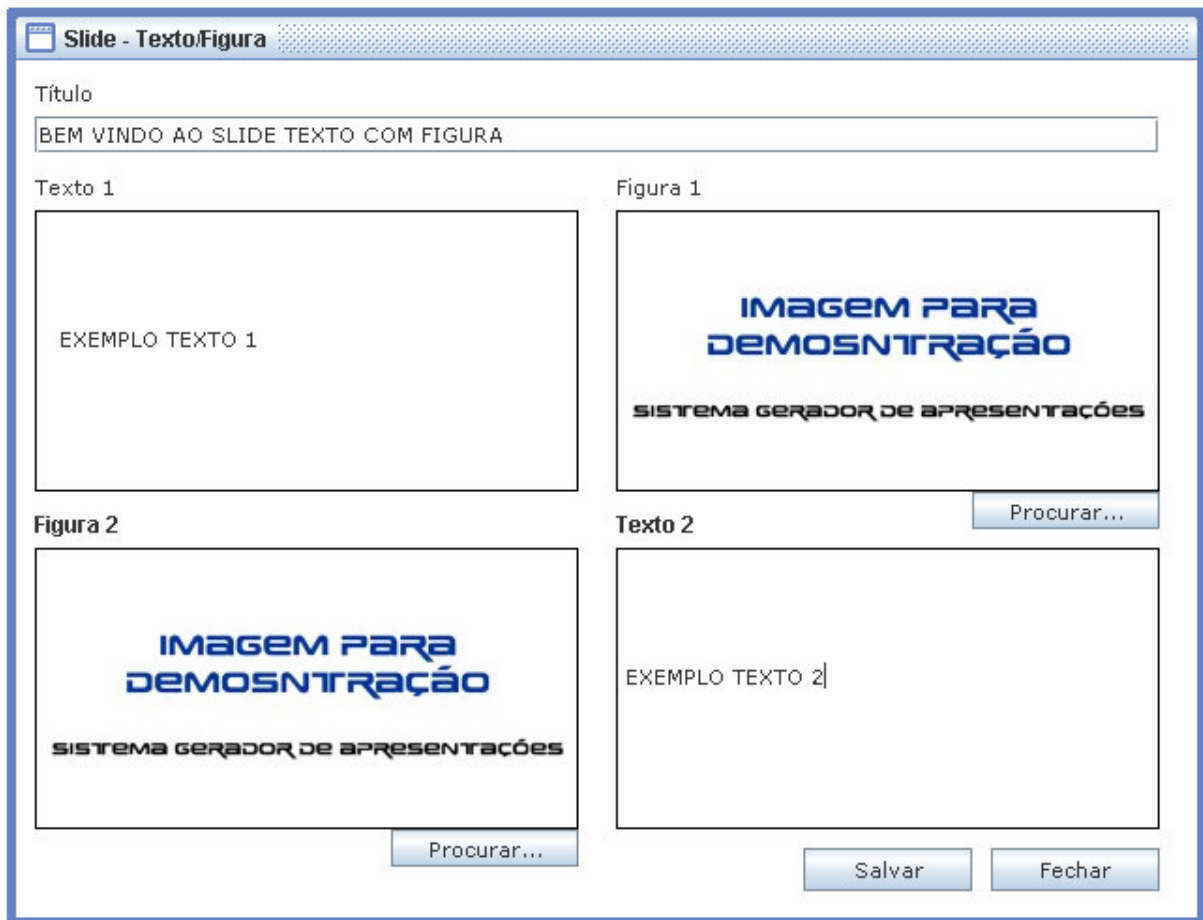
Para a criação do slide Texto com Figuras o usuário tem uma interface muito simples, onde ele escreve seus dois textos e seleciona a figura clicando no botão procurar, abrindo uma janela para a procura da imagem, como mostrado na Figura 9.



**Figura 9 – Janela procura de arquivos**

Com esta opção de procura de arquivo o sistema fica muito prático, no momento da criação do slide o usuário busca a figura e insere o texto, podendo assim visualiza-la no momento da criação como mostra a Figura 10.

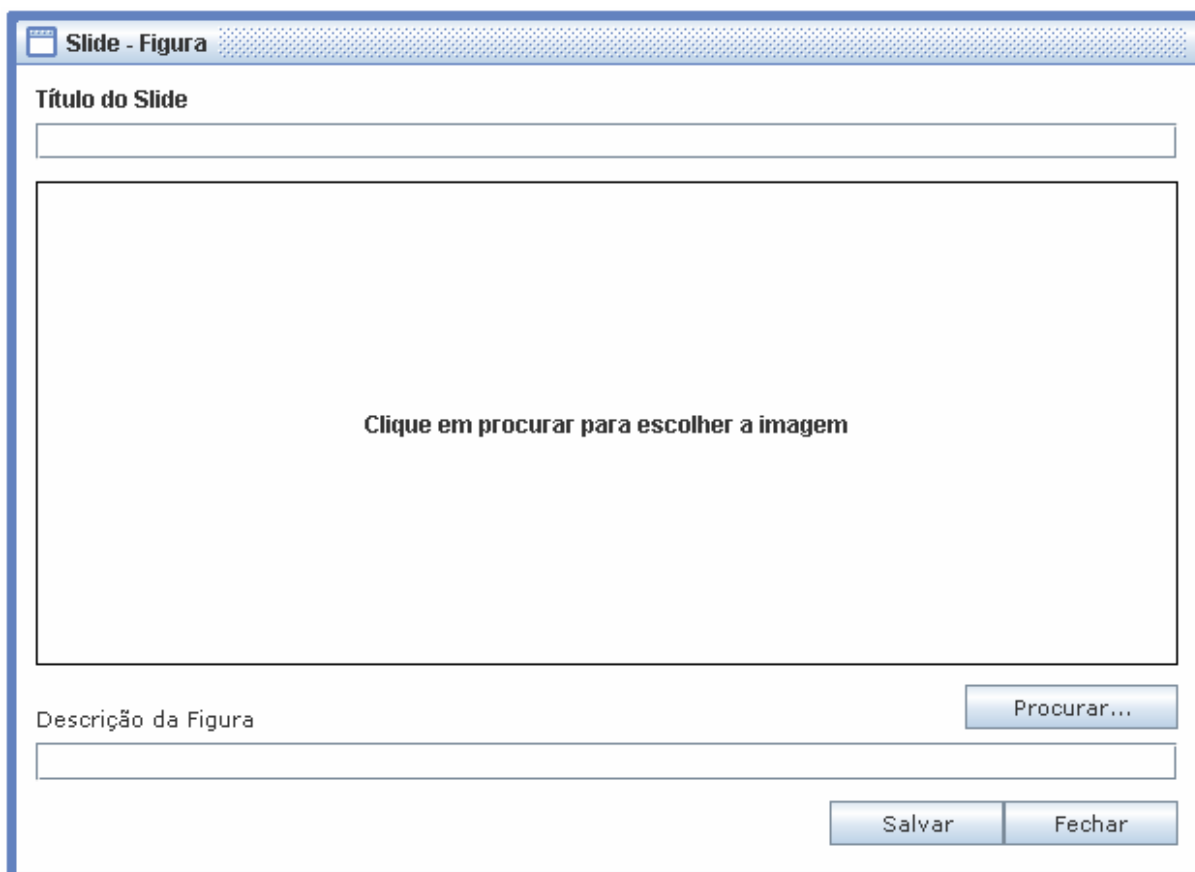
Esta característica de visualização na criação do slide Texto com Figura, torna o sistema muito agradável e proporciona ao usuário uma interação mais empolgante, pois ele pode abusar de sua criatividade.



**Figura 10 – Visualização na criação do slide**

## **5.6 Criando um Slide de Figura**

Após escolher a opção Figura, a janela de confecção do slide será aberta e o usuário terá as opções para a criação. O slide de figuras é muito simples, onde o usuário só tem a opção de escolher a figura e escrever uma breve descrição da mesma, como mostra a Figura 11.



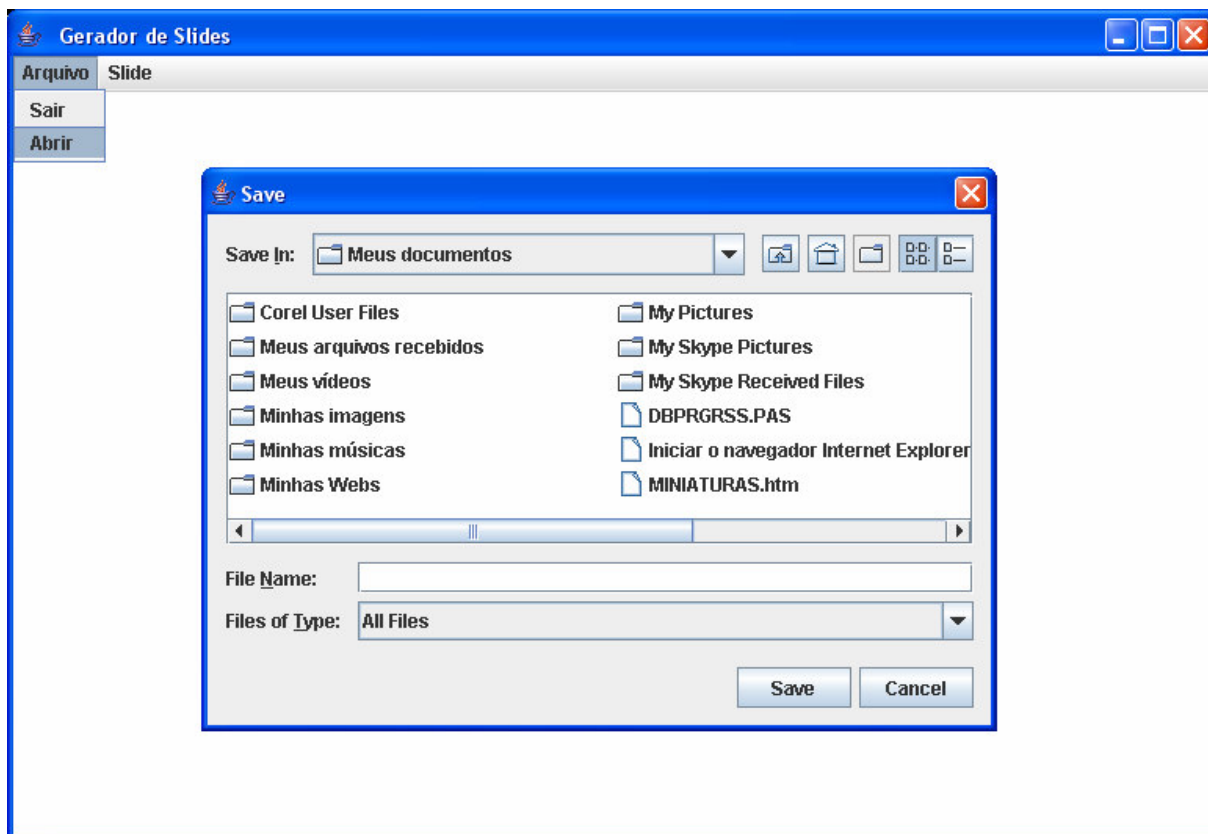
**Figura 11 – Janela para criação do Slide Figura**

O slide de figura possui as mesmas características do slide Figura & Texto, como busca de imagem, visualização em tempo de criação. É um modelo de slide muito simples pois muito usado, principalmente para amostras de planilhas e ou imagens de tamanhos grandes.

### **5.7 Abrindo um Slide**

O Sistema Gerador de Apresentações, possui uma opção para que o usuário possa abrir um slide gerado. Esta opção serve para que o usuário possa fazer futuras alterações em seus slides. Após aberto, o slide criado anteriormente pode ser alterado e salvo com as novas características, como mostra a figura 12.





**Figura 12 – Abrindo um Slide**

## **5.8 Modelo Slide Texto Gerado**

A Figura 13 mostra um slide do tipo Texto gerado pelo sistema, após as configurações passadas pelo usuário.

## O FUNCIONAMENTO DO SISTEMA

O sistema gerador de apresentações está dividido em três partes, que são os tipos de slides que poderão ser gerados, estes tipos são:

- **Slide Texto** : O slide de Texto, é caracterizado por conter o título e o corpo, onde o usuário poderá configurar o estilo e tamanho da fonte e a cor de fundo do slide.

- **Slide Figura** : O slide de Figura, é caracterizado por conter o título uma figura e a descrição desta figura.

- **Slide Texto & Figura** : O slide de Texto & Figura, é caracterizado por conter duas figuras e dois corpos de notícias.


< Indice> < Anterior> < Proxima>

**Figura 13 – Slide tipo Texto gerado pelo sistema**

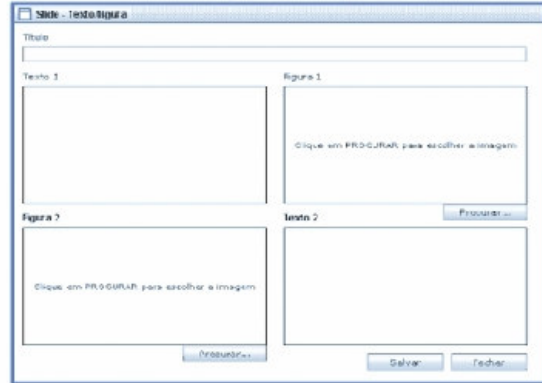
### 5.9 Modelo Slide Texto & Figura Gerado

A Figura 14 mostra um slide do tipo Texto & Figura gerado pelo sistema, após as configurações passadas pelo usuário.

No slide do tipo TEXTO, o usuário poderá configurar várias opções do seu slide, proporcionando assim a oportunidade do usuário criar o slide com as suas características desejadas, não limitando aos modelos já definidos.



Para a criação do slide Texto com Figuras o usuário tem uma interface muito simples, onde ele escreve seus dois textos e seleciona a figura clicando no botão procurar, abrindo uma janela para a procura da imagem.



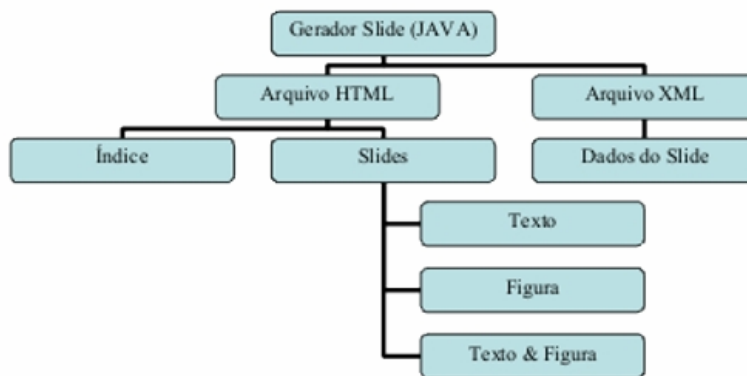
< Índice > < Anterior > < Próxima >

**Figura 14 – Slide tipo Texto & Figura gerado pelo sistema**

### 5.10 Modelo Slide Figura Gerado

A Figura 15 mostra um slide do tipo Figura gerado pelo sistema, após as configurações passadas pelo usuário.

## PROCESSO DE TRANSFORMAÇÃO



**Figura 15 – Slide tipo Figura gerado pelo sistema**

## **6. CONCLUSÕES**

Após todos os estudos e experimentos realizados através do desenvolvimento do aplicativo, pode-se concluir que é de grande valia uma ferramenta para geração de slides Multi-Plataforma. Dessa forma, fica caracterizada uma portabilidade que seria ideal a todo tipo de aplicativo, tornando-o compatível com todas as plataformas operacionais e facilitando aos usuários finais o aprendizado de uma única ferramenta para cada determinada finalidade independente do Sistema Operacional.

### **6.1 Aprendizado**

Com a criação desta ferramenta foi necessário serem feitos estudos sobre Java, XML, HTML, e desses estudos foi concluído que a orientação a objetos e a interface gráfica é muito utilizado pela maioria dos sistemas criados pelos desenvolvedores de software de hoje em dia, e Java é uma tecnologia que possibilita aos desenvolvedores uma grande facilidade para esse tipo de programação, uma vez que Java é completamente orientada a objetos e possui várias APIs para o desenvolvimento da interface gráfica.

Sobre XML, pode-se ver como uma linguagem de marcação estendida facilita o intercambio de informações e dados, servindo como linguagem intermediária para transferência de dados de uma linguagem para outra.

### **6.2 Trabalhos futuros**

Após todo o desenvolvimento do projeto ficam destacados alguns trabalhos para o futuro, sendo eles:

- Importação e Exportação de arquivos com formato PPT (arquivos do PowerPoint).
- Inserção de modelos de *Background*, planos de fundo para criação de diferentes modelos de *slides*.
- Inserção de multimídia, animações e sons para os *slides* criados.
- Redimensionamento de imagens, redimensionar as figuras no momento da criação dos *slides*.
- Formatação para impressão, formatar os *slides* criados para que sejam impressos.

## 7. BIBLIOGRAFIA

ALMEIDA, M. **Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares**. Minas Gerais: UFMG, 2002.

CORNELL, G. ; HORSTMANN, C. C. **Core Java 2, Volume II: Advanced Features**. Prentice Hall: BK&Cd Rom edition, 1999.

DOM W3C – Word Wide Web Consortium. Disponível em: <http://www.w3.org/DOM/>. Acessado em: 09/08/2005.

DEITEL, J.A. ; DEITEL, P.J. **Java Como Programar**. Porto Alegre: Makron Books, 1998.

RAMALHO, J.A. **HTML Avançado**. São Paulo: Makron Books, 1997.

RAMALHO, J.A. **HTML Referência Completa**. Rio de Janeiro: Compus, 1998.

SUN, **The Java Tutorial: A practical guide for programmers**. Disponível em: <http://java.sun.com/docs/books/tutorial/>. Acessado em: 17/10/2005.

## APÊNDICE A

### Principal.java

```
/*
----- - FUNDAÇÃO DE ENSINO EURÍPIDES SOARES DA ROCHA - UNIVEM - -----
----- -- GERADOR DE SLIDES -- -----
----- - TRABALHO DE CONCLUSÃO DE CURSO 2005 - -----
----- - ALUNO: CARLOS EDUARDO SANVIDO - -----
----- - PROF. ORIENTADOR: JOSE REMO FERREIRA BREGA - -----
*/

package geradorslide;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import javax.swing.JOptionPane;
import javax.swing.event.*;

/*

AUTOR : CARLOS EDUARDO SANVIDO

*/

public class Principal extends javax.swing.JFrame {

    public int contador=0;
    public int contadorAlteracao = 0; // CONTADOR PARA QDO O SLIDE SER ALTERADO RECEBE
O CONTADOR ANTIGO

    private String titulo, corpo, indice, indice1, nomeArqTexto, nomeArqIndice;
    private String aTitulo, aIndice, fontTituloTexto, tamanhoFontTituloTexto;
    private String comboFont[]={ "Verdana", "Times New Roman", "Arial" };
    private String comboCores[]={ "Branco", "Azul", "Amarelo", "Preto", "Vermelho", "Verde" };
    private String comboTamanhoFont[]={ "12", "14", "18", "22", "26", "32", "40", "48", "72" };
    private Font italicoNegritoTituloTexto, negritoTituloTexto, italicoTituloTexto, normalTituloTexto;

    public String caminhoFigura, caminhoFiguraTF1, caminhoFiguraTF2, nomeProjeto;

    FiguraTexto figuraTexto = new FiguraTexto();
    Indice chamaIndice = new Indice();
    FigurasTextos figurasTextos = new FigurasTextos();
    Ultimo ultimoSlide = new Ultimo();
    GeraXml geraXml = new GeraXml();
    LeituraXML leXml = new LeituraXML();

    public String nomeSlideAlterar; // Recebe o nome do slide a ser alterado.
    public String acao = "Inserir"; // Para saber se a Acao eh de gravacao ou alteracao.

    // CRIA FORMULARIO PRINCIPAL
    public Principal() {
        initComponents();
        java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        setBounds(0, 0,(screenSize.width), (screenSize.height));
    }
}
```



```

        indice1=""; // inicializa com nda

        nomeProjeto = JOptionPane.showInputDialog("Entre com o nome do PROJETO");
    }

//-----
//----- CONFIGURAS AS TELAS DO SISTEMA -----
//-----

private void initComponents() {
    buttonGroup1 = new javax.swing.ButtonGroup();
    jPanel1 = new javax.swing.JPanel();
    frameFigura = new javax.swing.JFrame();
    btProcuraFigura = new javax.swing.JButton();
    lbFigura = new javax.swing.JTextField();
    jLabel4 = new javax.swing.JLabel();
    btSalvarFigura = new javax.swing.JButton();
    btFecharFigura = new javax.swing.JButton();
    jLabel5 = new javax.swing.JLabel();
    lbTituloFigura = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    jInternalFrame2 = new javax.swing.JFrame();
    jTextField2 = new javax.swing.JTextField();
    textAreaTF1 = new javax.swing.JTextArea();
    figura1 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    figura2 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    textAreaTF2 = new javax.swing.JTextArea();
    ProcurarTF1 = new javax.swing.JButton();
    ProcurarTF2 = new javax.swing.JButton();
    btSalvarTF = new javax.swing.JButton();
    btFecharTF = new javax.swing.JButton();
    jLabel13 = new javax.swing.JLabel();
    jInternalFrame1 = new javax.swing.JFrame();
    jTextField1 = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jComboBox2 = new javax.swing.JComboBox();
    jComboBox3 = new javax.swing.JComboBox();
    jRadioButton2 = new javax.swing.JRadioButton();
    jRadioButton3 = new javax.swing.JRadioButton();
    jRadioButton4 = new javax.swing.JRadioButton();
    jComboBox1 = new javax.swing.JComboBox();
    jTextArea1 = new javax.swing.JEditorPane();
    jComboBox4 = new javax.swing.JComboBox();
    jComboBox5 = new javax.swing.JComboBox();
    jComboBox6 = new javax.swing.JComboBox();
    jComboBox7 = new javax.swing.JComboBox();
    jLabel3 = new javax.swing.JLabel();
    jRadioButton1 = new javax.swing.JRadioButton();
    jRadioButton5 = new javax.swing.JRadioButton();
    jRadioButton6 = new javax.swing.JRadioButton();
    jRadioButton7 = new javax.swing.JRadioButton();
    jRadioButton8 = new javax.swing.JRadioButton();
    jMenuBar1 = new javax.swing.JMenuBar();

```

```

jMenu2 = new javax.swing.JMenu();
jMenuItem4 = new javax.swing.JMenuItem();
jMenuItem5 = new javax.swing.JMenuItem();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenuItem2 = new javax.swing.JMenuItem();
jMenuItem3 = new javax.swing.JMenuItem();

getContentPane().setLayout(null);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle(" Gerador de Slides");
setFont(new java.awt.Font("Verdana", 0, 10));
jPanel1.setLayout(null);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
frameFigura.getContentPane().setLayout(null);

frameFigura.setBackground(new java.awt.Color(255, 255, 255));
frameFigura.setTitle("Slide - Figura");
btProcuraFigura.setFont(new java.awt.Font("Verdana", 0, 11));
btProcuraFigura.setText("Procurar...");
btProcuraFigura.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btProcuraFiguraActionPerformed(evt);
    }
});

frameFigura.getContentPane().add(btProcuraFigura);
btProcuraFigura.setBounds(490, 320, 110, 23);

lbFigura.setFont(new java.awt.Font("Verdana", 0, 11));
frameFigura.getContentPane().add(lbFigura);
lbFigura.setBounds(10, 350, 590, 20);

jLabel4.setFont(new java.awt.Font("Verdana", 0, 11));
jLabel4.setText("Descri\u00e7\u00e3o da Figura");
frameFigura.getContentPane().add(jLabel4);
jLabel4.setBounds(10, 330, 180, 15);

btSalvarFigura.setFont(new java.awt.Font("Verdana", 0, 11));
btSalvarFigura.setText("Salvar");
btSalvarFigura.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btSalvarFiguraActionPerformed(evt);
    }
});

frameFigura.getContentPane().add(btSalvarFigura);
btSalvarFigura.setBounds(420, 380, 90, 23);

btFecharFigura.setFont(new java.awt.Font("Verdana", 0, 11));
btFecharFigura.setText("Fechar");
btFecharFigura.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btFecharFiguraActionPerformed(evt);
    }
});

frameFigura.getContentPane().add(btFecharFigura);

```

```

btFecharFigura.setBounds(510, 380, 90, 23);

jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel5.setText("Clique em procurar para escolher a imagem");
jLabel5.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0)));
jLabel5.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
frameFigura.getContentPane().add(jLabel5);
jLabel5.setBounds(10, 60, 590, 250);

frameFigura.getContentPane().add(lbTituloFigura);
lbTituloFigura.setBounds(10, 30, 590, 19);

jLabel6.setText("\u00edtulo do Slide");
frameFigura.getContentPane().add(jLabel6);
jLabel6.setBounds(10, 10, 190, 14);

jPanel1.add(frameFigura);
frameFigura.setBounds(10, 130, 620, 450);

jInternalFrame2.getContentPane().setLayout(null);

jInternalFrame2.setBackground(new java.awt.Color(255, 255, 255));
jInternalFrame2.setTitle("Slide - Texto/Figura");
jInternalFrame2.setVisible(true);
jTextField2.setFont(new java.awt.Font("Verdana", 0, 11));
jInternalFrame2.getContentPane().add(jTextField2);
jTextField2.setBounds(10, 30, 600, 20);

textAreaTF1.setFont(new java.awt.Font("Verdana", 0, 11));
textAreaTF1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0)));
jInternalFrame2.getContentPane().add(textAreaTF1);
textAreaTF1.setBounds(10, 80, 290, 150);

figura1.setFont(new java.awt.Font("Verdana", 0, 10));
figura1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
figura1.setText("Clique em PROCURAR para escolher a imagem");
figura1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0)));
jInternalFrame2.getContentPane().add(figura1);
figura1.setBounds(320, 80, 290, 150);

jLabel8.setFont(new java.awt.Font("Verdana", 0, 11));
jLabel8.setText("Texto 1");
jInternalFrame2.getContentPane().add(jLabel8);
jLabel8.setBounds(10, 60, 100, 15);

jLabel9.setFont(new java.awt.Font("Verdana", 0, 11));
jLabel9.setText("Figura 1");
jInternalFrame2.getContentPane().add(jLabel9);
jLabel9.setBounds(320, 60, 140, 15);

jLabel10.setText("Figura 2");
jInternalFrame2.getContentPane().add(jLabel10);
jLabel10.setBounds(10, 240, 140, 14);

figura2.setFont(new java.awt.Font("Verdana", 0, 10));
figura2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
figura2.setText("Clique em PROCURAR para escolher a imagem");
figura2.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0)));
jInternalFrame2.getContentPane().add(figura2);
figura2.setBounds(10, 260, 290, 150);

```

```

jLabel12.setText("Texto 2");
jInternalFrame2.getContentPane().add(jLabel12);
jLabel12.setBounds(320, 240, 100, 14);

textAreaTF2.setFont(new java.awt.Font("Verdana", 0, 11));
textAreaTF2.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0)));
jInternalFrame2.getContentPane().add(textAreaTF2);
textAreaTF2.setBounds(320, 260, 290, 150);

ProcurarTF1.setFont(new java.awt.Font("Verdana", 0, 11));
ProcurarTF1.setText("Procurar...");
ProcurarTF1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ProcurarTF1ActionPerformed(evt);
    }
});

jInternalFrame2.getContentPane().add(ProcurarTF1);
ProcurarTF1.setBounds(510, 230, 100, 20);

ProcurarTF2.setFont(new java.awt.Font("Verdana", 0, 11));
ProcurarTF2.setText("Procurar...");
ProcurarTF2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ProcurarTF2ActionPerformed(evt);
    }
});

jInternalFrame2.getContentPane().add(ProcurarTF2);
ProcurarTF2.setBounds(200, 410, 100, 20);

btSalvarTF.setFont(new java.awt.Font("Verdana", 0, 11));
btSalvarTF.setText("Salvar");
btSalvarTF.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btSalvarTFActionPerformed(evt);
    }
});

jInternalFrame2.getContentPane().add(btSalvarTF);
btSalvarTF.setBounds(420, 420, 90, 23);

btFecharTF.setFont(new java.awt.Font("Verdana", 0, 11));
btFecharTF.setText("Fechar");
btFecharTF.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btFecharTFActionPerformed(evt);
    }
});

jInternalFrame2.getContentPane().add(btFecharTF);
btFecharTF.setBounds(520, 420, 90, 23);

jLabel13.setFont(new java.awt.Font("Verdana", 0, 11));
jLabel13.setText("\u00edtulo");
jInternalFrame2.getContentPane().add(jLabel13);
jLabel13.setBounds(10, 10, 60, 15);

jPanel1.add(jInternalFrame2);

```

```

jInternalFrame2.setBounds(430, 20, 640, 490);

jInternalFrame1.getContentPane().setLayout(null);

jInternalFrame1.setBackground(new java.awt.Color(255, 255, 255));
jInternalFrame1.setTitle(" Slide - Texto");
jTextField1.setFont(new java.awt.Font("Verdana", 0, 12));

jInternalFrame1.getContentPane().add(jTextField1);
jTextField1.setBounds(20, 30, 600, 21);

jLabel1.setFont(new java.awt.Font("Verdana", 0, 12));
jLabel1.setText("T\u00edtulo");
jInternalFrame1.getContentPane().add(jLabel1);
jLabel1.setBounds(20, 10, 80, 16);

jLabel2.setFont(new java.awt.Font("Verdana", 0, 12));
jLabel2.setText("Corpo");
jInternalFrame1.getContentPane().add(jLabel2);
jLabel2.setBounds(20, 100, 50, 16);

jButton1.setFont(new java.awt.Font("Verdana", 0, 11));
jButton1.setText("Salvar");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jButton1);
jButton1.setBounds(450, 320, 80, 23);

jButton2.setFont(new java.awt.Font("Verdana", 0, 11));
jButton2.setText("Fechar");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jButton2);
jButton2.setBounds(540, 320, 80, 23);

jComboBox2.setFont(new java.awt.Font("Verdana", 0, 11));
jComboBox2 = new JComboBox(comboCores);
jComboBox2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox2ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jComboBox2);
jComboBox2.setBounds(140, 70, 70, 23);

jComboBox3.setFont(new java.awt.Font("Verdana", 0, 11));
jComboBox3.setMaximumRowCount(5);
jComboBox3 = new JComboBox(comboTamanhoFont);
jComboBox3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox3ActionPerformed(evt);
    }
});

```

```

    }
});

jInternalFrame1.getContentPane().add(jComboBox3);
jComboBox3.setBounds(220, 70, 50, 23);

jRadioButton2.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton2.setFont(new java.awt.Font("Verdana", 1, 11));
jRadioButton2.setText("Negrito");
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton2ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton2);
jRadioButton2.setBounds(360, 70, 73, 23);

jRadioButton3.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton3.setFont(new java.awt.Font("Verdana", 2, 11));
jRadioButton3.setText("Italico");
jRadioButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton3ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton3);
jRadioButton3.setBounds(430, 70, 61, 23);

jRadioButton4.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton4.setFont(new java.awt.Font("Verdana", 3, 11));
jRadioButton4.setText("Italico/Negrito");
jRadioButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton4ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton4);
jRadioButton4.setBounds(490, 70, 160, 23);

jComboBox1.setFont(new java.awt.Font("Verdana", 0, 11));
jComboBox1 = new JComboBox(comboFont);
jComboBox1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox1ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jComboBox1);
jComboBox1.setBounds(20, 70, 110, 23);

jTextArea1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0)));
jInternalFrame1.getContentPane().add(jTextArea1);
jTextArea1.setBounds(20, 120, 600, 150);

jComboBox4.setFont(new java.awt.Font("Verdana", 0, 11));
jComboBox4 = new JComboBox(comboFont);
jInternalFrame1.getContentPane().add(jComboBox4);

```

```

jComboBox4.setBounds(20, 280, 110, 23);

jComboBox5.setFont(new java.awt.Font("Verdana", 0, 11));
jComboBox5 = new JComboBox(comboCores);
jComboBox5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox5ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jComboBox5);
jComboBox5.setBounds(140, 280, 70, 23);

jComboBox6.setFont(new java.awt.Font("Verdana", 0, 11));
jComboBox6.setMaximumRowCount(5);
jComboBox6 = new JComboBox(comboTamanhoFont);
jInternalFrame1.getContentPane().add(jComboBox6);
jComboBox6.setBounds(220, 280, 50, 23);

jComboBox7.setFont(new java.awt.Font("Verdana", 0, 11));
jComboBox7.setForeground(new java.awt.Color(0, 0, 255));
jComboBox7 = new JComboBox(comboCores);
jComboBox7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox7ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jComboBox7);
jComboBox7.setBounds(20, 330, 110, 23);

jLabel3.setFont(new java.awt.Font("Verdana", 0, 11));
jLabel3.setText("Cor de Fundo");
jInternalFrame1.getContentPane().add(jLabel3);
jLabel3.setBounds(20, 310, 80, 15);

jRadioButton1.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton1.setFont(new java.awt.Font("Verdana", 0, 11));
jRadioButton1.setText("Normal");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton1ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton1);
jRadioButton1.setBounds(290, 70, 67, 23);

jRadioButton5.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton5.setFont(new java.awt.Font("Verdana", 0, 11));
jRadioButton5.setText("Normal");
jRadioButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton5ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton5);
jRadioButton5.setBounds(290, 280, 67, 23);

```

```

jRadioButton6.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton6.setFont(new java.awt.Font("Verdana", 1, 11));
jRadioButton6.setText("Negrito");
jRadioButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton6ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton6);
jRadioButton6.setBounds(360, 280, 73, 23);

jRadioButton7.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton7.setFont(new java.awt.Font("Verdana", 2, 11));
jRadioButton7.setText("Italico");
jRadioButton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton7ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton7);
jRadioButton7.setBounds(430, 280, 61, 23);

jRadioButton8.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton8.setFont(new java.awt.Font("Verdana", 3, 11));
jRadioButton8.setText("Italico/Negrito");
jRadioButton8.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton8ActionPerformed(evt);
    }
});

jInternalFrame1.getContentPane().add(jRadioButton8);
jRadioButton8.setBounds(490, 280, 160, 23);

jPanel1.add(jInternalFrame1);
jInternalFrame1.setBounds(220, 180, 650, 400);

getContentPane().add(jPanel1);
jPanel1.setBounds(0, 0, 1380, 700);

jMenu2.setText("Arquivo");
jMenuItem4.setText("Sair");
jMenu2.add(jMenuItem4);

jMenuItem5.setText("Abrir");
jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem5ActionPerformed(evt);
    }
});

jMenu2.add(jMenuItem5);

jMenuBar1.add(jMenu2);

jMenu1.setText("Slide");

jMenuItem1.setText("Texto");

```



```

jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem1ActionPerformed(evt);
    }
});

jMenu1.add(jMenuItem1);

jMenuItem2.setText("Texto com Figura");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});

jMenu1.add(jMenuItem2);

jMenuItem3.setText("Figura");
jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem3ActionPerformed(evt);
    }
});

jMenu1.add(jMenuItem3);

jMenuBar1.add(jMenu1);

setJMenuBar(jMenuBar1);

pack();
}

//-----
//----- ABRE SLIDES EM XML -----
//-----

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {

    // CHAMA A CLASSE QUE ABRE A CAIXA DE BUSCA DE ARQUIVOS.
    FileDialogo f = new FileDialogo();
    String pathXML = f.aux; // RECEBE O CAMINHO DO ARQUIVO ESCOLHIDO

    leXml.abreXML(pathXML);

    String tipo = leXml.tipo;

    // ----- SLIDE TIPO TEXTO -----
    if (tipo.equals("Texto"))
    {
        // CASO UM DOS FRAMES ESTEJA ABERTO, SERAO FECHADOS PARA
        APARECER O FRAME ESPECIFICO.
        jInternalFrame2.setVisible(false);
        frameFigura.setVisible(false);

        // CASO O FRAME DO TEXTO ESTEJA FECHADO, ELE SERA ABERTO
        jInternalFrame1.setVisible(true);
        jTextField1.setText(leXml.titulo);
        jTextArea1.setText(leXml.corpo);
    }
}

```

```

// RECEBE O NOME DO SLIDE QUE SERA ALTERADO, PARA NAO SALVAR COMO UM
NOVO SLIDE E SIM SUBESCREVER.
    nomeSlideAlterar = leXml.nome;

// RECEBE O CONTADOR QUE PAROU NA CRIACAO DO SLIDE
contadorAlteracao = Integer.parseInt(leXml.contador);

// ACAO RECEBE ALTERA PARA QDO CLICAR EM SALVAR SABER QUE É
ALTERACAO.
    acao = "Alterar";
    }
    // ----- SLIDE TIPO FIGURA -----
    else if (tipo.equals("Figura"))
    {
        // CASO UM DOS FRAMES ESTEJA ABERTO, SERAO FECHADOS PARA
        APARECER O FRAME ESPECIFICO.
        jInternalFrame2.setVisible(false);
        jInternalFrame1.setVisible(false);

        // CASO O FRAME DO FIGURA ESTEJA FECHADO, ELE SERA ABERTO
        frameFigura.setVisible(true);
        lbTituloFigura.setText(leXml.titulo);
        lbFigura.setText(leXml.texto1);

        // RECEBE O PATH DA IMG PARA QUE SEJA MOSTRADA NA TELA
        caminhoFigura = leXml.pathFoto1;
        jLabel5.setText("");
        jLabel5.setIcon(new javax.swing.ImageIcon(caminhoFigura));

// RECEBE O NOME DO SLIDE QUE SERA ALTERADO, PARA NAO SALVAR COMO UM
NOVO SLIDE E SIM SUBESCREVER.
    nomeSlideAlterar = leXml.nome;

// RECEBE O CONTADOR QUE PAROU NA CRIACAO DO SLIDE
contadorAlteracao = Integer.parseInt(leXml.contador);

// ACAO RECEBE ALTERA PARA QDO CLICAR EM SALVAR SABER QUE É
ALTERACAO.
    acao = "Alterar";
    }
    // ----- SLIDE TIPO TEXTO/FIGURA -----
    else if (tipo.equals("TextosFiguras"))
    {
        // CASO UM DOS FRAMES ESTEJA ABERTO, SERAO FECHADOS PARA
        APARECER O FRAME ESPECIFICO.
        frameFigura.setVisible(false);
        jInternalFrame1.setVisible(false);

        // CASO O FRAME DO FIGURA ESTEJA FECHADO, ELE SERA ABERTO
        jInternalFrame2.setVisible(true);
        jTextField2.setText(leXml.titulo);
        textAreaTF1.setText(leXml.texto1);
        textAreaTF2.setText(leXml.texto2);

        // RECEBE O CAMINHO DA PRIMEIRA IMAGEM.
        caminhoFiguraTF1 = leXml.pathFoto1;
        figura1.setText("");
        figura1.setIcon(new javax.swing.ImageIcon(caminhoFiguraTF1));

        // RECEBE O CAMINHO DA SEGUNDA IMAGEM.

```

```

        caminhoFiguraTF2 = leXml.pathFoto2;
        figura2.setText("");
        figura2.setIcon(new javax.swing.ImageIcon(caminhoFiguraTF2));

        // RECEBE O NOME DO SLIDE QUE SERA ALTERADO, PARA NAO SALVAR COMO UM
        NOVO SLIDE E SIM SUBESCREVER.
        nomeSlideAlterar = leXml.nome;

        // RECEBE O CONTADOR QUE PAROU NA CRIACAO DO SLIDE
        contadorAlteracao = Integer.parseInt(leXml.contador);

        // ACAO RECEBE ALTERA PARA QDO CLICAR EM SALVAR SABER QUE É
ALTERACAO.
        acao = "Alterar";
    }

//-----
//----- TEXTO COM FIGURAS -----
//-----

// ABRE A FORMULARIO DE SLIDE DE TEXTO COM FIGURA...
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    jInternalFrame2.setVisible(true);
}

// FECHA O FORMULARIO DO SLIDE TEXTO COM FIGURA
private void btFecharTFActionPerformed(java.awt.event.ActionEvent evt) {
    jInternalFrame2.setVisible(false);
    acao = "Inserir";
}

// SALVA O SLIDE DE TEXTO COM FIGURA...
private void btSalvarTFActionPerformed(java.awt.event.ActionEvent evt) {

    String nome; // O NOME Q SERA PASSADO PARA O XML

    // CASO O TITULO DO SLIDE NAO FOR INFORMADO ABRIR UM AVISO PEDINDO PARA Q
    SEJA.
    if (jTextField2.getText().trim().length()==0)
        JOptionPane.showMessageDialog(null,"Informe o título do
        SLIDE","AVISO",JOptionPane.INFORMATION_MESSAGE);

    else
    {

        if (acao.equals("Alterar"))
        {
            nome = "slide"+contadorAlteracao; // PASSA O NOME PARA O XML

            figurasTextos.gravaFigurasTextos(jTextField2.getText(), caminhoFiguraTF1,
            textAreaTF1.getText(), caminhoFiguraTF2, textAreaTF2.getText(), contadorAlteracao);

            geraXml.gravaFigurasTextos(jTextField2.getText(), caminhoFiguraTF1,
            textAreaTF1.getText(), caminhoFiguraTF2, textAreaTF2.getText(), nome, contadorAlteracao);
        }
        else
        {
            contador++; // incrementa contador... fica tipo do slide + o contador
        }
    }
}

```

```

        figurasTextos.gravaFigurasTextos(jTextField2.getText(), caminhoFiguraTF1,
textAreaTF1.getText(), caminhoFiguraTF2, textAreaTF2.getText(), contador);
        ultimoSlide.gravaUltimo(contador);// CHAMA A CLASSE QUE GERA O
SLIDE DE FINALIZACAO

        nome = "slide"+contador; // PASSA O NOME PARA O XML
        String link = "slide"+contador+".htm";

        // CHAMA A CLASSE QUE GRAVA O INDICE.
        chamaIndice.gravaIndice(nomeProjeto , jTextField2.getText(), link);

        // CHAMA A CLASSE QUE GERA O XML.
        geraXml.gravaFigurasTextos(jTextField2.getText(), caminhoFiguraTF1,
textAreaTF1.getText(), caminhoFiguraTF2, textAreaTF2.getText(), nome, contador);

    }

    // LIMPA OS CAMPOS LOGO APOS EFETUAR A GRAVACAO.
    jTextField2.setText("");
    textAreaTF1.setText("");
    textAreaTF2.setText("");

    // LIMPA A FIGURA DEPOIS DE GRAVAR O SLIDE
    figura1.setText("Clique em procurar para escolher a imagem");
    figura1.setIcon(null);

    // LIMPA A SEGUNDA FIGURA DEPOIS DE GRAVA O SLIDE.
    figura2.setText("Clique em procurar para escolher a imagem");
    figura2.setIcon(null);
}

}

    // PROCURA A SEGUNDA FIGURA DO SLIDE TEXTO COM FIGURAS
private void ProcurarTF2ActionPerformed(java.awt.event.ActionEvent evt) {
    FileDialogo f = new FileDialogo();
    caminhoFiguraTF2 = f.aux;
    figura2.setText("");
    figura2.setIcon(new javax.swing.ImageIcon(caminhoFiguraTF2));
}

    // PROCURA A PRIMEIRA FIGURA DO SLIDE TEXTO COM FIGURAS
private void ProcurarTF1ActionPerformed(java.awt.event.ActionEvent evt) {
    FileDialogo f = new FileDialogo();
    caminhoFiguraTF1 = f.aux;
    figura1.setText("");
    figura1.setIcon(new javax.swing.ImageIcon(caminhoFiguraTF1));
}

//-----
//----- FIGURAS -----
//-----

// GRAVA SLIDE DE TEXTO
private void btSalvarFiguraActionPerformed(java.awt.event.ActionEvent evt) {

    String nome;

    if (lbTituloFigura.getText().trim().length()==0)

```

```

        JOptionPane.showMessageDialog(null,"Informe o título do
SLIDE","AVISO",JOptionPane.INFORMATION_MESSAGE);

        else
        {
            if (acao.equals("Alterar"))
                {
                    nome = "slide"+contadorAlteracao;

                    figuraTexto.gravaFiguraTexto(lbTituloFigura.getText(), caminhoFigura,
lbFigura.getText(), contadorAlteracao);
                    geraXml.gravaFigura(lbTituloFigura.getText(), caminhoFigura,
lbFigura.getText(), nome, contadorAlteracao);
                }
            else
            {
                contador++; // incrementa contador... fica tipo do slide + o contador

                figuraTexto.gravaFiguraTexto(lbTituloFigura.getText(), caminhoFigura, lbFigura.getText(),
contador);
                ultimoSlide.gravaUltimo(contador);

                nome = "slide"+contador;
                String link = "slide"+contador+".htm";
                chamaIndice.gravaIndice(nomeProjeto ,lbTituloFigura.getText(), link);

                // CHAMA A CLASSE QUE GERA O XML.
                geraXml.gravaFigura(lbTituloFigura.getText(), caminhoFigura,
lbFigura.getText(), nome, contador);
            }

            // LIMPA OS CAMPOS LOGO APOS A GRAVACAO.
            lbTituloFigura.setText("");
            lbFigura.setText("");

            jLabel5.setText("Clique em procurar para escolher a imagem");
            jLabel5.setIcon(null);
        }
    }

    // PROCURA A FIGURA DO SLIDE DE FIGURAS
private void btProcuraFiguraActionPerformed(java.awt.event.ActionEvent evt) {
    FileDialogo f = new FileDialogo();
    caminhoFigura = f.aux;
    jLabel5.setText("");
    jLabel5.setIcon(new javax.swing.ImageIcon(caminhoFigura));
}

    // FECHA A TELA DO SLIDE DE FIGURAS
private void btFecharFiguraActionPerformed(java.awt.event.ActionEvent evt) {
    frameFigura.setVisible(false);
    acao = "Insere";
}

    // ABRE A TELA DE SLIDE DE FIGURAS
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    jInternalFrame1.setVisible(false);
    jInternalFrame2.setVisible(false);
    frameFigura.setVisible(true);
}

```

```

//-----
//----- CONFIGURACOES DO SLIDE TEXTO -----
//-----

    // SELECIONA A COR DA FONTE DO SLIDE TEXTO
private void jComboBox5ActionPerformed(java.awt.event.ActionEvent evt) {
    corCorpoFonte();
}

    // SELECIONA A COR DE FUNDO DO SLIDE TEXTO
private void jComboBox7ActionPerformed(java.awt.event.ActionEvent evt) {
    corFundo();
}

private void jComboBox3ActionPerformed(java.awt.event.ActionEvent evt) {
    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

    i = jComboBox3.getSelectedIndex();
    tFont = Integer.parseInt(comboTamanhoFont[i]);

    if (jRadioButton4.isSelected())
    {
        italicoNegritoTituloTexto = new Font(aux, Font.ITALIC + Font.BOLD, tFont);
        jTextField1.setFont(italicoNegritoTituloTexto);
    }

    if (jRadioButton1.isSelected())
    {
        normalTituloTexto = new Font(aux, Font.PLAIN, tFont);
        jTextField1.setFont(normalTituloTexto);
    }

    if (jRadioButton2.isSelected())
    {
        negritoTituloTexto = new Font(aux, Font.BOLD, tFont);
        jTextField1.setFont(negritoTituloTexto);
    }

    if (jRadioButton3.isSelected())
    {
        italicoTituloTexto = new Font(aux, Font.ITALIC, tFont);
        jTextField1.setFont(italicoTituloTexto);
    }

    corTitulo();
}

private void jComboBox2ActionPerformed(java.awt.event.ActionEvent evt) {
    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

    i = jComboBox3.getSelectedIndex();
    tFont = Integer.parseInt(comboTamanhoFont[i]);
}

```

```

if (jRadioButton4.isSelected())
{
    italicoNegritoTituloTexto = new Font(aux, Font.ITALIC + Font.BOLD, tFont);
    jTextField1.setFont(italicoNegritoTituloTexto);
}

if (jRadioButton1.isSelected())
{
    normalTituloTexto = new Font(aux, Font.PLAIN, tFont);
    jTextField1.setFont(normalTituloTexto);
}

if (jRadioButton2.isSelected())
{
    negritoTituloTexto = new Font(aux, Font.BOLD, tFont);
    jTextField1.setFont(negritoTituloTexto);
}

if (jRadioButton3.isSelected())
{
    italicoTituloTexto = new Font(aux, Font.ITALIC, tFont);
    jTextField1.setFont(italicoTituloTexto);
}

corTitulo();
}

private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {

    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

    i = jComboBox3.getSelectedIndex();
    tFont = Integer.parseInt(comboTamanhoFont[i]);

    if (jRadioButton4.isSelected())
    {
        italicoNegritoTituloTexto = new Font(aux, Font.ITALIC + Font.BOLD, tFont);
        jTextField1.setFont(italicoNegritoTituloTexto);
    }

    if (jRadioButton1.isSelected())
    {
        normalTituloTexto = new Font(aux, Font.PLAIN, tFont);
        jTextField1.setFont(normalTituloTexto);
    }

    if (jRadioButton2.isSelected())
    {
        negritoTituloTexto = new Font(aux, Font.BOLD, tFont);
        jTextField1.setFont(negritoTituloTexto);
    }

    if (jRadioButton3.isSelected())
    {
        italicoTituloTexto = new Font(aux, Font.ITALIC, tFont);
        jTextField1.setFont(italicoTituloTexto);
    }
}

```

```

    corTitulo();
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    jButton2.setSelected(false);
    jButton1.setSelected(false);
    jButton3.setSelected(false);
    jButton4.setSelected(true);

    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

    i = jComboBox3.getSelectedIndex();
    tFont = Integer.parseInt(comboTamanhoFont[i]);

    italicoNegritoTituloTexto = new Font(aux, Font.ITALIC + Font.BOLD, tFont);
    jTextField1.setFont(italicoNegritoTituloTexto);

    corTitulo();
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    jButton2.setSelected(false);
    jButton1.setSelected(false);
    jButton4.setSelected(false);
    jButton3.setSelected(true);

    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

    i = jComboBox3.getSelectedIndex();
    tFont = Integer.parseInt(comboTamanhoFont[i]);

    italicoTituloTexto = new Font(aux, Font.ITALIC, tFont);
    jTextField1.setFont(italicoTituloTexto);

    corTitulo();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    jButton2.setSelected(false);
    jButton3.setSelected(false);
    jButton4.setSelected(false);
    jButton1.setSelected(true);

    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

    i = jComboBox3.getSelectedIndex();
    tFont = Integer.parseInt(comboTamanhoFont[i]);

    normalTituloTexto = new Font(aux, Font.PLAIN, tFont);
    jTextField1.setFont(normalTituloTexto);
}

```



```

    corTitulo();
}

public void corTitulo()
{
    int i = jComboBox2.getSelectedIndex();
    String aux = comboCores[i];

    if (aux == "Vermelho")
    {
        jTextField1.setForeground(new java.awt.Color(204, 0, 0));
    }

    else if (aux == "Preto")
    {
        jTextField1.setForeground(new java.awt.Color(0, 0, 0));
    }
    else if (aux == "Azul")
    {
        jTextField1.setForeground(new java.awt.Color(0, 0, 255));
    }
    else if (aux == "Amarelo")
    {
        jTextField1.setForeground(new java.awt.Color(255, 204, 0));
    }
    else if (aux == "Verde")
    {
        jTextField1.setForeground(new java.awt.Color(0, 153, 0));
    }

    else if (aux == "Branco")
    {
        jTextField1.setForeground(new java.awt.Color(255, 255, 255));
    }
}

public void corCorpoFonte()
{
    int i = jComboBox5.getSelectedIndex();
    String aux = comboCores[i];

    if (aux == "Vermelho")
    {
        jTextArea1.setForeground(new java.awt.Color(204, 0, 0));
    }

    else if (aux == "Preto")
    {
        jTextArea1.setForeground(new java.awt.Color(0, 0, 0));
    }
    else if (aux == "Azul")
    {
        jTextArea1.setForeground(new java.awt.Color(0, 0, 255));
    }
    else if (aux == "Amarelo")
    {
        jTextArea1.setForeground(new java.awt.Color(255, 204, 0));
    }
}

```

```

else if (aux == "Verde")
{
    jTextArea1.setForeground(new java.awt.Color(0, 153, 0));
}

else if (aux == "Branco")
{
    jTextArea1.setForeground(new java.awt.Color(255, 255, 255));
}
}

public void corFundo()
{
    int i = jComboBox7.getSelectedIndex();
    String aux = comboCores[i];

    if (aux == "Vermelho")
    {
        jTextArea1.setBackground(new java.awt.Color(204, 0, 0));
        jTextField1.setBackground(new java.awt.Color(204, 0, 0));
    }

    else if (aux == "Preto")
    {
        jTextArea1.setBackground(new java.awt.Color(0, 0, 0));
        jTextField1.setBackground(new java.awt.Color(0, 0, 0));
    }
    else if (aux == "Azul")
    {
        jTextArea1.setBackground(new java.awt.Color(0, 0, 255));
        jTextField1.setBackground(new java.awt.Color(0, 0, 255));
    }
    else if (aux == "Amarelo")
    {
        jTextArea1.setBackground(new java.awt.Color(255, 204, 0));
        jTextField1.setBackground(new java.awt.Color(255, 204, 0));
    }
    else if (aux == "Verde")
    {
        jTextArea1.setBackground(new java.awt.Color(0, 153, 0));
        jTextField1.setBackground(new java.awt.Color(0, 153, 0));
    }

    else if (aux == "Branco")
    {
        jTextArea1.setBackground(new java.awt.Color(255, 255, 255));
        jTextField1.setBackground(new java.awt.Color(255, 255, 255));
    }
}

private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    jRadioButton1.setSelected(false);
    jRadioButton3.setSelected(false);
    jRadioButton4.setSelected(false);
    jRadioButton2.setSelected(true);

    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

```

```

        i = jComboBox3.getSelectedIndex();
        tFont = Integer.parseInt(comboTamanhoFont[i]);

        negritoTituloTexto = new Font(aux, Font.BOLD, tFont);
        jTextField1.setFont(negritoTituloTexto);

        corTitulo();
    }

    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
        jButton5.setSelected(false);
        jButton6.setSelected(false);
        jButton8.setSelected(false);
    }

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        jButton5.setSelected(false);
        jButton8.setSelected(false);
        jButton7.setSelected(false);
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        jButton8.setSelected(false);
        jButton6.setSelected(false);
        jButton7.setSelected(false);
    }

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
        jButton5.setSelected(false);
        jButton6.setSelected(false);
        jButton7.setSelected(false);
        italicoNegritoTituloTexto = new Font("Verdana", Font.BOLD + Font.ITALIC, 11);
        jTextField1.setFont(italicoNegritoTituloTexto);
        int i = 0;
        i = jComboBox7.getSelectedIndex();
        String teste = comboCores[i];
        jLabel1.setText(teste);

        i = jComboBox1.getSelectedIndex();
        teste = comboFont[i];
        jLabel2.setText(teste);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        if (jTextField1.getText().trim().length()==0)
            JOptionPane.showMessageDialog(null,"Informe o titulo do
SLIDE", "AVISO",JOptionPane.INFORMATION_MESSAGE);
        else
            gravaTexto();
    }
//-----

    private static FileWriter open;
    private static PrintWriter dado;

    private static void FechaArquivo()
    {
        dado.close(); //Fechando arquivo
    }

```

```

//-----
public void gravaTexto()
{
    titulo = jTextField1.getText();
    corpo = jTextArea1.getText();

    jTextField1.setText("");
    jTextArea1.setText("");

        if (acao.equals("Alterar"))
        {
            gravaArquivoTexto();

            // RECEBE O NOME DO ARQ XML QUE SERA ALTERADO TB.
            String nomeXmlAux = "slide"+contadorAlteracao;
            geraXml.texto(titulo, corpo, nomeXmlAux, contadorAlteracao);
            acao = "Inserir"; // limpo a variavel de acao
        }
        else
        {
            contador++;
            gravaArquivoTexto();

            String nome = "slide"+contador;
            String link = "slide"+contador+".htm";
            chamaIndice.gravaIndice(nomeProjeto , titulo, link);

            // CHAMA A CLASSE QUE GERA O XML.
            geraXml.texto(titulo, corpo, nome, contador);

            // CHAMA A CLASSE QUE GERA O ULTIMO SLIDE
            ultimoSlide.gravaUltimo(contador);
        }
    }
}

//-----
public void gravaArquivoTexto()
{
    int i = 0, tFont = 0;

    i = jComboBox1.getSelectedIndex();
    String aux = comboFont[i];

    i = jComboBox3.getSelectedIndex();
    tFont = Integer.parseInt(comboTamanhoFont[i]);

    Integer anterior;
    Integer proximo;

    // CASO A ACAO SEJA ALTERACAO.
    if (acao.equals("Alterar"))
    {
        anterior = contadorAlteracao - 1;
        proximo = contadorAlteracao + 1;
        nomeArqTexto = nomeSlideAlterar;
    }
    else
    {

```

```

        anterior = contador - 1;
                proximo = contador + 1;
                nomeArqTexto="slide"+contador+".htm"; // Nome do Arquivo
        }

        // -----
        //----- SELECIONA A COR DO TITULO -----
--
        //-----
        i = jComboBox2.getSelectedIndex();
String corTitulo = comboCores[i];

if (corTitulo == "Vermelho")
{
    corTitulo="#CC0000";
}

else if (corTitulo == "Preto")
{
    corTitulo="#000000";
}
else if (corTitulo == "Azul")
{
    corTitulo="#003366";
}
else if (corTitulo == "Amarelo")
{
    corTitulo="#E8E800";
}
else if (corTitulo == "Verde")
{
    corTitulo="#006600";
}

else if (corTitulo == "Branco")
{
    corTitulo="FFFFFF";
}

        // -----
        //----- SELECIONA A COR DO FUNDO -----
--
        //-----

        i = jComboBox7.getSelectedIndex();
String corFundo = comboCores[i];

if (corFundo == "Vermelho")
{
    corFundo="#CC0000";
}

else if (corFundo == "Preto")
{
    corFundo="#000000";
}
else if (corFundo == "Azul")
{
    corFundo="#003366";
}
}

```

```

else if (corFundo == "Amarelo")
{
    corFundo="#E8E800";
}
else if (corFundo == "Verde")
{
    corFundo="#006600";
}

else if (corFundo == "Branco")
{
    corFundo="#FFFFFF";
}

// -----
//----- CRIA ESTILOS NEG - NORMAL -----
//-----

String estiloAbre, estiloFecha;
estiloFecha="";
estiloAbre="";
if (jRadioButton1.isSelected())
{
    estiloAbre="";
    estiloFecha="";
}
else if (jRadioButton2.isSelected())
{
    estiloAbre="<b>";
    estiloFecha="</b>";
}
else if (jRadioButton3.isSelected())
{
    estiloAbre="<i>";
    estiloFecha="</i>";
}
else if (jRadioButton4.isSelected())
{
    estiloAbre="<b><i>";
    estiloFecha="</b></i>";
}

// -----
//----- FIM DAS CONFIGURACOES -----
//-----

```

```

aTitulo = "<html><HEAD><style type=text/css>" +
        ".ativolink { font-family: verdana; font-size : 9pt; margin-left: 2pt;
color: #003366; text-decoration:NONE}" +
        ".ativolink:hover { font-family: verdana; font-size : 9pt; margin-left:
2pt; color: #CC0000; text-decoration:NONE}" +
        "</STYLE><title>Gerador de Slide</title></head>";

aTitulo = aTitulo + "<body topmargin='0' leftmargin='0' cellpadding='0' cellspacing='0'
bgcolor=#E1E1E1'>";
aTitulo = aTitulo + "<table border='1' align='center' cellpadding='0' cellspacing='0'
style='border-width:0; border-collapse: collapse' bordercolor='#111111' width='780' id='AutoNumber1'>";

```

```

        aTitulo = aTitulo + "<tr><td width='100%' bgcolor='#CBD5D9' style='border-style: none;
border-width: medium'>";
        aTitulo = aTitulo + "<p align='right'><font face='Verdana' size='1'>PG.
"+contador+"</font></td></tr>";
        aTitulo = aTitulo + "<tr><td width='100%' bgcolor='#CBD5D9' style='border-style: none;
border-width: medium'>";

        aTitulo = aTitulo + "<p align='center'><font size='2' face='"+aux+"
color='"+corTitulo+"'">+estiloAbre+titulo+estiloFecha+"</font><br>&nbsp;</td></tr>";
        aTitulo = aTitulo + "<tr><td width='100%' bgcolor='"+corFundo+"'" style='border-style: none;
border-width: medium'><p align='justify'><br><font face='Verdana' size='2'>";
        aTitulo = aTitulo + corpo+"</font><br><br>&nbsp;</td></tr></table>";

        aTitulo = aTitulo + "<table border='1' align='center' cellpadding='0' bgcolor='#FFFFFF'
cellspacing='0' style='border-width:0; border-collapse: collapse' bordercolor='#111111' width='780'
id='AutoNumber1'>";
        aTitulo = aTitulo + "<tr><td bgcolor='#CBD5D9' width='100%' align='center' style='border-
style: none; border-width: medium'>";

        // CRIA O LINK QUE VOLTA PARA O INDICE
        aTitulo = aTitulo + "<br>&lt;<a href='index.htm' class='ativolinki'>Indice</a>&gt;";

        // SE ANTERIOR FOR MENOR QUE ZERO ENTAUM EH O PRIMEIRO SLIDE E NAUM
TEM ANTERIOR
        if (anterior > 0)
        {
            aTitulo = aTitulo + " &lt;<a href='slide"+anterior+".htm'
class='ativolinki'>Anterior</a>&gt;";
        }

        // CRIA O LINK QUE DIRECIONA PARA A PROXIMA PAGINA
        aTitulo = aTitulo + " &lt;<a href='slide'+proximo+".htm' class='ativolinki'>Proxima</a>&gt;";

        aTitulo = aTitulo + "<br>&nbsp;</font></td></tr></table></body></html>";

        try
        {
            open=new FileWriter(nomeArqTexto); //Abrindo Arquivo
            dado=new PrintWriter(open,true); //Buffer para gravar dados no arquivo

            dado.println(aTitulo);// Grava aki

            }catch(Exception e)
            {
                JOptionPane.showMessageDialog(null,"Erro ao abrir
arquivo","Erro",JOptionPane.ERROR_MESSAGE);
            }
        }

//-----

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    jInternalFrame1.setVisible(false);
    acao = "Altera";
}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    frameFigura.setVisible(false);
    jInternalFrame2.setVisible(false);
}

```

```

jInternalFrame1.setVisible(true);

        corFundo();
jTextArea1.setForeground(new java.awt.Color(0, 0, 0));
jTextField1.setForeground(new java.awt.Color(0, 0, 0));

jRadioButton5.setSelected(true);
jRadioButton1.setSelected(true);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Principal().setVisible(true);
        }
    });
}

private javax.swing.JButton ProcurarTF1;
private javax.swing.JButton ProcurarTF2;
private javax.swing.JButton btFecharFigura;
private javax.swing.JButton btFecharTF;
private javax.swing.JButton btProcuraFigura;
private javax.swing.JButton btSalvarFigura;
private javax.swing.JButton btSalvarTF;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JLabel figura1;
private javax.swing.JLabel figura2;
private javax.swing.JInternalFrame frameFigura;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JComboBox jComboBox3;
private javax.swing.JComboBox jComboBox4;
private javax.swing.JComboBox jComboBox5;
private javax.swing.JComboBox jComboBox6;
private javax.swing.JComboBox jComboBox7;
private javax.swing.JInternalFrame jInternalFrame1;
private javax.swing.JInternalFrame jInternalFrame2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JMenu jMenuItem1;
private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenuBar jMenuItemBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenuItem jMenuItem5;

```



```
private javax.swing.JPanel jPanel1;  
private javax.swing.JRadioButton jButton1;  
private javax.swing.JRadioButton jButton2;  
private javax.swing.JRadioButton jButton3;  
private javax.swing.JRadioButton jButton4;  
private javax.swing.JRadioButton jButton5;  
private javax.swing.JRadioButton jButton6;  
private javax.swing.JRadioButton jButton7;  
private javax.swing.JRadioButton jButton8;  
private javax.swing.JEditorPane jTextArea1;  
private javax.swing.JTextField jTextField1;  
private javax.swing.JTextField jTextField2;  
private javax.swing.JTextField lbFigura;  
private javax.swing.JTextField lbTituloFigura;  
private javax.swing.JTextArea textAreaTF1;  
private javax.swing.JTextArea textAreaTF2;
```

```
}
```

## APÊNDICE B

### GeraXml.java

```
package geradorslide;

import java.io.*;
import javax.swing.JOptionPane;

public class GeraXml {

    private static FileWriter open;
    private static PrintWriter dado;

    public String conteudo;
    public String nomeArq;

    /** Creates a new instance of FiguraTexto */
    public void gravaFigurasTextos(String titulo, String pathFoto1, String descFoto1, String pathFoto2, String
descFoto2, String nome, int cont) {

        nomeArq = nome+".xml";

        conteudo = "<?xml version='1.0' encoding='UTF-8'?>";
        conteudo = conteudo + "<slide nome='GeradorSlide'>";
        conteudo = conteudo + "<sld>";
        conteudo = conteudo + "<tipo>TextosFiguras</tipo>";
        conteudo = conteudo + "<nome>"+nome+".htm</nome>";
        conteudo = conteudo + "<titulo>"+titulo+"</titulo>";
        conteudo = conteudo + "<texto1>"+descFoto1+"</texto1>";
        conteudo = conteudo + "<texto2>"+descFoto2+"</texto2>";
        conteudo = conteudo + "<path_img1>"+pathFoto1+"</path_img1>";
        conteudo = conteudo + "<path_img2>"+pathFoto2+"</path_img2>";
        conteudo = conteudo + "<contador>"+cont+"</contador>";
        conteudo = conteudo + "</sld>";
        conteudo = conteudo + "</slide>";

        Grava();
    }

    public void gravaFigura(String titulo, String pathFoto, String descFoto, String nome, int cont){

        nomeArq = nome+".xml";

        conteudo = "<?xml version='1.0' encoding='UTF-8'?>";
        conteudo = conteudo + "<slide nome='GeradorSlide'>";
        conteudo = conteudo + "<sld>";
        conteudo = conteudo + "<tipo>Figura</tipo>";
        conteudo = conteudo + "<nome>"+nome+".htm</nome>";
        conteudo = conteudo + "<titulo>"+titulo+"</titulo>";
        conteudo = conteudo + "<texto>"+descFoto+"</texto>";
        conteudo = conteudo + "<path_img>"+pathFoto+"</path_img>";
        conteudo = conteudo + "<contador>"+cont+"</contador>";
        conteudo = conteudo + "</sld>";
        conteudo = conteudo + "</slide>";

        Grava();
    }

    public void texto(String titulo, String corpo, String nome, int cont){
```

```

        nomeArq = nome+".xml";

        conteudo = "<?xml version='1.0' encoding='UTF-8'?>";
        conteudo = conteudo + "<slide nome='GeradorSlide'>";
        conteudo = conteudo + "<sld>";
        conteudo = conteudo + "<tipo>Texto</tipo>";
        conteudo = conteudo + "<nome>"+nome+".htm</nome>";
        conteudo = conteudo + "<titulo>"+titulo+"</titulo>";
        conteudo = conteudo + "<corpo>"+corpo+"</corpo>";
        conteudo = conteudo + "<contador>"+cont+"</contador>";
        conteudo = conteudo + "</sld>";
        conteudo = conteudo + "</slide>";

        Grava();
    }

    private void Grava()
    {
        try
        {
            open=new FileWriter(nomeArq); //Abrindo Arquivo
            dado=new PrintWriter(open,true); //Buffer para gravar dados no arquivo

            dado.println(conteudo); // Grava aki

        } catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,"Erro ao abrir
arquivo","Erro",JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

## APÊNDICE C

### LeituraXML.java

```
package geradorslide;

import java.io.File;
import java.io.IOException;
import java.util.Iterator;
import java.util.List;

import org.jdom.Document;
import org.jdom.Element;
import org.jdom.JDOMException;
import org.jdom.input.SAXBuilder;
import javax.swing.*.*;

public class LeituraXML {

    public String tipo;
    public String nome;
    public String corpo;
    public String titulo;
    public String contador;
    public String pathFoto1;
    public String pathFoto2;
    public String texto1;
    public String texto2;

    public void abreXML(String pathXML) {

        //Aqui informa o nome do arquivo XML.
        File f = new File(pathXML);

        //Criamos uma classe SAXBuilder que vai processar o XML4
        SAXBuilder sb = new SAXBuilder();

        //Este documento agora possui toda a estrutura do arquivo.
        Document d = null;
        try {
            d = sb.build(f);
        } catch (JDOMException e) {
            System.out.println(e.getMessage());
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }

        //Recuperamos o elemento root
        Element supermercado = d.getRootElement();

        //Recuperamos os elementos filhos (children)
        List elements = supermercado.getChildren();
        Iterator i = elements.iterator();

        while (i.hasNext()) {
```

```

Element element = (Element) i.next();

tipo      = element.getChildText("tipo");
nome      = element.getChildText("nome");
contador = element.getChildText("contador");
titulo    = element.getChildText("titulo");

if (tipo.equals("Texto"))
{
    corpo = element.getChildText("corpo");
}
else if (tipo.equals("Figura"))
{
    texto1      = element.getChildText("texto");
    pathFoto1   = element.getChildText("path_img");
}
else if (tipo.equals("TextosFiguras"))
{
    texto1      = element.getChildText("texto1");
    pathFoto1   = element.getChildText("path_img1");
    texto2      = element.getChildText("texto2");
    pathFoto2   = element.getChildText("path_img2");
}
}
}
}

```

## APÊNDICE D

### Indice.java

```
package geradorslide;

import java.io.*;
import javax.swing.JOptionPane;

public class Indice {

    private static FileWriter open;
    private static PrintWriter dado;
    private String local = "";
    /** Creates a new instance of Indice */
    public void gravaIndice(String nomeProjeto, String textoIndice, String link) {

        String localAux = "<a href="+link+" class=ativolinki>- " + textoIndice + "</a><br>";
        local = local + localAux;

        String nomeArqIndice="index.htm"; // Nome do Arquivo

        String conteudo = "<html><HEAD><style type=text/css>" +
            ".ativolinki { font-family: verdana; font-size : 9pt; margin-left: 2pt;
color: #003366; text-decoration:NONE}" +
            ".ativolinki:hover { font-family: verdana; font-size : 9pt; margin-left:
2pt; color: #CC0000; text-decoration:NONE}" +
            "</STYLE><title>Gerador de Slide</title></head>";

        conteudo = conteudo + "<body topmargin='0' leftmargin='0' cellpadding='0' cellspacing='0'
bgcolor=#E1E1E1>";
        conteudo = conteudo + "<table border='1' align='center' cellpadding='0' bgcolor=#6262FF'
cellspacing='0' style='border-width:0; border-collapse: collapse' bordercolor='#111111' width='780'
id='AutoNumber1'>";
        conteudo = conteudo + "<tr><td width='100%' style='border-style: none; border-width:
medium' bgcolor=#CBD5D9'><p align='center'><br>";
        conteudo = conteudo + "<b><font color='#0072A8' face='Arial'
size='4'>ÍNDICE<BR>"+nomeProjeto+"</font><font face='Arial'><br>&nbsp;</font></b></td></tr>";
        conteudo = conteudo + "<tr><td width='100%' style='border-style: none; border-width:
medium' bgcolor=#EBEFF0' height='500' valign='top'><BR><BR>";

        // PASSAS OS LINKS DOS SLIDES
        conteudo = conteudo + local + "</td></tr></table></body></html>";

        try
        {
            open=new FileWriter(nomeArqIndice); //Abrindo Arquivo
            dado=new PrintWriter(open,true); //Buffer para gravar dados no arquivo

            dado.println(conteudo); // Grava aki

        }catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,"Erro ao abrir
arquivo","Erro",JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

## APÊNDICE E

### FileDialogo.java

```
package geradorslide;

import javax.swing.*.*;
import java.io.*;

public class FileDialogo extends javax.swing.JFrame {
// Principal p = new Principal();
    public String aux;
    /** Creates new form FileDialogo */
    public FileDialogo() {
        initComponents();

        jFileChooser1.setSelectionMode(JFileChooser.FILES_ONLY);

        int result = jFileChooser1.showSaveDialog(null);

        if (result == JFileChooser.CANCEL_OPTION)
            return;
        else {
            File caminho = jFileChooser1.getSelectedFile();
            aux = caminho.getPath();
            return;
        }
    }
    private void initComponents() { //GEN-BEGIN: initComponents
        jFileChooser1 = new javax.swing.JFileChooser();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().add(jFileChooser1, java.awt.BorderLayout.CENTER);

        pack();
    } //GEN-END: initComponents

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new FileDialogo().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify //GEN-BEGIN: variables
    private javax.swing.JFileChooser jFileChooser1;
    // End of variables declaration //GEN-END: variables
}
```

## APÊNDICE F

### FiguraTexto.java

```
package geradorslide;

import java.io.*;
import javax.swing.JOptionPane;

public class FiguraTexto {

    private static FileWriter open;
    private static PrintWriter dado;

    /** Creates a new instance of FiguraTexto */
    public void gravaFiguraTexto(String titulo, String pathFoto, String descFoto, int cont) {
        String conteudo;

        String nomeArq = "slide"+cont+".htm";
        Integer anterior = cont - 1;
        Integer proximo = cont + 1;

        conteudo = "<html><head>";
        conteudo = conteudo + "<style type=text/css>";
        conteudo = conteudo + ".ativolink { font-family: verdana; font-size : 9pt; margin-left: 2pt; color: #003366; text-decoration:NONE}";
        conteudo = conteudo + ".ativolink:hover { font-family: verdana; font-size : 9pt; margin-left: 2pt; color: #CC0000; text-decoration:NONE}</STYLE>";
        conteudo = conteudo + "<title>GERADOR SLIDE</title></head>";

        conteudo = conteudo + "<body topmargin='0' leftmargin='0' cellpadding='0' cellspacing='0' bgcolor='#E1E1E1'>";
        conteudo = conteudo + "<table border='1' align='center' cellpadding='0' bgcolor='#FFFFFF' cellspacing='0' style='border-collapse: collapse; border-left-width:0; border-right-width:0; border-top-width:0; bordercolor='#111111' width='780' id='AutoNumber1'>";

        conteudo = conteudo + "<tr><td bgcolor='#CBD5D9' width='100%' style='border-style:none; border-width:medium; '>";
        conteudo = conteudo + "<p align='right'><font size='1' face='Verdana'>PG. "+cont+"</font></td></tr>";
        conteudo = conteudo + "<tr><td bgcolor='#CBD5D9' width='100%' style='border-left-style: none; border-left-width: medium; border-right-style: none; border-right-width: medium; border-top-style: none; border-top-width: medium; border-bottom-style: solid; border-bottom-width: 1'>";
        conteudo = conteudo + "<p align='center'><b><font face='Verdana'>"+titulo+"</font></b><br>&nbsp;</td></tr>";
        conteudo = conteudo + "<tr><td width='100%' style='border-left-style: solid; border-left-width: 1; border-right-style: solid; border-right-width: 1; border-top-style: solid; border-top-width: 1'><p align='center'>";
        conteudo = conteudo + "<img border='0' src='"+pathFoto+"'></td></tr>";
        conteudo = conteudo + "<tr><td width='100%' style='border-left-style: solid; border-left-width: 1; border-right-style: solid; border-right-width: 1; border-bottom-style: solid; border-bottom-width: 1'><p align='center'>";
        conteudo = conteudo + "<font face='Verdana' size='2'><br>"+descFoto+"<br>&nbsp;</font></td></tr></table>";
        conteudo = conteudo + "<table border='1' align='center' cellpadding='0' bgcolor='#FFFFFF' cellspacing='0' style='border-width:0; border-collapse: collapse' bordercolor='#111111' width='780' id='AutoNumber1'><tr>";
        conteudo = conteudo + "<td bgcolor='#CBD5D9' width='100%' align='center' style='border-style: none; border-width: medium'><font face='Verdana' style='font-size: 9pt'><br>";
```



```

// CRIA O LINK QUE VOLTA PARA O INDICE
conteudo = conteudo + "<a href='index.htm' class='ativolinki'>Indice</a>&gt;";

// SE ANTERIOR FOR MENOR QUE ZERO ENTÃO É O PRIMEIRO SLIDE E NÃO
TEM ANTERIOR
if (anterior > 0)
{
    conteudo = conteudo + " <a href='slide"+anterior+".htm'
class='ativolinki'>Anterior</a>&gt;";
}

// CRIA O LINK QUE DIRECIONA PARA A PRÓXIMA PÁGINA
conteudo = conteudo + " <a href='slide"+proximo+".htm'
class='ativolinki'>Próxima</a>&gt;";

conteudo = conteudo + "<br>&nbsp;</font></td></tr></table></body></html>";

try
{
    open=new FileWriter(nomeArq); //Abrindo Arquivo
    dado=new PrintWriter(open,true); //Buffer para gravar dados no arquivo

    dado.println(conteudo); // Grava aki

} catch(Exception e)
{
    JOptionPane.showMessageDialog(null,"Erro ao abrir
arquivo","Erro",JOptionPane.ERROR_MESSAGE);
}

private static void FechaArquivo()
{
    dado.close(); //Fechando arquivo
}
}

```

## APÊNDICE G

### FigurasTextos.java

```
package geradorslide;

import java.io.*;
import javax.swing.JOptionPane;

public class FigurasTextos {

    private static FileWriter open;
    private static PrintWriter dado;

    // pathFoto recebe por parametro o local da figura
    // descFoto recebe por parametro o texto que aparece ao lado da figura
    // cont é o contador que monta os links...
    public void gravaFigurasTextos(String titulo, String pathFoto1, String descFoto1, String pathFoto2, String
descFoto2, int cont) {
        String conteudo;

        String nomeArq = "slide"+cont+".htm";
        Integer anterior = cont - 1;
        Integer proximo = cont + 1;

        conteudo = "<html><head>";
            conteudo = conteudo + "<style type=text/css>";
            conteudo = conteudo + ".ativolink { font-family: verdana; font-size : 9pt; margin-left: 2pt; color: #003366;
text-decoration:NONE}";
            conteudo = conteudo + ".ativolink:hover { font-family: verdana; font-size : 9pt; margin-left: 2pt; color:
#CC0000; text-decoration:NONE}</STYLE>";
            conteudo = conteudo + "<title>GERADOR SLIDE</title></head>";
            conteudo = conteudo + "<body topmargin='0' leftmargin='0' cellpadding='0' cellspacing='0'
bgcolor=#E1E1E1>";
                conteudo = conteudo + "<table border='1' align='center' cellpadding='0' bgcolor=#FFFFFF'
cellspacing='0' style='border-collapse: collapse; border-left-width:0; border-right-width:0; border-top-width:0'
bordercolor=#111111' width='780' id='AutoNumber1'>";
                    conteudo = conteudo + "<td bgcolor=#CBD5D9' width='100%' colspan='2' style='border-
style:none; border-width:medium; '>";
                        conteudo = conteudo + "<p align='right'><font face='Verdana' size='1'>PG.
"+cont+"</font></td></tr>";
                            conteudo = conteudo + "<tr><td bgcolor=#CBD5D9' width='100%' colspan='2' style='border-
left-style: none; border-left-width: medium; border-right-style: none; border-right-width: medium; border-top-
style: none; border-top-width: medium'><p align='center'>";
                                conteudo = conteudo + "<font
face='Verdana'><b>"+titulo+"</b><br>&nbsp;</font></td></tr>";
                                    conteudo = conteudo + "<tr><td width='50%' style='border-left-style: solid; border-left-width:
1; border-top-style: solid; border-top-width: 1'>";
                                        conteudo = conteudo + "<p align='justify'><font face='Verdana'
size='2'>"+descFoto1+"</font></td>";
                                            conteudo = conteudo + "<td width='50%' align='center' style='border-right-style: solid; border-
right-width: 1; border-top-style: solid; border-top-width: 1'>";
                                                conteudo = conteudo + "<img border='0' src='"+pathFoto1+"></td></tr>";
                                                    conteudo = conteudo + "<tr><td width='50%' align='center' style='border-left-style: solid;
border-left-width: 1; border-bottom-style: solid; border-bottom-width: 1'>";
                                                        conteudo = conteudo + "<img border='0' src='"+pathFoto2+"></td>";
                                                            conteudo = conteudo + "<td width='50%' style='border-right-style: solid; border-right-width: 1;
border-bottom-style: solid; border-bottom-width: 1'>";
```

```

        conteudo = conteudo + "<p align='justify'><font face='Verdana'
size='2'>"+descFoto2+"</font></td></tr></table>";
        conteudo = conteudo + "<table border='1' align='center' cellpadding='0' bgcolor='#FFFFFF'
cellspacing='0' style='border-width:0; border-collapse: collapse' bordercolor='#111111' width='780'
id='AutoNumber1'>";
        conteudo = conteudo + "<tr><td bgcolor='#CBD5D9' width='100%' align='center'
style='border-style: none; border-width: medium'>";
        conteudo = conteudo + "<font face='Verdana' style='font-size: 9pt'><br>";

// CRIA O LINK QUE VOLTA PARA O INDICE
        conteudo = conteudo + "&lt;<a href='index.htm' class='ativolinki'>Indice</a>&gt;";

// SE ANTERIOR FOR MENOR QUE ZERO ENTÃO É O PRIMEIRO SLIDE E NÃO TEM ANTERIOR
        if (anterior > 0)
        {
                conteudo = conteudo + " &lt;<a href='slide"+anterior+".htm'
class='ativolinki'>Anterior</a>&gt;";
        }

// CRIA O LINK QUE DIRECIONA PARA A PRÓXIMA PÁGINA
        conteudo = conteudo + " &lt;<a href='slide"+proximo+".htm'
class='ativolinki'>Proxima</a>&gt;";

        conteudo = conteudo + "<br>&nbsp;</font></td>";
        conteudo = conteudo + "</tr></table></body></html>";

try
{
        open=new FileWriter(nomeArq); //Abrindo Arquivo
        dado=new PrintWriter(open,true); //Buffer para gravar dados no arquivo

        dado.println(conteudo);

        }catch(Exception e)
        {
                JOptionPane.showMessageDialog(null,"Erro ao abrir
arquivo","Erro",JOptionPane.ERROR_MESSAGE);
        }
}

private static void FechaArquivo()
{
        dado.close(); //Fechando arquivo
}
}

```

## APÊNDICE H

### Ultimo.java

```
package geradorslide;

import java.io.*;
import javax.swing.JOptionPane;

public class Ultimo {

    private static FileWriter open;
    private static PrintWriter dado;

    /** Creates a new instance of FiguraTexto */
    public void gravaUltimo(int cont) {
        String conteudo;

        cont++;

        String nomeArq = "slide"+cont+".htm";

        conteudo = "<html><head>";
        conteudo = conteudo + "<title>GERADOR SLIDE</title></head>";
        conteudo = conteudo + "<body topmargin='0' leftmargin='0' cellpadding='0' cellspacing='0'
bgcolor='#000000'>";
        conteudo = conteudo + "<table width='100%'><tr><td width='100%' align='center'><font
face='verdana' size='2' color='#FFFFFF'>";
        conteudo = conteudo + "<BR><BR><BR><BR>FIM DA
APRESENTAÇÃO<BR><BR><BR><BR>";
        conteudo = conteudo + "</font></td></tr></table></body></html>";

        try
        {
            open=new FileWriter(nomeArq); //Abrindo Arquivo
            dado=new PrintWriter(open,true); //Buffer para gravar dados no arquivo

            dado.println(conteudo); // Grava aki

        }catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,"Erro ao abrir
arquivo","Erro",JOptionPane.ERROR_MESSAGE);
        }
    }

    private static void FechaArquivo()
    {
        dado.close(); //Fechando arquivo
    }
}
```